An Information Theoretic Evaluation Criterion for 3D Reconstruction Algorithms

Rama Chellappa
University of Maryland,
College Park, MD 20742.
rama@cfar.umd.edu

Amit K. Roy Chowdhury
University of California
Riverside, CA 92521.
amitrc@ee.ucr.edu

Abstract

Even though numerous algorithms exist for estimating the structure of a scene from its video, the solutions obtained are often of unacceptable quality. To overcome some of the deficiencies, many application systems rely on processing more information than necessary with the hope that the redundancy will help improve the quality. This raises the question about how the accuracy of the solution is related to the amount of information processed by the algorithm. Can we define the accuracy of the solution precisely enough that we automatically recognize situations where the quality of the data is so bad that even a large number of additional observations will not yield the desired solution? This paper proposes an information theoretic criterion for evaluating the quality of a 3D reconstruction in terms of the statistics of the observed parameters (i.e. the image correspondences). The accuracy of the reconstruction is judged by considering the change in mutual information (or equivalently the conditional differential entropy) between a scene and its reconstructions and its effectiveness is shown through simulations. A brief discussion on the applicability of information theoretic criteria for other vision algorithms concludes the paper.

1 Introduction

Obtaining accurate 3D models from video using the structure from motion (SfM) approach [1], [2], is extremely important because of its diverse applications, ranging from multimedia to medical diagnosis. Yet the quality of many of the automatic 3D reconstructions leave much to be desired. This has led many researchers to analyze the sensitivity, robustness and statistical error characterization of the existing algorithms, trying to understand algorithm behavior and the characteristics of the natural phenomenon that is being modeled [3], [4], [5], [6], [7], [8], [9]. To overcome these errors, the tendency has been to add redundancy in the information processed. This raises the question as to how the redundant information affects the quality of the final solution. In this paper, we consider the situation where multiple reconstructions of the same scene are available (called intermediate or individual reconstructions, in this paper), that are combined together to obtain the final estimate (Figure (1)). We compute the incremental mutual information between the unknown 3D structure and increasing numbers of intermediate reconstructions.

Before proceeding to give a detailed description of the idea, we would like to draw the attention of the reader briefly to the area of model selection in statistics (AIC, BIC, MDL etc. [10]). The idea of fitting models to geometric data was formalized by Kanatani using a Geometric Information Criterion (GIC) [11]. However, a large number of SfM algorithms are not model based; they reconstruct individual point features of the scene. Our work tries to define the *quality of reconstruction* from point features in information theoretic terms. We also provide a discussion on the usefulness of information theoretic measures for evaluating computer vision algorithms.

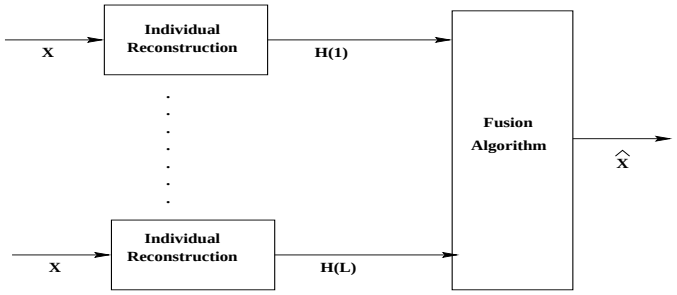


Figure 1: Block diagram representation of the reconstruction framework. $\mathbf{X}$ is the inverse depth that we want to estimate, $(\mathbf{H}(1), ..., \mathbf{H}(L))$ are the intermediate reconstructions (e.g. from each individual camera), and $\hat{\mathbf{X}}$ is the final fused estimate.

2 An Information Theoretic Criterion for 3D Reconstruction

2.1 Problem Formulation

We assume that all the depth values are aligned to a common frame of reference. Feature points will be represented by subscripts, separate reconstructions will be within parenthesis. The vector of estimates of the inverse depth [1] $[H_i(1), ..., H_i(N)]'$ will be denoted by $\mathbf{H}_i^{(N)}$. The boldface notation $\mathbf{H}(i)$ will represent all the features in the i^{th} reconstruction. The final estimate $\hat{\mathbf{X}}$ of $\mathbf{X} = [X_1, ..., X_M]'$ is obtained by fusing the individual reconstructions $(\mathbf{H}(1), ..., \mathbf{H}(L))$. To keep the notation simple, the subscript for the feature point will not be mentioned, unless required. The individual estimates are modeled as

$$H(i) = X + V(i) \tag{1}$$

where X is the inverse depth value of the particular feature.

2.2 Main Result

We will now present an information theoretic measure for evaluating the quality of a 3D reconstruction algortihm by analyzing the contribution of each of the individual reconstructions. Our entire analysis is for a particular point and thus the subscript will be dropped, unless required for clarity. Our criterion for evaluating the quality of reconstruction depends on estimating the difference in mutual information for the two sets of observations, $\mathbf{H}^{(L)}$ and $\mathbf{H}^{(L-1)}$. We term this as the *incremental mutual information* (IMI), i.e.

$$\Delta I(L) = I(X, \mathbf{H}^{(L)}) - I(X, \mathbf{H}^{(L-1)}). \tag{2}$$

The term gives us an idea of the contribution of the L^{th} observation to the reconstruction strategy with respect to the previous $(L-1)$ observations. As the number of observations increase, the effect of an additional observation decreases and approaches zero in the limit. In order to be assured that the reconstruction quality is actually improving, we need to consider only those situations where the mutual information $I(X, \mathbf{H}^{(L)})$ is non-decreasing. This ensures that we remove cases where the reconstruction is actually getting worse, and further observations are not improving it any more.

Using the relationship between mutual information and entropy, it is possible to obtain a different interpretation of the IMI. Denoting by $h(X)$ the entropy of the random variable X, we know that [12] $I(X;Y) = h(X) + h(Y) - h(X,Y)$. Thus $\Delta I(L)$ in (2) can be written as

$$\Delta I(L) \;\; = \;\; I(X; \mathbf{H}^{(L)}) - I(X; \mathbf{H}^{(L-1)}) \;\; = \;\; h(X|\mathbf{H}^{(L-1)}) - h(X|\mathbf{H}^{(L)}). \tag{3}$$

The quantity defined as the IMI can also be referred to as the incremental conditional entropy. Since entropy of a random variable is a measure of its uncertainty, ΔI measures the reduction in the uncertainty as we add an extra

[1] The inverse depth is used throughout this paper since it is the quantity that is estimated from the SfM equations for reconstruction from a video and its statistics can be obtained in an analytic form more easily than for the depth.

observation. Since the IMI tends to zero in the limit, the difference in the conditional entropy also approaches zero. Thus we will consider more and more images from the video sequence till the uncertainty in the final structure estimate can be reduced no further. This is the intuitive idea behind our criterion in (2).

The rate at which the IMI decreases is also an important measure of the progress of the algorithm. An extremely slow rate of fall indicates that more images will be necessary to achieve an acceptable level of quality. Since there is motion between adjacent frames of the video, a particular point will move out of the field of view of the camera after a certain amount of time. A very slow rate of fall of ΔI might mean that the quality of the reconstruction is not good enough even when the point is no longer visible. The rate of change of ΔI can be obtained as

$$\begin{aligned} \Delta^2 I(L) &= \Delta I(L) - \Delta I(L-1) \\ &= I(X, \mathbf{H}^{(L)}) + I(X, \mathbf{H}^{(L-2)}) - 2I(X, \mathbf{H}^{(L-1)}). \end{aligned} \tag{4}$$

Combining (2) and (4), we can state that an acceptable reconstruction quality has been achieved when $I(X, \mathbf{H}^{(L)})$ is non-decreasing **and** the following conditions are satisfied simultaneously:

$$\begin{aligned} \Delta^2 I(L) &\leq 0, \quad \forall L > L_0, \\ \Delta I(L) &< \tau, \end{aligned} \tag{5}$$

where L_0 is a constant and τ is a threshold defining an acceptable quality of reconstruction. Since $\Delta I(L)$ is monotone non-increasing for $L > L_0$ and is bounded below by zero, the monotone convergence theorem [13] applied to (3) implies that $h(X|\mathbf{H}^{(L-1)}) \to h(X|\mathbf{H}^{(L)}) \to h_0$ for some $L > L_0$. Thus, h_0 is the minimum level of uncertainty in a scene described by L observations.

Since the criterion does not depend on how the intermediate reconstructions are obtained, it is, in principle, independent of the 3D reconstruction strategy. However, the procedure for estimation of IMI may be optimized for a particular algorithm. Details on the estimation process can be found in [14].

2.3 IMI Computation Under Gaussian Distributions

Assume that $X \sim \mathcal{N}(0, \sigma_x^2 = P_X)$ and $\{V(i), i = 1, ..., N\}$ is a sequence of independent random variables distributed as $\mathcal{N}(0, \sigma_{V(i)}^2)$. Let $P_V = \text{diag}\,[P_V(i)]_{i=1,...,N} = \text{diag}\left[\sigma_{V(1)}^2, ..., \sigma_{V(N)}^2\right]$ [2].

From (1), $E[H(i)] = 0$ and

$$\begin{aligned} E[H(i)H(j)] &= E[(X + V(i))(X + V(j))] \\ &= P_X + P_V(i)\delta_{ij}, \end{aligned} \tag{6}$$

where δ_{ij} is a Kronecker delta function. Thus the covariance of $\mathbf{H}^{(N)}$ is $P_{\mathbf{H}^{(N)}} = P_V^{(N)} + \mathbf{1}_N P_X \mathbf{1}_N^T$, where $\mathbf{1}_N$ is a vector of N ones. Then the mutual information between X and $H(i)$,

$$\begin{aligned} I(X; H(i)) &= h(H(i)) - h(H(i)|X) \\ &= \frac{1}{2} \log\left(1 + \frac{P_X}{P_V(i)}\right). \end{aligned} \tag{7}$$

Next, consider the mutual information between the unknown X and the vector of observations $\mathbf{H}^{(N)}$. We will denote by $|K|$ the determinant of a matrix K.

$$\begin{aligned} I(X; \mathbf{H}^{(N)}) &= h(\mathbf{H}^{(N)}) - h(\mathbf{H}^{(N)}|X) \\ &\overset{(a)}{=} h(\mathbf{H}^{(N)}) - \sum_{i=1}^{N} \tfrac{1}{2} \log(2\pi e P_V(i)) \\ &\overset{(b)}{=} \tfrac{1}{2} \log\left(\frac{|P_V + \mathbf{1}_N P_X \mathbf{1}_N^T|}{|P_V|}\right). \end{aligned} \tag{8}$$

[2] Where necessary to distinguish a particular feature point, we will use the notation $\sigma_{x_j}^2$ and $P_{V_j}(i)$ or $\sigma_{V_j(i)}^2$ for the j^{th} point.

(a) is a result of applying the chain rule of entropy and substituting the expression for the differential entropy of a Gaussian random variable [12]; (b) is due to the fact that $|P_V| = \prod_{i=1}^{N} P_V(i) = \prod_{i=1}^{N} \sigma^2_{V(i)}$. Using the method of induction and the properties of determinants, it can be shown that $|P_V + \mathbf{1}_N P_X \mathbf{1}_N^T| = \prod_{i=1}^{N} \sigma^2_{V(i)} + \sigma_x^2 \sum_{i=1}^{N} \prod_{\substack{j=1 \\ j\neq i}}^{N} \sigma^2_{V(j)}$. Then from (8), the expression for the mutual information becomes

$$I(X; \mathbf{H}^{(N)}) = \tfrac{1}{2} \log \left(1 + \sum_{i=1}^{N} \frac{\sigma_X^2}{\sigma^2_{V(i)}} \right). \tag{9}$$

Let us compute the difference in the mutual information for the two sets of observations, $\mathbf{H}^{(N)}$ and $\mathbf{H}^{(N-1)}$. We shall call this the *incremental mutual information*, ΔI. Thus,

$$\begin{aligned}
\Delta I &= I(X; \mathbf{H}^{(N)}) - I(X; \mathbf{H}^{(N-1)}) \\
&= \tfrac{1}{2} \log \left(\frac{|P_{V^{(N)}} + \mathbf{1}_N P_X \mathbf{1}_N^T|}{|P_{V^{(N-1)}} + \mathbf{1}_{N-1} P_X \mathbf{1}_{N-1}^T|} . \frac{|P_{V^{(N-1)}}|}{|P_{V^{(N)}}|} \right) \\
&= \tfrac{1}{2} \log \left(\frac{\prod_{i=1}^{N} \sigma^2_{V(i)} + \sigma_x^2 \sum_{i=1}^{N} \prod_{\substack{j=1 \\ j\neq i}}^{N} \sigma^2_{V(j)}}{\prod_{i=1}^{N} \sigma^2_{V(i)} + \sigma_x^2 \sum_{i=1}^{N-1} \prod_{\substack{j=1 \\ j\neq i}}^{N} \sigma^2_{V(j)}} \right) \\
&= \tfrac{1}{2} \log \left(1 + \frac{1/\sigma^2_{V(N)}}{\frac{1}{\sigma_x^2} + \sum_{i=1}^{N-1} \frac{1}{\sigma^2_{V(i)}}} \right) \\
&= \tfrac{1}{2} \log \left(1 + \frac{1/P_V(N)}{\frac{1}{\sigma_x^2} + \sum_{i=1}^{N-1} \frac{1}{P_V(i)}} \right).
\end{aligned} \tag{10}$$

Equation (10) gives us a measure of the extra information that would be obtained by including an additional observation into the fusion process. Also, since

$$I(X; \mathbf{H}^{(N)}) - I(X; \mathbf{H}^{(N-1)}) = h(X|\mathbf{H}^{(N-1)}) - h(X|\mathbf{H}^{(N)}), \tag{11}$$

the quantity defi ned as the incremental mutual information can also be referred to as the incremental conditional entropy. Thus we are measuring the reduction in the uncertainty of the solution as we consider an extra observation. The difference in the differential entropy determines the decrease in the coding length of the scene structure as the number of observations increases [12].

The above calculation requires computing the variances of the intermediate reconstructions. Any method to compute them is perfectly suitable. In an earlier work [15], we have shown how to do this for the case of 3D reconstruction using optical flow. It should be remembered that all the geometric quantities have to be with respect to a particular frame of reference; hence it may be necessary to transform the variances appropriately.

An Estimation Theoretic Interpretation: We will now present an alternative interpretation of the result in (10) from an estimation theoretic perspective. The mean squared distortion is defi ned as

$$D(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{M} \sum_{j=1}^{M} E[(X_j - \hat{X}_j)^2]. \tag{12}$$

Let $p(X_j, H_j(1), ..., H_j(N))$ denote the joint density function of the parameter and observations. The mean square error estimator $\hat{X}_j$ of X_j, obtained from $\mathbf{H}^{(N)}$, is $\hat{X}_j(N) = E\left[X_j | H_j^{(N)}\right]$. From the Cramer-Rao lower bound

(CRLB) we can write the following set of inequalities.

$$\begin{aligned} D &\geq \frac{1}{M}\sum_{j=1}^{M}\frac{1}{E\left[-\frac{\partial^2}{\partial X^2}\log p(X_j, H_j(1), ..., H_j(N))\right]} \\ &= \frac{1}{M}\sum_{j=1}^{M}\frac{1}{\frac{1}{\sigma_{x_j}^2}+\sum_{i=1}^{N}E\left[-\frac{\partial^2}{\partial X^2}\log p(H_j(i)|X)\right]} \\ &\geq \frac{1}{\frac{1}{M}\sum_{j=1}^{M}\left(\frac{1}{\sigma_{x_j}^2}+\sum_{i=1}^{N}\frac{1}{P_{V_j}(i)}\right)} \\ &\triangleq \frac{1}{\frac{1}{M}\sum_{j=1}^{M}\frac{1}{D_j(N)}}. \end{aligned} \tag{13}$$

The last step is a result of the application of Jensen's inequality [16] and the fact that $E\left[-\frac{\partial^2}{\partial X^2}\log p(H_j(i)|X)\right] = \frac{1}{P_{V_j}(i)}$. Recalling that (10) is for a particular feature point where the subscript has been suppressed for clarity of notation, let us denote $\Delta I_j \triangleq I(X_j; \mathbf{H}_j^{(N)}) - I(X_j; \mathbf{H}_j^{(N-1)})$. Then from (13) and the last expression of (10), we get

$$\Delta I_j = \tfrac{1}{2}\log\left(\frac{D_j(N-1)}{D_j(N)}\right). \tag{14}$$

Alternatively, the innovations at the N^{th} stage, $\gamma_N = X_N - \hat{X}_N$. Then following the standard derivation for the Kalman filter [16], it can be shown that variance of the innovations

$$P_{\gamma_N} = \sigma_{V(N)}^2\left(1+\frac{1/\sigma_{V(N)}^2}{\frac{1}{\sigma_x^2}+\sum_{i=1}^{N-1}\frac{1}{\sigma_{V(i)}^2}}\right), \tag{15}$$

which shows that, for each feature point, the incremental mutual information is related to P_{γ_N} as

$$\Delta I = \tfrac{1}{2}\log\left(\frac{P_{\gamma_N}}{\sigma_{V(N)}^2}\right). \tag{16}$$

These relationships provide an alternative estimation theoretic interpretation to our result. Taken together (10), (14) and (16) demonstrate the use of statistical evaluation techniques to the SfM problem, when it is suitably formulated.

3 Analysis and Experiments

3.1 Analysis:

Present methods to evaluate the quality of a reconstruction involve computing the distortion in (12). For a fusion algorithm, this means that we need to compute (12) at every stage of the fusion and decide when to stop. This is computationally intensive, distortion measures are not always very useful in practical experiments since the choice of an acceptable threshold if often arbitrary and the source of the error (whether in the intermediate reconstructions or in the fusion algorithm) is difficult to identify. In our approach, (10) gives a direct way to measure the contribution of the intermediate solutions and the accuracy of the final solution as the algorithm progresses. The statistics of the error can be computed using the SfM equations and its solutions, as described in [15]. If the solution is far from its desired values, the error would be larger than if the solution is close to its true value. When the error in the intermediate reconstructions is small, D_j is small and hence the difference in the mutual information is small. Ideally, this difference should go to zero as we include more and more observations. If the error is large, D_j would be large and ΔI_j would not decrease appreciably with the number of observations. Another salient feature of our method is that we measure the information content between the true structure and the reconstructions *before* the

fusion. This allows us to understand the source of the error better since the effect of intermediate reconstructions and fusion algorithm are separated.

One scenario where this idea can be applied is reconstruction from a video sequence where intermediate reconstructions, $\mathbf{H}(1), ..., \mathbf{H}(L)$, obtained from a few frames (two or three) are combined together. Another application would be where partial reconstructions have been obtained from multiple cameras [3]. These partial models would have common overlapping regions which can be combined together to form the single estimate. In this case, $\mathbf{H}(1), ..., \mathbf{H}(L)$ would represent these common sub-regions from L separate reconstructions.

The statistical assumptions of independence and Gaussianity are necessary in order to derive closed form expressions for the quantities of interest. The independence of the intermediate estimates $H(1), ..., H(L)$ may be valid when these are obtained from separate imaging systems and then combined. When the same camera is used, the intermediate reconstructions should be obtained with non-overlapping frames; otherwise the common frames increase the dependencies. Regarding the Gaussianity assumptions, it has been pointed out by Zhang in [7] that the correspondence errors in SfM are usually normally distributed, if we can get rid of the outliers in the matches.

3.2 Experiments:

Experiment 1: A set of 3D points were generated so that we know their true positions. The perspective projections of these points were generated and Gaussian noise with zero mean and known variance was added to these 2D locations. The projections were taken for different positions of the camera, so that in the end a set of tracked features was obtained. From every pair of such tracked features, the positions of the original 3D points were estimated, which results in a set of 3D reconstructions. The first plot of Figure 2(a) shows the true value of the 3D points and their estimated reconstruction from all the frames over which the features could be tracked. [4] The second diagram in Figure 2(a) plots the decrease in the incremental mutual information with the increasing number of intermediate reconstructions.

Experiment 2: As in the previous simulation, a set of features were tracked over a number of frames. However, the level of noise added to the feature positions was higher and it led to a mismatch of some of the features. The 3D positions of the points were estimated using the SfM algorithm and the results were erroneous as is clear from the first plot of Figure 2(b). The second plot of Figure 2(b) depicts this case where the incremental mutual information remains large and does not follow any trend.

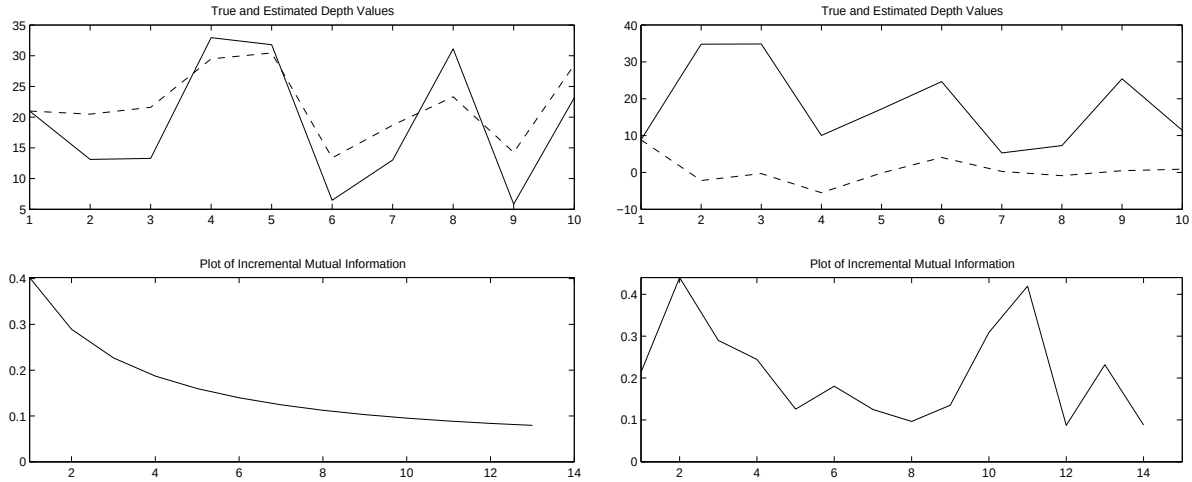
Experiment 3: We will now present our result on a real video sequence. The video consists of a person moving his head in front of a static camera. The aim was to reconstruct the model of the head of the person from this video. The focal length of the camera was known. Figure (3)(a) represents an image from the video along with some of the feature points which were tracked. Figure (3)(b) represents the change in the incremental mutual information between the unknown 3D structure and the intermediate reconstructions from every pair of frames. Based on this measure, the 3D model was reconstructed using 25 frames and Figure (3)(c) shows one particular view of this model.

4 A Discussion on the Usefulness of an Information Theoretic Criterion for Vision Algorithms

The statistical quality analysis of computer vision algorithms has been studied quite extensively (see [14] for a detailed literature survey on this topic). However, most of the methods have relied on computing the second order statistical moments, like covariance of the estimate. The covariance is a preferred measure because of its relation to the Cramer-Rao lower bound (CRLB), which dictates the minimum variance that an estimator can achieve [16]. If the variance of a sequence of estimates (say, of the 3D structure) tends towards the CRLB, then the estimate is said to be asymptotically efficient. However, computation of the CRLB often assumes that the estimate is unbiased (see [6]). This is because, computing the bias of an estimator is not an easy task. Hence, even though expressions exist

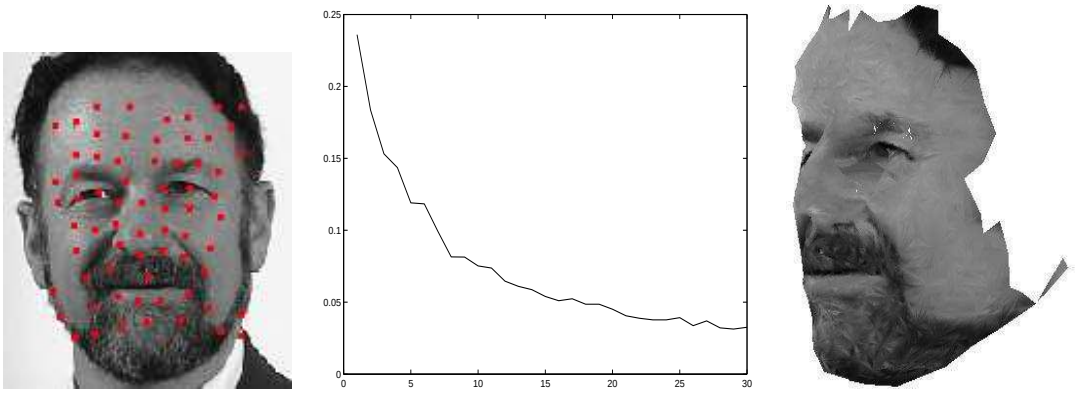
[3] This is the set-up in the "Eye Vision" technology developed by Carnegie Mellon University (CMU) and CBS Television (http://www.ri.cmu.edu/events/sb35/tksuperbowl.html).

[4] The first point was used to set the scale of the reconstruction, so that the geometric indeterminacies do not affect the result.



(a) (b)

Figure 2: (a): The upper plot shows the true value of the depth of the 3D points using the solid line and the fused estimate from the intermediate reconstructions from all the frames using the dotted lines. The second diagram plots the decrease in the incremental information with the increasing number of frames. (b): The upper plot shows the true value of the depth of the 3D points using the solid line and the fused estimate from the intermediate reconstructions from all the frames using the dotted lines. The lower plot is the change in the mutual information with increasing number of frames. This is the case where the estimated reconstruction does not converge to the true value even with increasing observations.



(a) (b) (c)

Figure 3: The above figures represent a 3D reconstruction from video using the method of measuring the IMI to judge the quality of the result. (a) is one of the images from the video along with the set of tracked features used for the reconstruction. (b) represents the change in the IMI with the number of images; (c) depicts one view from the reconstructed model.

for the CRLB of a biased estimator (known as the generalized CRLB), it is rarely used. The other main objection to the use of variance as a measure of quality is that it neglects the effect of higher order statistics. This is often a major approximation because the outliers, which are the source of many problems in computer vision algorithm, are often not modeled accurately by second order statistics.

Recent work [17, 18] has shown that the motion and depth estimates are statistically biased, and the bias is significant. This bias often propagates through later stages of the computation that rely on the motion and depth estimates. Also, as we have shown in [15], the noise in the SfM estimates is significantly non-Gaussian. Hence we propose that an information theoretic criterion which works by estimating the probability distribution function (pdf) of the concerned physical quantities (e.g. the depth), rather than concentrate on certain moments only, is a more suitable measure for a number of vision problems. The method of estimating the pdf will depend upon the particular algorithm and underlying assumptions. The major limitation of an information theoretic criterion is its efficient, robust and accurate estimation. This is because it is often difficult, and computationally expensive, to estimate the probability density functions of the parameters of interest. However, estimation of MI has received some attention

among researchers in signal processing and information theory [19]. It is our hope that such information theoretic criteria, as proposed in this paper, will become practically applicable as progress is made on robustly estimating them.

5 Conclusion

In this paper, we have introduced a method to evaluate the quality of 3D reconstruction from a video sequence. Existing methods rely on computing the distortion between the projections of the reconstructions and the original images and deciding that the reconstruction is of acceptable quality when the distortion is below a certain empirically chosen threshold. In this paper, we have shown that it is possible to evaluate the quality of the 3D structure estimate as the algorithm proceeds by computing the incremental mutual information, which determines the importance of considering an additional observation. It is related to the decrease in the coding length of the actual structure conditioned on the increasing number of observations. Finally, experimental results have been provided to justify these claims.

References

[1] O.D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[3] J. Oliensis, "A critique of structure from motion algorithms," *NECI Tech. Report*, vol. http://www.neci.nj.nec.com/homepages/oliensis/, 2000.

[4] Y. Ma, J. Kosecka, and S. Sastry, "Linear differential algorithm for motion recovery: A geometric approach," *International Journal of Computer Vision*, vol. 36, pp. 71–89, January 2000.

[5] J. Weng, N. Ahuja, and T.S. Huang, "Optimal motion and structure estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 864–884, September 1993.

[6] G.S. Young and R. Chellappa, "Statistical analysis of inherent ambiguities in recovering 3-d motion from a noisy fbw field," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 995–1013, October 1992.

[7] Z.Y. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision*, vol. 27, pp. 161–195, March 1998.

[8] D.D. Morris, K. Kanatani, and T. Kanade, "3d model accuracy and gauge fixing," in *CMU-RI-TR*, 2000.

[9] Z. Sun, V. Ramesh, and A.M. Tekalp, "Error characterization of the factorization method," *Computer Vision and Image Understanding*, vol. 82, pp. 110–137, May 2001.

[10] M. Hansen and B. Yu, "Model selection and the principle of minimum description length," *To appear in Journal of the American Statistical Association.*

[11] K. Kanatani, "Geometric information criterion for model selection," *IJCV*, vol. 26, no. 3, pp. 171–189, February 1998.

[12] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.

[13] R. Walter, *Principles of Mathematical Analysis, 3rd Edition*, McGraw-Hill, 1976.

[14] A. Roy-Chowdhury and R. Chellappa, "An information theoretic criterion for evaluating the quality of 3D reconstructions," *IEEE Trans. on Image Processing*, In press.

[15] A. Roy-Chowdhury and R. Chellappa, "Stochastic approximation and rate-distortion analysis for robust structure and motion estimation," *Intl. Journal of Computer Vision*, vol. 55, pp. 27–53, October 2003.

[16] H.V. Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1988.

[17] C. Fermuller, D. Shulman, and Y. Aloimonos, "The statistics of optical fbw," *Computer Vision and Image Understanding*, vol. 82, 2001.

[18] A. Roy Chowdhury and R. Chellappa, "Statistical error propagation in 3D modeling from monocular video," in *CVPR Workshop on Statistical Analysis in Computer Vision*, 2003.

[19] G.A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Trans. on Information Theory*, vol. 45, pp. 1315–1321, 1999.

Modular Programming Techniques for Distributed Computing Tasks

Anthony Cowley, Hwa-Chow Hsu, Camillo J. Taylor
GRASP Laboratory
University of Pennsylvania, Philadelphia, PA, USA, 19104

ABSTRACT

This paper describes design patterns used in developing a software platform for mobile robot teams engaged in distributed sensing and exploration tasks. The goal of the system presented is to minimize redundancy throughout the development and execution pipelines by exploring the application of a strong type system to both the collaborative development process and runtime behaviors of mobile sensor platforms. The solution we have implemented addresses both sides of this equation simultaneously by providing a system for self-describing inputs and outputs that facilitates code reuse among human developers and autonomous agents. This well-defined modularity allows us to treat executable code libraries as atomic elements that can be automatically shared across the network. In this fashion, we improve the performance of our development team by addressing software framework usability and the performance and capabilities of sensor networks engaged in distributed data processing. This framework adds robust design templates and greater communication flexibility onto a component system similar to TinyOS and NesC while avoiding the development effort and overhead required to field a full-fledged web services or Jini-based infrastructure. The software platform described herein has been used to field collaborative teams of UGVs and UAVs in exploration and monitoring scenarios.

KEYWORDS: *sensor network, distributed computing, software design*

1. INTRODUCTION

As efforts to field sensor networks, or teams of mobile robots, become more ambitious [5], [11], [4], communication constraints rapidly become the bottleneck both in the development effort and execution environment. From a development standpoint, human networking becomes clumsy as team sizes grow, putting team communications at a premium. Therefore, effort should be spent to optimize away the time developers must spend explaining things to each other, specifically, how to write code that has already been written or how to reuse existing code. If this aspect of collaborative development is not explicitly addressed, the team runs the risk of either losing the ability to reuse code, due to a lack of shared understanding, or drastically curtailing productivity by devoting excessive time to documentation efforts. Ideally, each developer's efforts will be documented extensively enough for others to easily reuse the existing code without placing an undesirable documentation burden on the original developer.

The desire for software agents to autonomously exploit existing code is a subtly parallel goal. Should an agent be able to specify its requirements, it ought to be able to identify any existing code that would meet this need. This applies both in the sense of agents discovering new sources of data, and that of interactive data processing requests. We wish to field a sensor network wherein one sensor can tap into a potentially live data stream without any a priori knowledge of other nodes or their capabilities, while also giving each node on the network the ability to ask questions that require the processing of large amounts of data. In the first case, we need to give our agents the ability to identify the types of data being exported by other agents. This is addressed by having communication endpoints describe the data they trade in. The latter case involves not only finding the correct type of data, but also sending an active query to the data rather than saturating the network by bringing the data to the query. Such behavior requires descriptions of data sources and sinks, as well as the ability to move, command, and control executable code across the network.

2. IMPLEMENTATION

A crucial aspect in the development of this framework design philosophy is the relationship between the new software and that which it is built upon. We chose to develop our high level environment on top of an already full-featured platform. In our case, this platform was Microsoft's .NET technology, which includes a strong type system in the .NET CLR (Common Language Runtime), an object-oriented language in the form of

C#, and many varieties of network functionality in the .NET Class Library. Our design then focused both on what functionality we wished to add and that which we wished to remove. Simply put, we want to impose some structure on our developers that is not inherent to C#, .NET, or any existing platform. This structure is a fundamental part of the ROCI (Remote Objects Control Interface) [6], [9] philosophy, and is imposed on the ROCI developer as a form of design control that we believe adds a level of reliability to the resultant system. By imposing a prescribed design on developers, we are better able to isolate potential weaknesses and build in error detection and handling functionality.

ROCI itself is a high level operating system useful for programming and managing sensor networks. The core control element in the ROCI architecture is the ROCI kernel. A copy of the kernel runs on every entity that is part of the ROCI network (robots, remote sensors, etc.). The kernel is responsible for handling program allocation and injection. It allows applications to be specified and executed dynamically by forming communication connections and transferring code libraries to the nodes as needed. The kernel is also responsible for managing the network and maintaining an updated database of other nodes in the ROCI network. In this way, ROCI acts as a distributed peer-to-peer system. Nodes can be dynamically added and removed from the network, and information about these nodes and the code running on them is automatically propagated throughout the system without the need for a central repository.

The control functionality needed by such a kernel is made possible by self-contained, reusable modules. Each module encapsulates a process which acts on data available on its inputs and presents its results on well defined outputs. Thus, complex tasks can be built by connecting inputs and outputs of specific modules. These connections are made through a pin architecture that provides a strongly typed, network transparent communication framework. A good analogy is to view each of these modules as an integrated circuit (IC) that has inputs and outputs and does some processing. Complex circuits can be built by wiring several ICs together, and individual ICs can be reused in different circuits. ROCI modules have been developed for a wide range of tasks such as: interfacing to low level devices like GPS units and cameras, computing position estimates based on GPS, IMU and odometry data, acquiring stereo panoramas, platform motion control, online map building and GPS waypoint navigation.

ROCI modules are further organized into tasks (Figure 1). A ROCI task is a way of describing an instance of a collection of ROCI modules to be run on a single node, and how they interact at runtime. Tasks represent a family of modules that work together to accomplish some end goal – a chain of building blocks that transforms input data through intermediate forms and into a useful output. A task can be defined in an XML file which specifies the modules that are needed to achieve the goal, any necessary module-specific parameters, and the connectivity between these modules. Tasks can also be defined and changed dynamically by starting new modules and connecting them with the inputs and outputs of other modules.

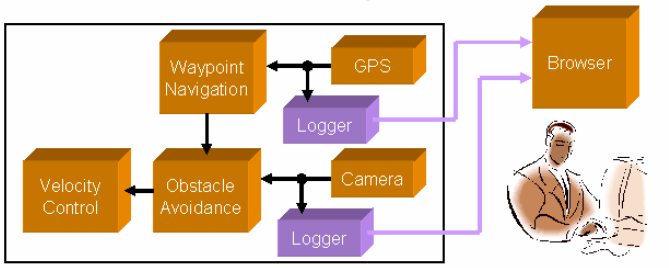


Figure 1. A typical ROCI task: a collection of behavior modules with loggers connected to specific pin connections. A human operator interfaces with the logs via the browser, which may be running on a different machine from the task.

The wiring that connects ROCI modules is the pin communication architecture. Pin communications in ROCI are designed to be network transparent yet high performance. Basically, a pin provides the developer with an abstract communications endpoint. These endpoints can either represent a data producer or a data consumer. Pins in the system are nothing more than strongly typed fields of the module class, and so are added to modules with a standard variable declaration statement. Pin communication allows modules to communicate with each other within a task, within a node or over a network seamlessly. The base Pin type will optimize the connection based on whether or not it is local and handle all error detection and handling, bandwidth utilization requirements, and optional buffering. The type system enforces pin compatibility at

run time which makes it impossible to connect inputs and outputs of incompatible types.

This compatibility evaluation is done in an object-oriented fashion such that, when necessary, output data will be transparently up-cast before being transmitted to a data sink. This negotiated compatibility allows for what we call "blind polymorphism," which does not require that both nodes have all the same types loaded. That is to say, if data can be cast up its inheritance hierarchy to the type that the data sink requires, then this cast will be done on the source side of the connection, thereby not requiring that the sink be aware of the inherited type.

Importantly, the modules in the system are self describing so that the kernel can automatically discover their input and output pins along with any user-settable parameters. These features of the ROCI architecture facilitate automatic service discovery since a module running on one ROCI node can query the kernel database to find out about services offered by modules on other nodes and can connect to these services dynamically.

The self describing behavior of module inputs, outputs, and parameters is achieved automatically through the use of the underlying type system. This is an important element of ROCI's ability to limit the potential for developer error. In the process of identifying necessary input and output pins, the module developer naturally defines certain data structures that the module takes as input and generates as output. These data structures represent a form of design contract that tells other users what type of input the module can parse, and what type of output it generates. This information is what the pin type system is built upon: a particular type of pin is designed to transfer a particular type of data. These types can then be used to verify potential connections between pins. By relying on type information that the developer necessarily creates by designing module-appropriate data structures, we are able to obviate the need for any separate developer-generated description of a module's inputs and outputs. Such descriptions run the risk of becoming out of date, and are not always easily checked. Relying on the type system, however, means that if a module incorrectly parses an input data structure, for example, it will not compile. In this way we guarantee that if a Module compiles, then it must be compatible with the associated data description.

2.1. *The Task Programming Model*

The abstraction gained by treating modules as primitive components allows us to bring compiler-level features to bear on ROCI tasks. Specifically, the idea of type checking the input/output connections between modules has already been covered, but type checking the parameters that govern the behavior of these modules is also provided at the task level.

Individual module authors are able to decorate class-scope variable declaration statements with attributes that specify whether or not a variable is a startup parameter, or even if it is a control parameter that should be modifiable at run time. These attributes are extracted from compiled code, and are used by the ROCI to kernel to expose these variables when appropriate.

Variables marked as startup parameters will be displayed in the browser UI when a user wishes to start a task. Type checking is performed as the user enters new values for these parameters, thus making it far less likely that a module will start with invalid parameters. Furthermore, the type of the parameter can be used to intelligently populate the parameter-setting UI by dynamically creating UI elements such as drop-down boxes with only valid values as options, as opposed to a text field for every parameter. Variables marked as control parameters (dynamic over the course of execution) can be modified by another standard browser interface. A running module can be selected, and any variables marked as control parameters will populate a parameter-setting UI similar to the one described for startup parameters. This functionality, built atop the strong type system in .NET, provides a compiler-like layer of type checking at all phases of execution, while simultaneously making the UI used to interact with a ROCI deployment more intuitive for the end user.

2.2. *General Instrumentation*

The notion of task as program allows for varied interesting forms of system-level instrumentation and control. First, by sufficiently isolating individual modules such that they can be treated as atomic operations, we are able to treat tasks as programs built on a language that uses the specified modules as statements. Second, by virtue of its role as provider of

all inter-module communications, the ROCI kernel is capable of rich monitoring and control of all data transactions. These two points both deal with the notion of program flow control.

Program flow control is primarily controlled by the sequence of operations specified in the program. In our case, a schedule of modules makes up the procedural part of a task program. As described above, a task is a collection of concurrently running modules. The order in which these modules run is not explicitly defined, but instead is effectively governed by data dependencies between modules. In general, if module alpha uses data from module beta, then module alpha will block until that data is available, thus creating a very loose schedule in which each iteration of module alpha's processing loop is preceded by at least one iteration of module beta. There are no guarantees on the efficiency of this schedule; if only module alpha uses module beta's output, then it may be wasteful for module beta to run at a higher rate than alpha.

This issue is addressed by having a task schedule. The task schedule merely specifies a linear sequence of module iterations, but can be leveraged to obtain far greater efficiency that a schedule governed solely by dependency blocking. This schedule is specified in the task XML file as a sequence of module names. The names are checked when the task file is loaded to ensure that all statements in the schedule are defined module names. This schedule can be used simply to eliminate wasted iterations of data producers, but it can also be used to obtain non-obvious gains in overall program efficiency. A schedule can include a bias to run a particular module more frequently than another if it would give the task, taken as a whole, greater efficiency. Furthermore, since this schedule is not encoded in compiled code, it is fully dynamic. That is, a user or automated process can adjust a task's schedule at runtime to meet changing resource availability or execution priorities.

Such behavior is dependent on information. This information is made available by the instrumentation built into task schedules. The mechanisms that govern the execution of a ROCI task are in good position to monitor the iteration frequency of the task schedule in its entirety, and the resources being used by individual modules. This information can be used to raise alarms when a task frequency drops below a specified threshold, to throttle iteration frequency, or to modify the schedule to make better use of available resources. Furthermore, application specific efficacy metrics can be utilized by task monitoring modules to initiate new schedules to improve efficiency.

The distributed nature of ROCI deployments suggests a form of program flow throttling apart from the usual method of CPU resource allocation: network resource allocation. While the scheduling system can be used to monitor and control the rate at which a task schedule iterates, ROCI's pin system can throttle network communications on a connection-by-connection basis. Individual pin connections can be monitored to examine the type of data being transmitted, the frequency of transmissions, and the bandwidth used. Both the frequency of transmission and the overall bandwidth used are controllable by the ROCI kernel. This allows a controller, human or automated, to give network precedence to certain connections, potentially allowing greater system effectiveness with limited resources. Note that by throttling network communications, the speed at which a networked task runs can be controlled. Especially in a schedule-free execution environment, wherein a collection of modules have their iteration frequencies mediated by data dependencies, the throttling of individual connection bandwidth can be used to control the iteration frequency of individual modules. Thus there are two distinct methods of controlling performance in an on-demand fashion based on mediating CPU or network resource allocation.

2.3. *Logger Modules*

Our sensor database [12], [10] system is implemented on top of ROCI through the addition of logger modules. These logger modules can be attached to any output pin and record the outputs of that pin's owner module in a time-stamped log which can be accessed by external processes. These logger modules appear to the system as regular ROCI modules which means that they can be started and stopped dynamically and can be discovered by other ROCI nodes on the system. This last point is particularly salient since it means that robots can learn about the records available in other parts of the network at run time as those resources become available. Since logger modules can be attached to any output pin, there is no meaningful distinction between "low level" sensor data such as images returned by a camera module and "high level" information such as the output of a position

estimation module. Any data that is relevant to a task can easily be logged through the addition of a logger module.

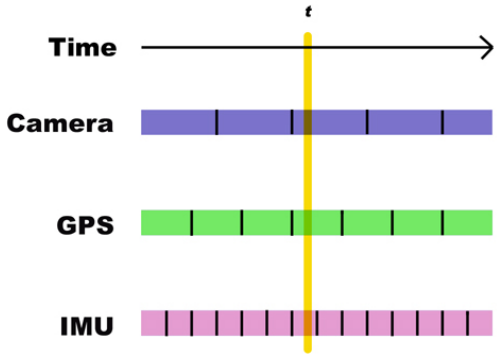


Figure 2. Time is a useful index for synchronizing data concurrently collected from multiple sources.

The generic logger module logs all incoming data based on time, an index relevant and meaningful regardless of the data type (Figure 2). Additional indexing methods that are specific to a particular data type are easily implemented by creating a new type of logger module that inherits from the general logger and is explicitly usable only with the expected data type. For example, a logger module that records the output of a GPS unit may also support efficient indexing based on position. Using time as a common key provides a simple mechanism for correlating information from different channels. Consider, for example, the problem of obtaining all of the images that a robot acquired from a particular position. This can be implemented efficiently by first indexing into the GPS log to find the times at which the robot was at that location and then using those times to index the image log to pull out the images taken from that vantage point. Using time as a common index also eliminates the need for a fixed database schema on the robots: different logger modules can be added or removed from a node as needed without having to perform complex surgery on a global table of sensor readings. Since the logger processes do not interact directly, they can be started and stopped, added and removed independently of each other.

2.4. *Query Processing*

Once a relevant data log has been found on the network, one must then face the problem of executing a query to extract information from that archive. It is often the case that the volume of data stored in a log makes it unattractive to transfer the data over the network for processing. In these situations we can take advantage of the fact that the facilities provided by ROCI can be used to support distributed query processing. Consider the example of a UAV that stores a log of images acquired as it flies over a site. If a process on a UGV wanted to access this data to search for particular targets in the scene, it would be impractical to transfer every image frame to the ground unit for processing. Here it makes sense to consider sending an active query to the UAV requesting it to process the images and send the target locations back to the UGV. This can be accomplished by developing a ROCI module that extracts the targets of interest from UAV imagery and then sending this module to the UAV as part of a query. The ROCI kernel on the UAV would then instantiate a task and use this module to process the data in the image log returning the results to the UGV.

Sophisticated queries that involve chaining together the results of many processing operations or combining information from several logs can be handled through precisely the same mechanism. The query takes the form of a network of ROCI modules that carry out various phases of the query. The modules in this task are distributed to appropriate nodes on the network and the final output is returned to the node that initiated the request. This approach allows us to dynamically distribute the computation throughout the network in order to make more efficient use of the limited communication bandwidth.

Another feature of this approach is that it promotes code re-use since the modules that are developed for carrying out various data processing and analysis operations online can also be used to implement queries on stored data logs (Figure 3). This is important not just by virtue of facilitating rapid development, but also by the robustness and familiarity users have with the component modules used in all aspects of a ROCI deployment. By making the same framework pervasive throughout the development pipeline, users are able to concentrate their efforts on improving core techniques because the code only needs to be written once. Once the code has been written, users setting up robot behaviors work from the same toolbox as those formulating queries at run time and throughout post-processing.

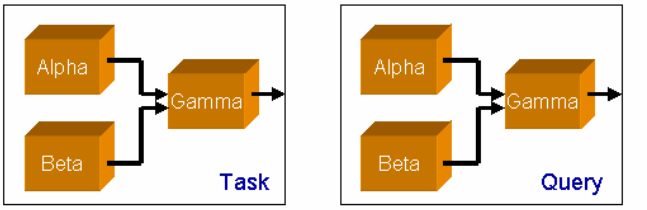


Figure 3. A ROCI query is, in many ways, very similar to a real-time task. In many cases, the inputs of the task come from live sensors, while the query gets data from logs. This distinction is transparent to the component modules.

The notion of query stages combined with the strong type system underlying ROCI module inputs and outputs immediately opens the door for a multitude of queries that make use of functionality already used by robot behaviors. For example, a robust localization routine may be run on all robots as they move around the environment. This routine must update relatively quickly to allow the robot to navigate in real-time, thus necessitating that it only consider readily available data. However, a user or autonomous agent may require an alternate estimation of a robot's location at a particular time in the past, perhaps utilizing newly acquired data. This can be achieved by designing a query wherein a localization routine, possibly another instance of the original routine, is connected to not only locally collected data, but also to any number of data processing routines, also specified by the query body, running on any number of other nodes. This localization may take a relatively long time to execute, and may not be suitable for real time control, but it is available to any programmed behavior or human operator that requests it. This query, while complex, automatically benefits from the shared toolbox provided by the consistent design framework. Processing modules that already exist on data hosts need not be transmitted, while others are downloaded from peers on an as-needed basis. The query itself is analogous to a behavior task: it specifies processing modules and how they connect. The ROCI kernel handles the work of ensuring that modules exist on the nodes that need them, and that those modules are properly connected.

By applying distributed database methods and techniques, the architecture presented here frees designers from having to create a static, all-encompassing communications scheme capable of satisfying a set of pre-specified query types. Instead, individual developers are able to utilize all sensor network resources in a modular, dynamic fashion through the use of active distributed database queries.

3. APPLICATIONS

ROCI technology is being used throughout the GRASP Lab to power a variety of robotics projects. The structure supported by ROCI facilitates the design of complex single-platform systems, high-performance real-time behaviors, and relatively simple static sensors. Projects such as the Smart Wheelchair utilize ROCI to organize and make sense of the data collected by dozens of sensors on a single mobile platform. Teams of small truck-like robots (Clodbusters) use ROCI for everything from collaborative error minimization to vision-based obstacle avoidance. Even a fixed camera becomes far more useful when plugged into a computer running ROCI. ROCI immediately provides logging capabilities as well as the ability to expose the camera's data stream to the network. Teams of ROCI-powered vehicles made up of Clodbusters, fixed wing UAVs, and an autonomous blimp have been successfully fielded in exploration and navigation experiments under adverse network conditions as part of the DARPA-funded MARS2020 program.

Current database-related work involves visualization and exploitation of data generated by a heterogeneous team of ground and air robots equipped with cameras, GPS receivers, IMU readers, altimeters and other sensors. For visualization purposes, this data can be fused in an on-demand fashion through visualization modules a human operator can interact with. In this way, one can quickly bring up images taken by a UAV flying over a particular location by joining a GPS log with an image log over a time index. Of note is what data is sent over the network to meet a particular demand. To minimize network usage, one might use a map location selected by the user to index into a GPS log to see when the robot was at the desired location, if it ever was. The resultant time indices can be used to index into the image database, thus avoiding the need to transfer unnecessary images.

An alternate formulation of this scenario that still maintains network efficiency, while improving usability, is to obtain the time indices of all images taken within some timeframe. These indices can be used to index into the GPS log to present the user with a map marked

up with the locations where pictures were taken. The user can select one of these locations, thus providing the database system with a time index to use in obtaining a particular image. This solution exploits the fact that both time indices and GPS data are far more compact than image data. The goal is to transmit as narrow a subset of the largest data log, in this case the image log, as possible. This setup is what is used at the GRASP Lab to intuitively scan data collected during a team operation.

A behavior-oriented application of the logging functionality can be found in a mobile target acquisition behavior. In this scenario, periodically placed overhead camera nodes log their image data which is made accessible to mobile robots when a network route to the camera node exists. Given a piece of code for visually identifying a target, a mobile robot can move to within routed radio range of overhead camera nodes and inject the target identification code as part of an image log query. The results of this query can simply be the time indices when the target was visible to the overhead camera. This information can be used to improve the efficacy of visual target searches – an extremely data-intensive process -- while minimizing the burden placed on the network. Under lab conditions, a two-node network, using a technology based on 802.11b ad-hoc networks, may be expected to manage 300KB/sec data transfer rates. This would mean that a single, uncompressed 1024x768 color image (2.25MB) would take over 7 seconds to transfer. While compression can greatly help, any resultant artifacts could cripple the effectiveness of a given processing algorithm. Regardless, a factor of 10 gained in compression is more than lost when faced with an image log of thousands of images. Compare this to the 20-50KB size of a typical ROCI module DLL, and it is clear that transferring the code rather than the data often presents considerable advantages.

4. EVALUATION

The primary benefit of ROCI is the development process it suggests. Developing high level applications from reusable, modular components is a well-understood concept, but one whose acceptance has faced real difficulties as popular programming technologies have not kept up with the requirements of modern design techniques. ROCI represents an attempt to push the field forward by taking full advantage of powerful hardware as well as relatively modern programming techniques such as object-oriented programming and strong type systems. By building consistent support for the type system into our high level framework we have successfully allowed loosely structured development teams to collaborate on large-scale projects with more reliable results than is usual. The task-module-pin design structure encourages engineers without strong computer science backgrounds to contribute to larger projects without having to worry about their lack of understanding of the underlying system. Most developers concentrate on the specifics of what their module does, not how it fits into a larger system, or how any of the internal mechanisms – such as scheduling, communications, or user interface – work.

4.1 *Related Work*

Similar systems exist for other application scenarios. TinyOS is an open-source effort to provide OS-level support for sensor platforms with extremely limited hardware. In fact, the fundamental design concepts of TinyOS and ROCI have much in common, primarily the encouraging of modular software design [1]. However, TinyOS specifically targets limited hardware platforms, which imposes limits on what can be attempted with it. We have chosen to target much more capable hardware – we use consumer-level laptop computers on many of our robots – and we are therefore able to distance ourselves from many of the difficulties faced by the mote programmer.

Distributed computing infrastructures that target more powerful hardware can also be found. Several Grid computing efforts are making large strides towards harnessing the computational power of thousands of computers over the Internet [3]. These efforts tend to be of a much more general slant than what we have undertaken. We have found that by focusing on the needs of our developers, we are better able to define constraints on the development process that significantly improve reliability. While the event-based pin communications infrastructure ROCI employs works well for sensor platforms exchanging data, it is not necessarily optimal for all computing needs. Further, we do not provide any tools for automating the distribution of a single computation over a very large network.

Sun's Jini system for Java is an architecture that attempts to bridge the gap between embedded systems and services running on general purpose computers [2]. While this system boasts many of the same benefits as ROCI, we feel that it requires somewhat more effort on the part of the developer to make use of. The simplest ROCI deployments involve minimal usage of ROCI API calls. There is a template module that an author fills out, and then each pin connection in a task is specified in one line of XML. This type of deployment is an example of how the ROCI kernel is designed to handle most common usage patterns with minimal developer action.

4.2 *Final Words*

The ROCI system is evidence that a strong type system paired with solid software design fundamentals can yield substantial improvements in software reliability, reuse, and ease of use. While still primarily used in robotics efforts, projects that seek to stretch ROCI design methods in new directions, such as limited hardware devices and schedule optimization, are now underway. By defining the ROCI kernel itself in a modular fashion with well-defined interfaces, we are able to extend the offered functionality, usually without breaking backwards compatibility. This extensibility, both in terms of novel task-level applications and kernel extensions, is a validation of the design methods presented above.

5. REFERENCES

[1] Philip Levis, Sam Madden, David Gay, Joe Polastre, Robert Szewczyk, Alec Woo, Eric Brewer and David Culler, "The Emergence of Networking Abstractions and Techniques in TinyOS," Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004).

[2] Jim Waldo, "The Jini Architecture for Network-centric Computing," *Communications of the ACM*, pp. 76-82, July 1999.

[3] R. Bramley, K. Chiu, S. Diwan, D. Gannon, M. Govindaraju, N. Mukhi, B. Temko, and M. Yechuri, "A Component Based Services Architecture for Building Distributed Applications," In Proc. 9th IEEE International Symposium on High Performance Distributed Computing, Pittsburgh, PA, Aug. 2000.

[4] R. Alur, A. Das, J. Esposito, R. Fierro, G. Grudic, Y. Hur, V. Kumar, I. Lee, J. Ostrowski, G. Pappas, B. Southall, J. Spletzer, and C. Taylor, "A framework and architecture for multirobot coordination," In D. Rus and S. Singh, editors, *Experimental Robotics VII, LNCIS 271*. Springer Verlag, 2001.

[5] Sarah Bergbreiter and K.S.J. Pister, "Cotsbots: An off-the-shelf platform for distributed robotics," In *IROS*, pp. 1632, October 2003.

[6] Luiz Chaimowicz, Anthony Cowley, Vito Sabella, and Camillo J. Taylor, "Roci: A distributed framework for multi-robot perception and control," In *IROS*, pp. 266, 2003.

[7] A. Das, J. Spletzer, V. Kumar, and C. J. Taylor, "Ad hoc networks for localization and control," Proceedings of the IEEE Conference on Decision and Control, 2002.

[8] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of Supercomputer Applications*, 15(3), 2001.

[9] A. Cowley, H. Hsu and C.J. Taylor, "Distributed Sensor Databases for Multi-Robot Teams," Proceedings of the 2004 IEEE Conference on Robotics and Automation (ICRA), April 2004.

[10] Joseph M. Hellerstein, Wei Hong, Samuel Madden, and Kyle Stanek, "Beyond average: Towards sophisticated sensing with queries," In *Information Processing in Sensor Networks*, March 2003.

[11] D. MacKenzie, R. Arkin, and J. Cameron, "Multiagent mission specification and execution," *Autonomous Robots*, 4(1): pp. 29-52, 1997.

[12] Samuel R. Madden, Michael J. Franklin, Joseph Hellerstein, and Wei Hong, "The design of an acquisitional query processor for sensor networks," In *SIGMOD*, June 2003.

[13] D. Martin, A. Cheyer, and D. Moran, "The open agent architecture: a framework for building distributed software systems," *Applied Artificial Intelligence*, 13(1/2): pp. 91-128, 1999.

Evaluating Reliability of Motion Features in Surveillance Videos

Longin Jan Latecki[1], Roland Miezianko[1], Dragoljub Pokrajac[2]
[1]Temple University, CIS Dept., Philadelphia, PA, latecki@temple.edu, rmiezian@temple.edu
[2]Delaware State University, CIS Dept., Dover, DE, dpokraja@desu.edu

ABSTRACT

Although a tremendous effort has been made to perform a reliable analysis of images and videos in the past fifty years, the reality is that one cannot rely 100% on the analysis results. The only exception is applications in controlled environments as dealt in machine vision, where closed world assumptions apply. However, in general, one has to deal with an open world, which means that content of images may significantly change, and it seems impossible to predict all possible changes. For example, in the context of surveillance videos, the light conditions may suddenly fluctuate in parts of images only, video compression or transmission artifacts may occur, a wind may cause a stationary camera to tremble, and so on. The problem is that video analysis has to be performed in order to detect content changes, but such analysis may be unreliable due to the changes, and thus fail to detect the changes and lead to "vicious cycle".

The solution pursuit in this paper is to monitor the reliability of the computed features by analyzing their general properties. We consider statistical properties of feature value distributions as well as temporal properties. Our main strategy is to estimate the feature properties when the features are reliable computed, so that any set of features that does not have these properties is detected as being unreliable. This way we do not perform any direct content analysis, but instead perform analysis of feature properties related to their reliability.

The main effort in video analysis nowadays is still in making the feature computation more reliable. Our claim is that we need to accept the fact that the computed features will never be 100% reliable, and focus our attention on computing reliability measures. This way system decisions will only be made when features are sufficiently reliable. This means for an intelligent system for video analysis that in addition to feature computation, it should perform instantaneous evaluation of their reliability, and adapt its behavior in accordance to the reliability. For example, if the goal of a system is to monitor motion activity, and to signal an alarm if the activity is high, the system is allowed to make reliable decisions only if there exist a subset of the computed motion activity features that is sufficiently reliable. The monitoring of features reliability and adjusting the system behavior accordingly, seems to be the only way to deploy autonomous video surveillance systems.

1. INTRODUCTION

We begin by showing two examples of video content changes that cause the existing motion detection approaches to inaccurately detect the presence of substantial motion. Clearly, the detected motion is present in videos, but it is due to some content artifacts and is not due to the actual presence of moving objects. Consequently, human observant ignores such "motion" as irrelevant, while standard video analysis systems detect it as significant activity. We will show that the feature reliability methods proposed in this paper allow us to identify the unreliable motion features, and to ignore the irrelevant artifacts. This is possible without reducing the detection rate of real moving objects. Consequently, we eliminate false alarms without reducing the detection rate. We stress that this is obtained without any direct video content analysis (e.g., using different features), but by monitoring the reliability of computed features. As stated in the introduction, direct video content analysis with further features does not solve the problem, since these features may also become unreliable.

Our first example illustrates motion artifacts in *Campus 3*[1] video introduced by some reflections in windows that are probably caused by cars passing by. In Fig. 1, we show two frames from Campus 3 video, one showing real motion, and the second showing the motion artifacts in addition to the normal motion. Our second example, in Fig. 2(a), shows motion artifacts introduced by video compression. The same scene without such artifacts is shown in Fig. 2(b). This video, which we call *Temple 2*, was recorded in real-world environment by the video surveillance system of the Campus Police Division of the Temple University.

In Section 2, we first describe a simple temporal method to determine the reliability of motion detection. In Section 3, we present a more sophisticated statistical method based on distribution analysis of feature values and information theory [21]. Both methods monitor features computed by our motion detection approach presented in [17], which we summarize in Section 4. The motion features are computed for gray level or infrared videos using 3D spatiotemporal blocks of spatial size 8x8 pixels, and temporal size of 3 frames. The blocks are disjoint in space and overlap by one frame in time. As result we obtain *motion activity* values for each 8x8 block in every video frame. By thresholding the motion activity values, we

[1] *Campus 3* can be obtained from the Performance Evaluation of Tracking and Surveillance (PETS) repository:
ftp://pets.rdg.ac.uk/PETS2002//DATASET1/TESTING/CAMERA3_JPEGS/

obtain a binary feature, called *motion detection*, with 1 standing for 'motion detected' and 0 for 'no motion detected'. Both videos as well as our motion detection results can be viewed on [12].

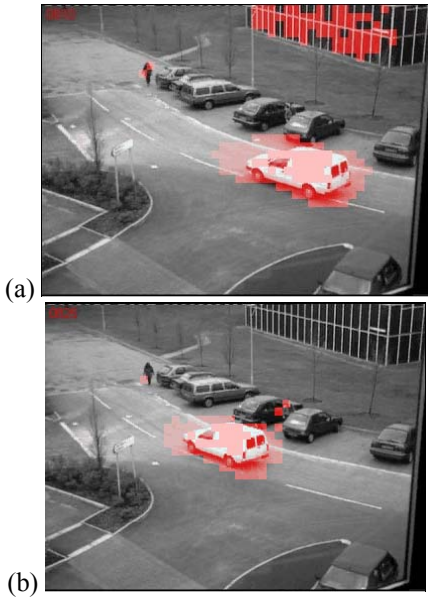


Figure 1. Two frames from *Campus 3* video with moving blocks highlighted red: (a) motion artifacts due to reflections in the windows, (b) the same scene (a few frames later) without the artifacts.

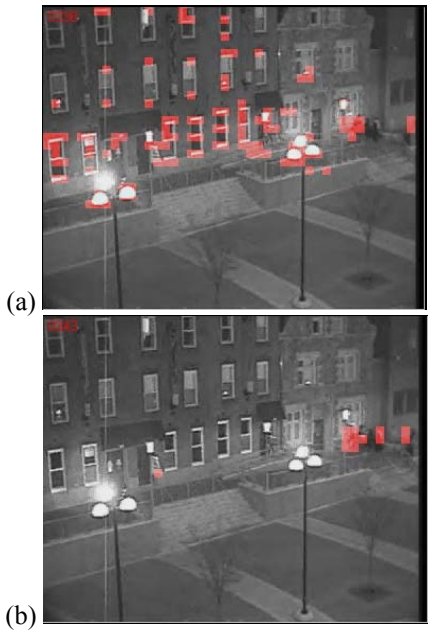


Figure 2. Two frames from *Temple 2* video with moving blocks highlighted red: (a) motion artifacts introduced by video compression, (b) the same scene (a few frames later) without the artifacts.

A temporal reliability analysis introduced in Section 2 is applied to the motion detection feature, while a statistical reliability analysis introduced in Section 3 is applied to the motion activity feature.

2. TEMPORAL ANALYSIS OF FEATURE RELIABILITY

In this section, we describe a simple temporal method to determine the reliability of motion features. The input motion feature has binary values for each 8x8 block for each video frame with 1 for 'motion detected' and 0 for 'no motion detected'. The algorithm described in Section 4.2 computes this feature vector. The 8x8 blocks are disjoint. Let $f(n)$ be the number of 1s in the frame number n, i.e., $f(n)$ is the number of detected moving blocks as function of frame number. We use the finite difference approximation of first derivative of f to monitor the reliability of our motion detection. In simple words, if the jump in values of f is above a certain threshold for a given time interval, the binary feature is unreliable in this interval. The threshold necessary to detect the unreliable features is not static. We propose a dynamic thresholding algorithm described in Section 2.1 to learn and vary this threshold. However, some other learning techniques could also be used.

This reliability property works under the assumption that there exists an upper bound on the size of moving objects whose motion we want to detect (measured in the number of moving blocks). This assumption holds for most surveillance videos. Now we consider an example video, called *Temple 2,* that satisfies this assumption. This video is recorded by a roof mounted, stationary camera, so that a certain minimal distance to moving objects is guaranteed. Typical moving objects there, humans and vehicles, cannot get arbitrarily large. Hence, the fraction of the scene occupied by a moving object is limited. Observe that the actual value of the upper bound on the size of moving objects needs not to be known, since our algorithm learns it automatically.

In Fig. 3(a), we see the graph of function f for *Temple 2* video. Time intervals with significant jumps of f that are correctly identified by our dynamic thresholding are marked with red lines in Fig. 3(b). The graph of modified feature f, when f was set to zero within the time intervals when motion detection was detected as unreliable is shown in Fig. 3(c). Fig. 3(c) shows that the proposed method is able to identify and exclude the unreliable results of motion detection. By excluding these time intervals from further processing, we not only eliminate false alarms, but make possible to correctly detect alarm situations, although the input motion detection in not 100% reliable. For example, a significant increase in the number of motion blocks after the frame 1700 indicates an alarm situation. This is a correct prediction, since a street fight is recorded on the video after the frame 1700, see the *Temple 2* video [12].

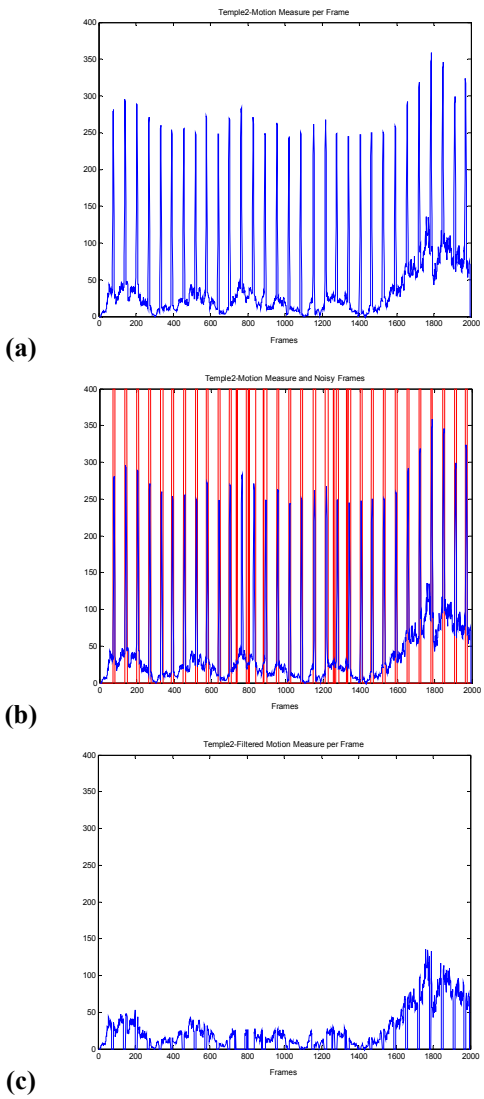


Figure 3. (a) The graph of *f(n)*, which is the number of moving blocks as function of frame number *n*. (b) Significant jumps of *f* (caused by feature unreliability) correctly identified by our dynamic thresholding. (c) The graph of *f* padded by zeros for frames with unreliable motion detection.

2.1. DYNAMIC THRESHOLDING ALGORITHM

Now we describe a dynamic thresholding algorithm used to detect the jumps of function *f*. First we compute the initial values of mean *meanl* and standard deviation *stdl* using all previous values of *f*(*x*) for *x*=1, ..., *t*-1 and some time instance *t*. The actual dynamic thresholding starts at time *x*=*t*. A jump up is detected at points $x \in \{x+1, x+2, .., x+w\}$ for a window size *w* if

$$meanrw(x) - meanl(x) > C1*stdl(x),$$

where *C1* is a constant. A dynamic threshold values *meanl* and *stdl* are updated if

$$meanrw(x) - meanl(x) < C2*stdl(x),$$

where *C2* < *C1* is a second constant. The updated values are:

$$meanl(x) = u*meanl(x) + (1-u)*meanrw(x)$$

$$stdl(x) = u*stdl(x) + (1-u)*stdrw(x)$$

where *u* is a learning constant and

$$meanrw(x) = \frac{1}{w}\sum_{\tau=1}^{w} f(x+\tau)$$

$$stdrw(x) = \sqrt{\frac{1}{w-1}\sum_{\tau=1}^{w}\left(d(x+\tau) - meanrw(x)\right)^2}\,.$$

The symmetric window constant *w* was set to 3, giving us a sliding window of 7 frames (2**w*+1). The learning constant was *u*=0.9. Constants *C1*, *C2* of function *f* used in detecting jumps of the *Temple 2* video were selected based on the initial running average *meanl* and *stdl*. The value of *meanl* was 10.3 and the value of *stdl* was 7.4. Constant *C1* was set to 15 and constant *C2* was set to 3 providing the initial jump detected threshold to 154.5 and reset to no-jump detected threshold of 30.9.

3. STATISTICAL ANALYSIS OF FEATURE RELIABILITY

To determine whether a particular feature is reliable, we assume that the feature bears more information if its distribution differs more significantly from a normal (Gaussian) distribution. Similar heuristics are used e.g., in Independent Components Analysis [20]. The follow-up assumption is that the feature becomes unreliable if an addition random noise is superimposed, which would lead the distribution of such noisy feature to become more Gaussian like. Hence, by measuring to what extent a feature distribution differs from a Gaussian distribution, one can not only get information to what extent the feature is useful but also when such usefulness drops (e.g., due to some external and often non-observed factor).

The Information Theory proposes *negentropy* as the measure of this discrepancy. Given a probability density *p*(*x*) of a feature, Differential Entropy is defined [18, 19] as:

$$H(x) = \int_{-\infty}^{\infty} -p(x)\log p(x)dx \qquad (1)$$

For a given class of distributions *p*(*x*) that have the same variance σ^2, differential entropy is maximal for a Gaussian distribution where it is equal

$$H_{Gauss}(\sigma^2) = \frac{1}{2}\log 2\pi e\sigma^2\,. \qquad (2)$$

Hence, a negentropy, which defined as

$$J(x) = H_{Gauss}(\sigma^2) - H(x) \qquad (3)$$

or its normalized value

$$J(x)/H_{Gauss}(\sigma^2) = 1 - \frac{H(x)}{H_{Gauss}(\sigma^2)} \quad (4)$$

may be used to measure usefulness and reliability of a feature. Observe that the minimal value of negentropy is 0 (when $p(x)$ is Gaussian).

A naïve approach to compute negentropy would be to employ histograms to approximate $p(x)$ with piecewise linear function p'(x) such that:

$$p'(x) = Kp(x_i), x \in [x_i, x_i + \Delta x]$$

where K is a normalization constant (chosen such that $p'(x)$ is a distribution). However, as shown in [21] this non-parametric technique is very unstable since dependent on a proper choice of a histogram bin size Δx and histogram centers x_i. Hence we use parametric approach suggested in Hyvarinen's NIPS 1997 paper [18]. The main ideas of this approach are:

1) Instead of original feature x, use a standardized feature $x^*=(x\text{-}mean(x))/\text{std}(x)$ that have zero mean and unit standard deviation. This way, we could directly use negentropy to compare reliability with no need to normalize with the entropy of a Gaussian.

2) Use a first-order Taylor approximation of a logarithmic function in eq. (1) that leads to: $(1+\varepsilon)\log(1+\varepsilon) \approx \varepsilon + \varepsilon^2/2$;

3) Use conveniently chosen set of orthogonal functions of $G_i(x)$ of a feature x to expand probability density function $p(x)$ in vicinity of a Gaussian probability density.

In practice, the choice of orthogonal functions is based on practicability and sensitivity on outliers of the computation of estimates for expectations $E(G_i(x))$, integrability of the obtained probability density function approximation and last, but not the least, the properties of non-Gaussian distributions we want to capture.

Based on such consideration, [18] proposes the following two approximations of negentropy, that we use in this paper:

$$J_a(x^*) = k_1 E\left(x^* \cdot e^{-x^{*2}/2}\right)^2 + k_{2a}\left(E(|x^*|) - \sqrt{\frac{2}{\pi}}\right)^2 \quad (5a)$$

$$J_b(x^*) = k_1 E\left(x^* \cdot e^{-x^{*2}/2}\right)^2 + k_{2b}\left(E(e^{-x^{*2}/2}) - \sqrt{\frac{1}{2}}\right)^2 \quad (5b)$$

where the coefficients are determined as:

$$k_1 = \frac{36}{8\sqrt{3}-9}, k_{2a} = \frac{24}{2-\frac{6}{\pi}}, k_{2b} = \frac{24}{16\sqrt{3}-27}. \quad (6)$$

The proposed technique is applicable on any continuous feature. In this paper, we evaluate the reliability of the *motion activity* feature, defined in [17] (see Section 4) as the largest eigenvalue of texture vectors in a small time window. For each frame, we standardize the feature values x^*, compute expectations $E(|x^*|)$, $E\left(x^* \cdot e^{-x^{*2}/2}\right)^2$ and finally compute the negentropy approximations eq. (5a), (5b) per frame.

We evaluated the proposed techniques for assessing feature reliability on a set of videos [12]. This set includes infrared videos, for which the same settings of parameters as for visual light videos were used. Here we focus on our results on two video sequences from the Performance Evaluation of Tracking and Surveillance (PETS) repository: a sequence from PETS2001[2] here referred to as *Campus 1* sequence, a sequence from PETS2002, here referred to as the *Campus 3* sequence and on a *Temple 2* sequence from Temple University.

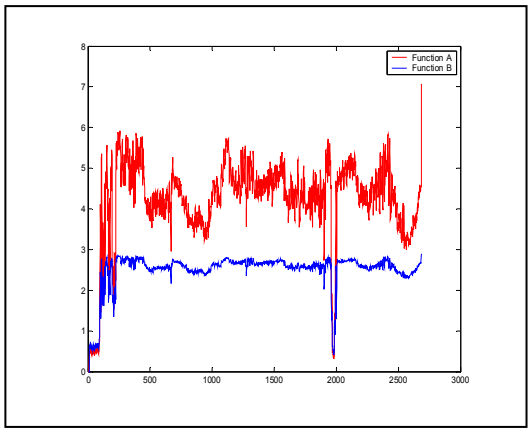
Campus 1. At the beginning of the sequence, there is no movement, so changes in the motion activity (an observed feature) are random, which reflects small negentropy values in approximately first 100 frames, see Fig 4(a). Both negentropy approximations (eq. 5a, 5b) demonstrate strong drop between frames 1960 and 2000 which corresponds to the higher level of noise that can be visually observed between these frames. Function B (eq. 5b) provides more stable approximation values, which makes it potentially more useful.

Campus 3. Both methods identified drop around frames 330, 660, a strong drop around 700, a drop around 720 and the relatively long-term drop between 800 and 900, see Fig. 4(b). Finally, there were some small oscillations between 1200 and 1300 and one drop around 1400. All these events correspond to frames in the video sequence when our algorithm has difficulties in properly identifying moving objects based on observed feature (e.g., due to reflections in the upper right part of the frame, cp. Fig. 1). Again Function B (eq. 5b) performed better, by having less oscillations and fluctuations.

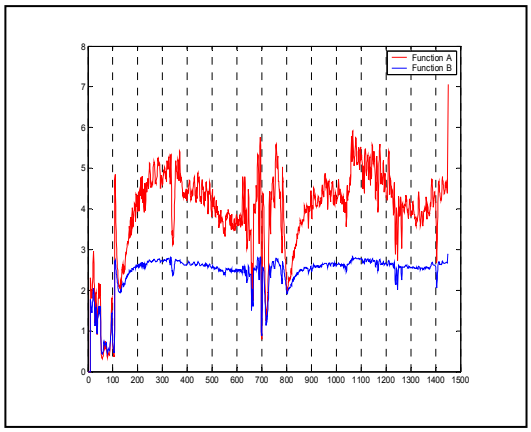
Temple 2. On this video, there is evident instability (manifested as flicker) that can be traced to applied compression technique. The period of this disturbance, which has negative effects on motion detection, is around 62 frames. Using the proposed technique, we obtained negentropy values that reflect this periodicity. Both functions eq. (5a) and (5b) have strong periodical components, see Fig. 4(c), and demonstrate oscillations which period can be correctly determined using a Fast Fourier Transform [22], as approximately 62 frames. Function (5b) is again more stable and provides better automatic period estimation. The results of the statistical method agree with those of the temporal methods, cp. Fig. 3.

A common denominator of the results shown is that the proposed negentropy-based technique can help in determining frames when the observed feature is unreliable (periodic or pulse flicker, noise, etc.). Since both eq. 5a and 5b are only relatively rough approximations of negentropy, there is no wonder they do not provide the same values, especially when a negentropy is relatively high. As expected, when a negentropy is low, the feature probability distribution is closer to a Gaussian so both approximations would give similar results. Generally, eq. (5b) provides better performance. It is more stable and has less fluctuations. Hence is potentially more suitable for automatic thresholding.

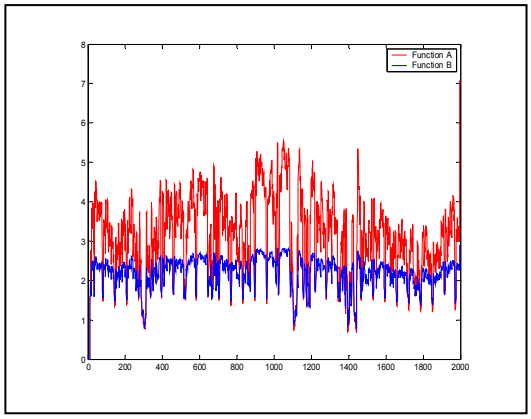
[2] ftp://pets.rdg.ac.uk/PETS2001/DATASET1/TESTING/CAMERA1_JPEGS/



(a)



(b)



(c)

Figure 4. Estimated negentropy per frame of each video using (eq. 5a) in red and (eq. 5b) in blue for (a) *Campus 1*; (b) *Campus 3*; (c) *Temple 2* videos.

4. FEATURE GENERATION AND MOTION DETECTION

We shortly describe our motion detection method proposed in [17]. It is based on change analysis of texture vectors computed for 3D, spatiotemporal (sp) blocks. In our previous paper [11] we have shown that the use of sp texture vectors of 3D blocks in the framework of Stauffer and Grimson [14] can improve the detection of moving objects while potentially cutting back the processing time due to the reduction of the number of input vectors per frame. Our experimental results in [17] (videos be viewed on [12]) show that our motion detection technique leads to further performance improvements.

4.1 *Video representation with spatiotemporal (sp) texture vectors*

We represent videos as three-dimensional (3D) arrays of gray level or monochromatic infrared pixel values $g_{i,j,t}$ at a time instant t and a pixel location i, j. A video is characterized by temporal dimension Z corresponding to the number of frames, and by two spatial dimensions, characterizing number of pixels in horizontal and vertical direction of each frame.

We divide each image in a video sequence into disjoint $N_{BLOCK}\times N_{BLOCK}$ squares (e.g., 8x8 squares) that cover the whole image. Spatiotemporal (3D) blocks are obtained by combining squares in consecutive frames at the same video plane location. In our experiments, we used 8x8x3 blocks that are disjoint in space but overlap in time, i.e., two blocks at the same spatial location at times t and t+1 have one square in common.

The fact that the 3D blocks overlap in time allows us to perform successful motion detection in videos with very low time frequency, e.g., in our experimental results [12] videos with 2 fps (frame per second) are included. The obtained 3D blocks are represented as 192-dimensional vectors of gray level or monochromatic infrared pixel values. We then zero mean these vectors and project them to three dimensions using principal component analysis (PCA). The obtained 3-dimensional vectors form a compact spatiotemporal texture representation for each block. The PCA projection matrices are computed separately for all video plane locations (a set of disjoint 8x8 squares in our experiments).

The blocks are represented by N-dimensional vectors $\mathbf{b}_{I,J,t}$, specified by spatial indexes (I,J) and time instant t. Vectors $\mathbf{b}_{I,J,t}$ contain all values $g_{i,j,t}$ of pixels in the corresponding 3D block.

To reduce dimensionality of $\mathbf{b}_{I,J,t}$ while preserving information to the maximal possible extent, we compute a projection of the normalized block vector to a vector of a significantly lower length $K<<N$ using a PCA projection matrix $\boldsymbol{P}^{K}_{I,J}$ computed for all $\mathbf{b}_{I,J,t}$ at video plane location (I,J). The resulting sp texture vectors $\mathbf{b}^{*}_{I,J,t} = \boldsymbol{P}^{K}_{I,J}*\mathbf{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos

and are used as input of algorithms for detection of moving objects. We used K=3 in our experiments.

To compute $\boldsymbol{P}^K_{I,J}$ we employ the principal values decomposition following [4,5]. A matrix of all normalized block vectors $\mathbf{b}_{I,J,t}$ at video plane location (I,J) is used to compute the $N\times N$ dimensional covariance matrix $\boldsymbol{S}_{I,J}$. The PCA projection matrix $\boldsymbol{P}_{I,J}$ for spatial location (I,J) is computed from the $\boldsymbol{S}_{I,J}$ covariance matrix. The projection matrix $\boldsymbol{P}_{I,J}$ of size $N\times N$ represents N principal components. By taking only the principal components that corresponds to the K largest eigenvalues, we obtain $\boldsymbol{P}^K_{I,J}$.

4.2 *Moving objects detection based on local variation*

The assumption of the proposed technique is that the variation of location vectors—corresponding to the same location within a small number of consecutive frames— will increase if the vectors correspond to a moving object. In practice, for each location (x,y), we consider vectors

$$b^*_{x,y,t-W}, b^*_{x,y,t-W+1}, \ldots, b^*_{x,y,t}, \ldots, b^*_{x,y,t+W}$$

corresponding to a symmetric window of size $2W+1$ around the temporal instant t. For these vectors, we compute the covariance matrix $\mathbf{C}_{x,y,t}$. We assign the largest eigenvalue of $\mathbf{C}_{x,y,t}$, denoted as $\Lambda_{x,y,t}$, to a given spatiotemporal video position to define a local *variance measure,* which we will also refer to as *motion activity*

$ma(x,y,t) = \Lambda_{x,y,t}$.

The larger the variance measure $ma(x,y,t)$, the more likely is the presence of a moving object at position (x,y,t). Finally, we label each video position as moving or stationary (background) depending whether the motion activity is larger or smaller than a suitably defined threshold. We use a dynamic thresholding algorithm (described in Section 2) to determine the threshold value at position *(x,y,t)* based on the history of $ma(x,y,s)$ values over time (s=1, ..., t-1).

5. CONCLUSIONS

In this paper, we proposed and evaluated two methods to monitor the reliability of features applied in video surveillance and motion detection. The methods have been evaluated on real-life surveillance videos. Both methods correctly identified time intervals when an observed feature becomes non useful for motion detection (e.g., due to flicker, artifacts introduced by compression algorithm, etc.). The proposed methodology is potentially applicable to other domains where unsupervised learning is performed under open-world assumption (where we cannot anticipate all the events which could occur during the operational life of an automated intelligent system).

6. ACKNOWLEDGEMENTS

D. Pokrajac has been partially supported by NIH-funded Delaware Biomedical Research Infrastructure Network (BRIN) Grant (P20 RR16472), and DoD HBCU/MI Infrastructure Support Program (45395-MA-ISP Department of Army).

7. REFERENCES

[1] Buttler, D., Sridharan, S., and Bove, V. M. Real-time adaptive background segmentation. *In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, Baltimore 2003.

[2] R.T. Collins, A.J. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance", *IEEE PAMI* 22(8) (2000), pp. 745–746.

[3] Devore, J. L., *Probability and Statistics for Engineering and the Sciences*, 5th edn., Int. Thomson Publishing Company, Belmont, 2000.

[4] Duda, R., P. Hart, and D. Stork, *Pattern Classification,* 2nd edn., John Wiley & Sons, 2001.

[5] Flury, B. A *First Course in Multivariate Statistics*, Springer Verlag, 1997.

[6] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-Time Surveillance of People and Their Activities", *IEEE PAMI* 22(8) (2000), pp. 809–830.

[7] Jain, R., Militzer, D., and Nagel, H. Separating nonstationary from stationary scene components in a sequence of real world TV images. In Proc. IJCAI, 612–618, Cambridge, MA, 1977

[8] Jolliffe, I. T, *Principal Component Analysis*, 2nd edn., Springer Verlag, 2002.

[9] Javed, O., Shafique, K., and Shah, M. A. Hierarchical approach to robust background subtraction using color and gradient information. In Proc. IEEE Workshop on Motion and Video Computing (MOTION), 22-27, Orlando, 2002,.

[10] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions", *IEEE PAMI* 22(8) (2000), pp. 831–843.

[11] D. Pokrajac and L. J. Latecki: Spatiotemporal Blocks-Based Moving Objects Identification and Tracking, *IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, October 2003.

[12] R. Miezianko, L. J. Latecki, D. Pokrajac. Link to test results. *http://knight.cis.temple.edu/~video/VA*

[13] Remagnino, P., G. A. Jones, N. Paragios, and C. S. Regazzoni, eds., *Video-Based Surveillance Systems*, Kluwer Academic Publishers, 2002.

[14] C. Stauffer, W. E. L. Grimson, "Learning patterns of activity using real-time tracking", *IEEE PAMI* 22(8) (2000), pp. 747–757.

[15] Westwater, R., Furht, B., *Real-Time Video Compression: Techniques and Algorithms*, Kluwer Academic Publishers, 1997.

[16] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-time Tracking of the Human Body", *IEEE PAMI* 19(7) (1997), pp. 780–785.

[17] L. J. Latecki, R. Miezianko, and D. Pokrajac. Motion Detection Based on Local Variation of Spatiotemporal Texture. *CVPR Workshop on Object Tracking and*

Classification Beyond the Visible Spectrum (OTCBVS), Washington, July 2004.
[18]A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, volume 10, pages 273-279. MIT Press, 1998.
[19]T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
[20] A. Hyvärinen, J. Karhunen, and E. Oja. Independent Component Analysis. Wiley, 2001.
[21] D. Pokrajac and L. J. Latecki. Entropy-Based Approach for Detecting Feature Reliability. *Invited Paper, 48th Conf. for Electronics, Telecommunications, Computers, Automation, and Nuclear Engineering* (*ETRAN)*. Cacak, Serbia, June 2004.
[22] E. Oran Brigham. *The Fast Fourier Transform: An Introduction to Its Theory and Application*. Prentice Hall, 1973.

A Framework for Measuring Tracking Performance of a Sensor Network

K Madhava Krishna Henry Hexmoor
Dept. of Computer Science and Computer Engineering
University of Arkansas
Fayetteville, AR 72701
mkrishna@uark.edu hexmoor@uark.edu

Abstract: *A framework and analysis for a distributed sensor network based surveillance system is presented here. In a previous effort [1] we have presented methodologies for coordination and decision-making amongst sensors for tracking targets while in [2] we presented the results of endowing the sensor network with autonomy. Sensors monitor targets that crisscross a rectangular surveillance zone. When a sensor pursuits a target it leaves areas unguarded through which other targets can get past undetected. In this paper we presents a methodology that computes the tracking time for a sensor that guarantee detection of a required fraction of the targets expected to crisscross its home area to an arbitrary probabilistic threshold. The home area of the sensor is the area guarded by it when it is stationed at its home position, its default position when it is not in pursuit of a target. Simulations are presented that corroborate with the probabilistic model developed and hence vindicate its correctness. A framework for extending the probabilistic model to a system where multiple sensors guard the same area is also presented.*

1. Introduction

This paper describes a methodology that guarantees probabilistic completeness for sensors that track targets in a multi-sensor setting. Each sensor guards in its default stationary state an area called the home area of the sensor. For a sensor s_j, its home area is denoted by H_j. The robots perform surveillance over a rectangular (square) surveillance zone. The surveillance zone is divided into number of square shaped cells as shown in figure 1 for the sake of modeling. The figure shows the sensors placed in their home positions. The radius of vision of the sensor equals the length of the diagonal of the cell. However the sensor only considers those targets that lie within its four neighboring cells as targets within its field of vision. This area representing its field of vision in its home position is also called as the home area of that sensor for the remaining of this paper. The home area of each sensor is depicted by thick squares. The simulation environment used for testing our strategies has been developed through *Borland's JBuilder IDE* on the Java platform.

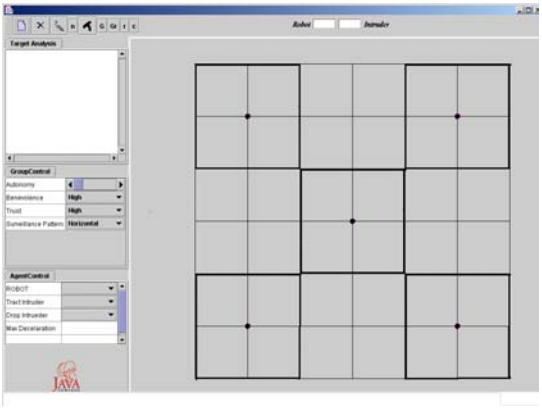


Figure 1: The rectangular surveillance zone with sensors depicted as circles ensconced in their home positions. The home area of each sensor is denoted by thick squares

A sensor allocates itself to one of the targets within its field of vision through a resource allocation process modeled through fuzzy rules [1]. The sensor further decides if it would monitor the target by remaining stationary or by pursuing (tracking) it. When a sensor tracks a target it leaves areas in its home position unguarded. The tracking time for a sensor s_j, denoted by t_j represents the time for which the target would be away from its home position H_j. The tracking time can be modulated based on the fraction of the number of targets that a sensor is expected to detect within a probabilistic threshold. If $\hat{N}_T$ denote the number of targets expected to

crisscross H_j within a temporal window Γ and f be the lower bound on the fraction of the number of targets, $\hat{N}_\Gamma$, to be detected and Ω represent the threshold the paper presents a framework for computing $t_j = g\left(\Gamma, f, \hat{N}_T, \Omega\right)$. Here g is a function that guarantees that at-least a fraction f of the targets, N_Γ, are detected to a probabilistic guarantee of Ω. In other words $g : P\left(n \geq f \cdot \hat{N}_\Gamma\right) \geq \Omega$, where P is the probability computation over the random variable n that denotes the number of targets detected.

The rest of the paper is organized as follows. Section 2 presents the current work in the context of similar works found in the literature. Section 3 presents the formulation of the methodology and section 4 depicts the efficacy of the methodology in simulations. Section 5 extends the formulation to an environment where multiple sensors are placed with the responsibility of guarding the same home area. Section 6 concludes and provides further scope of this work.

2. Background Review

The problem of multi sensor surveillance involves detection of multiple intrusions and/or tracking through coordination between the sensors. Detection and target tracking has been researched from multiple viewpoints. Some efforts have focused on the problem of identifying targets from a given set of data through particle filters [3, and probabilistic methods [4]. The problem of data association or assigning sensor measurements to the corresponding targets were tackled by Joint Probabilistic Data Association Filters by the same researchers such as in [4]. Kluge and others [5] use dynamic timestamps for tracking multiple targets. Krishna and Kalra [6] presented clustering based approaches for target detection and further extended it to tracking and avoidance in [7]. The focus of these approaches has been on building reliable estimators and trackers. They do not use distributed sensors and are not directly useful for the problem of large area surveillance.

In the context of distributed task allocation and sensor coordination Parker [8] proposed a scheme for delegating and withdrawing robots to and from targets through the ALLIANCE architecture. The protocol for allocation was one based on "impatience" of the robot towards a target while the withdrawal was based on "acquiescence". Jung and Sukhatme [9] present a strategy for tracking multiple intruders through a distributed mobile sensor network. Lesser's group have made significant advances to the area of distributed sensor networks [10] and sensor management [11]. In [9] robots are distributed across a region using density estimates in a manner that facilitates maximal tracking of targets in that region. The decision for a robot to move to another region or to stay in its current region is based on certain heuristics. The authors of this paper present their scheme for resource allocation and coordination in a distributed sensor system through a set of fuzzy rules in [1] and further analyze the behavior of system by varying the autonomy of the sensors in [2].

In none of the above efforts is a strategy for guaranteeing some form of completeness is presented. This paper is unique from other efforts in multi-sensor systems in that it offers a tracking strategy for sensors that modulates their tracking time such that a required number of targets are detected within probabilistic guarantees. The authors in [12] present a framework that provides for meeting a targeted search time within probabilistic guarantees for a cooperated UAV search. However the computations and the basis for their framework is disparate from this effort and is presented for a different application and motivation.

3. The Methodology

Targets are modeled percolating in a Poisson fashion at the rate λ through each cell, which has one of its edges on the boundary or the perimeter of the surveillance area. For a surveillance zone such as in figure 1 consisting of six cells along each side of the perimeter, the rate of entry is 6λ. λ is fixed at 0.1 for all the examples discussed in this paper. Then the apparent rate at which each sensor would see a target, λ_{SJ} provided it is stationary is given by the following approximation:

$\lambda_{SJ} = \lambda \sum_{k=1}^{P} \frac{\theta_k}{\pi}$, where, θ_k is the angle subtended from the point where the target enters the boundary at the home position of sensor sj.

Since the entry points of the arriving targets are not known a-priori, θ_k is computed assuming that the target infiltrates at the midpoint of the cell edge that coincides with the perimeter of the zone. In the figure below (figure 2) the targets are assumed to enter at points p1, p2, p3, … along the perimeter of the surveillance zone. For the sensor centered at 'b', the angle subtended by the target entering at p4 is shown marked θ. This angle covers the span of all the targets that will cross the region of surveillance of the sensor at 'b' by targets entering at p4. The total span of the angle for a target entering at all those points is π radians or in other words all targets that enter the surveillance zone have to necessarily be within a span of π radians from the point of entry for them to be within the surveillance zone.

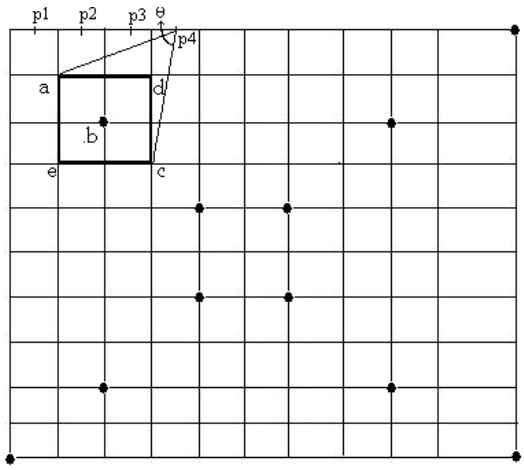


Figure 2: Targets are modeled as entering at locations p1, p2, …, p4 at the midpoints of the cell edges that coincides with the perimeter of the zone.

Let t_j be the time for which a sensor sj is away from its home position in pursuit of a target as mentioned in section 1. We compute the apparent time T_a (the time for which a target that would have been in the field of vision of sj had sj been stationary at its home position perceives sj to be away) as:

$$T_a = \begin{cases} 2\int_0^{t_j/2} \frac{t}{T_\kappa} dt \quad ; t_j \le 2T_\kappa \\ 2\int_0^{T_\kappa} \frac{t}{T_\kappa} dt + 2\left(t - T_\kappa\right)_{T_\kappa}^{t_j} ; t_j > 2T_\kappa \end{cases} \quad (1)$$

Here, κ is the fraction of the home area left unguarded by sj as it moves away from its home.

The upper limit of the integral T_κ denotes the time at which the sensor leaves its original area completely unguarded. If T_{esc} represents the average time for which a target stays in the home area of a sensor, the probability that a target is detected by a sensor is given by:

$$p = \frac{T_{esc}}{T_a}. \quad (2)$$

Let n be the random variable denoting the number of detections made over a temporal window Γ as before. The probability of detecting exactly k of the N_Γ targets expected to arrive in Γ is given by the familiar binomial distribution:

$$P\left(n = k / X = \hat{N}_\Gamma\right) = \binom{\hat{N}_\Gamma}{k} p^k \left(1-p\right)^{\hat{N}_\Gamma - k}$$

Here X is a Poisson random variable that measures the number of targets arriving. The resultant probability of detecting k such targets then becomes

$$P\left(n = k\right) = P\left(n = k / X = \hat{N}_\Gamma\right) \cdot P\left(X = \hat{N}_\Gamma\right)$$

It can be shown that the above resultant probability once again has a poisson distribution with parameter $\lambda_{SJ} p$. Hence the probability of detecting k targets has the representation

$$P\left(n = k\right) = e^{-\lambda_{SJ} p} \frac{\left(\lambda_{SJ} p\right)^k}{k!} \quad (3).$$

The tracking time t_j is eventually computed by making use of equations (1), (2) and (3) and that which would satisfy the following guarantee condition:

$$P\left(n = k \ge f \cdot \hat{N}_t\right) \ge \Omega \quad (4)$$

Since double precision arithmetic does not allow computation of factorials beyond 20 the normal approximation to poisson distribution is used in our computations.

4. Simulation Results

The first objective is to evaluate empirically the validity of equation (2) that ascertains the probability of detecting a target by a sensor while it is in motion. For this purpose a single cell environment with one sensor such as in figure 4 is

considered. Targets are introduced in poisson fashion at the midpoints of the four boundaries of the cell. The sensor's home position is at the center of the cell. The wandering time t_j is calculated for a given value of Ω and f. The sensor is away from its home position for t_j units. The sensor does not track a particular target. It is merely away from its home position. The number of targets that crisscrossed the cell during this time interval and the number of those detected were recorded.

The results are tabulated in table 1. The first column represents the desired fraction of the targets that need to be detected and the second the minimum probabilistic threshold of detecting the fraction. The third column is the actual fraction of the targets detected averaged over twenty runs. The fourth column signifies the relative frequency of times a fraction greater than or equal to the desired fraction was detected. The fourth column is then a means of evaluating whether the desired probabilistic threshold was obtained. If the desired fraction to be obtained is 0.6 and thirteen times out of twenty a fraction more than 0.6 was detected, the entry in the last column of the table is 0.65 that signifies the required performance was met. It is seen that the average obtained fraction is above the desired fraction whenever the minimum probabilistic threshold is high indicating that the desired fraction was detected in most of the runs as entailed by the threshold. The average fraction obtained is lesser than the desired fraction when the desired probabilistic threshold is low. This is indeed expected as a low desired threshold indicates that the sensor is entailed to detect the desired fraction of targets only in a few of those twenty runs. The relative frequencies in the fourth column also do not fall significantly below the desired minimum probabilistic threshold in any of the runs. That the relative frequency is within $5-10\%$ of the desired threshold in all the runs validates the probability definition of equation (2) and the computation of apparent time in equation (1).

Desired minimum fraction (f)	Desired minimum probabilistic threshold (Ω)	Obtained average fraction (20 runs)	Obtained relative frequency (20 runs)
0.9	0.9	0.91	0.95
0.9	0.1	0.23	0.05
0.6	0.9	0.73	0.85
0.6	0.1	0.38	0.2
0.4	0.6	0.42	0.65
0.4	0.4	0.32	0.3
0.1	0.9	0.28	1
0.1	0.1	0.08	0.2

Table 1: Tabulation of the results obtained for the environment of figure 4 for different desired fraction and threshold values.

In simulations T_{esc} is computed as the average of the minimum and maximum a target is within the home area of a sensor. The minimum time is the time taken by the target to traverse half the edge of a cell. The maximum time is the time taken to traverse from midpoint of a side to the farthest opposite vertex of the cell. Similarly T_κ is the average of the minimum and maximum time taken by the sensor to result in its home area becoming completely unguarded. The details of computing T_κ are omitted here for brevity.

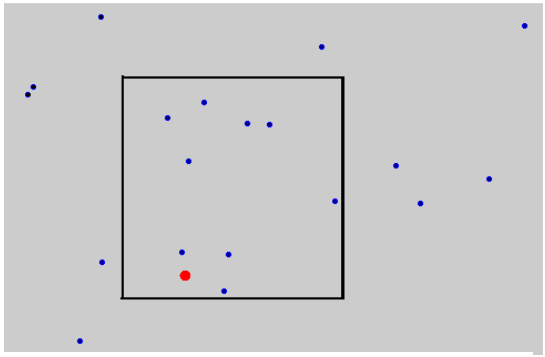


Figure 4: A single cell environment used for validating the definitions of equations (2) and (1). The bigger circle denotes the sensor and the smaller circles the crisscrossing targets introduced in poisson fashion.

The framework developed in section 2 is now tested for an environment shown in figure 1 with multiple sensors. Each sensor tracks targets such that the probabilistic guarantee is maintained with respect to its home area. The overall quality of track (*QoT)* at the end of a simulation interval Γ is defined by:

$$QoT = \frac{1}{N_\Gamma}\sum_{i=1}^{N_\Gamma}\frac{d_i(\Gamma)}{c_i(\Gamma)} \qquad (5)$$

Here N_Γ is the actual number of targets that crossed the surveillance zone in Γ. The

numerator d_i indicates the number of sensors that detected the target i in Γ while the denominator, c_i denotes the number of sensor home areas that the target i had got past in Γ. Essentially QoT is an average measure on whether a target that crossed the home area of a sensor was detected by that sensor with the difference that the QoT would also reflect cases of targets that are detected by sensors whose home area it did not cross elsewhere in the surveillance zone. In the summation of (5) if the fraction $\frac{d_i}{c_i}$ exceeds unity it is clipped to unity. This is done since a fraction greater than unity tends to offset for fractions less than unity and does not reflect those cases. A stricter definition of QoT that specifically captures the number of targets missed by a sensor that crossed its home area is given by: $QoT = \frac{1}{N_S}\sum_{j=1}^{N_S}\frac{n_{dj}}{n_{cj}}$ (6).

Here N_S denotes the total number of sensors in the environment, n_{dj} denotes the number of targets detected by the sensor among those that had visited its home area H_j, while n_{cj} is the number of targets that had been through H_j. Thus $\frac{n_d}{n_c}$ represents the fraction of the targets that entered a sensor's H_j and were detected by it and can never exceed unity. However we found that at the end of the simulation interval the values as computed by (5) and (6) vary only marginally. Hence in this paper the results of QoT are those computed as in (5).

In the simulations that follow a sensor leaves its environment in pursuit of a target. Sensors can reallocate themselves to other targets during the course of a track as dictated by the resource allocation strategy. The time t_j for a sensor is updated after every fixed number of samples based on the fraction of the targets that were detected thus far and the fraction that need to be detected in the remaining time window to meet the objective of (4).

Figure 5 shows a snapshot of a simulation run. The bigger circles represent the sensors and the smaller circles the targets. The dashed rectangles enclosing the sensor and target identify the sensor-target pair (the target to which the sensor has allocated itself to). It is to be noted while a sensor tracks a target it also detects all other targets within its field of vision. Currently the problems of data association and target occlusion are not considered.

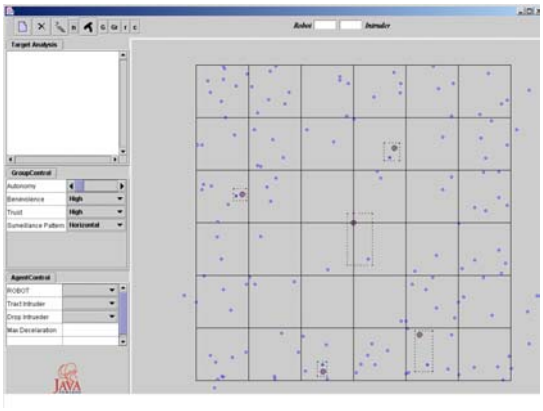


Figure 5: A snapshot of a simulation run. The dotted rectangles enclosing a sensor-target pair indicate the target to which that sensor is currently allocated to. Sensors are shown by larger circles while targets are depicted smaller

4.1 Analysis:

Figure 5a shows two graphs that plot f_r, the fraction remaining and wander time, t_j, along the y-axis. In both the graphs the abscissa denotes the time in samples. Sampling measurements on f_r, t_j are done once in every ten cycles of a simulation run. The total number of simulation cycles is 150 or in other words $\Gamma = 150$ in these simulations. Each cycle is repeated every 500ms. The graphs cover the entire simulation run of 150 cycles or 15 samples of measurements. The plot of figure 5b depicts QoT on y-axis and sample time on x. Both graphs 5a and 5b are for a simulation run with parameter $f = 0.8, \Omega = 0.75$. The graphs of figure 5a are for one of the sensors of figure 5 only the graph of 5b depicts QoT of the entire system. Graphs in figure 5c and 5d have the same connotations as in figures 5a and 5b except that they are for parameters $f = 0.8, \Omega = 0.3$. The horizontal dashed line in the graph of 5b and 5d indicate the desired

fraction of target fraction of targets, f, expected to be detected at the end of the simulation. Since QoT as defined in (6) computes the fraction of targets detected averaged across all the sensors in the system, the horizontal line serves as an indicator if the QoT was achieved or not.

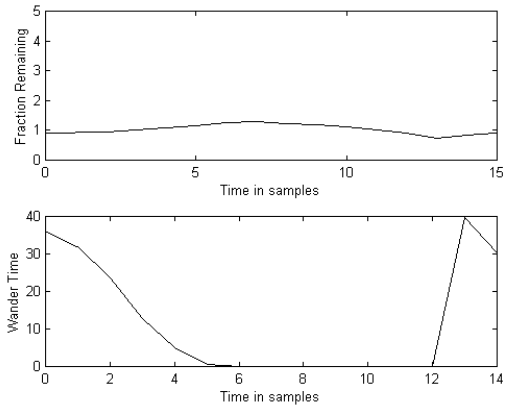


Figure 5a: The top graph shows f_r plotted against sample time while the bottom graph is a plot of t_j against samples. The plots are for a simulation such as in figure 5 run with $f = 0.8, \Omega = 0.75$

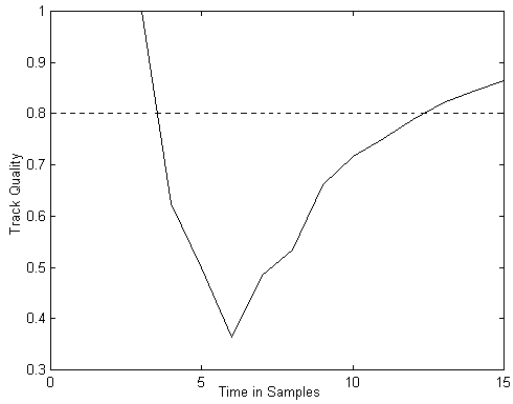


Figure 5b: A plot of track quality QoT. The dashed horizontal line denotes the desired fraction of the total targets that need to be detected.

For $\Omega = 0.75$ the track time is modulated such that the desired fraction f averaged over all sensors is detected at the end of a simulation run for majority of such runs. Figure 5b corresponds to one such run where the track quality at the end of the simulation is 0.85 and is above the expected criterion of 0.8 and lies above the horizontal. Figure 5d corresponds where the QoT at the end of the simulation does not achieve the desired fraction. This is expected for a run with $\Omega = 0.3$ where most of the runs are not required to detect a fraction greater than 0.8.

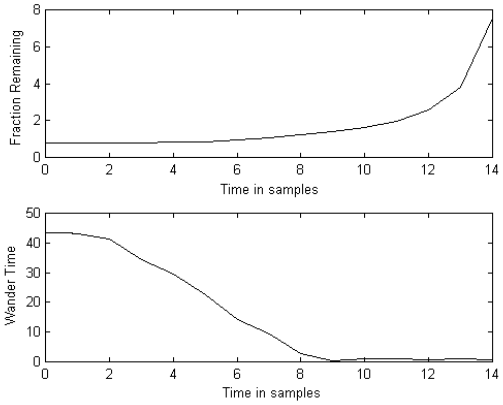


Figure 5c: Graph same as figure 5a for parameters $f = 0.8, \Omega = 0.3$

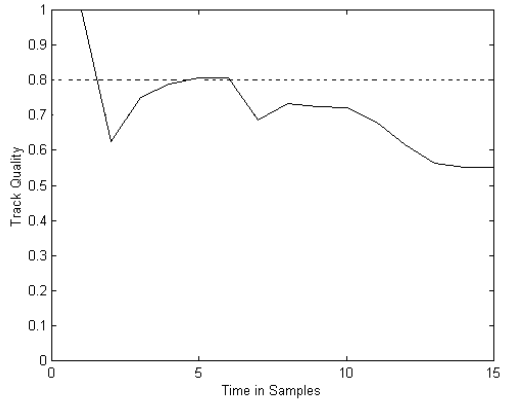


Figure 5d: Graph with same connotations as figure 5b for parameters $f = 0.8, \Omega = 0.3$. At the end of simulation time the track quality is below the horizontal line indicating that performance criterion was not met

For a marginal increase in f_r in the top plot of figure 5a the corresponding decrease in t_j is steeper in figure 5a when compared with figure 5c. The decrease in wander time t_j as f_r is less steep in 5c than in figure 5a. For a given f the variations in t_j are due to the variations in Ω. A higher Ω entails that the sensor cannot move too far away from its home due to lower values of t_j. As the sensor moves away from its home and misses targets the required remaining fraction to be detected f_r may tend to increase. In such a case the decrease in wander time also tends to be steeper for a similar increase in f_r for a higher

Ω. A steeper increase translates as quicker returns to home by the sensor to detect more targets.

6 Extension to Multi-sensor Surveillance

The benefits of having more than one sensor guard the same home area is now considered. For example let each of the home area in figure 1 (shown by thick squares) be guarded by 3 sensors. One such area is shown in figure 6.

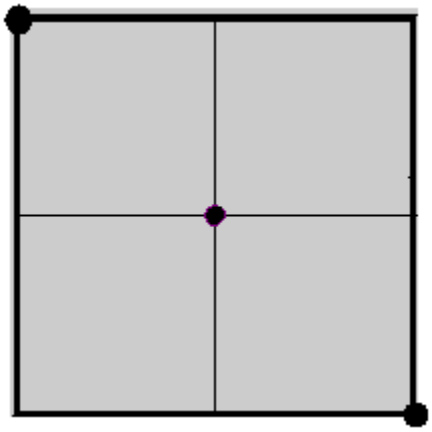


Figure: An area guarded by three sensors, one at the center and two at the corners

It can be shown that T_κ, the time taken by the sensor to leave the area completely unguarded is the same on an average for all the three sensors. Let the sensors be labeled as A, B and C. The probability that sensor A detects a target is denoted as p_A and has the same form as (2). Similarly the individual probabilities of detection for sensors B and C are denoted as p_B and p_C. Since T_κ and hence T_{app} are the same for the three sensors $p_A = p_B = p_C$. The probability that at-least one of the sensors detect a target is given by $p = p_A \cup p_B \cup p_C$. Determining p from equations (3) and (4) and along with the condition $p_A = p_B = p_C$ leads to the following cubic in the individual probability of a sensor:

$$p_A^3 - 3p_A^2 + 3p_A - p = 0 \qquad (7).$$

The solution to the cubic solves the individual probability of a sensor detecting a target.

Figure 7a shows two graphs. The top graph depicts the individual probability of a sensor detecting a target for a fixed Ω (here 0.7) and varying f. The lower graph plots t_j versus f for a constant Ω (here 0.7). Each of these graphs shows two plots. The plot with a dashed line corresponds to the case where a single sensor guards an area. The plot with solid line corresponds to the case where multiple sensors guard an area. The graphs indicate that the individual probability of detection is consistently lower when multiple sensors guard an area than a single sensor guarding and the wander time is correspondingly high for a multi-sensor case. Hence a sensor can wander away from its home area for a longer duration when if there are more than one guarding its home area. In [2] we have shown longer tracking time enhances performance criteria based on median and mean number of detections of targets.

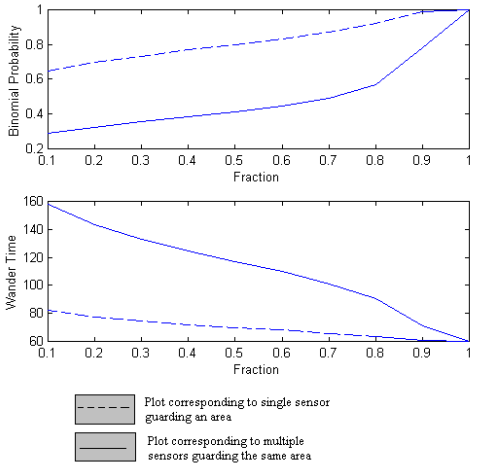


Figure 7a: Plots of individual probability and wander time against varying values of desired fraction f. Solid lines are plots corresponding to a single sensor while dashed lines correspond to multi sensor case

7 Conclusions:

A framework that provides for probabilistic guarantees for a multi-sensor based multi-target tracking system is provided here. Sensors modulate their tracking time based on the desired fraction of targets that need to be detected and the minimum probabilistic threshold of detection. Simulation results corroborate the efficacy of the formulation of the scheme for probabilistic guarantees. Extension of the scheme to multiple sensors guarding the same area enables longer tracking time for a sensor and hence better

performance based on mean and median number of detections.

References:

[1] K Madhava Krishna, H Hexmoor and S Pasupuleti, "A Surveillance System Based on Multiple Mobile Sensors", *to appear in Proceedings of FLAIRS,* 2004, Special Track in Sensor Fusion.

[2] K Madhava Krishna, H Hexmoor and S Pasupuleti, "Role of Autonomy in a Distributed Sensor Network", *to appear in Proceedings of ICAI,* 2004.

[3] D Schulz; W Burgard, D Fox and A Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association", *IEEE International Conference on Robotics and Automation*, 1165-1170, 2001

[4] D Schulz and W Burgard, "Probabilistic state estimation of dynamic objects with a moving mobile robot", *Robotics and Autonomous Systems,* 2001.

[5] B. Kluge, C Kohler and E Prassler, "Fast and robust tracking of multiple objects through a laser range finder", *IEEE International Conference on Robotics and Automation*, 1165-1170, 2001

[6] K Madhava Krishna and P K Kalra, "When does the robot perceive a dynamic object", *Journal of Robotic Systems*, 19(2), 2002

[7] K Madhava Krishna and P K Kalra, "Detection tracking and avoidance of multiple dynamic objects", *Journal of Intelligent and Robotic Systems*, 33(4): 371-408, 2002

[8] L Parker, "Cooperative robotics for multi-target observation", *Intelligent Automation and Soft Computing,* 5[1]:5-19, 1999

[9] B Jung and G.S. Sukhatme, "Multi-Target Tracking using a Mobile Sensor Network", Proc., *IEEE Intl. Conf. On Robotics and Automation,* 2002

[10] B Horling, R Vincent, R Miller, J Shen, R Becker, K Rawlins, and V Lesser, "Distributed Sensor Network for Real Time Tracking", In *Proceedings of the 5th International Conference on Autonomous Agents*:. 417-424, 2001.

[11] B Horling, R Miller, M Sims, and V Lesser, "Using and Maintaining Organization in a Large-Scale Distributed Sensor Network", In *Proceedings of the Workshop on Autonomy, Delegation, and Control, (AAMAS 2003)*.

[12] P Vincent and I Rubin, "A Framework and Analysis for Cooperative Search Using UAV Swarms", *Proceedings of SAC'04*, March 14-17, Nicosia, Cyprus

Goal-Oriented Intelligence in Optimization of Distributed Parameter Systems

Shlomo Orr, MRDS, Inc. (PERMIS, 2004)

Abstract

Models of complex systems can be differentiated by their ability to reproduce or generate system behavior, by their prediction power, by their robustness, or, conversely, by their sensitivity to inputs and parameters; by their uncertainty (if captured); and by their intelligence. Even the term "prediction" is not unique. First, a first-principle (physically based) distributed parameter model could be an excellent predictor if (a) it captures the main system behavior, and (b) its parameters and inputs are known accurately; otherwise, it would fail, possibly drastically. Second, predictive power depends on the data, on the goal, and on the time scale. For example, scheduling of pumping and injection in an oilfield for maximum profit over the next 5 years; or pumping from a contaminated aquifer in order to maintain certain (low) concentration at a compliance point for the next 20 years, vs. prediction of plume migration in groundwater towards a nearby river, over time: in each case, the model has a slightly different expected function, as well as different intelligence type. The paper reviews the recent developments in subsurface fluid flow management such as optimization of oil production and groundwater remediation (both sharing similar practices, though for different purposes) as a continuous struggle to increase intelligence by (a) adapting new tools such as artificial intelligence and dynamic stochastic control; (b) attempting to integrate these tools; and (c) reducing uncertainty. Although the systems discussed seem specific to the (mathematical) geosciences (specifically to oil reservoirs and contaminated aquifers), and although these systems are very different from man-made machines, similar rigid structure and reliance on differential-integral calculus, as well as the serial processing, knowledge evolution, and uncertainty propagation from one discipline to the next exist in most science and engineering fields, and so does the need for a paradigm shift. Given the need in adaptive, intelligent control/planning/optimization of such systems, the progress of these segregated efforts towards a multiresolutional decision support system is inevitable, highly desirable, and highly promising. However, we are still facing the challenge of performing and defining optimal integration between PDE models and multiresolutional representations, and since such integration depends on model quality and data quantity and quality, this is an adaptive integration process as well. One criterion for such "optimal" integration would be uncertainty reduction (resulting from the integrated MR system), which brings up yet another challenge: to define the metrics for uncertainty reduction.

1. Introduction

Management of distributed-parameter systems, particularly where complex natural processes intersect with human industrial practice and theoretical knowledge, is extremely difficult to analyze and optimize. From a knowledge organization perspective, this area of practical knowledge is in disarray. For example, knowledge in the petroleum industry relates to many scales of representation, yet this fact is not taken into account in an organized manner.

Indeed, the oil industry is a live example of the need to integrate multiple disciplines presently not integrated, including huge volumes of raw data (particularly in geophysics) and multiple computational and conceptual models of geology, geophysics, and reservoir dynamics. Although the need in unification of the bodies of knowledge in these disciplines and the associated (tremendous) benefits of analysis enhancement capability has been widely recognized, progress in this direction has been very slow. The main reason for this is the rigid, segregated, serial process of knowledge and model building – from geophysical signals to simulation-optimization of fluid dynamics in porous media – a process that has evolved naturally during the last century, where the simulator has become the focal junction where all the knowledge and understanding of the physical processes and material properties are being filtered and concentrated in the form of partial differential equations (PDE that describe mass balance of oil, water, and gas in 3D space) whose coefficients (or distributed parameters) suppose to capture the physical-chemical properties of the medium on a particular scale, everywhere in the modeled subsurface/reservoir domain, assuming some "known" boundary conditions and initial conditions. In the following, we will use subsurface flow systems, particularly, oil reservoirs, as examples of managed, distributed parameter systems.

Oil reservoirs are complex systems on all scales. Decisions such as pumping and injection (schedule and rates), new well placement, and (directional) drilling in an active oil field, are typical of the complex relationships between reservoir characterization and oil field/reservoir management. The solutions to such problems involve a complex system of multiphase flow equations (linked PDE) in a heterogeneous domain (reservoir), as well as economical factors such as short-term and long-term oil price, worth of information, inventory/storage/delivery, cost of drilling, maintenance, production, etc. Well drilling and construction equipment are costly and cannot be afforded as frequently as necessary, while prevalent information gaps render decision-making uncertain and hence, risky.

Yet, the advantages of reservoir simulations should not be underestimated either. Models or simulators based on PDE solutions provide physical insight into various important flow phenomena, as well as the general behavior of the fluid movements in the reservoirs, under scarcity of spatial data typical of both old and young reservoirs, particularly the latter. The ability to capture the essence of the complex physics behind the reservoir responses to pumping and injection is the strength of the simulator and the essence of its intelligence. However, this strength could promptly become its weakness where (a) natural geological heterogeneities on certain scales are being missed, or (b) physical/chemical/thermodynamic processes are being missed (e.g., leaching geochemistry; instability of the oil-water interface), or (c) uncaptured (or erroneous) boundary conditions, all of which would lead to wrong predictions. In other words, wherever an essential physical phenomenon on any scale is being missed, the simulator

looses its intelligence, i.e., its ability to predict short-term reservoir responses and long-term oil reserves and revenues.

The main question we attempt to address in this paper is how to merge the advantages and use the intelligence of existing models and interpretations in a comprehensive intelligent system that could take advantage of such physically based intelligence, while eliminating its limitations. In order to answer this question, we first need to understand the structure and limitations of current approaches to optimization/control of subsurface fluid flow and solute transport. The reader could notice that although the problems discussed here seem to be specific to optimization of oil production and groundwater remediation, similar rigid structure and reliance on differential-integral calculus exist in most engineering fields, and so is the need for a paradigm shift when planning/control/optimization become the focus. Yet, we should also keep in mind some major difference between the geosciences (or natural systems in general) and man-made machines. In hydrogeology and oil reservoirs, we deal with multiphase flow in heterogeneous formations, with transient flow and transport phenomena occurring on all scales, with nonlocal dependency on (unknown) fluid flow everywhere. However, whenever we focus on operations' scheduling, feedback and feed-forward, this complex system becomes similar to other complex operations; finding a new well location, however, requires considering the complete heterarchy of transient flow phenomena in space, often with major data gaps. Such data gaps challenge all models, and consequently, any integration of PDE models with MR knowledge representations (and/or MRDS). Such integration seems to be data-dependent, and requires optimization on its own merit, weighting the robustness, prediction/anticipation (goal-oriented) power, the uncertainty associated with different representations, as well as uncertainty reduction produced by such integration.

2. Current approaches to simulation-optimization-control of distributed-parameter, subsurface flow systems

The cutting edge subsurface fluid management such as oil reservoir optimization and groundwater remediation control under uncertainty has been moving in three major fronts: (1) operations research (including stochastic models and risk assessment); (2) stochastic-dynamic control; and (3) artificial intelligence (AI), particularly artificial neural networks (ANN), genetic algorithms (GA), and fuzzy logic (FL). However, under the current structure of serial, segregated, and isolated "disciplines" that process the information from geophysics/explorations to reservoir characterization, reservoir simulations (or flow and transport models in hydrogeology), and optimization/control, it is impossible for these three fronts to merge into a unified, integrated approach, nor could a major progress in oil field management be made. Under the current paradigm, optimization/planning/control of these complex systems have been handicapped by uncertainty on one hand, and prohibitive computer power & time requirements on the other hand, without benefiting from all available information.

The typical approach to reservoir characterization and management sketched in Figure 1 shows the different subsystems that constitute both exploration and production. The figure is highly simplified, with many subsystems not shown, such as subdivision of

exploration (remote sensing, surface- and borehole-geophysics; seismic, electrical, electromagnetic, micro-gravity, SP, etc); geologic investigations (structural geology, geochemistry, lithology, bio-stratigraphy); subdivision of production (well pattern design, injection and pumping, gas, water, and displacing fluids); and well construction, - all of which are complex, interdependent, and require real time updating and decision making. The figure illustrates current model construction and subsequent optimization.

Blocks 1 and 2, and, to some extent, Block 3 (conceptual model), represent reservoir characterization, which plays a crucial role in exploration and subsequent reservoir management. Typically, the conceptual model (Block 3) of the reservoir is an undeclared part of the simulator; this is where all the geology is filtered, upscaled, and translated into the simulator's parameters, which inherently entails averaging and discarding of information (acting as a low-band filter), including small scale features that may be crucial (in which case, their large-scale influences would be modeled as different, large-scale parameters – e.g., *dispersion coefficient*). Most of the assumptions and decisions related to reservoir representation are made at the conceptual model stage, and are subject to modeler's understanding and experience.

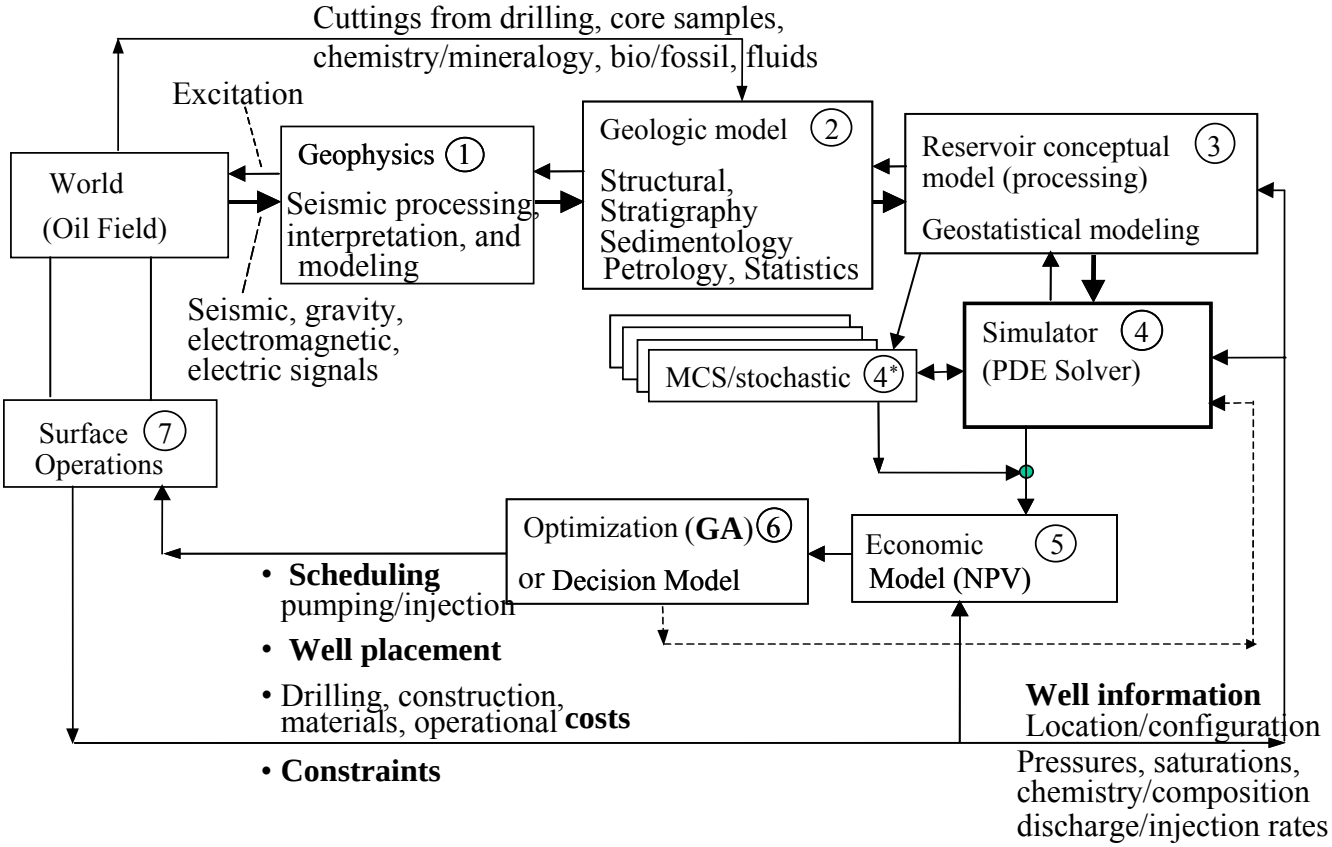


Figure 1. A schematic view of current approaches to reservoir optimization.

Block 4 is the current quantitative "brain" (or predictor), typically a distributed-parameter, complex PDE solver that may include several linked PDE with their auxiliary constitutive functions (mostly determined in the lab, on a lab-scale), or less commonly, a cell model, represented by Ordinary Differential Equations (ODE), implying a larger

scale, i.e., lower resolution). Within this computational block, additional analytical forms are utilized for computing parameters and constitutive relationships, as well as local modeling (e.g., the Buckley-Leverett model of displacement). Ideally, the simulator contains multiple analytical models functioning at different time scales, and demonstrates processes working at different levels of resolution. However, current models are far from this ideal.

Due to the embedded upscaling and loss of information, feedback from the reservoir simulator (4) to the geological model (2) is not reliable, except for special cases where certain disparity exists between the measuring window and the scale of the geologic feature, and where only a piece of the puzzle is missing (e.g., in well testing with an appropriate monitoring system in place, or in dual porosity systems where the rock properties are well characterized on all scales). Typical inverse or automatic calibration procedures determine some local reservoir-fluid interaction "properties" that fit a particular (and hence, uncertain) model. Subsequent interpretations of geologic features based on inverse modeling (or calibration of the simulator) are, therefore, speculative.

2.1 Reservoir Simulations and Groundwater Modeling Problems

Historically, modeling of fluids flow in porous media using PDE started by Muskat in the 1930's [52, 53]. Until that point, predictions of reservoir behavior over time was merely extrapolations from a (local) "production curves" or "well performance models", which describe cause-effect relationships between production, fluid content (oil, water, gas), and pressures in producing and injection wells, over time, using curve fitting/regression [66]. The introduction of PDE suddenly provided insight and extended the prediction power significantly, providing intelligence to an otherwise a black box model. This addition of intelligence has had a sweeping effect, and was extended to groundwater and multiphase flows in porous media, including geothermal reservoirs, unsaturated flow in soils, and contaminant transport in aquifers (e.g., [9]), while it has been further reinforced by a rapid development of numerical methods and ever-increasing computer power. The success of the numerical model that could explain and predict the subsidence of Venice in the early 70's [84-86] has been used as a live example of the ultimate intelligence of this new tool. By the early 80's, reservoir engineers and hydrogeologists have developed numerous numerical models (or simulators), which, with the help of new visualization tools and mainframe computers, could both predict and visualize the movement of oil, gas, water, and contaminants (in groundwater), with ultimate confidence and optimism due to the exponential growth of computer power and the prospect of optimal management. However, before long, it has become clear that once applied to geological formations on a scale where the heterogeneity cannot be neglected, the strength of the PDE-based model becomes its weakness; this sensitivity, overlooked and even welcomed initially (because sensitivity to certain inputs and parameters is consistent with the physics of the phenomenon), came to haunt the modelers later on; the exaggerated expectations have turned into disappointment and distrust.

How could this be explained? A reservoir simulator based on PDE requires accurate definitions of reservoir properties on assumed certain scales, everywhere in the reservoir, in order to reliably represent the flow, and predict reservoir responses. Such near-ideal conditions could occur in cases where the scales of heterogeneities are much smaller than the simulated domain, and given the particular question being asked, such heterogeneities

could be lumped under unique, measurable reservoir characteristics that could be assumed uniform on a particular scale. While this could be the case in many sandy reservoirs and aquifer, on a certain scale, many more reservoirs and aquifers exhibit non-uniformity on all scales, and thereby, drastically degrade the prediction capability of PDE-based models. This has forced reservoir modelers and hydrogeologists to account for heterogeneity in any possible way. Due to lack of spatial data, heterogeneity translates to uncertainty, and uncertainty translates to randomness, and thus, the PDE become random (or stochastic) PDE [e.g., 87], severely complicating and inhibiting prediction capability. Moreover, since the uncertainty is on all scales and in all parameters, structural model errors add severe, unquantifiable uncertainty to the already uncertain solution [58, 54, 82]. Before we reach this bleak conclusion, let us further explore current treatment of heterogeneity/uncertainty in reservoir simulations.

2.2 Dealing with data gaps and uncertainty

We recognize that knowledge of all reservoir flow properties on all scales everywhere in the reservoir is impossible even without considering drilling costs. Due to the high cost of drilling, there is typically only sparse information on reservoir behavior, while geophysical (esp. seismic) data are by far more abundant. As a separate discipline, reservoir modelers have no choice but to rely on geological interpretations and limited observations of reservoir behavior, while being forced to average and upscale reservoir properties using ad-hoc estimates and geostatistical tools.

The prevalent method to account for heterogeneity (of reservoir properties) and the resulting uncertainty is to treat all the data or interpretations related to one or two dominant parameters (typically, saturated permeability) statistically, i.e., transforming uncertainty into randomness, typically under the assumptions of underlying joint probability distribution (PDF) that (assuming ergodicity) represents space and time as well. Once a certain parameter is considered random, the PDE solution becomes random (or stochastic) as well, and additional theoretical difficulties emerge [56-58, 87]. This approach has been developed and used for the last five decades in different areas of science and engineering, including stochastic optimization of oil reservoirs [1, 10-12, 20, 28, 36, 51, 59, 65-66, 80-81] and groundwater remediation [2, 3, 14-15, 19, 21-23, 25-27, 31, 39-42, 49, 68-73, 77, 83]. Despite remarkable theoretical advances in this difficult yet essential extension of the deterministic approach, developments have been limited to simple geometries and far-reaching simplifications.

The use of Monte Carlo Simulations (MCS), where “equally probable” high-resolution worlds (or realizations) are generated and simulated in order to compute ensemble statistics has been used extensively in the areas of reservoir simulations and hydrogeology (e.g., [56-58]); however, in practice, such a procedure results in enormous (if not prohibitive) computational burden for predictions alone, and becomes practically prohibitive for optimization, unless far-reaching simplifications are being made (e.g., [28]). In addition to these limitations, the traditional stochastic approach suffers from the following drawbacks and inconsistencies: (a) it cannot overcome, nor assess the major uncertainty in the model structure (which remain rigid); and (b) it leads to additional (now statistical) models with new parameters that are also uncertain; (c) using interpretations of well tests that assume homogeneity on a “near well” scale as the basis for conditional (stochastic) simulations; (d) using a single “dominant” parameter (on a

single scale) as the only random property (otherwise, computations are prohibitive even for limited cases); (e) assuming a PDF based on sparse spatial data; (f) assuming deterministic boundary conditions despite the significant uncertainty in it; (g) the inability to capture the linked physics and chemistry on all relevant scales (thereby, missing important phenomena such as front instability (between displacing and displaced fluids during enhanced oil recovery, where micro-scale variations trigger and promote fingering and bypassing due to capillary and viscosity differences [e.g., 35] and various geochemical reactions. Nevertheless, reservoir simulations and groundwater modeling are an important basis for approximations, correlations, and physical interpretations, including understanding and highlighting of the gaps and limitations of these interpretations.

Further, repeating the MCS chain of simulations and optimization as soon as new information arrives is practically impossible under the current scheme. Thus, despite the powerful theoretical framework and insight provided by the stochastic approach, this approach is yet in infancy, and does not extend beyond a certain definition of parameter uncertainty. Indeed, when optimization is attempted, e.g., for a new well placement, the computation-intensive stochastic approach becomes impossible, while a partial use of the approach (e.g., using only a few Monte Carlo simulations, as in [28, 12, 71], not accounting for uncertainty in other parameters, in the conceptual models, and in all interpretations and decisions along the path in Figure 1, leads to largely non-optimal decisions. If the reduction of intelligence can be measured by the amount of error between optimum and non-optimal operation, such a difference implies a significant reduction in intelligence gained by physically based models.

In conclusion, we recognize relationships between uncertainty, model robustness, and intelligence; an ideal PDE-based model is highly intelligent in a sense that it can predict reservoir behavior at all points in space and time; however, as soon as the model structure is inaccurate or model parameters are uncertain, it looses its intelligence to a large degree. Practically, this implies that a sensitive model would be "intelligent" as long as all necessary data exist and are accurate, but drastically looses its intelligence where data are uncertain, inaccurate, or insufficient; on the other hand, a less sensitive (more robust) PDE-based model would also be less intelligent to begin with. Thus, we seem to face an optimization problem: what model would be the optimal model for a particular problem; or better, what combination of models would be optimal in terms of data use, (maximum) intelligence, and robustness.

2.3 Current use of Artificial Intelligence

Before we answer this question, we should be aware two other sets of models: one based on artificial intelligence (AI) methods (also commonly termed *soft computing*), mostly in the context of geologic analysis and oil explorations, and one based on a statistical framework, particularly geostatistics and Bayesian statistics. As to AI, tools such as artificial neural networks, fuzzy logic, genetic algorithms, and probabilistic reasoning, have been used in reservoir characterization [75-76], subsurface flow [29, 63-64], and well field development and optimization [1, 10-11, 13, 28-30, 59, 62-64]. Consistent with the AI approach is the excessive use of geostatistics, such as the search for best next well placement described in [30], where the authors bypassed the simulator altogether, and used indicator kriging, instead, to interpolate expected production and

make corresponding decisions. Nevertheless, one of the important outcomes from these developments is the recognition of the need in integration of methodologies rather than using them in isolation [75]. In particular, the need to address the issues of (a) integrating information from various sources with varying degrees of uncertainty; (b) finding relationships between measurements and reservoir properties; (c) reducing uncertainty and risk; and (d) using all of these to optimize reservoir development and management, in real time. However, the progress in this direction has been slow, and fragmented results still dominate the field. The main reason for this is the need to translate information among the subsystems that constitute an oil reservoir, from geophysics to geology, and from geology to reservoir flow properties, and perform all of these translations on different scales of information, with different geometric and stratigraphic representations. These integration problems, and the overwhelming problem of uncertainty due to lack of data in the presence of inherent heterogeneity, have been unresolved, to date, with only scattered use of the various computational tools for limited characterization and prediction purposes; hence, leading to non-optimal management of oil reservoirs, water resources, and environmental cleanup (groundwater remediation) operations.

How could these capable AI models be combined with all other models in an adaptive framework that will (a) account for all the information (old and new, without initial filtering), and (b) allow continual updating and improvement due to continual data accumulation. The challenge is, thus, to integrate various measurements/data and models in a comprehensive, flexible, adaptive knowledge representation that will use all the available information for optimal decision making in the most intelligent way possible. Before trying to answer this question, let us review the third approach to control/optimization of subsurface flow and transport problems. One step in that direction was made by Rogers and co-workers [63, 64] who 'trained' a ANN by using multiple deterministic flow and transport simulations of a complex aquifer under a pump & treat operation, and later [29] for oil reservoir simulation-optimization, and then used the efficient ANN as a replacement ("proxy") for the cumbersome, slow simulator. Due to the limited extrapolation power of the ANN, many model runs were needed for the training to cover the expected span of possibilities (in the search space), to enable optimization of pumping and injection schedules. Although the method used is deterministic, as pointed out by the authors, it could, in principle become stochastic by generating multiple realizations and running Monte Carlo simulations (MCS), which, however, would result in prohibitive computer power. In other words, since the unsupervised ANN used is relatively less intelligent than the PDE models, it was used only as a minor auxiliary function. We would like to reverse this ranking of intelligence in a way that will enable broader conceptualization and knowledge representation.

2.4 Applying Dynamic Control

Dynamic, stochastic control has been used and further developed mainly for groundwater remediation purposes, specifically for pump & treat operations, which are similar to oil production operations; while the goal in oil production is to maximize production profit (over a certain period), the goal of groundwater remediation is to maximize extraction of contaminated water from the aquifer over a reduced period. The dynamic-stochastic control approach in this field [6-7, 16-18, 24, 32-34, 37-38, 60-61, 74] is an extension of the more general stochastic control theories of [8, 50, 67, 78-79].

While under the dynamic control approach, inverse modeling (i.e., updating/calibrating of uncertain parameters in a simulator or in a flow and transport model) is done jointly with the optimization process (hence, dual control), while feedback control rules enable changing of pumping rates (control variables) in response to changing hydraulic heads or contaminant concentrations (state variables). The stochastic simulator varies from extended Kalman filter to PDE, using perturbation methods and dividing the cost function into deterministic and stochastic parts, with the goal of minimizing remediation or plume-containment cost while optimizing both sampling and control actions. Typically, differential dynamic programming (DDP) is used to compute the deterministic control [6] while the solution of the stochastic part of the cost function is obtained analytically using stochastic control techniques applied to the governing flow and transport equations (PDE), with challenging mathematical derivation that requires a twice differentiable cost function. The on-line parameter estimation fed into the flow equation enables updating of both state variable estimates and state covariances. In terms of formulation of the cost function, the following highlights are worth mentioning: (a) the goal is to minimize the average (estimated, probabilistic) cost function; (b) the cost function is separable in stages, and according to the dynamic programming approach, whatever the initial state and initial decision are, the remaining (future) decisions/solutions should constitute an optimal solution based on the current state; i.e., the problem is reduced to finding a current optimal control variable, given a cost function over the remaining (future) periods, and given the current information state which includes all relevant a priori knowledge of the system and its history of observations and control; probabilistically, this information state is the conditional probability density function of the state at the current period conditioned on all past information; consequently, the cost function depends on uncertainty, directly. The two hidden elements in this procedure are: (a) the Bayesian approach, and (b) learning (from past experience).

Similar works [24] emphasize the use of all available information to estimate all present and future uncertainties, solving the management problem over the designated control horizon, applying the optimal control action (pumping or injection) during the current time period, and repeating this process at the next decision time, with PDE (flow equations) treated as a dynamical state-space system using finite element and finite difference techniques, considering (both) transmissivities and boundary conditions uncertain, and hence, perturbed in a highly simplified aquifer system, with the goal of minimizing pumping (and treatment) costs while maintaining hydraulic heads that guarantee containment of the contaminant plume. The results (a) provided insight into system response under uncertainty; (b) assessed trade-offs between satisfying goals and minimizing uncertainty (based on a simplified uncertainty model); (c) emphasized the effect of management decisions at any stage on model predictions in the next step. Explicit optimization combined with sensitivity analysis appeared to be an effective management approach. Other works [37-38, 60-61] extended the methodology of optimal estimation and scheduling of aquifer remediation under uncertainty, by allowing more complexity to be introduced, while performing real time (dynamic) feedback from measurements, as well as joint (on-line) parameter estimation - optimization and stochastic optimization. Subject to constraints and a specified reliability of meeting water quality requirements for a current period, the method minimizes the expected value of the

cost in the next (remaining) periods. A comparison between (adaptive) deterministic feedback control and the stochastic control formulated by [37] showed a clear cost reduction using the stochastic control formulation, with increasing difference as the uncertainty increases. Despite the accommodation of more complexity, and more general constraints, dynamic control methods that rely on PDE models are not yet suitable for complex real world problems.

One of the important insights that emerged in this implementation of stochastic control is the "probing" and "caution" effects highlighted by Bar Shalom [8]; the effect of the stochastic/perturbation part in the dual-control example [of Lee and Kitanidis] is that of sensitivity analysis and system excitation (the "probing" effect) followed by measurements and gaining information about system parameters that resulted in a substantial improvement. A paradigm shift is embedded here: rather than focusing on general predictive power (or lack of it), the dynamic control approach anticipates how the actual (future) state will deviate from estimated state currently in hand, and steers the system to mitigate possible losses (the "caution" effect). These two effects (of probing and anticipation/caution) imply yet another effect – that of goal-oriented learning.

The advantage of dynamic control was demonstrated by [16] who used differential dynamic programming to determine the benefits of time-varying optimal groundwater pumping policies, with the goal to reduce groundwater concentrations (of a contaminant) to acceptable levels. They demonstrated that static pumping policies would cost 45-75% more than policies that allow time-varying pumping rates, where the management model can "chase" the contaminant plume. Another set of developments along this line [17-18, 74] made use of the "transition function" (TF) that models (or transforms) the system from one state to the next (in the groundwater contamination case, the TF consists of the matrices generated by the finite element model at each time step) in order to reduce the number of iterations needed for convergence and overall computational time in the differential dynamic programming.

A substantial use of the second derivatives of the transition function in a constrained differential dynamic programming (DDP in a complete form) was made [74] with respect to a general case pump & treat remediation, including pumping scheduling and finding best well location. In this work [74], these derivatives were used to generate feedback laws with the aid of the penalty function method (which converts the constrained optimal control problem to unconstrained optimization, and consequently, allows flexibility in the response of the feedback laws to violation of constraints). These feedback laws describe relationships between required corrections of the control variables and weighted deviations of observed states from the predicted states. The goal was to find the relationships between the second derivatives of the transition function and evolutionary feedback laws, where the latter relate deviations from (and hence, required corrections to) optimal pumping scheduling and deviations of heads and concentrations (state variables) from their anticipated states, through weights discovered/assigned to these state deviations. The methods requires, as a first step, to employ a (deterministic) model and initial "optimal" pumping policy, which enables to build the first transition function, and find relationships between control and state deviations. The feedback laws are obtained by adjusting the relative weight assigned to each penalty function (corresponding to each control variable).

If we disregard the evolutionary nature of the feedback laws, the simple linear relationships expressed by the feedback laws (between observed deviation and required action) resembles the inverse of action-response functions used in different works [39-40]. It is interesting to note that while the transition function is derived from the governing PDE (flow and transport) model (which could be viewed as an elaborated response function model), the feedback law represents cause-effect rules (much like the inverse of the transition function) that compensate for model errors, regardless of the source of the errors. It is also interesting to note that the evolution of the feedback laws over time has an element of memory and learning (from past cause-effect relationships). Results from a simplistic example [74] showed to be robust and efficient in terms of reducing cost (by 4-51% less than optimization without using a feedback law) as well as required computer time, for up to 25% deviations from mean parameter values (i.e., uncertainty up to CV = 0.25).

Although exercised with only small perturbations (hence, small uncertainty) and some other limitations, this particular (complete) DDP approach is the first control/optimization method that frees itself not only from the need in a rigorous, well defined statistical/uncertainty model (with assumed PDF, correlation structure, etc.) but also free from both parameter errors and model errors, yet without neglecting uncertainty, and indirectly, reducing it, which makes this work a milestone that calls for continuation. Other works have coupled optimization with network design (optimal monitoring and information extraction from new wells) [7, 31, 40-42, 68], where the former [7] coupled sequential development of the groundwater withdrawal management with sampling strategies, dynamically, which led to the solution of the withdrawal design using a closed-loop stochastic control (dual control) method that includes anticipation of future observation locations; the decomposition of the cost function into deterministic and stochastic parts, particularly, the inclusion of uncertainty in the cost function leads to trade-off between cost of new wells and uncertainty reduction. The sampling network design method sequentially selects new measurement locations based on the combined effect of the state variable (hydraulic head) uncertainty at that location, and the sensitivity of the cost function to that uncertainty. More specifically, new sampling locations are selected using the Bayesian approach (to condition new measurements on existing information) and based on the product of the sensitivity of the stochastic part of the cost function and the modeled (predicted) variance of the state-variable (hydraulic head) at that location; that is, the sensitivity of the cost function to the head uncertainty is weighted by the magnitude of the prediction error – and vice versa (the prediction error is weighted by the sensitivity of the cost function to this error). The head uncertainty is evaluated by first-order, second-moment groundwater flow model, where the head uncertainty is linked to uncertainty in hydraulic conductivity, boundary conditions, recharge, and leakage (all are inputs of the PDE).

3. Interim conclusions

The theoretical developments and adaptations of methods from the different disciplines of operations research, stochastic control theories, and artificial-intelligence/soft-computing for management of oil reservoirs and groundwater remediation have provided insight into

a) the effect of uncertainty (even if just in one parameter) on optimal management and cost;
b) the inseparability of the various components of optimal reservoir management, such as optimal scheduling and best new well location for either pumping/injection or new monitoring wells;
c) the inseparability between optimal management and characterization;
d) the relationships between parameter uncertainty, reliability, and risk;
e) the relationships between parameter uncertainty and cost;
f) the effect of probing the system, system anticipation, and the "caution" that follows;
g) the similarity between the components of sensitivity analyses, random perturbations, and response functions and their "inverse" - weighted feedback laws;
h) the ability to compensate for unknown model errors by determining appropriate weighted feedback policies, particularly under dynamic feedback control;
i) the hidden forms of memory and learning that exist in some statistical models (particularly Bayesian statistics), particularly where recursive/evolutionary information processing takes place, as is the case in some dynamic control systems, and particularly where such processing results in corresponding feedback;
j) the strength of Bayesian approaches in both estimation and uncertainty reduction.

The advantage of the control approach is in shifting the emphasis from one type of intelligence – that of predictions of first principle (physically-based) models to goal-oriented system anticipation (the anticipation of the effect that a control action would have on the goal, i.e., on the cost function), as well as shifting sensitivity analysis (of general model predictions) to sensitivity of the cost function (to parameter uncertainty and particularly, to state uncertainty), which changes the experimental design and overall planning. This goal-oriented intelligence is less "ambitious" than the "know-it-all" first-principle model. Our goal is to increase the intelligence of the goal-oriented anticipating model by combining/integrating knowledge and models from all disciplines in a multiresolutional decision support system (MR-DSS or MRDS), e.g., [4-5, 43-48].

4. Increasing intelligence with intelligent control

Fortunately, the area of intelligent control, particularly, the MRDS has been developing rapidly during the last two decades, combining the advantages and eliminating the limitations of control theories, operations research, and artificial intelligence. For example, an intelligent control agent such as MRDS could free the dynamic control from its ultimate dependence on the rigid PDE, and can increase its learning, accumulated memory, and speed of convergence to optimal solutions by orders of magnitude. Moreover, one of the appealing outcomes of the stochastic approach (including stochastic PDE) – the effect of conditioning on uncertainty reduction via correlations among variables – could be amplified significantly by extending the associations among variables on all relevant scales (through advanced MR clustering methods) far beyond linear the statistical correlation used in the traditional stochastic approach; the Multi-Resolutional (MR) knowledge representation in MRDS maximizes the information hidden in interdependencies among these variables on all levels of resolution, independent of any particular single-scale model. By maximizing extraction of information, the MR approach effectively reduces uncertainty and overcomes the problem of lacking and

corrupted information. Most importantly, by using intelligent control, specifically, a goal-oriented MR knowledge representation, we could eliminate the dependency on PDE models, and use them just as interpretations and general gap-filler in the process of MR rule building (the latter being based on experiences and cause-effect relationships). An adaptive MR knowledge representation is the only way to integrate all the methods from all disciplines – to benefit from the advantages of the different models and eliminate their limitations (particularly their rigid structure); to break through the rigid serial, model-building process (currently done in segregation and isolation) and (hence) to enable more powerful use of data and knowledge from all disciplines; to provide the highest uncertainty reduction possible, and efficient global stochastic optimal control of complex natural resources systems such as oil reservoirs and groundwater, with the highest intelligence and autonomy possible for particular goals.

However, we are still facing the challenge of performing and defining optimal integration between PDE models and multiresolutional representations, and since such integration depends on model quality and data quantity and quality, this is an adaptive integration process as well. One criterion for such "optimal" integration would be uncertainty reduction (resulting from the integrated MR system), which brings up yet another challenge: to define the metrics for uncertainty reduction.

REFERENCES

1. Aanonsen, S. I., Eide, A. L., Holden, L. and Aasen, J. O. (1995) "Optimizing Reservoir Performance Under Uncertainty with Application to Well Location," paper SPE 30710 presented at the SPE Annual Technical Conference & Exhibition, Dallas, Texas, October 22-25.
2. Ahlfeld, D., J. Mulvey, G. Pinder, and E. Wood, (1988a) Contaminated groundwater remediation design using simulation, optimization and sensitivity theory, 1, Model development, *Water Resour. Res*., *24*, 431–442.
3. Ahlfeld, D., J. Mulvey, G. Pinder, and E. Wood, (1988b) Contaminated groundwater remediation design using simulation, optimization and sensitivity theory, 2, Analysis of a field site, *Water Resour. Res*., *24*, 443–452.
4. Albus, J., A. Meystel, S. Uzzaman (1993) "Nested Motion Planning for an Autonomous Robot", Proc. IEEE Conference on Aerospace Systems, May 25-27, Westlake Village, CA.
5. Albus, J., A. Meystel (2001) *Engineering of Mind: An Introduction of the Science of Intelligent Systems*, Wiley, NY.
6. Andricevic, R., and P.K. Kitanidis (1990) Optimization of the pumping schedule in aquifer remediation under uncertainty, *Water Resour. Res., 26*(5), 875-885.
7. Andricevic, R. (1993) Coupled withdrawal and sampling designs for groundwater supply models, *Water Resour. Res., 29*(1), 5-16.
8. Bar-Shalom, Y. (1981) Stochastic dynamic programming: Caution and Probing, IEEE *Trans. Autom. Control*, 26(5), 1184-1195.
9. Bear, J., *Dynamics of Fluids in Porous Media*, American Elsevier, 1972.
10. Beckner, B. L. and Song, X.: "Field Development Planning Using Simulated Annealing - Optimal Economic Well Scheduling and Placement," paper SPE 30650 presented at the 1995 SPE Annual Technical Conference & Exhibition, Dallas, Texas, October 22-25.
11. Bittencourt, A. C. and Horne, R. N. (1997) "Reservoir Development and Design Optimization," paper SPE 38895 presented at the 1997 SPE Annual Technical Conference & Exhibition, San Antonio, Texas, October 5-8.

12. Bush, M.D., M. Cuypers, F. Roggero, A-R. Syversveen (2002) "Comparison of production forecast uncertainty quantification methods – An integrated study", Delft, The Netherlands. http://www.nitg.tno.nl/punq/cases/punqs3/PUNQS3paper/index.htm
13. Centilmen, A., Ertekin, T. and Grader, A. S.: "Applications of Neural Networks in Multiwell Field Devel-opment," paper SPE 56433 presented at the 1999 SPE Annual Technical Conference & Exhibition, Houston, Texas, October 3-6.
14. Chan, N. (1993) Robustness of the multiple realization method for stochastic hydraulic aquifer management, *Water Resour. Res., 29*(9), 3159-3167.
15. Chan, N. (1994) Partial infeasibility method for chance-constrained aquifer management, *ASCE J. Water Resour. Plan. and Manag., 120*(1), 70-89.
16. Chang, L.-C., C. A. Shoemaker, and P. L.-F. Liu (1992) Optimal time-varying pumping rates for groundwater remediation: Application of a constrained optimal control algorithm, *Water Resour. Res*., *28*, 3157– 3173.
17. Culver, T.B., and C.A. Shoemaker (1992) Dynamic optimal control for groundwater remediation with flexible management periods, *Water Resour. Res., 28*(3), 629-641.
18. Culver, T.B., and C.A. Shoemaker (1993) Optimal control for groundwater remediation by differential dynamic programming with quasi-newton approximations, *Water Resour. Res.,* 29(4), 823-831.
19. Dougherty, D.E., and R.A. Marryott (1991) Optimal groundwater management 1. Simulated Annealing, *Water Resour. Res., 27*(10), 2493-2508.
20. Eidi, A.L., Holden, L., Reiso, E., and Aanonsen, S. I. (1994) "Automatic History Matching by use of Response Surfaces and Experimental Design," paper presented at the 4th European Conference on the Mathematics of Oil Recovery, Roros, Norway, 7-10 June, 1994.
21. Feinerman, E., Bresler E., and Dagan G. (1985) Optimization of a spatially variable resource: An illustration for irrigated Crops. *Water Resour. Res., V. 12. pp. 793-800.*
22. Freeze, R..A., J. Massmann, L. Smith, T. Sperling, and B. James (1992) *Hydrogeological Decision Analysis*. 72 pp. Ground Water Publishing Company, Dublin, OH.
23. Freeze, R.A. and S.M. Gorelick (1999) Convergence of stochastic optimization and decision analysis in the engineering design of aquifer remediation, *Ground Water*, 37(6), p. 934-954.
24. Georgakakos, A.P. and D.A. Vlatsa (1991) Stochastic control of ground water systems, *Water Resour. Res., 27*(8), 2077-2090.
25. Gordon, E., Shamir, U. and Bensabat, J. (2000) "Optimal Management of a Regional Aquifer under Salinization Conditions", Water Resources Research, 36(11), p. 3193-3204.
26. Gorelick, S.M. (1997) Incorporating uncertainty into aquifer management models. In: *Subsurface flow and transport*, edited by G. Dagan and S.P. Neuman, Cambridge University Press p. 101-112.
27. Gorelick, S.M., R.A. Freeze, D. Donohue, and J.F. Keely (1993) *Groundwater Contamination: Optimal capture and containment,* Lewis Publishers, Chelsea, Michigan, 385 p.
28. Gűyagűler, R.N. Horne (2001) "Uncertainty Assessment of Well Placement Optimization", SPE 71625, 2001 [with reference to http://www.nitg.tno.nl/punq/index.htm : "Production forecasting with Uncertainty Quantification" (PUNQ) - A Comparative Study, Netherlands Institute of Applied Geosciences, TNO, National Geological Survey, Elf Exploration Production, 2002].
29. Gŭyagŭler, B., Horne, R.N., Rogers, L. and Rosenzweig, J.J. (2000) "Optimization of Well Placement in a Gulf of Mexico Waterflooding Project," paper SPE 63221 presented at the 2000 SPE Annual Technical Conference and Exhibition, Dallas, Texas, October 1-4.
30. Hohn, M.E. and R.R. McDowel, "Geostatistical Analysis of Oil production and Potential Using Indicator Kriging", in *Stochastic Modeling and Geostatistics – Principles, Methods, and Case Studies,* J.M. Yarus, and R.L. Chambers, editors, AAPG Computer Applications in geology, No. 3, The American Association of Petroleum Geologists, Tulsa, OK, USA, 1994.

31. James, B. and S.M. Gorelick (1994) When enough is enough: The worth of monitoring data in aquifer remediation design, *Water Resources Research*, vol. 30, no. 12, p. 3499-3513.
32. Jones, L., Willis, R, and Yeh, W.W.-G. (1987) Optimal control of nonlinear groundwater hydraulics using differential dynamic programming. *Water Resour. Res.*.,23(11), 2097-2106.
33. Jones, L. (1992) Adaptive control of ground-water hydraulics, *ASCE J. Water Resour. Plan, and Manag., 118*(1), 1-17.
34. Kessler A. and Glodberg I. (2004) Optimal Operation of the Water Resources in Ramat Hagolan, Proceedings of The Kinneret Lake Workshop, The Israeli Water Resources Association.
35. Lake, L.W., *Enhanced Oil Recovery*, Prentice Hall, 1989
36. Landa J.L. and B. Gűyagűler (2003) A Methodology for History Matching and the Assessment of Uncertainties Associated with Flow Prediction, SPE 84465.
37. Lee, S. -I., and Kitanidis, P. K. (1991) Optimal Estimation and Scheduling in Aquifer Remediation with Incomplete Information, *Water Resour. Res.,* 27(9), 2203-2217.
38. Lee, S.-I., and P. K. Kitanidis (1996) Optimization of monitoring well installation time and location during aquifer decontamination, *Water Resour. Management,* 10, 439-462.
39. Maddock, T. III (1972) Algebraic technological functions from a simulation model. *Water Resour. Res.*., 8(1), 129-134.
40. Maddock, T. III (1973) Management model as a tool for studying the worth of data. *Water Resour. Res.*., v. 9, pp. 270-280.
41. Massmann, J. and Freeze, R.A. (1987a) Groundwater contamination from waste management sites: The interaction between risk-based engineering design and regulatory policy, 1. *Water Resour. Res.*., v. 23, pp. 351-367.
42. Massmann, J. and Freeze, R.A. (1987b) Groundwater contamination from waste management sites: The interaction between risk-based engineering design and regulatory policy, 2. *Water Resour. Res.*., v. 23, pp. 368-380.
43. Meystel, A. (1986) "Planning in a hierarchical nested controller for autonomous robots", in Proc. IEEE 25th Conf. Decision Control, Athens, Greece. pp. 1237-1249.
44. Meystel, A. (1991) Autonomous Mobile Robots: Vehicles with Cognitive Control (World Scientific).
45. Meystel, A. (1995) Semiotic Modeling and Situation Analysis: An Introduction (AdRem, 1995)
46. Meystel, A., S. Uzzaman (2000) "Multiresolutional Decision Support System", US Patent, No. 6102958.
47. Meystel, A., J. Albus (2002) *Intelligent Systems: Architectures, Design, Control*, Wiley, NY.
48. Meystel, A.M., *Multiresolutional Hierarchical Decision Support Systems*, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviesa, Vol. 33, No. 1, February 2003, Vol. 33, No. 1, ITCRFH, p.p. 86-101.
49. Morgan, D.R., J.W. Eheart, and A. J. Valocchi (1993) Aquifer remediation design under uncertainty using a new chance constrained programming technique, *Water Resour. Res.,* 29(3), 551-561.
50. Murray, D., and S. Yakowitz (1979) Constrained differential dynamicprogramming and its application to multireservoir control. *Water Resour. Res.*, 15(4), 1017-1027.
51. Murty, C.R. K., and A. Al-Hadad: "Integrated Development Approach for a Mature Oil Field". SPE 81438, April, 2003.
52. Muskat, M. (1937, 1946) The Flow of Homogeneous Fluids Through Porous Media. McGraw-Hill, New York. (2nd printing by Edwards, Ann Arbor, Misc.)
53. Muskat, M. (1949) Physical Principles of Oil Production, McGraw-Hill, New York.
54. Neuman, S.P. (2003) Maximum likelihood Bayesian averaging of alternative conceptual-mathematical models, *Stochastic Environmental Research and Risk Assessment*, 17(5), 291-305, DOI: 10.1007/s00477-003-0151-7.

55. Neuman, S.P. and P.J. Wierenga (2003) A comprehensive strategy of hydrogeologic modeling and uncertainty analysis for nuclear facilities and sites, NUREG/CR-6805, U.S. Nuclear Regulatory Commission, Washington, D.C.
56. Neuman, S. P., and S. Orr, "Prediction of steady state flow in nonuniform geologic media by conditional moments: Exact non local formalism, effective conductivities, and weak approximations", *Water. Resources Research,* 29 (2), 341-364, 1993.
57. Orr, S., and S.P. Neuman, "Operator and Integro-Differential Representations of Conditional and Unconditional Stochastic Subsurface Flow". An invited paper, Journal of Stochastic Hydrology and Hydraulics, Vol 8(2), pp. 157-172, 1994
58. Orr, S. (2002) "Uncertain Predictions of Flow and Transport in Random Porous Media: The implications for Process Planning and Control", in Measuring the Performance and Intelligence of Systems: Proceedings of the PerMIS Workshop, NIST Spec. Pub. 990, edited by E.R. Messina and A.M. Meystel, August 13-15, 2002 (pp. 435-441).
59. Pan, Y. and Horne, R. N. (1998) Improved Methods for Multivariate Optimization of Field Development Scheduling and Well Placement Design, paper SPE 49055 presented at the 1998 SPE Annual Technical Conference & Exhibition, New Orleans, Texas, September 27-30.
60. Philbrick, C. R., and P. K. Kitanidis (1998.) Optimal Conjunctive-Use Operations and Plans, *Water Resour. Res.*, 34(5), 1307-1316.
61. Philbrick, C. R., and P. K. Kitanidis (2000) Improved Dynamic Programming Methods for Optimal Control of Lumped Parameter Stochastic Systems, Operations Research.
62. Rian, D. T. and Hage, A. (1994) "Automatic Optimization of Well Locations in a North Sea Fractured Chalk Reservoir Using a Front Tracking Reservoir Simulator," paper SPE 28716 presented at the 1994 SPE International Petroleum & Exhibition of Mexico, Veracruz, Mexico, October 10-13.
63. Rogers, L.L., and Dowla, F.U. (1994) "Optimal Groundwater Remediation Using Artificial Neural Networks with Parallel Solute Transport," Water Resources Research, 30(2): 458-481.
64. Rogers, LL, Dowla, F.U., and Johnson V.M. (1995) Optimal field-scale groundwater remediation using neural networks and the genetic algorithm. Environ. Sci. Technol., 29(5). Pp. 1145-1155.
65. Saputelli, L.A., S. Mochizuki, L. Hutchins, T. Cramer, M.B. Anderson, J.B. Mueller, A. Escorcia, A.L. Harms, C.D. Sisk, S. Pennebaker, J.T. Han, A. Brown, C.S. Kabir, R.D. Reese, G.J. Nuňez, K.M. Landgren, C.J. McKie, and A. Airlie: "Promoting Real Time Optimization of Hydrocarbon Producing Systems". SPE 83978, Sept. 2003.
66. Stoisits, R. F., Crawford, K. D., MacAllister, D. J., Lawal, A. S., and Ogbe, D. O. (1999) "Production Optimization at the Kuparuk River Field Utilizing Neural Networks and Genetic Algorithms," paper SPE 52177 presented at the 1999 Mid-Continent Operations Symposium, Oklahoma City, Oklahoma, March 28-31.
67. Tse, E., Bar-Shalom, Y., and Meier, L. (1973) Wides sense dual control for nonlinear stochastic systems. IEEE *Trans. Autom. Control*, 18(2), 98-108.
68. Tucciarelli, T., and G. Pinder (1991) Optimal data acquisition strategy for the development of a transport model for groundwater remediation, *Water Resour. Res., 27*(4), 577-588.
69. Tung, Y., Groundwater management by chance-constrained model, *ASCE J. Water Resour. Plan. and Manag., 112*(1), 1-19, 1986.
70. Wagner, B.J. and S.M. Gorelick (1987) Optimal groundwater quality management under uncertainty, *Water Resour. Res., 23*(7), 1162-1174.
71. Wagner, B.J. and S.M. Gorelick (1989) Reliable aquifer remediation in the presence of spatially variable hydraulic conductivity: From data to design, *Water Resour. Res., 25*(10).
72. Wagner, J.M., U.Shamir, and H.R. Nemati, (1992) Groundwater quality management under uncertainty: Stochastic programming approaches and the value of information, *Water Resour. Res., 28*(5), 1233-1246.

73. Wang, W., and D. P. Ahlfeld (1994) Optimal groundwater remediation with well location as a decision variable: Model development, *Water Resour. Res*., *30*(5), 1605–1618.
74. Whiffen, G.J., and C.A. Shoemaker (1993) Nonlinear weighted feedback control of groundwater remediation under uncertainty, *Water Resour. Res., 29*(9), 3277-3289.
75. Wong, P. F. Aminzadeh, and M. Nikravesh, editors (2002) *Soft Computing for reservoir Characterization and Model*ing, Pysica-Verlag, Heidelberg, NY.
76. Wong, P.M.F., Aminzadeh, and M. Nikravesh, "Intelligent Reservoir Characterization", in *Soft Computing for reservoir Characterization and Model*ing, P. Wong, F. Aminzadeh, and M. Nikravesh, editors, Pysica-Verlag, Heidelberg-NY, 2002.
77. Xiang, Y., J. Sykes, and N. Thomson (1995) Alternative formulations for optimal groundwater remediation design, *J. Water Resour. Plann. Manage*., *121*, 171–181.
78. Yakowitz, S.J. (1982) Dynamic programming applications in water resources, *Water Resour. Res*., *18*(4), 673–696.
79. Yakowitz, S.J. (1986) The stagewise Kuhn-Tucker condition and differential dynamic programming. IEEE *Trans. Autom. Control*, AC-31(1), 25-30.
80. Yang, D., Qi Zhang, Y. Gu, and L. Li: "Integrated Global optimization of Displacement Efficiency in Hydrocarbon reservoirs". SPE 81035, April, 2003.
81. Yarus, J.M., and R.L. Chambers, editors, *Stochastic Modeling and Geostatistics – Principles, Methods, and Case Studies,* AAPG Computer Applications in geology, No. 3, The American Association of Petroleum Geologists, Tulsa, OK, USA, 1994.
82. Ye, M, Neuman, S.P., Meyer, D.P. (2004) Maximum Likelihood Bayesian Averaging of spatial variability models in unsaturated tuff. Submitted to *Water Resour. Res*. (in press).
83. Zheng, C., and P. Wang (1999) An integrated global and local optimization approach for remediation system design. *Water Resour. Res*., 35(1), 137-148.
84. Gambolati, G. and R.A. Freeze (1973) Mathematical simulation of the subsidence of Venice: 1. Theory. *Water resourses Res.,* 9, pp. 721-733.
85. Gambolati, G., P. Gatto, and R.A. Freeze (1974a) Mathematical simulation of the subsidence of Venice: 2. Results. *Water resourses Res.,*10, pp. 563-577.
86. Gambolati, G., P. Gatto, and R.A. Freeze (1974b) Predictive simulation of the subsidence of Venice. *Science.,*183, pp. 849-851.
87. Zhang, D.(2002) Stochastic Methods for Flow in Porous Media: Coping with Uncertainties, Academic Press, San Diego, Calif., ISBN 012--7796215, pp.350.

AGENT WITH REASONING AND LEARNING. THE STRUCTURE DESIGN

Dr. Leonid M. Polyakov
Globe Institute of Technology
New York, NY
Leonid@globeinstitute.org

ABSTRACT
Reasoning and learning are the most powerful intellectual functions. It is not easy to emulate them. Main problem is determined by nature of reasoning that is based on computation with words instead of computation with numbers. There are a lot of different approaches to the knowledge representation in the agent's knowledge base. The most important languages of knowledge representation are preposition logic and predicate logic. Design of the agent models of intentional (conscious) and unintentional (unconscious) reasoning (intuition) with multi-knowledge base structure based on preposition and predicate logic, learning and heuristic generation are topics of this discussion.
Keywords: *agent, design, learning, reasoning, preposition logic, predicate logic, knowledge base, rules of reasoning, application rule, hypothesis, intuition, multi-KB system.*

APPROACH TO THE PROBLEM SOLUTION

Reasoning, as we know, is the process of drawing conclusion from facts. There is a lot of research dedicated to the problems of reasoning and the agent structure design [1,2,4,6]. All of them are based on representation of knowledge as rule-based, semantic net or frame structure knowledge base. These knowledge bases (KB) include just knowledge of application (AKB) (domain oriented KB). Rules of reasoning are applied on AKB in different ways for different agents. The theorem prover is the representative of the system with reasoning but it is design without of the KB. Most of the existing systems with reasoning are not universal theorem prover (http://www-formal.stanford.edu/clt/ARS/Entries/acl2). These systems are based on rules of reasoning and don't work with application knowledge ether. Some of them, like ACL2, are designed as multi-KB systems. However, all these systems are based just on preposition logic. The most interesting result in the area of reasoning is the Jess language (Jess, the Java Expert System Shell, http://herzberg.ca.sandia.gov/jess/demo.html). This system is not the multi-KB system and has just one KB-AKB. Information is presented by predicate logic. Rules of reasoning are incorporated into source code. Idea of a multi-KB in search engines also was described by Dr. Lotfi Zadeh in "The Prototype-Centered Approach to Adding Deduction Capability to Search Engines- The Concept of Protoform" (BISC letter, 21 Dec 2001) http://www.cs.berkeley.edu/People/Faculty/Homepages/zadeh.html In this letter: "The deduction database is assumed to consist of logical database and a computational database, with the rules of deduction…"

Multi-KB structure is only the possible way to increase level of universality of the agent up to the level of AI system [5]. Separation of AKB and RKB from the program converts a conventional system into system with ability to learn, creates conditions for teaching the system through delivery new rules of application by an expert in area of application and reasoning without knowledge of programming. It is important step from a conventional system to the AI system. New rules should have the same structure as existing rules. New processes can be added as new program modules. Multi-KB structure creates conditions to design a system with ability to generate rules as possible hypothesis in the AKB (Fig.8). The first KB is application knowledge base-AKB; the second one is a reasoning KB-RKB. RKB has rules of reasoning. RKB is universal KB. It can be used with different AKB. The number of areas of application or number of Goals determines the number of AKB. The Double-KB structure of a system is shown on Fig.1. The process of reasoning is shown on Fig. 2. Complicated application rules should be decomposed to simple structure by rules of reasoning (And-Elimination rule-RR on Fig.2) application.

Process of reasoning in preposition logic is determined by terms of application rules (AR). Process of reasoning in predicate logic is determined by predicate of application rules (AR). Choice of rules of reasoning (RR) is determined by the structure of the application rule. New knowledge is generated by application rules to the World Model (WM). Technically a

process of reasoning can be described as the chain of steps:

Data-AR activation-AR application to the WM (testing all chains of related knowledge)-RR activation

Fig. 3 shows four-steps algorithm. Fig.4 shows Forward-chain algorithm of reasoning that is based on rules RR15-RR17. Application of the rules RR1-RR14 is not shown. Fig.6 presents realization of multi-KB structure in the systems with preposition and predicate logic. Fig.7 presents realization of multi-KB structure in the systems with INTUITION. Definition and nature of intuition is described in [5]. Discover of the Dr. Ben Seymour (neuroscientist from London University College (http://news.bbc.co.uk/go/pr/fr/-/2/hi/health/3791357.stm) supports this hypothesis.

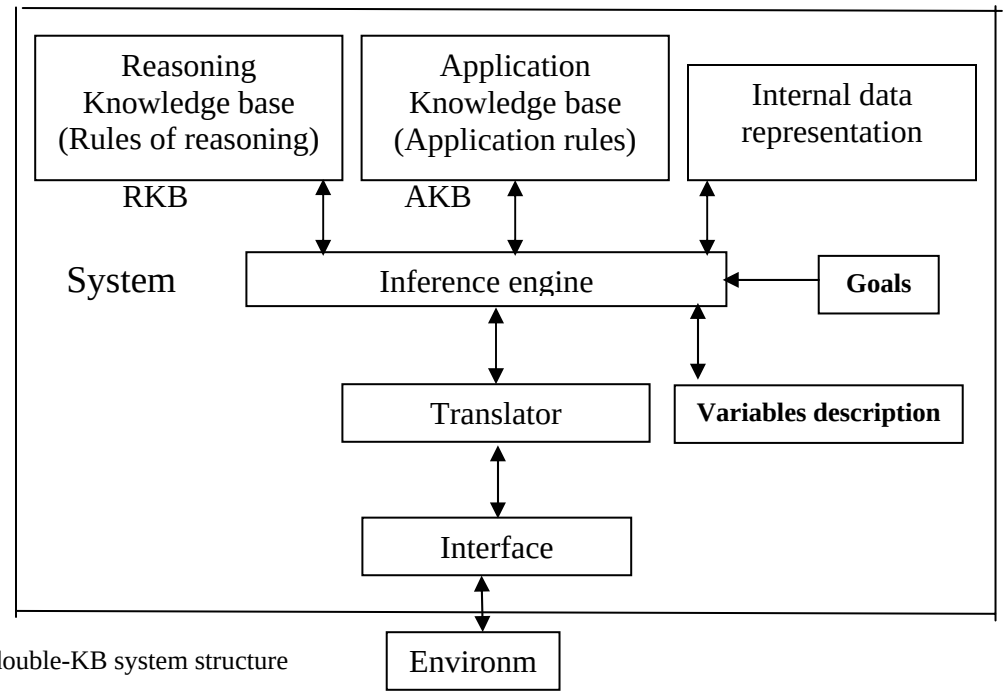


Fig. 1 The double-KB system structure

PREPOSITION AND PREDICATE LOGIC RULES OF REASONING

There is limited set of reasoning rules in preposition logic [1-5]:

RR1. Implication Elimination: $\alpha \Rightarrow \beta$, (modus ponens-mp) (IF α is in DB THEN $\beta \Rightarrow$ true)

RR2. And –Elimination: $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_n \Rightarrow \text{LIST}(\alpha_i)$, $\text{LIST}(\underline{\alpha_i}) = \alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n$

$\text{con}(\alpha_i) \Rightarrow \text{LIST}(\alpha_i)$, [i=1,n]

RR3. And-Introduction: $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n \Rightarrow \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \ldots \wedge \underline{\alpha_n}$

$\text{LIST}(\alpha_i) \Rightarrow \text{con}(\alpha_i)$, [i=1,n]

RR4. Or-Introduction: $\text{LIST}(\alpha_i) \Rightarrow \alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \ldots \vee \alpha_n$ $\alpha_i = \underline{\alpha_1}, \alpha_2, \alpha_3, \ldots, \alpha_n$

$\text{LIST}(\alpha_i) \Rightarrow \text{dis}(\alpha_i)$, [i=1,n]

RR5. Double-Negation Elimination: $\neg \neg \alpha \Rightarrow \alpha$

RR6. Unit Resolution: $\alpha \vee \beta \wedge \neg \beta \Rightarrow \alpha$

RR7. Resolution: $\alpha \vee \beta \wedge \beta \vee \gamma \Rightarrow \alpha \vee \gamma$

RR8. Universal Elimination: $\forall \nu \ \alpha(\nu) \Rightarrow \alpha(g)$, (from DB: $\nu = g$)

RR9. Existential Elimination: $\exists \nu \ \alpha(\nu) \Rightarrow \alpha(g)$ (from DB: $\nu = g$)

RR10..Existential Introduction: $\alpha(g) \Rightarrow \exists \nu \ \alpha(\nu)$ (from DB: $\nu = g$)

RR11. DeMorgan Laws

RR12 Universal Generalization: $(\forall x) P(x)$

RR13 Existential Generalization: $(\exists x) P(x)$

RR14 Rules of induction: $P(1)=T$

$(\forall k) \{[P(k)=T] \Rightarrow [P(k+1)=T]\} \Big| \rightarrow P(n) \Rightarrow T$

RR15 Associative law

This set of rules creates the universal RKB.

Example of the process of reasoning.
Suppose, DB initially includes facts A, B, C, D, and E, and AKB contains **application** rules:

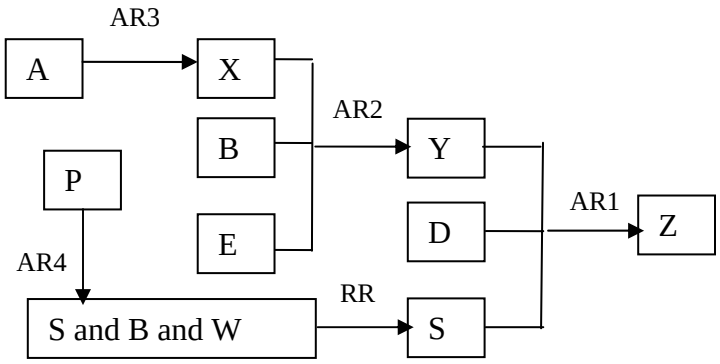
AR1: IF Y is true
AND S is true AR3

AND D is true
THEN Z is true

AR2: IF X is true
AND B is true
AND E is true
THEN Y is true

AR3: IF A is true
THEN X is true

AR4: IF P is true
THEN S ∧ B ∧ W is true



RR: IF S and B and W is true THEN S is true

Fig. 2 An inference (forward) chain in a system based on proposition logic.

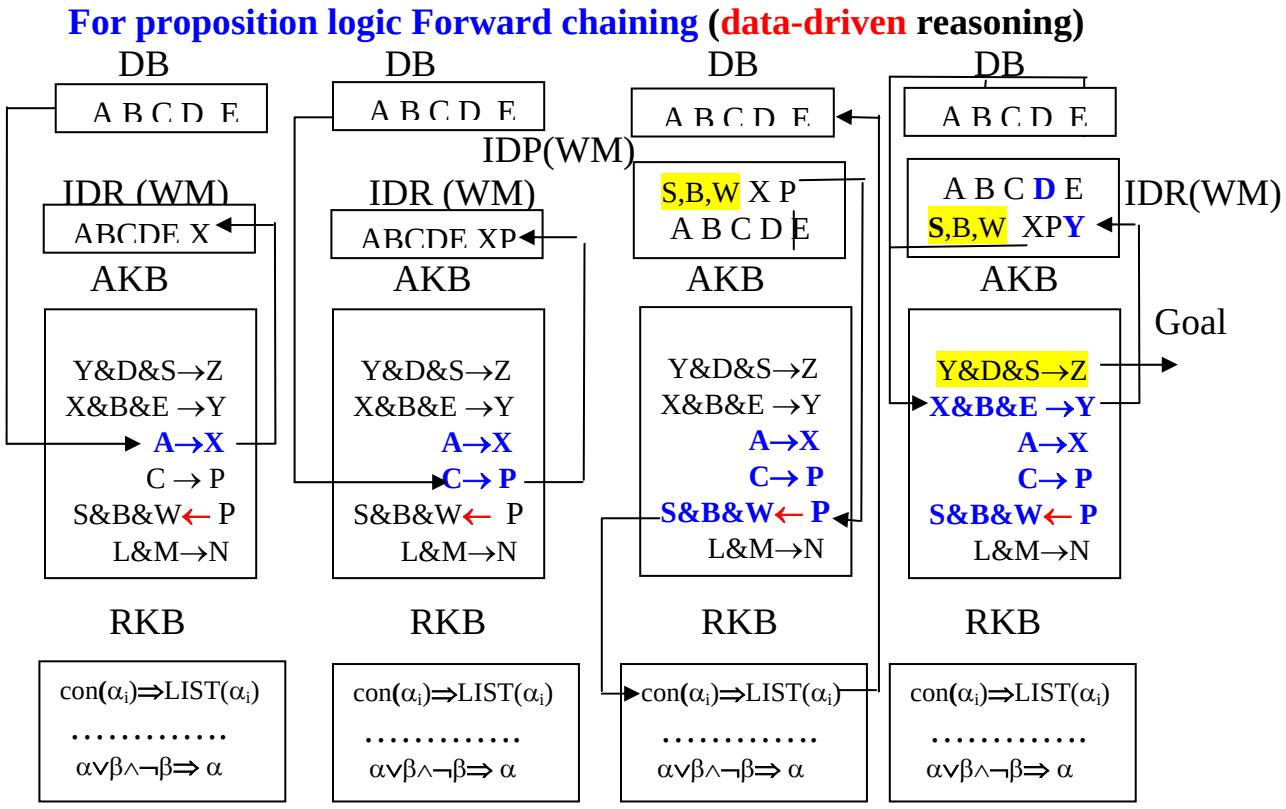


Fig.3 The system structure and algorithm. IDR-internal data representation, WM-World Model, AKB-application knowledge base, DB-Data base (external data representation), RKB-reasoning knowledge base

SYSTEMS ARE BASED ON PREDICATE LOGIC

Syntax in predicate logic can be presented as: PREDICATE (LIST OF TERMS - OBJECTS)
PREDICATES: RELATIONSHIP, PROPERTIES, and FUNCTIONS.
Suppose, the following facts in the predicate Logic using meaningful predicates and functions rules.

Rules of application

1) Anyone sane does not teach an AI course.
   $\forall x\ sane(x) \rightarrow \neg\ AIInsructor\ (x)\ )$
2) Every circus elephant is a genius.
   $\forall x CircusElephant(x) \rightarrow genius(x)$
3) Nothing is both male and a circus elephant.
   $\neg\ \forall x\ Male(x) \Leftrightarrow CircusElephant(x)$
4) Anything not male is female.
   $\neg \forall x\ Male(x) \Leftrightarrow Female(x)$

Data

1) Clyde is not an AI instructor. $\neg AIInsructor(Clyde)$
5) Clyde is a circus elephant. $CircusElephant(Clyde)$

Determine state of the following is true, false or cannot be determined based on the application rules:
Clyde is a genius.

Rules of reasoning include all rules of reasoning based on preposition logic and set of rules that are specific to the predicate logic:
RR16. Find all atomic sentences that related to the first term in DB
RR17. Find all atomic sentences with conclusion that related to the predicate of the result of RR1 action
RR18. Check each of them against solution question.

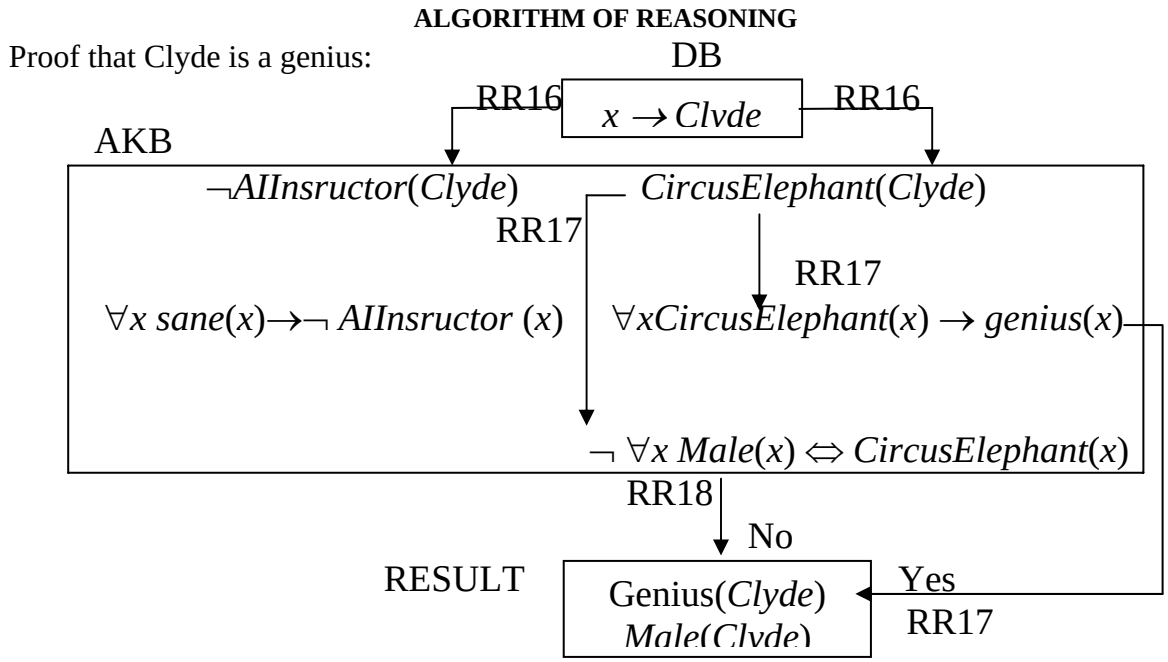


Fig.4 shows Forward-chain algorithm of reasoning based on rules RR15-RR17. Action of the RR1-RR14 is not shown.

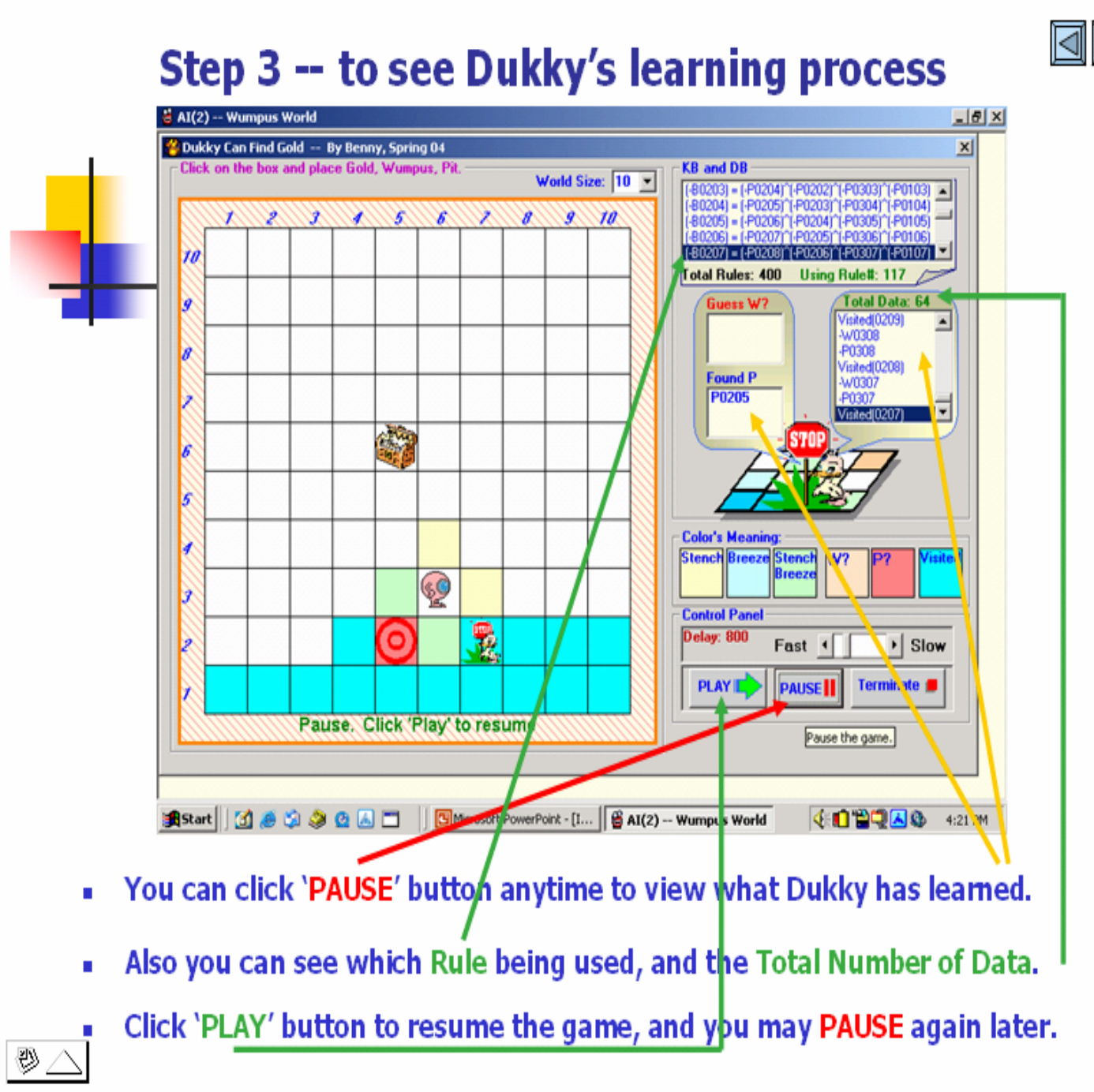


Fig.5 “Wumpus World” model (preposition logic).

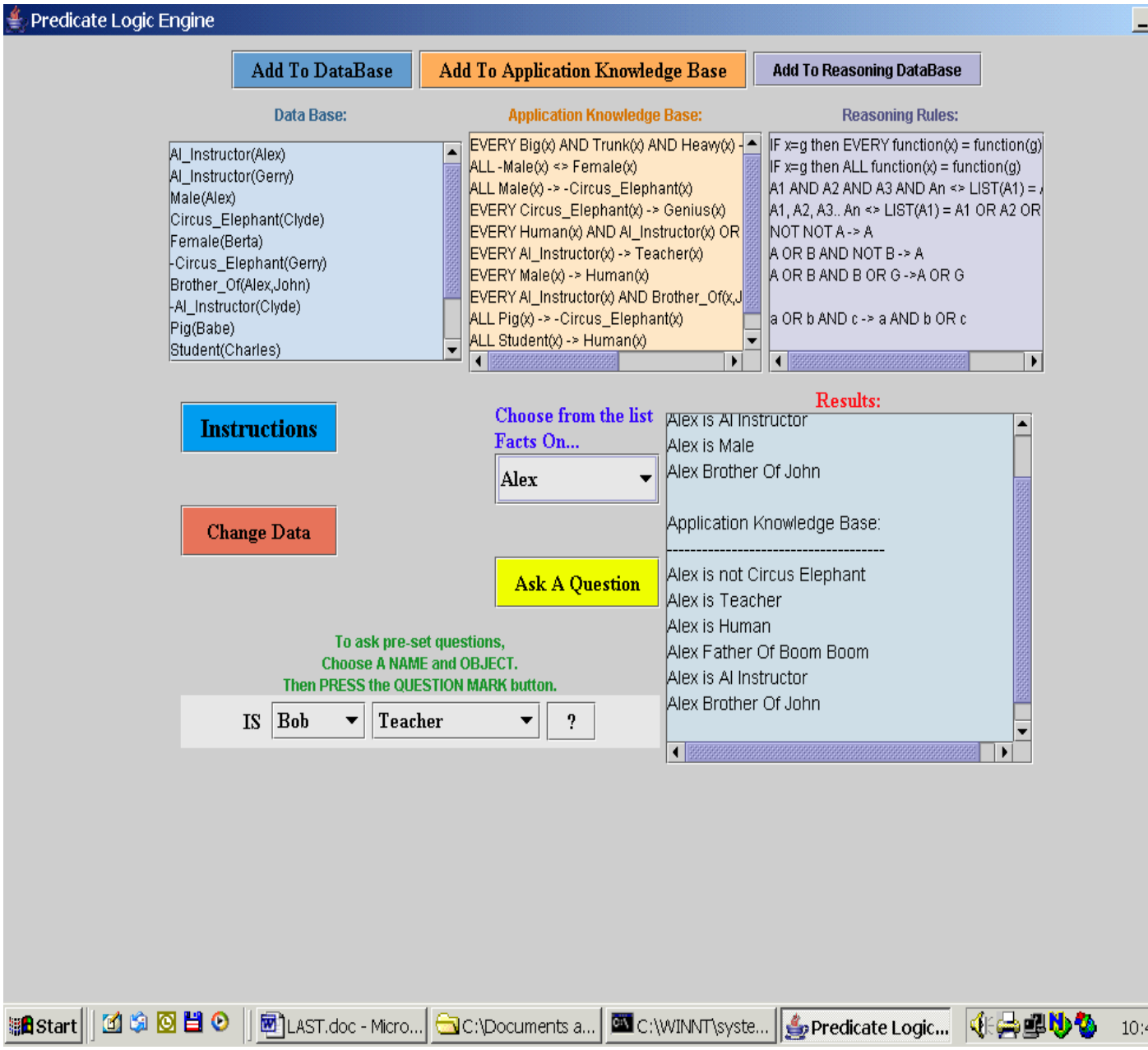


Fig. 6. This system (predicate logic language) is based on the algorithm presented by Fig, 4

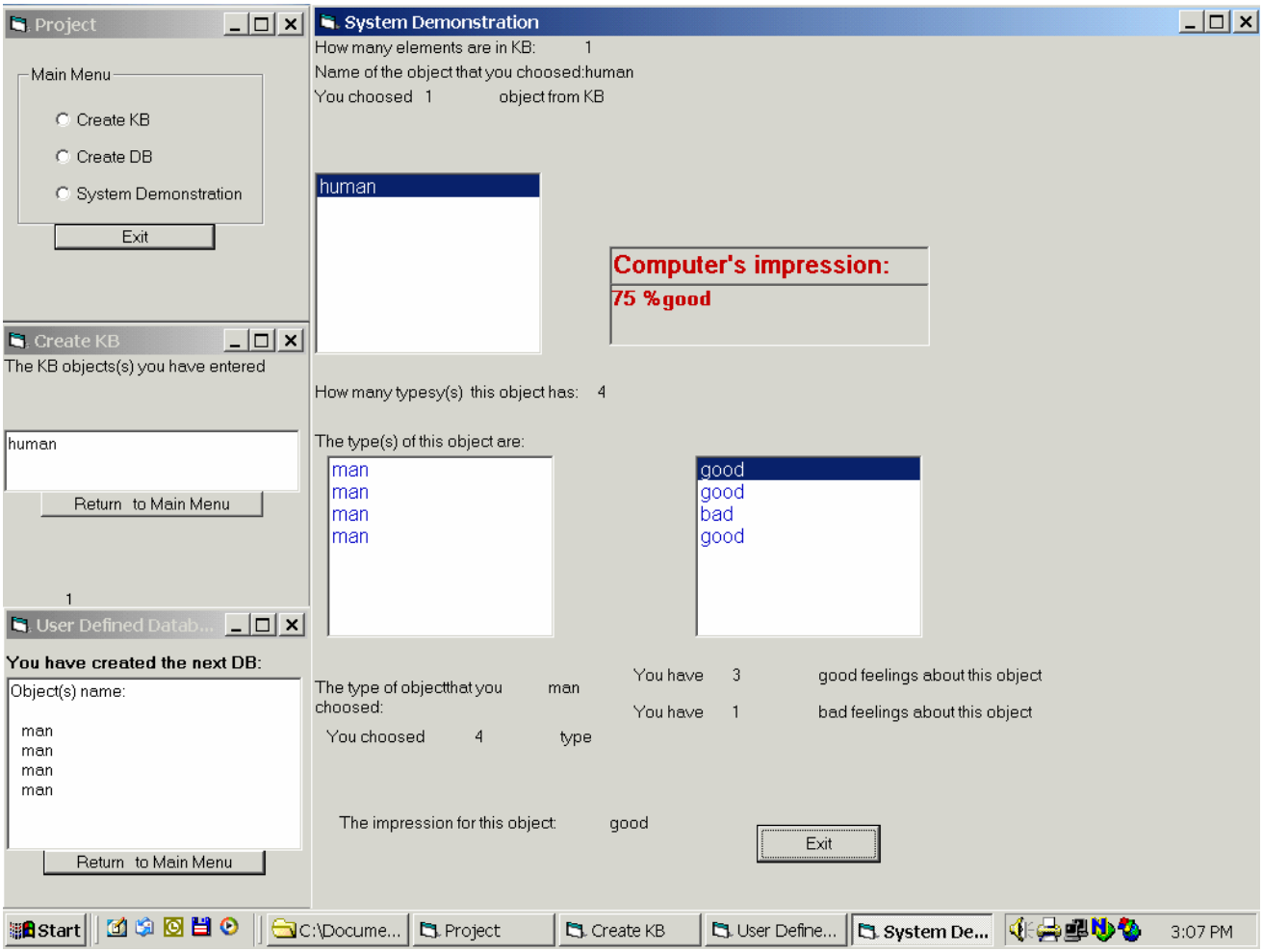


Fig 7. Modeling of INTUITION.

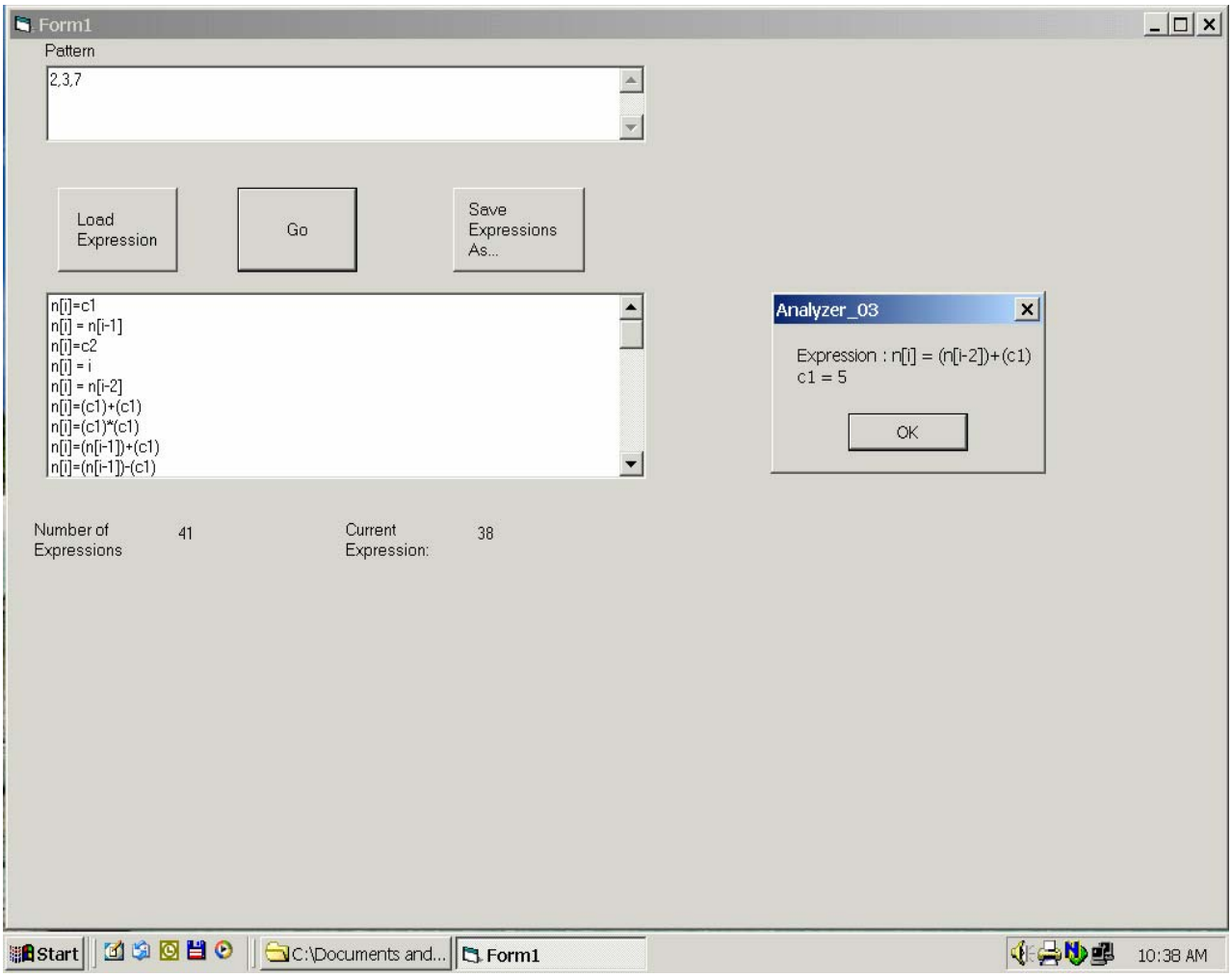


Fig 8. Hyphotheses generator. First 5 expressions are basic knowledge in math, rest of the expressions are generated as the heuristics to mach to the presented set of numbers. Analyzer shows correct expression. This system has a very limited basic math knowledge..

References

[1] Cawsey A. The Essence of Artificial Intelligence. Prentice Hall, 1995

[2] Dean T., Allen J., Aloimonos Y. Artificial Intelligence. Theory and Practice. The Benjamin/Cummings Publishing Company, 1995.

[3]. Gersting J. Mathematical Structures For Computer Science, W. H. Freeman and Co. 1999

[4] Negnevitsky M. Artificial Intelligence. A Guide to Intelligence Systems, Addison-Wesley, 2001

[5] Polyakov L. Structure Approach to the Intelligent System Design. Proceedings of the 2002 PerMIS Workshop August 13-15, 2002.

[6] Russell S. Norvig P. Artificial Intelligence. A Modern Approach. Prentice Hall, 1995

Miniaturized all-solid-state 3D camera for real-time range imaging

Thierry Oggier*, Peter Seitz and Nicolas Blanc

CSEM SA, Swiss Center for Electronics and Microtechnology, Badenerstrasse 569

CH 8048 Zurich, Switzerland

ABSTRACT

The perception of the world in three-dimensions is natural for human beings. Technical 3D imaging systems, however, have suffered until today from high complexity and severe practical limitations to obtain 3D-information of the environment. This is overcome by a new type of optical 3D camera based on the time-of-flight (TOF) principle: Light from an LED or laser diode array is RF modulated at a few tens of MHz and illuminates a scene. The light is diffusely reflected back by the objects in the scene, and it is imaged with a conventional lens onto a custom solid-state image sensor. Each of its pixels is capable of synchronous demodulation of the incident modulated light, for the precise local determination of the parameters offset, amplitude and phase. The phase information is a direct measure of the local distance.

This principle has been employed in a miniaturized 3D camera (SwissRanger) for the acquisition of range images in video real-time. Without any mechanical scanning parts and with eye-safe emitting power, the camera delivers distance data, intensity information as well as an estimation of the distance accuracy for each of its 124x160 pixels.

Due to the use of a combination of CCD principles and CMOS circuitry in each pixel, a distance resolution is obtained that is close to the physical noise limitations given by the photon shot noise. Under optimum conditions a distance resolution of a few mm over a measurement range of several meters is obtained. A large number of applications are envisaged for which our TOF range camera provides a cost-effective and simple solution.

Keywords: *Optical 3D imaging, range finder, SwissRanger, distance measurement, lock-in pixel, time-of-flight, range camera.*

1. INTRODUCTION

We live in a three-dimensional world. Human beings are able to perceive their environment thanks to the "human stereo vision" system, the eyes. Several technical approaches are known to render imaging systems more intelligent by letting conventional cameras obtain information about the third dimension. Methods based on the triangulation principle were developed in the past e.g. stereo vision systems or systems based on structured projection. So far, the required computational cost underlying such a triangulation system and the inherent need of a minimum base distance of the two sensors (the projection and sensing unit) prevented their introduction into markets that are demanding low cost and small camera size. Another approach based on interferometry achieves very high distance resolutions, but due to its short measurement range, interferometers are only employed in a very restricted number of applications. Very promising results are achieved with time-of-flight (TOF) measuring devices. Resolutions in the sub-centimeter range and measurement ranges of several hundred meters have been reported. The drawback of these TOF-systems lies mainly in the high emitted power that is required and the necessity to use moving mechanical parts (e.g. scanners), leading to system costs that are unacceptable for many applications.

Until recently, no reliable and cost-effective 3D imaging cameras have been available on the market. The optical 3D camera described in this work was developed targeting a new, cost-effective imaging system allowing to capture three-dimensional imagery of the world in video real-time. A cost-effective and robust camera dictates the use of an all-solid-state, application-specific image sensor obviating the need for any moving parts. Extensive optoelectronic characterization results prove that the achieved distance accuracies approach the ultimate resolution given by the photon shot noise [1].

In the first section of this work, the principle of optical distance-measuring cameras based on the TOF method is explained. An electromagnetic wave, modulated at a few tens of MHz is emitted by the camera's illumination unit, illuminating the entire scene. The light, diffusely reflected by the different objects in the scene, is imaged onto the camera's custom sensor by a conventional lens. Within each pixel of this sensor, the signal is demodulated synchronously. The local phase shift - caused by the imaged object's distance - as well as the intensity and the amplitude of the incoming electro-magnetic wave are obtained at each pixel site.

In the following section, the working principle and the requirements of the sampling process in the so-called "lock-

in pixel" [2] and the sensor are discussed. Extensive optoelectronic characterization results corroborate the theoretical predictions, proving that the SwissRanger camera yields distance data whose precision is close to the photon quantum noise limit.

In the fourth section different possible applications are described, in which the knowledge of the third dimension makes it possible to work with straightforward, simple algorithms. Some pictorial results of these algorithms illustrate the effectiveness of the chose approach based on range image data. The market potential of such cost-effective 3D-imaging devices in their different application fields is discussed and illustrated.

Finally an outlook for future improvements in the field of optical 3D cameras is given and their implications on the camera's performance is briefly discussed.

2. TOF MEASUREMENT PRINCIPLE

The TOF measurement principle is based on the finite speed of light $c \cong 3 \cdot 10^8\ ms^{-1}$ (in air). In order to reduce the required timing constraints on the system, instead of a pure pulse TOF measurement, a homodyne phase shift measurement is performed, as illustrated in Figure 1.

The principle is based on an illumination unit, which emits an intensity-modulated electromagnetic wave front. This emission signals can be described as

$$e(t) = e \cdot \left[1 + \sin\left(2\pi f \cdot t\right)\right] \qquad (1)$$

e: Emitted mean optical power
f: Modulation frequency

After reflection on the objects in the scene, the wave front reaches the sensor again. The power impinging on the sensor is reduced by different target characteristics (reflectivity, distance) and optical properties of the camera. The total signal attenuation can be summarized as a factor *k*. In addition to the modulated light, background light *BG* e.g. sun light or artificial light is sensed. Compared to the high frequency of the modulated signal, the background illumination can be considered as being constant. Therefore, the signal power impinging on a pixel can be described as

$$s(t) = BG + e \cdot k \cdot \left[1 + \sin\left(2\pi f \cdot t - \varphi\right)\right] \qquad (2)$$

BG: Background illumination power
k: Attenuation factor
φ: Phase delay arising from the object's distance

By sampling the incoming signal four times within a modulation period, the incident signal's modulation parameters can be completely determined, by making use of Equ. (3), (4), and (5). This calculation procedure is known as the four-bucket algorithm [3]. The four sampled and accumulated photo charge signals are denoted as A_0, A_1, A_2 and A_3.

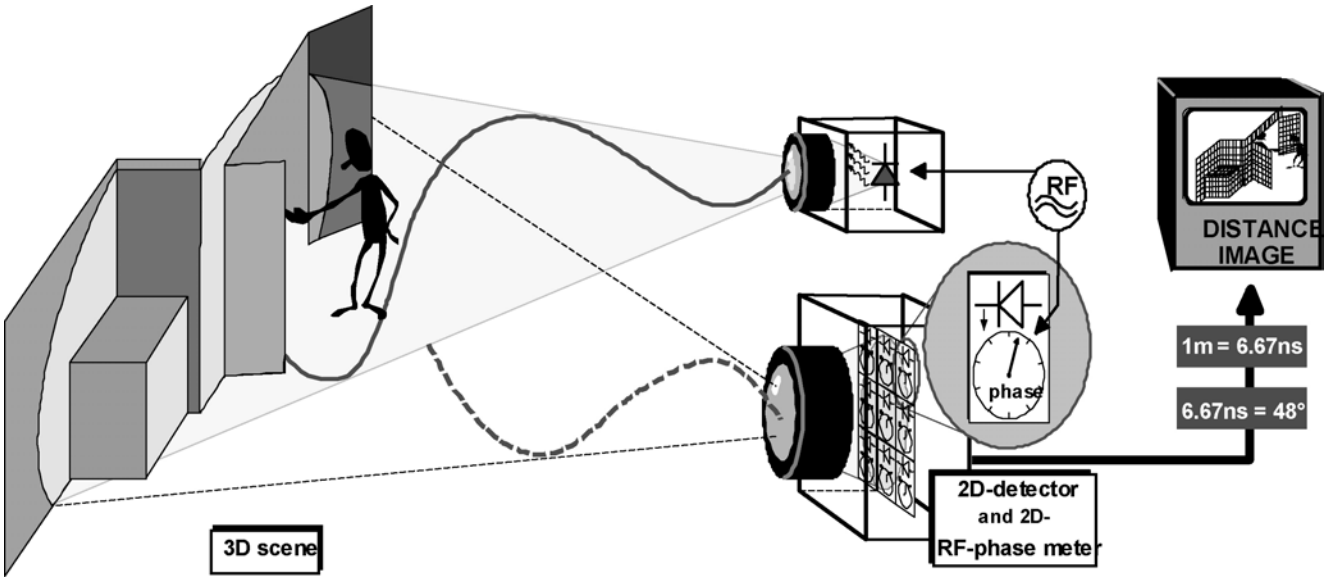


Figure 1: Measurement principle of the optical TOF 3D range camera. High-frequency modulated light is emitted by an LED array, it is reflected diffusely by the object, and it is demodulated synchronously in a lock-in image sensor.

$$\varphi = \operatorname{atan}\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \qquad (3)$$

$$B = \frac{A_0 + A_1 + A_2 + A_3}{4} \qquad (4)$$

$$A = \frac{\sqrt{[A_3 - A_1]^2 + [A_0 - A_2]^2}}{2} \qquad (5)$$

φ: Measured phase delay
B: Measured offset
A: Measured amplitude

An illustration of the sampling process is sketched in figure 2. The phase φ represents a direct measure of the acquired target distance, *B* corresponds to a conventional black and white intensity image and *A* is the amplitude of the incoming wave.

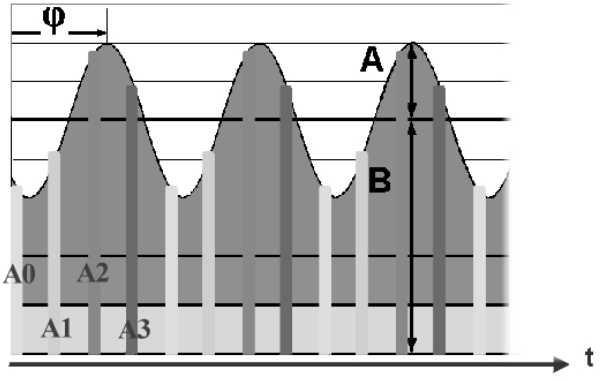


Figure 2: Illustration of the four samplings of the wave A0, A1, A2 and A3 and the reconstituted wave characteristics amplitude A, intensity B and phase φ.

The distance can be derived from the phase according to

$$L = \frac{L_0}{2\pi} \cdot \varphi \qquad (6)$$

L_0: Non-ambiguity range

L_0 represents the non-ambiguity range of the phase measurement and corresponds to half the wavelength of the modulation frequency (7).

$$L_0 = \frac{c}{2f} \qquad (7)$$

A more detailed discussion of homodyne phase measurement and its physical detection limit given by the photon shot noise is given in [4].

3. SWISSRANGER CAMERA DEMONSTRATOR

Figure 1 illustrates the different components required for the distance-measuring device. The camera consists of:

- Illumination unit
- Optical elements for imaging and filtering
- Custom sensor
- Sensor control electronics
- Data processing electronics
- Camera interface

Based on the described synchronous demodulation image sensor and the optical TOF range imaging principle, a miniaturized camera demonstrator (SwissRanger) was developed, shown in Fig. 3. Apart from the custom-made sensor, the SwissRanger camera only consists of commercially available components.

Its dimensions are 135 mm x 45 mm x 32 mm and the camera weighs less than 200 g, of which more than half is the weight of the metal case. The camera offers a lateral resolution of 160 x 124 pixels and during operation it consumes about 1.5 A at 12 V, depending on the particular measurement settings. Due to the use of a solid-state image sensor and the commercial availability of all other microelectronic components, the camera's measurement principle is well suited for the fabrication of cost-effective products.

The illumination unit emits an intensity-modulated light wave of a few 10 MHz into the entire field-of-view (FOV). The total emitted mean power amounts to up to 800 mW. LEDs emitting at a wavelength of 870 nm are employed. In general, laser diodes or LEDs can be used in the illumination unit. In any case, the maximal optical power limit is given by eye safety considerations. Implementing diffusive micro-optical elements in front of the emitting diodes eases on the one hand the eye safety limitations and on the other hand illuminates the FOV in a more controlled manner.

Figure 3: Photograph of the SwissRanger camera demonstrator.

The optical elements consist of the imaging optics in front of a bandpass filter. The imaging optics forms an image of the scene in the sensor plane, and the filter reduces possible background illumination in the scene. The imaging optics has to be adjusted carefully to the illumination unit and the sensor. Ideally, the illumination unit only emits light in the FOV determined by the optics and the sensor. The filter also needs to be designed according to the light source. LEDs require a broader bandpass filter than laser diodes. Therefore, for outdoor applications laser diodes allowing the implementation of a more narrow bandpass filter are preferred to LEDs.

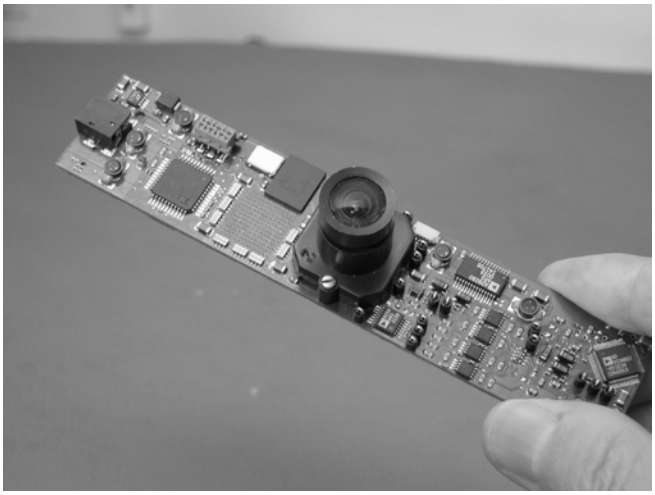


Figure 4: Electronics board of the SwissRanger camera demonstrator.

The electronics board provides the different signals to control the sensor. These signals are required for the readout, supply voltages or gate controls. The analog to digital conversion is performed on the printed circuit board, as well as the entire data processing, implementing the equations (3), (4) and (5). The camera settings allow the definition of a noise-level threshold to withhold inaccurate distance measurements from being made available outside the camera. The interface to the computer conforms to the widely used USB2.0 standard. Different parameters such as integration time, definition of region-of-interest (ROI) or spatial filtering can be programmed by the user through the USB2.0 interface.

4. 3D LOCK-IN IMAGE SENSOR

The application-specific image sensor has been designed, simulated and manufactured in a 0.8 µm CMOS/CCD technology from the silicon foundry ZMD in Dresden, Germany. The sensor contains 160 x 124 pixels. Each pixel can be addressed and read out individually, thus, any arbitrary ROI can be defined. The sensor is based on the so-called 2-tap pixel architecture, implying that each pixel contains two photocharge storage sites. This architecture represents a trade-off between speed (4-tap) and sensitivity (1-tap). A more detailed comparison about the different multi-tap pixel architectures is reported in [5].

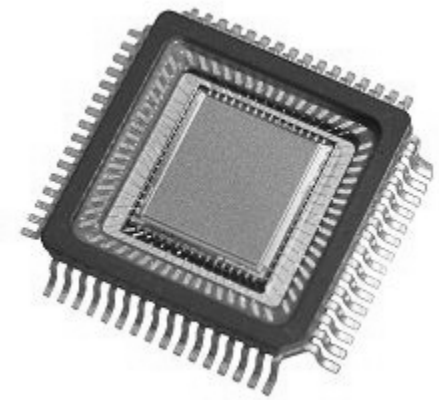


Figure 5: Picture of the SwissRanger sensor manufactured on a 0.8 µm CMOS/CCD technology.

The pixel has been created using ISE-TCAD simulation tools. The CCD-gates within each pixel allow fast photocharge separation, temporal sampling and accumulation. More details about CCD imaging have been published in [6] and [7]. Because of the high modulation frequencies and the requirements in TOF-systems regarding the shutter, dominating the demodulation characteristics, it is of highest importance to transfer and sum the photocharges with as little additional noise as possible. For this reason, we employ the CCD principle, realized with a buried channel CCD option in an otherwise standard CMOS process for virtually noise-free photocharge separation, sampling and accumulation. The same CMOS technology is used to address and read out the pixels.

This combination of CMOS and CCD technology does not only result in a pixel performance that is optimized for optical TOF range imaging, it also opens the way to future

system-on-chip (SoC) solutions of cost-effective single-chip TOF range cameras.

Extensive optoelectronic characterization confirms that the SwissRanger camera demonstrator approaches the physical detection limit given by the photon shot noise. Under favorable measurement conditions (little background illumination and pixel signals close to saturation) a distance resolution of a few mm can be obtained for a measurement range of several meters [1].

5. APPLICATION EXAMPLE

Thanks to the advantageous properties of the SwissRanger camera (high distance resolution of TOF-measurements, high lateral resolution and speed) completely new solutions to difficult measurement problems can be envisaged. A large number of applications fields can be covered in a more cost-efficient way.

In this paragraph, we describe a few application examples of optical 3D range cameras based on the TOF principle. Typical pictures illustrate the feasibility of the 3D imaging approach for these problems.

Figure 6 shows three views of one 3D picture of a human face taken by a SwissRanger camera.

Figure 6: Shot of the SwissRanger camera on a human face.

The 3D picture was taken with an exposure time of less than one second. The black-and-white reflectivity information is projected onto the measured three-dimensional surface. This results implies that optical 3D imaging can provide valuable additional information to the reflectivity map of a human face, without any special requirements, in a short time and at low cost. It seems obvious that biometrical applications could make good use of this additional information for increased security.

Another application example is motion tracking. The knowledge of the third dimension allows implementing much faster and easier detection algorithms. Figure 7 shows the hand tracking of a person in a room. Three pictures of a video-rate sequence are presented.

Figure 7: 1st row: b/w coded distance map. 2nd row: b/w reflectance image. 3rd row: hand tracking.

In the first row of figure 7, the distance map is intensity coded. Pixels considered not accurate enough for the hand tracking are withheld by the camera at pixel level. These pixels are represented in the first and second row as black pixels. Pixels in the first image row represent the measured distance. Dark pixels correspond to closer distance whereas bright pixels originate from objects farther away from the camera. The second row shows the conventional b/w reflectance images of the sequence. The third row shows those pixels in white that are extracted by a simple hand-tracking algorithm.

Information about the third dimension makes it much easier to develop fast, reliable algorithms of low complexity, for which the presented hand tracking is a good example: Algorithms based on the distance maps save a lot of computational power compared to pure 2D algorithms.

Figure 8 shows a sequence of range images of a scene, acquired at video-rate. The distance on the different maps is

b/w coded. Objects with darker pixels are closer to the camera; brighter objects are situated farther from the camera. For illustration, the used colour-coded distance scale is provided. The sequence shows a person entering a room. This person can easily be tracked and detected by processing the distance information. Any possible influences by lighting effects, such as shadows, do not influence the distance map, in strong contrast to the detection algorithms based on conventional 2D-sensors.

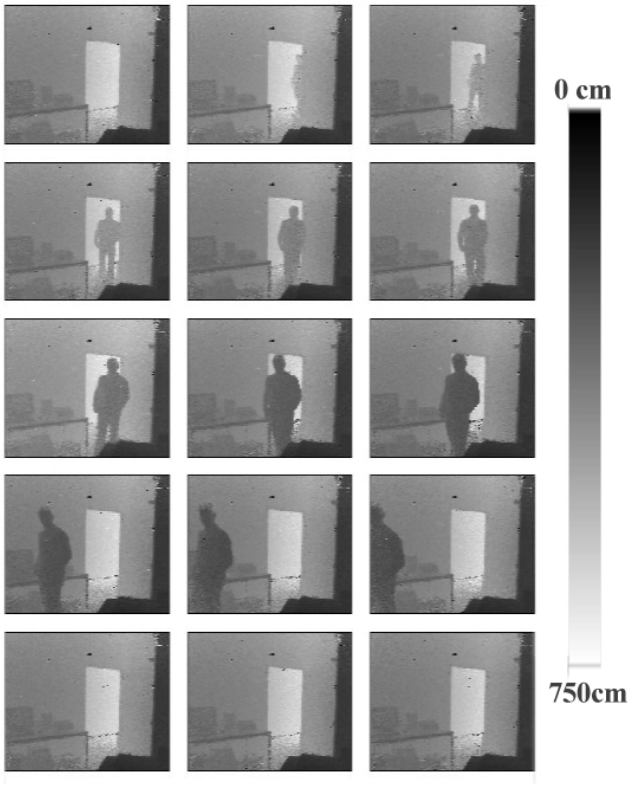


Figure 8: Sequence of distance maps acquired by the SR-2 camera for room observation.

6. OUTLOOK

The next generation of 3D-TOF cameras will be based on CMOS/CCD technologies with smaller feature sizes. This will increase the number of lock-in pixels and their density on one custom TOF image sensor. By using laser diodes and vertical cavity surface emitting lasers (VCSEL), outdoor applications are targeted, first at closer distances and then at distances of several tens of meters. New pixel architectures and technologies will allow higher modulation frequencies [8] and thus further improve the distance resolution. Finally, the use of microlenses is foreseen in order to increase the pixel sensitivity.

7. SUMMARY AND CONCLUSIONS

In this paper, an overview of the optical homodyne phase measurement technique used for distance measurements is given. The TOF-principle as implemented in the SwissRanger camera demonstrator is presented and the different required components are described.

The potential of the optical 3D TOF imaging principle is illustrated with three examples, face recognition, hand tracking and room observation, requiring only straight-forward and simple algorithms. Many more applications are foreseen in a wide range of applications, such as automotive, security, safety, door-and-gate control, robotics, autonomous vehicles, computer pointing devices, domotics, games, biometrics, toys, etc.

It is concluded that the described optical 3D imaging as implemented in the SwissRanger camera demonstrator represents an outstanding measurement concept for acquiring true images of our rich three-dimensional world, that can finally been acquired at high speed, high resolution, with miniature camera systems and at low cost.

ACKNOWLEDGMENT

The authors would like to thank all the members of the Image Sensing Group of CSEM SA for their continuing support in the various phases of this research and development project.

REFERENCES

[1] T. Oggier et al., "An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger^{TM})", *Proceedings of the SPIE,* Vol. 5249 No. 65, 2003.

[2] T. Spirig, P. Seitz, O. Vietze, F. Heitger, „The Lock-in CCD – Two-Dimensional Detection of Light", *IEEE journal of quantum electronics*, Vol.31, No. 9, 1995.

[3] K. Creath, "Phase-Measurement Interferometry Techniques", *Progress in Optics,* Vol. XXVI, E. Wolf (Ed.), Elsevier, 1988.

[4] R. Lange and P. Seitz, "Solid-State Time-of-Flight Range Camera", *IEEE J. Quantum Electronics*, Vol. 37 (3), 390-397, March 2001.

[5] W. Boyle and G. Smith, "Charge coupled semiconductor devices", *Bell Syst. Tech. Jour.*, Vol. 49, pp. 587-593, 1970.

[6] R. Kaufmann et al., "A Time-of-Flight Line Sensor – Development and Application", *Proceedings of the SPIE*, Vol. 5459 No. 24, 2004.
[7] A. Theuwissen, "Solid-State Imaging with Charge-Coupled Devices", *Kluwer Academic Publishers*, 1995.
[8] B. Büttgen, T. Oggier, R. Kaufmann, P. Seitz, N. Blanc, "Demonstration of a novel drift field pixel structure for the demodulation of modulated light waves with application in three-dimensional image capture", *Proceedings of the SPIE,* Vol. 5302 No. 2, 2004

* Contact: thierry.oggier@csem.ch

Obstacle Detection using a TOF Range Camera for Indoor AGV Navigation

T. Hong, R. Bostelman, and R. Madhavan
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230, U.S.A.
Tel: (301) 975-2865 Fax: (301) 990-9688
Email: {tsai.hong, roger.bostelman, raj.madhavan}@nist.gov

ABSTRACT— The performance evaluation of an obstacle detection and segmentation algorithm for Automated Guided Vehicle (AGV) navigation in factory-like environments using a 3D real-time range camera is the subject of this paper[1]. Our approach has been tested successfully on British safety standard recommended object sizes and materials placed on the vehicle path. The segmented (mapped) obstacles are then verified using absolute measurements obtained using a relatively accurate 2D scanning laser rangefinder.

Keywords:

Automated Guided Vehicle, 3D range camera, 2D Laser Rangefinder, Obstacle Detection and Segmentation.

1. INTRODUCTION

Obstacle detection and mapping are crucial for autonomous indoor driving. This is especially true for Automated Guided Vehicle (AGV) navigation in factory-like environments where safety of personnel and that of the AGV itself is of utmost importance. This paper describes the performance of an obstacle detection and segmentation algorithm using a 3D real-time range camera.

The 3D range image camera is based on the Time-Of-Flight (TOF) principle [8] and is capable of simultaneously producing intensity images and range information of targets in indoor environments. This range camera is extremely appealing for obstacle detection in industrial applications as it will be relatively inexpensive as compared to similar sensors and can deliver range and intensity images at a rate of 30 Hz with an active range of 7.5 m.

Since obstacle detection plays a basic function for autonomous driving, there has been much research on many different types of sensors, such as sonar [11], color/gray level cameras [2], FLIR (Forward Looking InfraRed) cameras [10], and stereo cameras [9], [1], [12], [6]. Most of the vision approaches are not applicable to indoor scenes due to lack of texture in the environment. Other researchers have proposed LADAR (Laser Detection And Ranging) sensors for detecting obstacles [4], [3], [5]. However, one dimension LADAR which has been used in AGV industry is not suitable for the 3D world of factory environments.

Our proposed approach to obstacle detection uses a low cost, 3D real-time range camera. First, we calibrate the camera with respect to the AGV so that we can convert the range values to 3D point clouds in the AGV coordinate frame. Second, we segment the objects which have high intensity and whose elevation values are above the floor of the operating environment on the AGV path. The segmented 3D points of the obstacles are then projected and accumulated into the floor surface-plane. The algorithm utilizes the intensity and 3D structure of range data from the camera and does not rely on the texture of the environment. The segmented (mapped) obstacles are verified using absolute measurements obtained using a relatively accurate 2D scanning laser rangefinder. Our approach has been tested successfully on British safety standard recommended object sizes and materials placed on the vehicle path. In this paper, the AGV remained stationary as the measurements were collected.

The U.S. American Society of Mechanical Engineers (ASME) B56.5 standard [13] was recently upgraded[2] to allow non-contact safety sensors as opposed to contact sensors such as bumpers on AGVs. Ideally, the U.S. standard can be upgraded further similar to the British safety standard requirements [14]. The British safety standard of industrial driverless trucks/robots requires that (a) sensors shall operate at least over the full width of the vehicle and load in every direction of travel, (b) sensors shall generate a signal enabling the vehicle to be stopped by the braking system under specified floor condition before contact between the rigid parts of the vehicle and/or load and a person, (c) sensors shall detect parts of a persons body as close as possible to the floor but at least the

[1]Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

[2]not cited here as the upgrade was not published prior to the date of this paper.

"test apparatus shall be detected", (d) the activation of such sensors shall not cause injury to persons, and (e) reflective characteristics of test apparatus for personnel detection means which work without physical contact shall be representative of human clothing. We anticipate the work described in this paper and the continuing research efforts to lay the groundwork towards further upgrade of the U.S. safety standards for AGVs in factory-like environments.

The paper is structured as follows: Section 2 describes an obstacle detection and segmentation algorithm using range camera images. Section 3 provides the experimental results when the proposed algorithm is employed for detection and segmentation of British standard test apparatus. Section 4 concludes the paper and indicates future research areas that are under investigation.

2. OBSTACLE DETECTION AND SEGMENTATION

In this section, we describe an algorithm to detect and segment obstacles in the path of the AGV using a solid-state Time-Of-Flight (TOF) range camera. The 3D range camera shown in Figure 1 is a compact, robust and cost effective solid state device capable of producing 3D images in real-time. The camera has a field-of-view of 42° (horizontal) × 46° (vertical) and is capable of producing range images of 160×124 pixels. For a brief overview of the characteristics and operating principles of the camera, see [8]. Approximately sized British standard test obstacles, shown in Figure 2, were placed on the travel path.

Fig. 1. The TOF 3D range image camera. The camera simultaneously generates intensity images and range information of targets in its field-of-view at a rate of 30 Hz with an active range of 7.5 m.

The obstacle detection and segmentation algorithm combines intensity and range images from the range camera to detect the obstacles and estimate the distance to the obstacles. The steps of the algorithm are illustrated for a sample image from the camera:

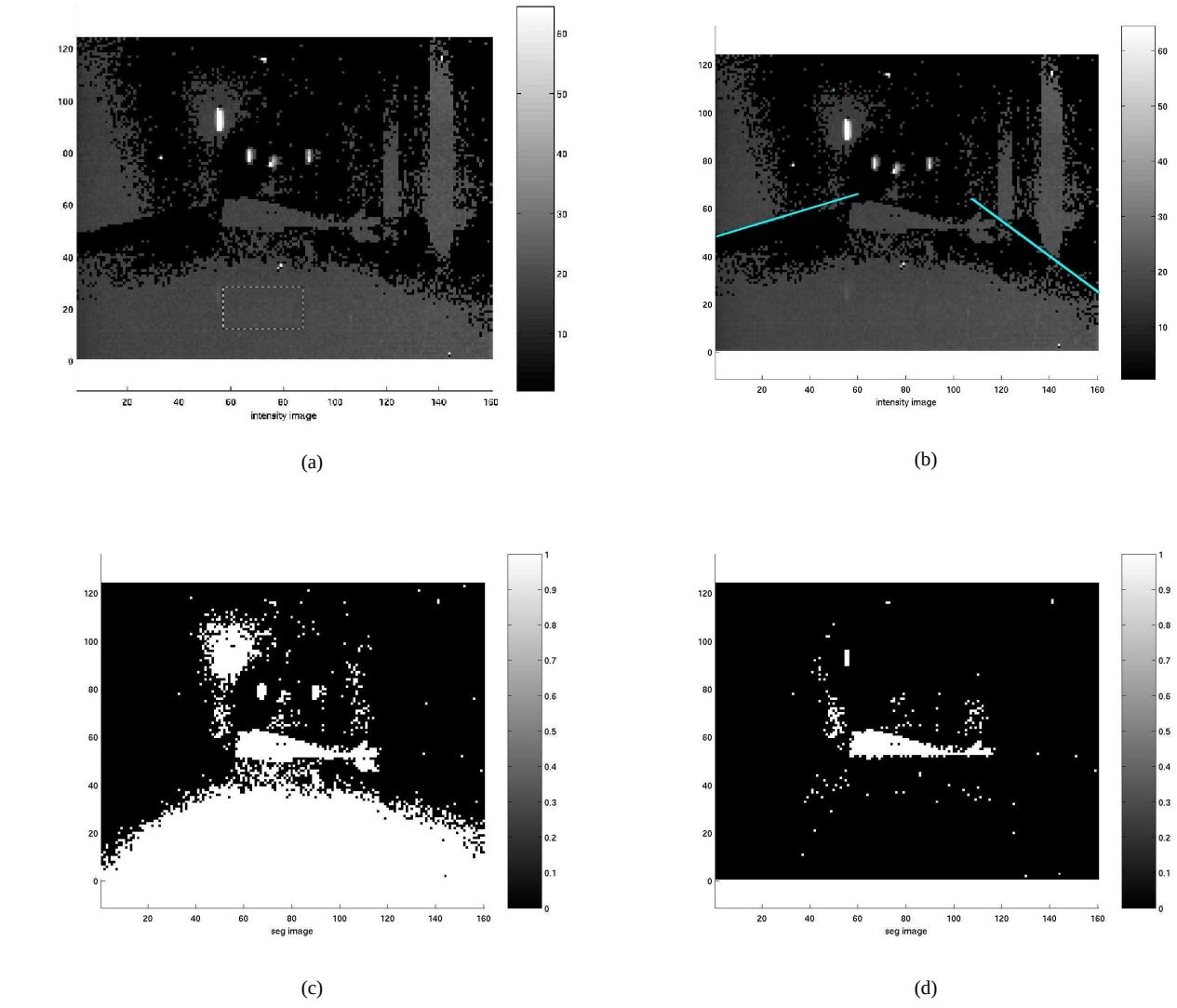
1) First, a patch data with high intensity values (i.e., the intensity value is greater than 20) in the front of the robot are used to fit a plane for estimating the floor surface as shown in Figure 3(a).
2) Second, the left and right edges of 3D robot paths are projected to the range and intensity images such that only obstacles on the path can be considered as shown in Figure 3(b).
3) Third, all the intensity pixels inside of the left and right edges are used to hypothesize the potential obstacle. If the intensity value of the pixel is greater than half of the average of the intensity in the image then the pixel is considered as a potential obstacle as shown in Figure 3(c).
4) Fourth, each potential obstacle pixel in the range image is used to find the distance to the floor plane when the distance to the floor is greater than some threshold as shown in Figure 3(d). The threshold is dependent on the traversability of the robot.

(a)

(b)

Fig. 2. Experimental setup. (a) and (b) depict the experimental setups that are described in this paper. See Section 3 for further details.

Potential obstacles in the world model can be accumulated as the AGV drives; Figure 4 shows an obstacle map representation that is part of the world model. The obstacles map is shown at 10 cm grid resolution. Nearly all the obstacles are found, but at the cost of false positives from the reflected objects. To increase the accuracy of our



(a) (b)

(c) (d)

Fig. 3. Obstacle segmentation algorithm illustration.

obstacle detection, the obstacles in the map and information obtained from an added color camera may be temporally integrated. Such integration has proven to be a very useful cue for obstacle detection [7].

3. EXPERIMENTAL SETUP AND RESULTS

The experiments were conducted under two scenarios as stated within the British Standard:

1) A test apparatus with a diameter of 200 mm and a length of 600 mm placed at right angles on the path of the AGV. The actuating force on this test apparatus shall not exceed 750 N.

2) A test apparatus with a diameter of 70 mm and a height of 400 mm set vertically within the path of the AGV. The actuating force on this test apparatus shall not exceed 250 N.

Figures 2(a) and (b) show the experimental setup for the two aforementioned scenarios. The center of the camera lens was centered approximately horizontal and vertical on the apparatus for all measurements. The scanning laser rangefinder was offset from the camera by 0 mm vertically, 250 mm horizontally, and to the left of the camera as viewed from the camera to the test apparatus. The range camera was used to detect known test apparatus mounted on a stand and moved to different locations with respect to the camera.

The obstacle detection and segmentation algorithm was tested on the British standard test apparatus as described in [14], and was evaluated against *ground truth*. A single-line scanning laser rangefinder, shown in Figure 5, mounted beside the range camera, was used to simultaneously verify the distance to the test apparatus for each data set and served as ground truth. The rangefinder produces 401 data

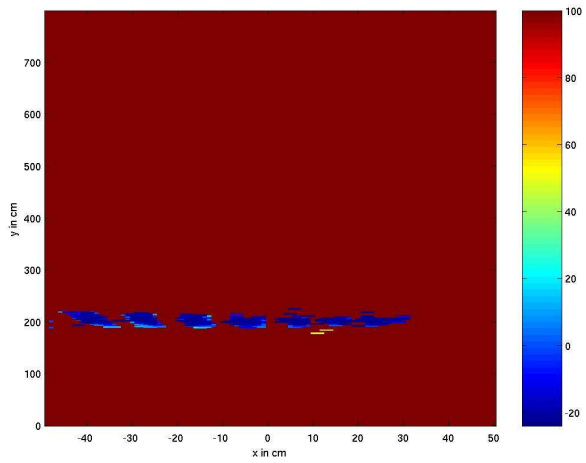


Fig. 4. Obstacle map.

Fig. 5. Experimental setup of the AGV, the scanning laser rangefinder, and the range camera.

points over a 100° semi-circular region in front of the robot.

The obstacle detection and segmentation algorithm was tested on British standard test apparatus which were placed in 0.5 meter to 7.5 m distances to the sensor. Table 1 shows the performance of the range camera for measuring the distance to the test apparatus placed at several distances from the range camera. As can be seen, the accuracy (mean) of the range decreases as the distance of the apparatus placed in front of the range camera is increased.

In Figure 6, the test apparatus was placed at a distance of 2.5 m from the range camera. Each object in the test apparatus was clearly detected even though the range camera was also sensitive to the reflectors on the wall of the hallway. The resultant intensity, range, and segmented images are shown in Figures 6(a), (b) and (c), respectively. The ground truth provided by the scanning laser rangefinder is shown in Figure 6(d) and has been rotated to show a top-down view.

In Figure 7, the test apparatus is a mannequin leg placed on the floor with an approximate diameter of 200 mm and a length of 600 mm. This test apparatus is more challenging for the algorithm because the entire object is close to the floor. As can be seen, the legs are detected, but at the cost of detecting reflectors. Since some reflectors (see Figure 7(c)) are at a distance of more than 7.5 m, these are modulated by the non-ambiguity distance range of the camera. This deficiency can be eliminated by using two different modulation frequencies (such as 10 MHz and 20 MHz) where the detected objects would be coarsely represented at a more appropriate distance. The control algorithm can then intelligently delete them.

4. CONCLUSIONS AND FURTHER WORK

An obstacle detection and segmentation algorithm for Automated Guided Vehicle (AGV) navigation in factory-like environments using a novel 3D range camera was described in this paper. The range camera is highly attractive for obstacle detection in industrial applications as it will be relatively cheap and can deliver range and intensity images in real-time. The performance of the algorithm was evaluated by comparing it with ground truth provided by a single-line scanning laser rangefinder.

We envisage the extension of the work detailed in this paper in the following areas:

- We believe that the range camera can be used for moving obstacle detection from a moving AGV. The detection of moving obstacles in the factory floor is a next critical step for AGV navigation in such dynamic environments. Additionally, this sensor can be combined with a color camera for detecting and tracking obstacles over long distances.
- We also believe that the range camera discussed in this paper holds good potential to be used in outdoor environments. Towards this, we have taken and analyzed some outdoor data and the preliminary results show good promise in using this sensor for outdoor forest environments. Some prospective applications include mapping factory environments ("lights-out") manufacturing, and even for use in space due to its compactness.

5. REFERENCES

[1] P. Batavia and S. Singh. Obstacle Detection Using Adaptive Color Segmentation and Color Stereo Homography. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, May 2001.

[2] M. Bertozzi, A. Broggi, A. Fascioli, and P. Lombardi. Artificial Vision in Road Vehicles. In *Proc. of the 28th IEEE Industrial Electronics Society Annual Conf.*, 2002.

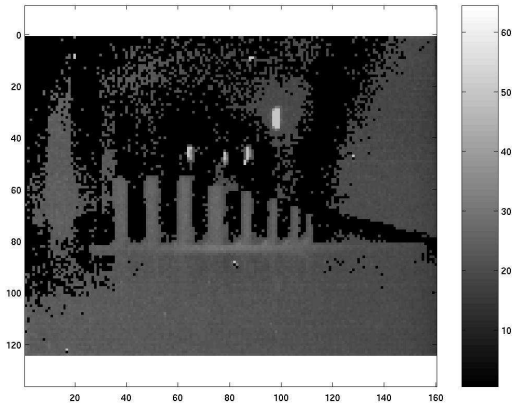
[3] T. Chang, T-H. Hong, S. Legowik, and M. Abrams. Concealment and Obstacle Detection for Autonomouos Driving. In *Proc. of the Intl. Association of Science and Technology for Development - Robotics and Application*, 1999.

[4] A. Ewald and V. Willhoeft. Laser Scanners for Obstacle Detection in Automotive Application. In *Proc. of the Intell. Vehicles Symp.*, 2000.

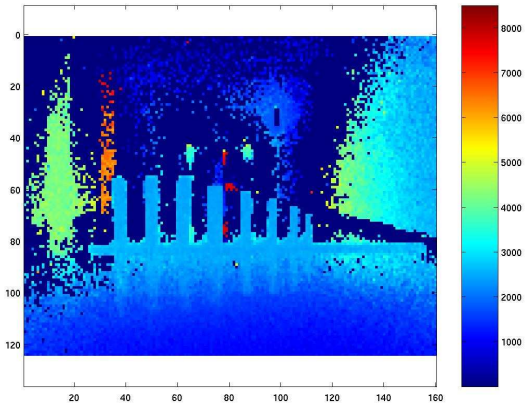
[5] J. Hancock, M. Hebert, and C. Thorpe. Laser Intensity-based Obstacle Detection. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 1998.

Table 1
Quantitative Comparison of Performance

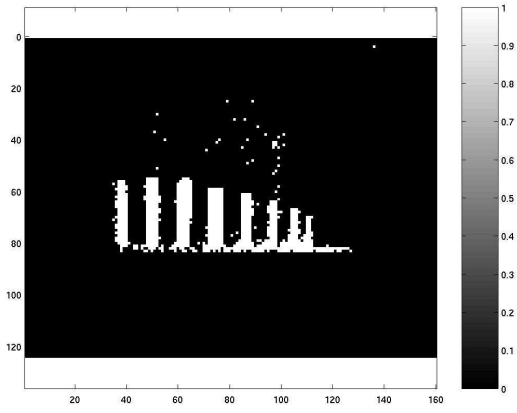
Nominal Obst. Dist. [cm]	3D Range Camera Mean [cm]	2D Rangefinder Mean [cm]
64	64	65
111	111	111
160	161	161
210	204	210
259	249	259
310	284	310



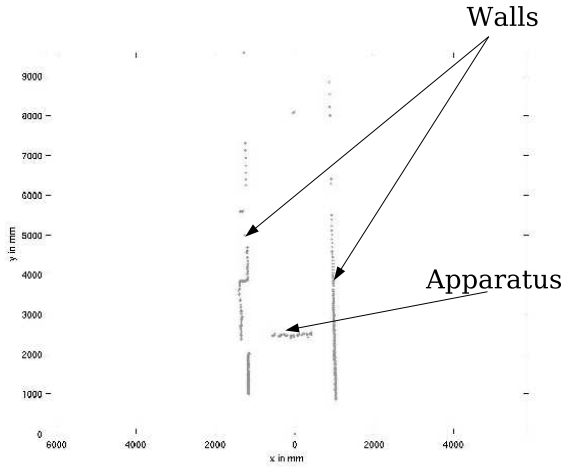
(a) Intensity Image



(b) Range Image

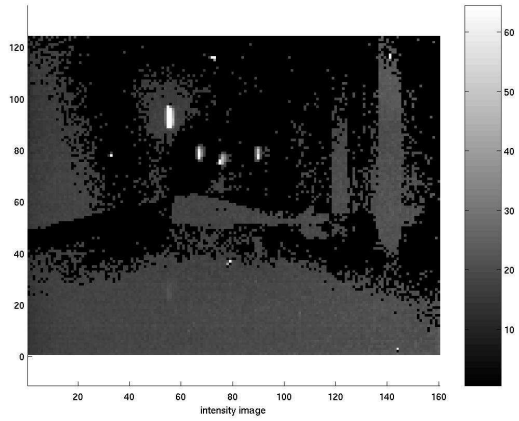


(c) Segmented Image

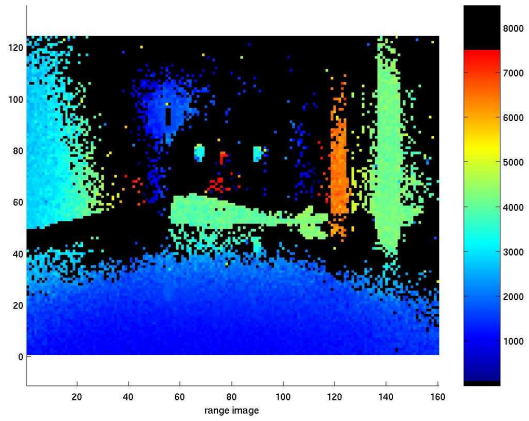


(d) Ground Truth

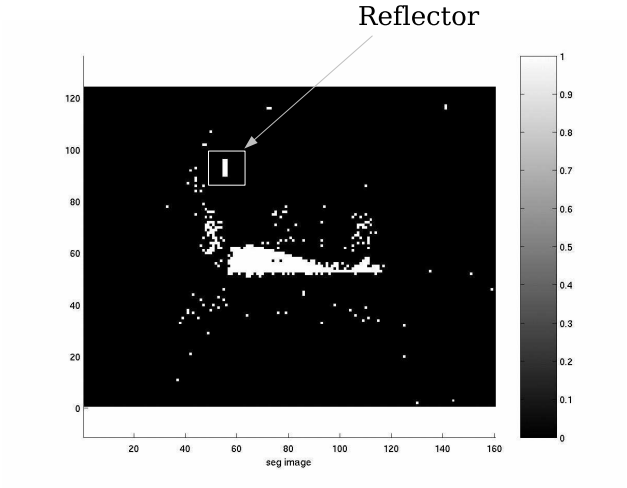
Fig. 6. Results of the obstacle detection and segmentation algorithm for the experimental setup shown in Figure 2(a). The resultant intensity, range, and segmented images are shown in (a), (b) and (c), respectively. The ground truth provided by the scanning laser rangefinder is shown in (d) and has been rotated to show a top-down view.



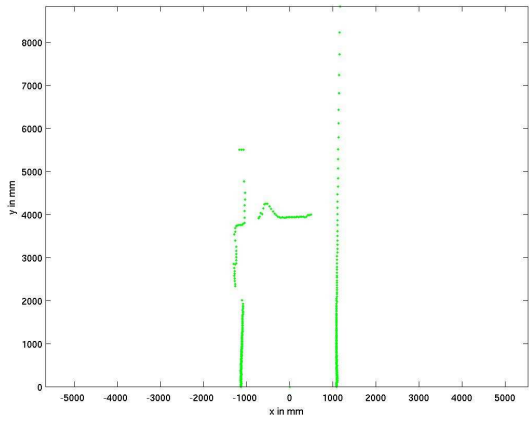
(a) Intensity Image



(b) Range Image



(c) Segmented Image



(d) Ground Truth

Fig. 7. Results of the obstacle detection and segmentation algorithm for the experimental setup shown in Figure 2(b). The resultant intensity, range, and segmented images are shown in (a), (b) and (c), respectively. The ground truth provided by the scanning laser rangefinder is shown in (d) and has been rotated to show a top-down view.

[6] M. Hariti, Y. Ruichek, and A. Koukam. A Voting Stereo Matching Method for Real-Time Obstacle Detection. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 2003.

[7] T-H. Hong, T. Chang, C. Rasmussen, and M. Shneier. Feature Detection and Tracking for Mobile Robots Using a Combination of Ladar and Color Images. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, May 2002.

[8] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc. An All-solid-state Optical Range Camera for 3D Real-time Imaging with Sub-centimeter Depth Resolution. In *Proc. of the SPIE Conf. on Optical System Design*, September 2003.

[9] C. Olson, L. Matthies, H. Schoppers, and M. Maimone. Robust Stereo Ego-motion for Long Distance Navigation. In *Proc. of the IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2000.

[10] K. Owens and L. Matthies. Passive Night Vision Sensor Comparison for Unmanned Ground Vehicle Stereo Vision Navigation . In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*.

[11] N. Sgouros, G. Papakonstantinous, and P. Tsanakas. Localized Qualitative Navigation for Indoor Environments. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1996.

[12] R. Williamson and C. Thorpe. A Trinocular Stereo System for Highway Obstacle Detection. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1999.

[13] American Society of Mechanical Engineers' Safety Standard for Guided Industrial Vehicle and Automated Functions of Manned Industrial Vehicle. Technical Report ASME B56.5, 1993.

[14] British Standard Safety of Industrial Trucks - Driverless Trucks and their Systems. Technical Report BS EN 1525, 1998.

Performance Evaluation of Temporal Range Registration for Unmanned Vehicle Navigation

R. Madhavan and E. Messina
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230.
Tel: (301) 975-2865 Fax: (301) 990-9688
Email: {raj.madhavan, elena.messina}@nist.gov

ABSTRACT— In this paper, we evaluate the performance of an iterative registration algorithm for position estimation of Unmanned Ground Vehicles (UGVs) operating in unstructured environments. Field data obtained from trials on UGVs traversing undulating outdoor terrain is used to quantify the performance of the algorithm in producing continual position estimates. These estimates are then compared with those provided by ground truth to facilitate the performance evaluation of the algorithm. Additionally, we propose performance measures for assessing the quality of correspondences. These measures, collectively, provide an indication of the quality of the correspondences thus making the registration algorithm more robust to outliers as spurious matches are not used in computing the incremental transformation.

Keywords:
UGV, range images, registration, uncertainty, performance measures.

1. INTRODUCTION

Active range sensing has become an integral part of any unmanned vehicle navigation system due its ability to produce unambiguous, direct, robust, and precise images consisting of range pixels. This is in direct contrast to passive sensing where the inference of range largely remains computationally intensive and not robust enough for use in natural outdoor environments. Depending on the speed of the vehicle, operating environment, and data rate, such range images acquired from a moving platform need to be registered to make efficient use of information contained in them for various navigation tasks including map-building, localization, obstacle avoidance, and control.

Iconic methods that attempt to minimize the discrepancies between sensed data and a model of the environment have been utilized for range registration. The attraction of these methods lies in the fact that the matching works directly on data points. Because the search is confined to small perturbations of the range images, it is computationally efficient. For example, Kanade et al. [3] compared elevation maps obtained from 3D range images to determine vehicle location. A similar iconic approach has also been adopted by Shaffer [10] but it does not take into account the uncertainty associated with the observation data.

We have developed a temporal iterative algorithm for registering range images obtained from unmanned vehicles. Formally, the process of registration is defined as follows: Given two sets of range images (model set: $\mathbf{M}$ and data set: $\mathbf{D}$), find a transformation (rotation and translation) which when applied to $\mathbf{D}$ minimizes a distance measure between the two point sets. Despite the apparent simplicity of the problem, to register range images from unmanned vehicles traversing unstructured environments, the terrain of travel, sensor noise and determination of accurate correspondences make it quite challenging.

The registration algorithm is a modified variant of the well-known Iterative Closest Point (ICP) algorithm [1]. At each iteration, the algorithm determines the closest match for each point and updates the estimated position based on a least-squares metric with some modifications to increase robustness. The modified algorithm has been shown to be robust to outliers and false matches during the registration of 3D range images obtained from a scanning LADAR rangefinder on an Unmanned Ground Vehicle (UGV) and also towards registering LADAR images from the UGV with those from an Unmanned Aerial Vehicle (UAV) that flies over the terrain being traversed [9]. For completeness, the temporal iterative registration algorithm is summarized in Section 2.

In this paper, we evaluate the performance of the registration algorithm for position estimation of UGVs operating in unstructured environments. Field data obtained from two trials on UGVs traversing undulating outdoor terrain is used to quantify the performance of the algorithm in producing continual position estimates. Using the data obtained from the first trial, the iterative registration algorithm aids the position estimation process whenever Global Positioning System (GPS) estimates are unavailable or are below required accuracy bounds. In the second trial, ICP is combined with a post-correspondence Extended Kalman Filter (EKF) to account for uncertainty present in the range images. For both the trials, the position estimates are then compared with those provided by ground truth to facilitate the performance evaluation of the registration algorithm. In addition,

we propose performance measures for assessing the quality of correspondences. These measures, collectively, provide an indication of the quality of the correspondences thus making the registration algorithm more robust to outliers as spurious matches are not used in computing the incremental transformation. The registration algorithm is then combined with proposed performance metrics and compared to the traditional ICP algorithm in terms of accuracy and speed.

The paper is structured as below: Section 2 describes the iterative temporal registration algorithm. Section 3 presents experimental results when the iterative algorithm is used for obtaining position estimates. Section 3.1 compares registration-aided position estimates with those provided by GPS. Section 3.2 details a map-aided registration algorithm for pose estimation. Section 4 develops performance measures for quality assessment of correspondences within the registration process and provides the associated results. Section 5 provides the conclusions and outlines areas of continuing research.

2. ITERATIVE TEMPORAL REGISTRATION ALGORITHM

The process of registration is stated formally as:

$$min_{(\mathbf{R},\mathbf{T})} \sum_i ||\mathbf{M}_i - (\mathbf{R}\mathbf{D}_i + \mathbf{T})||^2 \quad (1)$$

where $\mathbf{R}$ is the rotation matrix, $\mathbf{T}$ is the translation vector and the subscript i refers to the corresponding points of the sets $\mathbf{M}$ and $\mathbf{D}$.

2.1. Iterative Closest Point Algorithm

The ICP algorithm can be summarized as follows: Given an initial motion transformation between the two point sets, a set of correspondences are developed between data points in one set and the next. For each point in the first data set, find the point in the second that is closest to it under the current transformation. It should be noted that correspondences between the two point sets is initially unknown and that point correspondences provided by sets of closest points is a reasonable approximation to the true point correspondence. From the set of correspondences, an incremental motion can be computed facilitating further alignment of the data points in one set to the other. This find correspondence/compute motion process is iterated until a predetermined threshold termination condition.

In its simplest form, the ICP algorithm can be described by the following steps:

1. For each point in data set $\mathbf{D}$, compute its closest point in data set $\mathbf{M}$. In this paper, this is accomplished via nearest point search from the set comprising $N_{\mathbf{D}}$ data and $N_{\mathbf{M}}$ model points.
2. Compute the incremental transformation ($\mathbf{R}$,$\mathbf{T}$) using Singular Value Decomposition (SVD) via correspondences obtained in step **1.**
3. Apply the incremental transformation from step **2.** to $\mathbf{D}$.
4. If relative changes in $\mathbf{R}$ and $\mathbf{T}$ are less than a threshold, terminate. Else go to step **1.**

To deal with spurious points/false matches and to account for occlusions and outliers, we modify and weight the least-squares objective function in Equation (1) such that [11]:

$$min_{(\mathbf{R},\mathbf{T})} \sum_i w_i \ ||\mathbf{M}_i - (\mathbf{R}\mathbf{D}_i + \mathbf{T})||^2 \quad (2)$$

If the Euclidean distance between a point x_i in one set and its closest point y_i in the other, denoted by $d_i \triangleq d(x_i, y_i)$, is bigger than the maximum tolerable distance threshold $\mathcal{D}_{max}$, then w_i is set to zero in Equation (2). This means that an x_i cannot be paired with a y_i since the distance between reasonable pairs cannot be very big. The value of $\mathcal{D}_{max}$ is set adaptively in a robust manner by analyzing distance statistics.

Let $\{x_i, y_i, d_i\}$ be the set of original points, the set of closest points and their distances, respectively. The mean and standard deviation of the distances are computed as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} d_i; \quad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \mu)^2}$$

where N is the total number of pairs.

The pseudo-code for the *Adaptive Thresholding* (AT) of the distance $\mathcal{D}_{max}$ is given below:

$$\begin{array}{c} \texttt{if } \mu < \mathcal{D} \\ \mathcal{D}^{itn}{}_{max} = \mu + 3\sigma\texttt{;} \\ \texttt{elseif } \mu < 3\mathcal{D} \\ \mathcal{D}^{itn}{}_{max} = \mu + 2\sigma\texttt{;} \\ \texttt{elseif } \mu < 6\mathcal{D} \\ \mathcal{D}^{itn}{}_{max} = \mu + \sigma\texttt{;} \\ \texttt{else } \mathcal{D}^{itn}{}_{max} = \epsilon\texttt{;} \end{array}$$

where itn denotes the iteration number and $\mathcal{D}$ is a function of the resolution of the range data.

During implementation, $\mathcal{D}$ was selected based on the following two observations:

1) If $\mathcal{D}$ is too small, then several iterations are required for the algorithm to converge and several good matches will be discarded, and
2) If $\mathcal{D}$ is too big, then the algorithm may not converge at all since many spurious matches will be included. The interested reader is referred to [11] for more details on the effect and selection of $\mathcal{D}$ and ϵ on the convergence of the algorithm.

At the end of this step, two corresponding point sets, $\mathbf{P_M}$:$\{\mathbf{p}_i\}$ and $\mathbf{P_D}$:$\{\mathbf{q}_i\}$ are available.

The incremental transformation (rotation and translation) of step **2.** is obtained as follows:

- Calculate $\mathbf{H}$=$\sum_{i=1}^{N_D}(\mathbf{p}_i - \mathbf{p}_c)(\mathbf{q}_i - \mathbf{q}_c)^T$; ($\mathbf{p}_c$,$\mathbf{q}_c$) are the centroids of the point sets ($\mathbf{P_M}$,$\mathbf{P_D}$).

- Find the Singular Value Decomposition (SVD) of $\mathbf{H}$ such that $\mathbf{H} = \mathbf{U}\mathbf{\Omega}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices whose columns are the singular vectors and $\mathbf{\Omega}$ is a diagonal matrix containing the singular values.
- The rotation matrix relating the two point sets is given by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$.
- The translation between the two point sets is given by $\mathbf{T} = \mathbf{q}_c - \mathbf{R}\mathbf{p}_c$.

This process is iterated as stated in step **4.** until the mean Euclidean distance between the corresponding point sets $\mathbf{P_M}$ and $\mathbf{P_D}$ is less than or equal to a predetermined distance or until a given number of iterations is exceeded.

3. EXPERIMENTAL RESULTS

3.1. Registration-aided Position Estimation

In this section, we estimate the position of an UGV operating in an unknown outdoor environment. The registration algorithm is used for aiding position estimation whenever GPS errors are above a predetermined threshold[1].

An Extended Kalman Filter (EKF) was used to fuse encoder, GPS and compass observations to arrive at *ground truth* position estimates. It should be noted here that the EKF pose estimate is always superior than that provided by GPS alone and thus has been considered as ground truth. Consequently, a better position fix is guaranteed even when GPS is subject to multipathing errors. The ground truth was obtained in a similar fashion as reported in [8].

Figure 1 shows the results of the position estimation using the registration algorithm. As mentioned earlier, registration of range images is used to aid position estimation when GPS reported positional errors exceed a given threshold. In Figure 1(a), the registration-aided position estimates are denoted by '$+$' and that of the GPS by '$\circ$'. The wheel encoder estimates are also shown by '$\times$' for comparison. The error between the GPS and the registration-aided position estimates as compared with the ground truth are shown in Figure 1(b). It is evident that the registration-aided estimates are far superior than that of GPS alone.

3.2. 2D Map-aided Position Estimation

A map-aided position estimation algorithm for computing accurate pose estimates for a UGV operating in tunnel-like environments is described in this section. Using ground truth[2] together with the information from a range and bearing scanning laser rangefinder, a map of the operating domain, represented by a polyline that adequately approximates the geometry of the environment, is obtained. The map building process relies on position estimation provided by artificial landmarks.

An Iterative Closest Point-Extended Kalman Filter (ICP-EKF) algorithm is used to match range images from a scanning laser rangefinder to the line segments of the polyline map [6]. For this application, ICP alone does not provide sufficiently reliable and accurate vehicle motion estimates. These shortcomings are overcome by combining the ICP with a post-correspondence EKF. Once correspondences are established, a post-correspondence EKF, with the aid of a non-linear observation model, provides consistent vehicle pose estimates.

The ICP-EKF algorithm has several advantages. First, the uncertainty associated with observations is explicitly taken into account. Second, observations from a variety of different sensors can be easily combined as the changes are reflected only as additional observational states in the EKF. Third, laser observations that do not correspond to any line segment of the polyline map are discarded during the EKF update stage thus making the algorithm robust to errors in the map.

The estimated vehicle positions (solid line)[3] provided by the ICP-EKF algorithm along with the ground truth (dotted line) is shown in Figure 2(a). The vehicle travels a distance of 150 meters from right to left. The corresponding 2σ confidence bounds for the absolute error in x, y and ϕ are shown in Figure 2(b). It can be seen that the errors are bounded and thus the pose estimates are consistent. It is also clear that the estimated path is in close agreement with the ground truth.

4. PERFORMANCE MEASURES

The correspondence determination process is the most challenging step of the iterative algorithm. Establishing reliable correspondences is extremely difficult as the UGV is subjected to heavy pitching and rolling motion characteristic of travel over undulating terrain. This is further exacerbated by the uncertainty of the location of the sensor platform relative to the global frame of reference. In addition to these factors, noise inherently present in range images complicates the process of determining reliable correspondences.

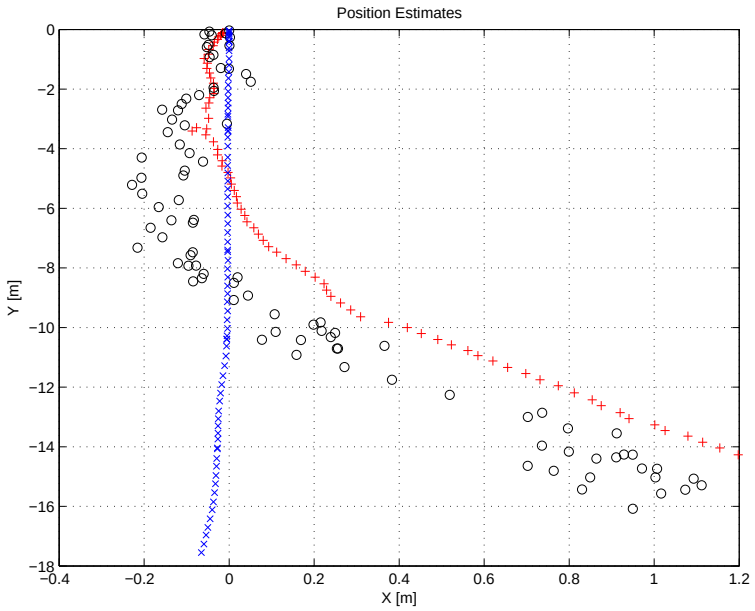
One solution to overcome the above deficiencies is to extract naturally occurring view-invariant features, for example, corners, from range images. Such *ground control points* can then be used for establishing reliable registration with the ICP algorithm converging to the global minimum. A hybrid approach to register aerial images obtained from a UAV with those from the UGV was developed by augmenting the modified ICP algorithm with a feature-based method. The feature-based hybrid approach was shown to be effective in producing reliable registration for UGV navigation [9].

For the map-aided position estimation scheme described in Section 3.2, the ICP-EKF algorithm failed to produce

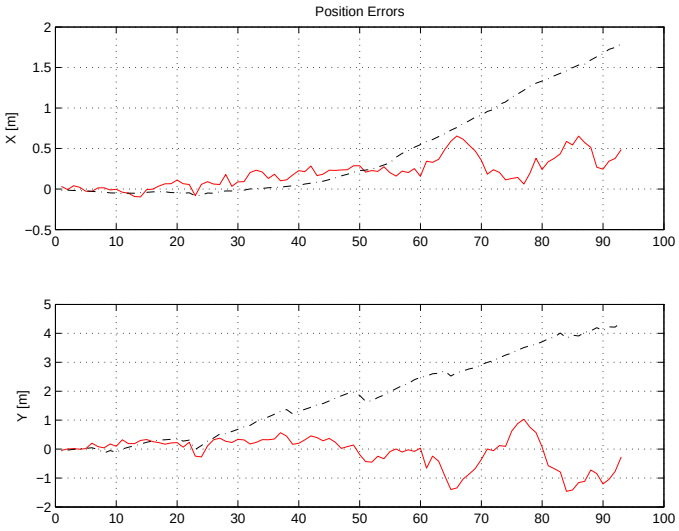
[1]The error in the GPS positions reported were obtained as a function of the number of satellites acquired. As an alternative, the so-called *dilution of precision* measure associated with the GPS can be used for the same purpose [2].

[2]The ground truth was obtained using a rotating laser scanner and known artificial landmarks placed at surveyed locations [7].

[3]As the estimates and the their corresponding ground truth are very close, extra effort is required on the part of the reader to distinguish between the two.



(a)



(b)

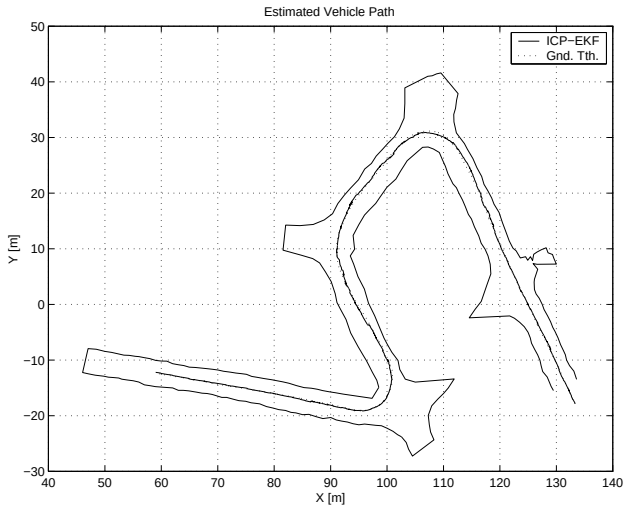
Fig. 1. Registration-aided position estimation. The aided estimates are shown by '+' and that of GPS by 'o'. The wheel encoder estimates shown by '×' are included for comparison. In (b), position errors as compared to the ground truth is depicted (GPS estimate is shown in dashed-dotted line).

unambiguous correspondences with the map whenever variations in data sets were not unique. To enable ICP to produce accurate correspondences, a strategy to augment the ICP-EKF algorithm with artificial/natural landmarks was devised to provide external aiding. To facilitate the selection of landmarks, an entropy-based metric was developed to enable the evaluation of information contained in a potential landmark. A curvature scale space algorithm was developed to extract natural landmarks from laser rangefinder scans [5]. The proposed landmark augmentation methodology has been verified for the localization of a Load-Haul-Dump truck and resulted in the ICP-EKF algorithm producing reliable and consistent estimates [6].

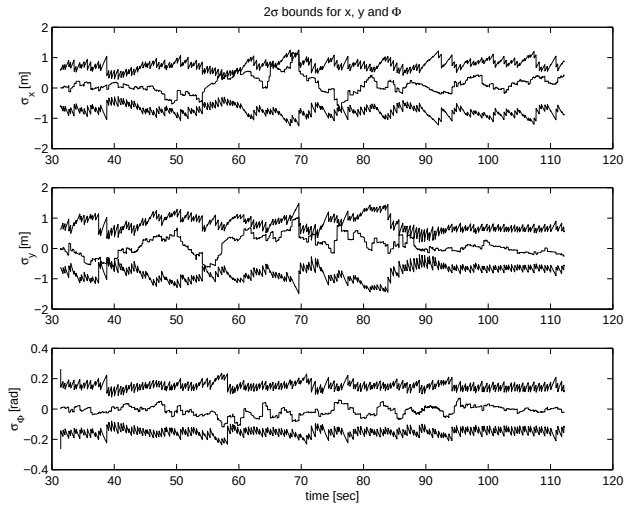
We propose the following measures towards performance evaluation of the registration algorithm for position estimation.

4.1. ICP Estimate and Dead-reckoning Prediction Measure

The ICP itself can be used to compute the estimates of the pose of the UGV. This can be compared with dead-



(a)



(b)

Fig. 2. 2D Map-aided Position Estimation. ICP-EKF estimated position the trial vehicle (solid line) and the ground truth (dotted line) are shown in (a). The 2σ confidence bounds are computed using the covariance estimate for the error in x, y and ϕ compared to the actual error computed with the ground truth estimates as depicted in (b).

reckoning estimates each time before the correspondences are computed. More specifically, the prediction covariance (from dead-reckoning) can be utilized as a check on the ICP estimates, since if the associated state covariances become large, this is an indication of the state estimation filter divergence as a result of the poor ICP estimates.

The largest Eigen value of the predicted state covariance matrix (that is a measure of the total positional uncertainty) can be used as a measure to check the quality of the ICP estimates.

Also the determinant of the predicted state covariance matrix can be used as a measure since it represents total predicted uncertainty and this can be observed to see if the ICP produces reliable and non-divergent estimates (since once the ICP estimates start behaving erratically, this is reflected by similar behavior in the correspondences).

4.2. Mean Squared Error Measure

To indicate if the correspondences make sense the following measure is proposed:

$$\mathcal{P}_{mse} \quad = \quad \frac{1}{n}\sum_{i=1}^{n}\left[d\left(p_i,\ell_i\right)\right]^2$$

where p_i and ℓ_i are the i^{th} of n range data points and $d\left(p_i,\ell_i\right)$ is the distance from the p_i^{th} point to the ℓ_i^{th} point. Global minimum of the function will occur at the *true* pose of the vehicle.

At the true pose, all or at least the majority of the range data points will be close to their corresponding points, thus yielding a very low value for the correct solution. The problem with this measure is that it is difficult to decide if the pose is true in the presence of outliers and occlusions.

4.3. Classification Factor

Similar to [4], we define *well defined data points* as those points that lie within some distance threshold from their corresponding points:

$$\mathcal{P}_{cf} \quad = \quad \frac{1}{n}\sum_{i=1}^{n}\left(1-\frac{d^m}{d^m+c^m}\right)$$

where d = d(p_i,ℓ_i), c = neighborhood size, m = sigmoid steepness[4]. At true pose, global maximum should approach close to unity and will be less in the neighborhood of well defined data points. Note that $\mathcal{P}_{cf}$ values can fall only between [0,1].

Indirectly this measure indicates the *future-goodness* of the pose estimate if a certain threshold is exceeded. The problem with this measure could be that it is not as sensitive as $\mathcal{P}_{mse}$ since it applies only for a certain local neighborhood. Thus $\mathcal{P}_{mse}$ can be used as a comparative performance measure whereas $\mathcal{P}_{cf}$ for pass/fail decisions for the correspondences before they are passed on for computing the incremental transformation.

4.4. Comparative Performance Measure

It is the ratio defined by

$$\mathcal{P}_{cpm} \quad = \quad \frac{{\mathcal{P}_{cf}}^2}{\mathcal{P}_{mse}}$$

The peak of this measure should occur at the true pose. In other words, this measure serves as a nonlinear scaling factor applied to the inverse of the measure, $\mathcal{P}_{mse}$.

[4]A sigmoid function is given by $f(a) = \frac{1}{1+e^{-ga}}$ where g denotes gain.

4.5. Results and Discussion

In this section, we use 3D LADAR data obtained during field trials to illustrate the utility of the proposed metrics in assessing the quality of correspondences. The LADAR was mounted on a UGV traversing rugged terrain on a pan/tilt platform to increase its narrow 20° field of view. The range of the tilt motion is $\pm 30^\circ$ resulting in an effective field of view of about 90° and providing a range image of 32 lines $\times$ 180 pixels where each data point contains the distance to a target in the operating environment. The angular resolution of this LADAR is $0.658^\circ \times 0.5^\circ$ in the horizontal and vertical directions, respectively.

We illustrate the combined utility of adaptive thresholding and the $\mathcal{P}_{mse}$ measure by using it to register 3D range images. We then compare the registration results with direct ICP (i.e., without AT and $\mathcal{P}_{mse}$). For the comparison, the same termination threshold condition is employed for both the algorithms.

Figures 3 and 4 summarize the comparative results. Figures 3(a) and (b) show the registered LADAR images via the direct and combined ICP algorithms, respectively. The combined ICP needed 39 iterations whereas the direct ICP took 82 iterations. The mean distances before and after registration were 0.19 m and 0.07 m for the two algorithms, respectively. Figures 4(a) and (b) show the closest point distance before and after registration. It is thus evident that the combined ICP algorithm is vastly superior than the direct ICP algorithm both in terms of accuracy and speed. Even though the $\mathcal{P}_{mse}$ metric is sensitive to outliers, we contend that the adaptive thresholding methodology in combination with the mean-squared error metric provides an acceptable means in inferring the validity of the position estimate.

5. CONCLUSIONS AND FURTHER WORK

The evaluation of performance of an iterative registration algorithm for position estimation of UGVs operating in unstructured environments was the main theme of this paper. The modified ICP algorithm was used to aid the position estimation process and the resulting estimates were compared with ground truth to facilitate the performance evaluation for two sets of field data. Field data obtained from trials on UGVs traversing undulating outdoor terrain was used to quantify the performance of the algorithm in producing continual position estimates.

In the first set of experimental results, registration-aided position estimates were generated whenever GPS estimates were unavailable or unreliable. For the second set of trials, the ICP-EKF algorithm was used for map-aided position estimation. In both cases, the presented results demonstrated the efficacy of the registration algorithm for position estimation.

The second part of the paper developed performance measures towards assessing the quality of correspondences required for accurate and efficient registration. The modified algorithm was combined with the mean-squared error metric to register 3D LADAR range images. The combined algorithm was then evaluated against the direct ICP algorithm. The accompanying results showed the superiority of the combined algorithm both in terms of speed and accuracy.

Future work includes combining the measures to achieve efficient 3D registration for mapping and position estimation tasks, both in indoor and outdoor environments. Currently, we are in the process of obtaining LADAR data in areas where GPS accuracy degrades and then approaches its best estimate. Such data sets would be of immense value in evaluating the utility of the registration algorithm and the proposed performance measures.

6. REFERENCES

[1] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] M.S. Grewal, L.R. Weill, and A.P. Andrews. *Global Positioning Systems, Inertial Navigation and Integration*. Wiley, 2001.

[3] I. Kweon and T. Kanade. High Resolution Terrain Map from Multiple Sensor Data. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, volume 1, pages 127–134, 1990.

[4] P. MacKenzie and G. Dudek. Precise Positioning using Model-based Maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1615–1621, 1994.

[5] R. Madhavan and H. Durrant-Whyte. Natural Landmark-based Autonomous Navigation using Curvature Scale Space. *Robotics and Autonomous Systems*, 46(2):79–95, February 2004.

[6] R. Madhavan and H. Durrant-Whyte. Terrain Aided Localization of Autonomous Ground Vehicles. *Special Issue of the Journal of Automation in Construction (Invited)*, 13(1):69–86, January 2004.

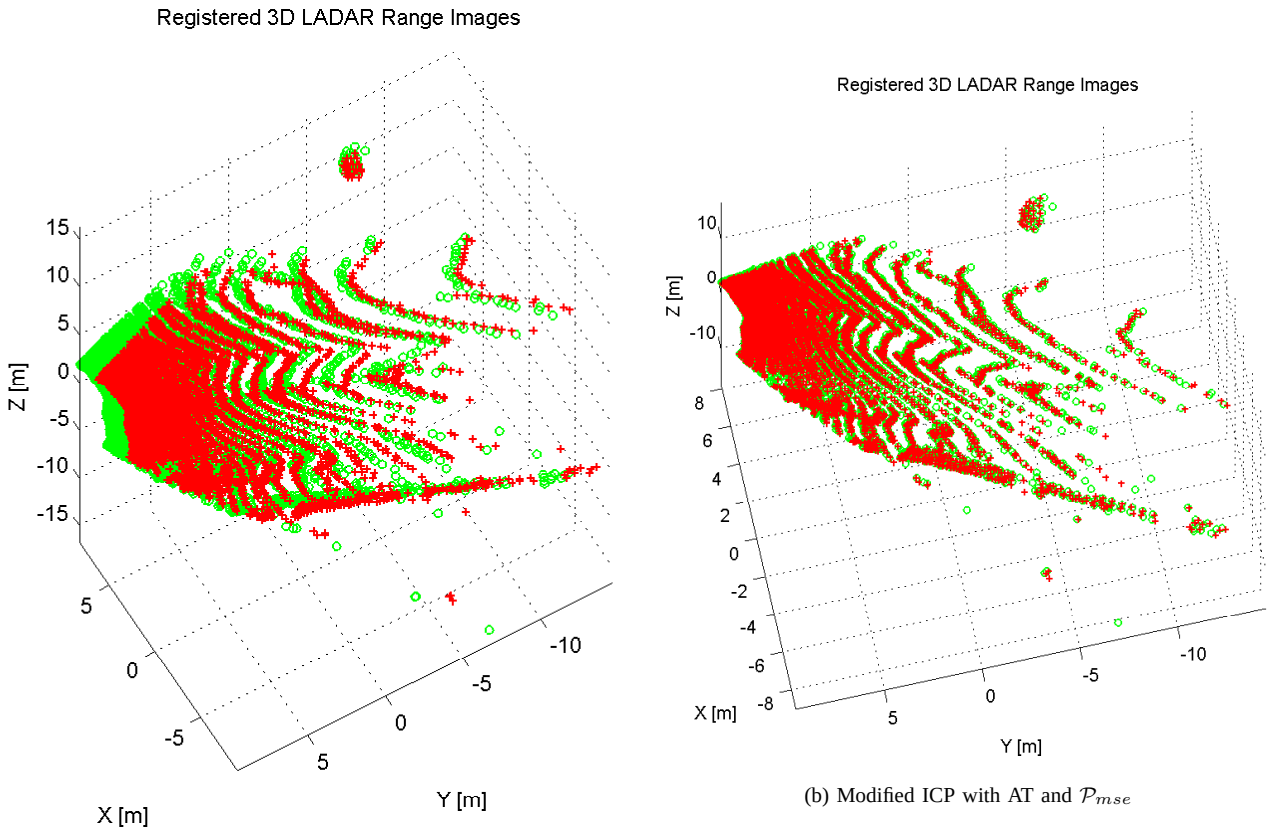
[7] R. Madhavan et al. Evaluation of Internal Navigation Sensor Suites for Underground Mining Vehicle Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 999–1004, May 1999.

[8] R. Madhavan, K. Fregene, and L.E. Parker. Distributed Cooperative Outdoor Multi-robot Localization and Mapping. *Autonomous Robots: Special Issue on Analysis and Experiments in Distributed Multi-Robot Systems*, 17(1):23–39, July 2004.

[9] R. Madhavan, T. Hong, and E. Messina. Temporal Range Registration for Unmanned Ground and Aerial Vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3180–3187, April 2004.

[10] G. Shaffer. *Two-Dimensional Mapping of Expansive Unknown Areas*. PhD thesis, Carnegie Mellon University, 1992.

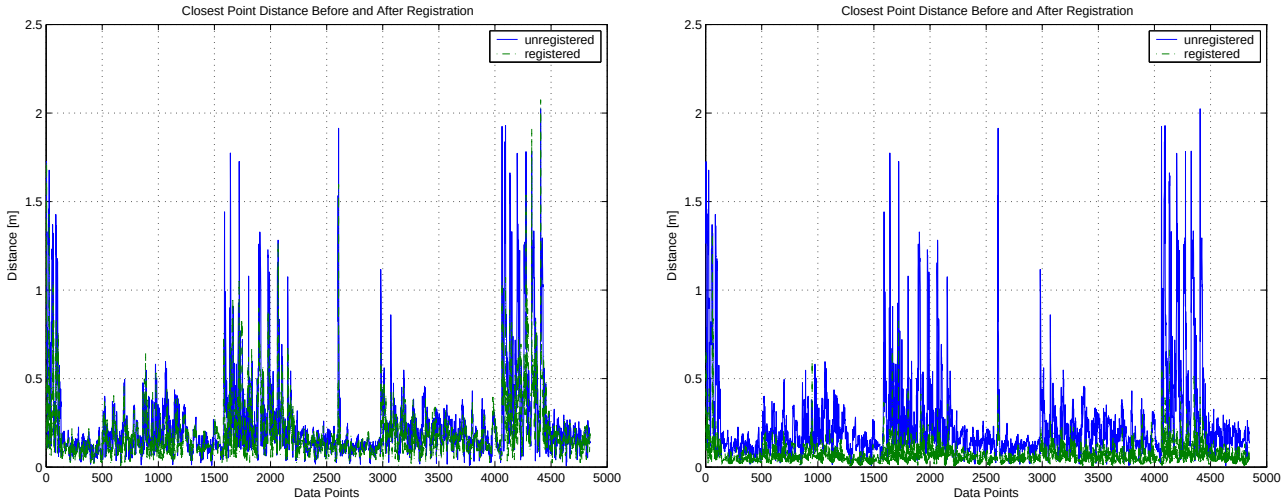
[11] Z. Zhang. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.



(a) Direct ICP

(b) Modified ICP with AT and $\mathcal{P}_{mse}$

Fig. 3. Illustration of 3D LADAR registration via the direct (w/o AT and $\mathcal{P}_{mse}$) and combined ICP algorithms. The model ('$\circ$') and data ('+') points after registration are shown.



(a)

(b)

Fig. 4. The registered (shown in dashed-dotted line) and unregistered closest point distances are shown corresponding to the registration of range images depicted in Figures 3(a) and 3(b), respectively.

Evaluating System Capabilities and User Performance in the Battlefield Augmented Reality System*

Mark A. Livingston† J. Edward Swan II† Simon J. Julier‡ Yohan Baillot‡
Dennis Brown† Lawrence J. Rosenblum† Joseph L. Gabbard§ Tobias H. Höllerer¶
Deborah Hix§
Naval Research Laboratory
Washington D.C.

Abstract

We describe a first experiment in evaluating the system capabilities of the Battlefield Augmented Reality System, an interactive system designed to present military information to dismounted warfighters. We describe not just the current experiment, but a methodology of both system evaluation and user performance measurement in the system, and show how both types of tests will be useful in system development. We summarize results in a perceptual experiment being used to inform system design, and discuss ongoing and future experiments to which the work described herein leads.

1 Introduction

One of the most challenging aspects of the design of intelligent systems is the user interface – how the user will perceive and understand the system. Our application presents military information to a dismounted warfighter. In order to both refine the system's capabilities and improve the warfighter's performance of tasks while using the system, we measure human performance using our system, even while early in the design phase of the user interface. This paper describes an early experiment in the context of system evaluation and describes implications for both system and human performance metrics as they apply to such systems.

1.1 Application context

Military operations in urban terrain (MOUT) present many unique and challenging conditions for the warfighter. The environment is extremely complex and inherently three-dimensional. Above street level, buildings serve varying purposes (such as hospitals or communication stations). They can harbor many risks, such as snipers or mines, which can be located on different floors. Below street level, there can be an elaborate network of sewers and tunnels. The environment can be cluttered and dynamic. Narrow streets restrict line of sight and make it difficult to plan and coordinate group activities. Threats, such as snipers, can continuously move and the structure of the environment itself can change. For example, a damaged building can fill a street with rubble, making a once-safe route impassable. Such difficulties are compounded by the need to minimize the number of civilian casualties and the amount of damage to civilian targets.

In principle, many of these difficulties can be overcome through better *situation awareness*. The Concepts Division of the Marine Corps Combat Development Command (MCCDC) concludes [2]:

> Units moving in or between zones must be able to navigate effectively, and to coordinate their activities with units in other zones, as well as with units moving outside the city. This navigation and coordination capability must be resident at the very-small-unit level, perhaps even with the individual Marine.

A number of research programs have explored the means by which navigation and coordinated information can be delivered to the dismounted warfighters. We believe a mobile augmented reality system best meets the needs of the dismounted warfighter.

1.2 Mobile Augmented Reality

Augmented reality (AR) refers to the mixing of virtual cues from the real three-dimensional environment into the user's perception. In this work, AR denotes the 3D merging

*Portions of this paper were originally published in [10] and [11].

†Virtual Reality Laboratory, Naval Research Laboratory. Corresponding email: livingston@ait.nrl.navy.mil
‡ITT Advanced Engineering and Sciences
§Systems Research Center, Virginia Polytechnic Inst. and State Univ.
¶Dept. of Computer Science, University of California, Santa Barbara

Figure 1. A sample view of our system, showing one physically visible building with representations of three buildings which it occludes.

of synthetic imagery into the user's natural view of the surrounding world, using an optical, see-through, head-worn display.

A *mobile augmented reality* system consists of a computer, a tracking system, and a see-through HMD. The system tracks the position and orientation of the user's head and superimposes graphics and annotations that are aligned with real objects in the user's field of view. With this approach, complicated spatial information can be directly aligned with the environment. This contrasts with the use of hand-held displays and other electronic 2D maps. With AR, for example, the name of a building could appear as a "virtual sign post" attached directly to the side of the building. To explore the feasibility of such a system, we are developing the Battlefield Augmented Reality System (BARS). Figure 1 is an example from BARS. This system will network multiple dismounted warfighters together with a command center.

Through the ability to present direct information overlays, integrated into the user's environment, AR has the potential to provide significant benefits in many application areas. Many of these benefits arise from the fact that the virtual cues presented by an AR system can go beyond what is physically visible. Visuals include textual annotations, directions, instructions, or "X-ray vision," which shows objects that are physically present, but occluded from view. Potential application domains include manufacturing [1], architecture [20], mechanical design and repair [7], medical applications [4, 17], military applications [11], tourism [6], and interactive entertainment [19].

1.3 Performance Measurement in BARS

BARS supports information gathering and human navigation for situation awareness in an urban setting [11]. A critical aspect of our research methodology is that it equally addresses both technical and human factors issues in fielding mobile AR. AR system designers have long recognized the need for standards for the performance of AR technology. As the technology begins to mature, we and some other research groups are also considering how to test user cognition when aided by AR systems.

We determined the task in which to measure performance first through consultation with domain experts [9]. They identified a strong need to visualize the spatial locations of personnel, structures, and vehicles occluded by buildings and other urban structures during military operations in urban terrain. While we can provide an overhead map view to view these relationships, using the map requires a context switch. We are designing visualization methods that enable the user to understand these relationships when directly viewing, in a heads-up manner, the augmented world in front of them.

The perceptual community has studied depth and layout perception for many years. Cutting [3] divides the visual field into three areas based on distance from the observer: near-field (within arms reach), medium-field (within approximately 30 meters), and far-field (beyond 30 meters). He then points out which depth cues are more or less effective in each field. Occlusion is the primary cue in all three spaces, but with the AR metaphor and the optical see-through, this cue is diminished. Perspective cues are also important for far-field objects, but this assumes that they are physically visible. The question for an AR system is which cues work when the user is being shown virtual representations of objects integrated into a real scene.

Our immediate goal is thus to determine methods that are appropriate for conveying depth relationships to BARS users. This requires measurement of the system's performance in presenting information that feed the users' perceptions of the surrounding environment. Then, we need to establish a standard for warfighter performance in the task of locating military personnel and equipment during an operation in urban terrain. For example, one goal of our work is to determine how many depth layers a user can understand.

2 Related Work

2.1 Perceptual Measures in AR Systems

A number of representations have been used to convey depth relationships between real and virtual objects. Partial transparency, dashed lines, overlays, and virtual cut-away views all give the user the impression of a difference in the depth [7, 16, 20, 12].

Furmanski et al. [8] utilized a similar approach in their pilot experiment. Using video AR, they showed users a stimulus which was either behind or at the same distance as an obstructing surface. They then asked users to identify whether the stimulus was behind, at the same distance as, or closer than the obstruction. The performance metric here is thus an ordinal depth measure. Only a single occluded object was present in the test. The parameters in the pilot test were the presence of a cutaway in the obstruction and motion parallax. The presence of the cutaway significantly improved users' perceptions of the correct location when the stimulus was behind the obstruction. The authors offered three possible locations to the users, even though only two locations were used. Users consistently believed that the stimulus was in front of the obstruction, despite the fact that it was never there.

Ellis and Menges [5] found that the presence of a visible (real) surface near a virtual object significantly influences the user's perception of the depth of the virtual object. For most users, the virtual object appeared to be nearer than it really was. This varied widely with the user's age and ability to use accommodation, even to the point of some users being influenced to think that the virtual object was further away than it really was. Adding virtual backgrounds with texture reduced the errors, as did the introduction of virtual holes, similar to those described above. Rolland et al. [13] found that occlusion of the real object by the virtual object gave the incorrect impression that the virtual object was in front, despite the object being located behind the real object and other perceptual cues denoting this relationship. Further studies showed that users performed better when allowed to adjust the depth of virtual objects than when making forced-choice decisions about the objects' locations [14].

2.2 Cognitive Measures in AR Systems

There have been few user studies conducted with AR systems; most such studies (including ours) have been at the perceptual level, such as those described above. The recent emergence of hardware capable of delivering sufficient performance to achieve stable presentation of graphics does enable such studies, however. One example of a cognitive-level study is the application of AR to medical interventions with ultrasound guidance [15]. In this trial, a doctor performed ultrasound-guided needle biopsies with and without the assistance of an AR system that had been designed for the task. A second physician evaluated the needle placement of the first. The analysis showed that needle localization was improved when using the AR system. The performance metrics in this trial were the standard for evaluating doctors' performance used by medical schools: needle placement at various locations within the target lesion. The physician uses the ultrasound to determine the ideal and actual needle locations. Thus the measure is tightly connected to the task, and in fact exists prior to the development of the AR system.

3 Experiment

As noted above, we have begun our performance measurements with the subsystem that depicts occluded surfaces. The first test we performed was a perceptual experiment to determine whether the system provides sufficient information for the user to understand three layers of depth among large objects that are occluded from view.

3.1 Design Methodology

From our initial design work and review by colleagues, we selected three graphical parameters to vary in our representations: drawing style, opacity, and intensity. These comprised a critical yet tenable set of parameters for our study. We used an urban environment that fit our laboratory facilities. By sitting in the atrium of our building, a user could wear an indoor-based version of our system (which is more powerful than the current mobile prototypes). The environment included one physically visible building and two occluded buildings. Among the two occluded buildings we placed one target to locate in one of three different positions: closer than the two occluded buildings, between the two, or behind both. This introduced the question of whether the ground plane (i.e. perspective) would provide the only cue that users would actually use. Because our application may require users to visualize objects that are not on the ground or are at a great distance across hilly terrain, we added the use of a consistent, flat ground plane for all objects as a parameter.

3.2 Hardware

The hardware for our AR platform consisted of three components. For the image generator, we used a Pentium IV 1.7 GHz computer with an ATI FireGL2 graphics card (outputting frame-sequential stereo). For the display device, we used a Sony Glasstron LDI–100B stereo optical see-through display (SVGA resolution, 20° horizontal field of view in each eye). The user was seated indoors for the experiment and was allowed to move and turn the head and upper body freely while viewing the scene, which was visible through an open doorway to the outdoors. We used an InterSense IS-900 6-DOF ultrasonic/inertial hybrid tracking system to track the user's head motion to provide a consistent 3D location for the objects as the user viewed the world. The IS-900 provides position accuracy to 3.0 mm and orientation accuracy to 1.0°.

The user entered a choice for each trial on a standard extended keyboard, which was placed on a stand in front of the seat at a comfortable distance. The display device, whose transparency can be adjusted in hardware, was set

for maximum opacity of the LCD, to counteract the bright sunlight that was present for most trials. Some trials did experience a mix of sunshine and cloudiness, but the opacity setting was not altered. The display brightness was set to the maximum.

The display unfortunately does not permit adjustment of the inter-pupillary distance (IPD) for each user. If IPD is too small, then the user will be seeing slightly cross-eyed and tend to believe objects are closer than they are. The display also does not permit adjusting the focal distance of the graphics. The focal distance of the virtual objects is therefore closer than the real object that we used as the closest obstruction. This would tend to lead users to believe the virtual objects were closer than they really were.

Stereo is considered a powerful depth cue at near-field distances (approximately 1.0 meters, or about at arm's length). At far-field distances, such as the task we gave our users, stereo is not considered to be a strong depth cue; however, we wanted to be able to provide some statistical evidence for this claim. Many practitioners of AR systems have noted that improper settings of parameters related to stereo imagery (such as IPD and vergence) can lead to user discomfort in the form of headaches or dizziness. None of users reported any such problems; they wore the device for an average of 30 minutes. These issues will need to be addressed in future versions of the hardware for AR systems, but are beyond the scope of our work.

3.3 Experimental Design

3.3.1 Independent Variables

From our heuristic evaluation and from previous work, we identified the following independent variables for our experiment. These were all *within-subject* variables; every user saw every level of each variable.

Drawing Style ("wire", "fill", "wire+fill"): Although the same geometry was visible in each stimulus (except for which target was shown), the representation of that geometry was changed to determine what effect it had on depth perception. We used three drawing styles (Figure 2). In the first, all objects are drawn as wireframe outlines. In the second, the first (physically visible) object is drawn as a wireframe outline, and all other objects are drawn with solid fill (with no wireframe outline). In the third style, the first object is in wireframe, and all other layers are drawn with solid fill with a white wireframe outline. Backface culling was on for all drawing styles, so that the user saw only two faces of any occluded building.

Opacity (constant, decreasing): We designed two sets of values for the α channel based on the number of occluding objects. In the "constant" style, the first layer (visible with registered wireframe outline) is completely opaque, and all other layers have the same opacity ($\alpha = 0.5$). In the "decreasing" style, opacity changes for each layer. The first

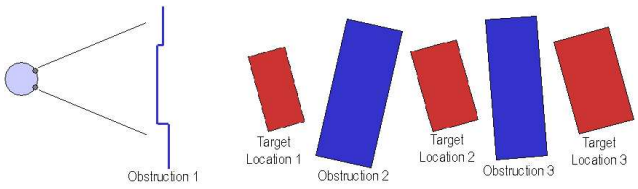


Figure 3. The experimental design (not to scale) shows the user position at the left. Obstruction 1 denotes the visible surfaces of the physically visible building. The distance from the user to obstruction 1 is approximately 60 meters. The distance from the user to target location 3 is approximately 500 meters, with the obstructions and target locations roughly equally spaced.

(physically visible, wireframe) layer is completely opaque. The successive layers are not opaque; the α values were 0.6, 0.5, and 0.4 for the successively more distant layers.

Intensity (constant, decreasing): We used two sets of intensity modulation values. The modulation value was applied to the object color (in each color channel, but not in the opacity or α channel) for the object in the layer for which it was specified. In the "constant" style, the first layer (visible with registered wireframe outline) has full intensity (modulator=1.0) and all other layers have intensity modulator=0.5. In the "decreasing" style, the first layer has its full native intensity, but successive layers are modulated as a function of occluding layers: 0.75 for the first, 0.50 for the second, and 0.25 for the third (final) layer.

Target Position (close, middle, far): As shown in the overhead map view (Figure 3), there were three possible locations for the target.

Ground Plane (on, off): From the literature and everyday experience, we know that the perspective effects of the ground plane rising to meet the horizon and apparent object size are a strong depth cues. In order to test the representations as an aide to depth ordering, we removed the ground plane constraint in half of the trials. The building sizes were chosen to have the same apparent size from the users' location for all trials. When the ground plane constraint was not present in the stimulus, the silhouette of each target was fixed for a given pose of the user. In other words, targets two and three were not only scaled (to yield the same apparent size) but also *positioned vertically* such that all three targets would occupy the same pixels on the 2D screen for the same viewing position and orientation. No variation in position with respect to the two horizontal dimensions was necessary when changing from using the ground plane to not using it. The obstructions were always presented with the same ground plane. We informed the users for which

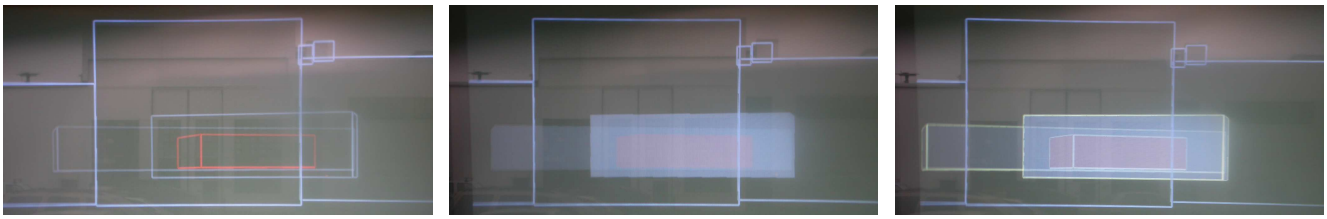


Figure 2. User's view of the stimuli. *Left*: **"wire" drawing style.** *Center*: **"fill" drawing style.** *Right*: **"wire+fill" drawing style. The target (smallest, most central box) is between (position "middle") obstructions 2 and 3 in all three pictures. These pictures were acquired by placing a camera to the eyepiece of the HMD, which accounts for the poor image quality. The vignetting and distortion are due to the camera lens and the fact that it does not quite fit in the exit pupil of the HMD's optics.**

half of the session the ground plane would be consistent between targets and obstructions.

We did this because we wanted to remove the effects of perspective from the study. Our application requires that we be able to visualize objects that may not be on the ground, may be at a distance and size that realistic apparent size would be too small to discern, and may be viewed over hilly terrain. Since our users may not be able to rely on these effects, we attempted to remove them from the study.

Stereo (on, off): The Sony Glasstron display receives as input left- and right-eye images. The IPD and vergence angle are not adjustable, so we can not provide a true stereo image for all users. However, we can present images with disparity (which we call "stereo" for the experiment) or present two identical images ("biocular").

Repetition (1, 2, 3): Each user saw three repetitions of each combination of the other independent variables. It is well-known that users will often improve their performance with repetition of the same stimulus within an experiment. By repeating the stimuli, we can gain some insight into whether the user needs to learn how the system presents cues or whether the system presents intuitive cues. If there is no learning effect with repetition of stimuli, then we can infer that the users had whatever collective performance they achieved intuitively.

3.3.2 Dependent Variables

For each trial, we recorded the user's (three-alternative forced) choice for the target location and the time the user took to enter the response after the software presented the stimulus. We opted to ask the user only to identify the ordinal depth, not an absolute distance between the graphical layers. This implied the forced-choice design.

All combinations of these parameters were encountered by each user; however, the order in which these were presented was also randomly permuted. Thus each user viewed 432 trials. The users ranged in time from twenty to forty minutes for the complete set of trials. The users were told

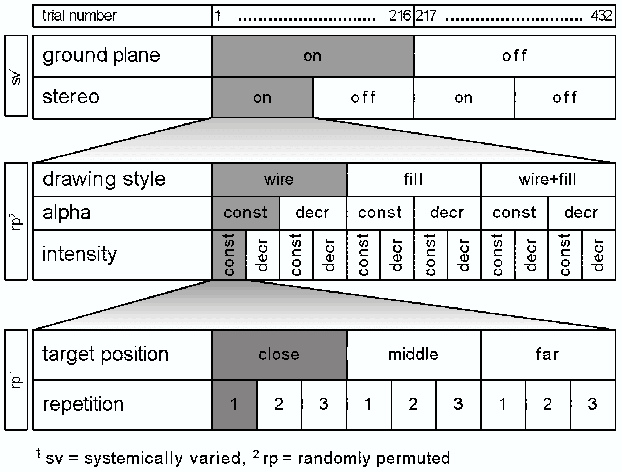


Figure 4. Experimental design and counterbalancing for one user. Systematically varied parameters were counterbalanced between subjects.

to make their best guess upon viewing the trial and not to linger; however, no time limit per trial was enforced. The users were instructed to aim for a balance of accuracy and speed, rather than favoring one over the other.

3.3.3 Counterbalancing

In order to reduce time-based confounding factors, we counterbalanced the stimuli. This helps control learning and fatigue effects within each user's trials and factors such as the amount of sunshine that change between subjects beyond our control. Figure 4 describes how we counterbalanced the stimuli. We observed (in conjunction with many previous authors) that the most noticeable variable was the presence of the ground plane [3, 18]. In order to minimize potentially confusing large-scale visual changes, we gave ground plane and stereo the slowest variation. Following this logic, we next varied the parameters which controlled the scene's visual appearance (drawing style, alpha, and in-

tensity), and within the resulting blocks, we created nine trials by varying target position and repetition.

3.4 Experimental Task

We designed a small virtual world that consisted of four buildings (Figure 3), with three potential target locations. The first building was an obstruction that corresponded (to the limit of our modeling accuracy) to a building that was physically visible during the experiment. The obstructions were always drawn in blue; the target always appeared in red. The target was scaled such that its apparent 2D size was equal, regardless of its location. Obstructions 2 and 3 roughly corresponded to real buildings. The three possible target locations did not correspond to real buildings.

The task for each trial was to determine the location of the target that was drawn. The user was shown the overhead view before beginning the experiment. This helped them visualize their choices and would be an aide available in a working application of our system. The experimenter explained that only one target would appear at a time. Thus in all of the stimulus pictures, four objects were visible: three obstructions and the target. For the trials, users were instructed to use the number pad of a standard extended keyboard and press a key in the bottom row of numbers (1–3) if the target were closer than obstructions 2 and 3, a key in the middle row (4–6) if the target were between obstructions 2 and 3, or a key in the top row (7–9) if the target were further than obstructions 2 and 3. A one-second delay was introduced between trials within sets, and a rest period was allowed between sets for as long as the user wished. We showed the user 48 sets of nine trials each. The users reported no difficulties with the primitive interface after their respective practice sessions. The users did not try to use head motion to provide parallax, which is not surprising for a far-field visualization task.

3.5 Subjects

Eight users completed the experiment (432 trials each). All subjects were male and ranged in age from 20 to 48. All volunteered and received no compensation. Our subjects reported being heavy computer users. Two were familiar with computer graphics, but none had seen our representations. Subjects did not have difficulty learning or completing the experiment.

Before the experiment, we asked users to complete a stereo acuity test, in case stereo had produced an effect. The test pattern consisted of nine shapes containing four circles each. For each set of four circles, the user was asked to identify which circle was closer than the other three. Seven users answered all nine test questions correctly, while the other user answered eight correctly.

4 Hypotheses

We made the following hypotheses about our independent variables.

1. The ground plane would have a strong positive effect on the user's perception of the relative depth.

2. The wireframe representation (our system's only option before this study) would have a strong negative effect on the user's perception.

3. Stereo imagery would not yield different results than biocular imagery, since all objects are in the far-field [3].

4. Decreasing intensity would have a strong positive effect on the user's perception for all representations.

5. Decreasing opacity would have a strong positive effect on the user's perception of the "fill" and "wire+fill" representations. In the case of wireframe representation the effect would be similar to decreasing intensity. Apart from the few pixels where lines actually cross, decreasing opacity would let more and more of the background scene shine through, thereby indirectly leading to decreased intensity.

5 Results

There are a number of error metrics we apply to the experimental data. Figure 5 categorizes the user responses. Subjects made 79% correct choices and 21% erroneous choices. We found that subjects favored the far position, choosing it 39% of the time, followed by the middle position (34%), and then by the close position (27%). We also found that subjects were the most accurate in the far position: 89% of their choices were correct when the target was in the far position, as compared to 76% correct in the close position, and 72% correct in the middle position.

As discussed above, we measured two dependent variables: ***user response time***, and ***user error***. For user response time, the system measured the time in milliseconds (ms) between when it drew the scene and when the user responded. Response time is an interesting metric because it indicates how intuitive the representations are to the user. We want the system to convey information as naturally as the user's vision does in analogous real-world situations.

For user error, we calculated the metric $e = |a - u|$, were a is the actual target position (between 1 and 3), and u is the target position chosen by the user (also between 1 and 3). Thus, if $e = 0$ the user has chosen the correct target; if $e = 1$ the user is off by one position, and if $e = 2$ the user is off by two positions.

We conducted significance testing for both response time and user error with a standard analysis of variance

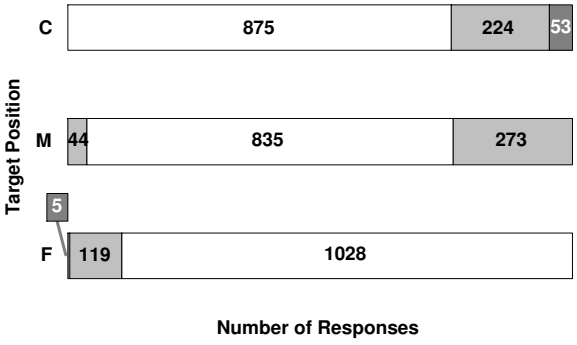


Figure 5. User responses by target position. For each target position, the bars show the number of times subjects chose the (C)lose, (M)iddle, and (F)ar positions. Subjects were either correct when their choice matched the target position (white), off by one position (light gray), or off by two positions (dark gray).

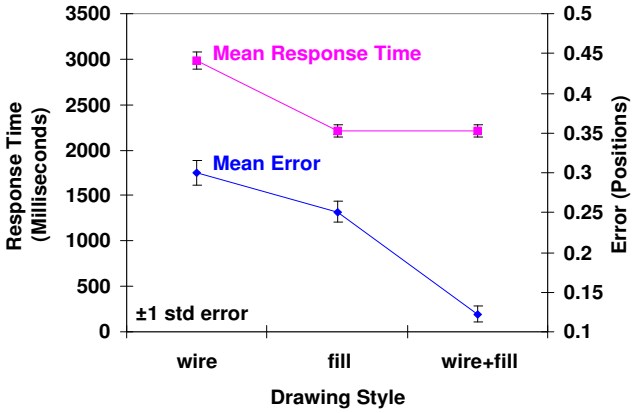


Figure 6. Main effect of drawing style on response time (□) and error (◇).

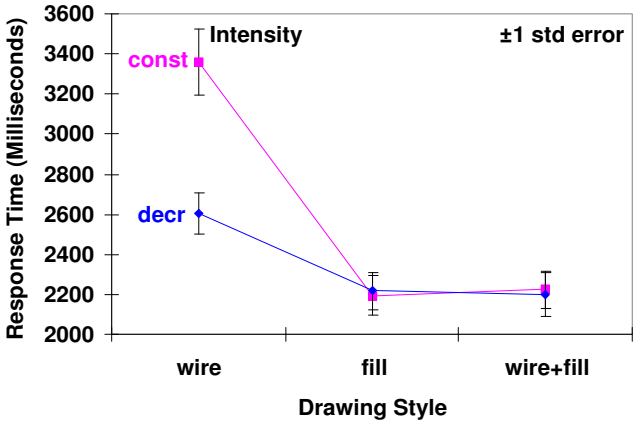


Figure 7. Drawing style by intensity (constant (□), decreasing (◇)) interaction on response time.

(ANOVA) procedure. In the summary below, we report user errors in positions (pos).

We briefly discuss the factors that affected user performance. As we expected, subjects were more accurate when a ground plane was present (.1435 pos) then when it was absent (.3056 pos). Interestingly, there was no effect of ground plane on response time ($F < 1$). This indicates that subjects did not learn to just look at the ground plane and immediately respond from that cue alone, but were in fact also attending to the graphics.

Figure 6 shows that subjects were slower using the "wire" style than the "fill" and "wire+fill" styles. Subjects had the fewest errors with the "wire+fill" style. These results verified our hypotheses that the "wire" style would not be very effective, and the "wire+fill" style would be the most effective, since it combines the occlusion properties of the "fill" style with the wireframe outlines, which help convey the targets' shapes.

Subjects were more accurate with decreasing opacity (.1962 pos) than with constant opacity (.2529 pos). This makes sense because the decreasing opacity setting made the difference between the layers more salient. Subjects were both faster (2340 versus 2592 ms) and more accurate (.1811 versus .2679 pos) with decreasing intensity. This result was expected, as decreasing intensity did a better job of differentiating the different layers. However, Figure 7 shows that the effect on response time is due to the difference between constant and decreasing intensity when the target is drawn in the "wire" style.

As expected from training effects, subjects became faster with repetition. However, repetition had no effect on absolute error ($F < 1$), so although subjects became faster, they did not become more accurate. This can be taken as a sign that the presented visuals were understandable for the subjects right from the outset. No learning effect took place regarding accuracy. Subjects became faster, though, which is a sign that their level of confidence increased.

6 Discussion

In a broad context, we believe that our methodology will enable us to evaluate both system capabilities and user performance with the system. Human perception is an innate ability, and variations in performance will reflect the system's appropriateness for use by dismounted warfighters. Thus, we are really evaluating the system's performance by measuring the user's performance on perceptual-level tasks. The evaluation of cognitive-level tasks will enable us to determine how users are performing. Such high-level metrics can only be measured after the results of the perceptual-

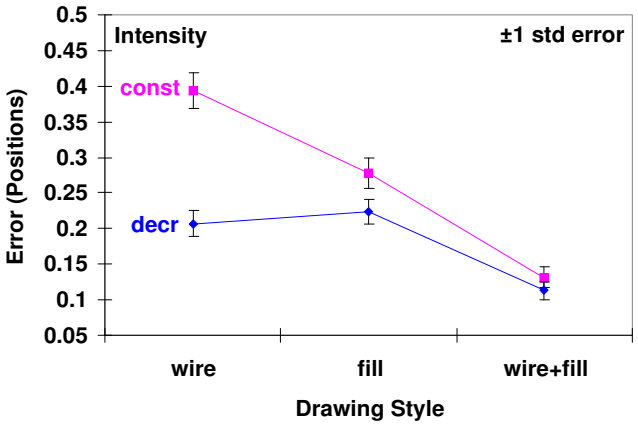


Figure 8. Drawing style by intensity (constant (▫), decreasing (◇)) interaction on absolute error.

level tests inform the system design.

Our first experiment has given insight into how users perceive data presented in the system. The application of our results to human perception and thus our system design are straightforward. It is well-known that a consistent ground plane (a perspective constraint) is a powerful depth cue. However, we can now provide statistical backing for our fundamental hypothesis that graphical parameters can provide strong depth cues, albeit not physically realistic cues. We found that with the ground plane on the average error was .144 pos, whereas the with the ground plane off and the following settings:

- drawing style: "wire+fill"
- opacity: decreasing
- intensity: decreasing

the average error was .111 pos. The data thus suggest that we did find a set of graphical parameters as powerful as the presence of the ground plane constraint. This would indeed be a powerful statement, but requires further testing before we can say for sure whether this is our finding. As a secondary result, the fact that there was a main effect of repetition on response time but not on accuracy indicates that the subjects could quickly understand the semantic meaning of the encodings. This validates that BARS is performing at a level that is sufficient for users to consistently (but not always) identify the ordinal depth among three occluded objects.

There are several next steps available to us. Further perceptual-level testing will demonstrate whether these results extend to more complex scenes (with more layers of depth). We are currently designing a follow-up study that will use not just an ordinal depth metric, but an absolute distance metric. This study will task the user to move a virtual object into depth alignment with real objects. We are developing metrics to apply to the user's control of the object, such as the number of oscillations they use to place the object into position, that will give us insight into their confidence in the depth estimates they perceive through BARS. We are also considering ways in which to measure the user's subjective reaction to the system, as this is also an important aspect of the system's capabilities.

Once these results inform our future system design, we will move up to cognitive-level testing, in which we hope to have multiple users wear prototype systems in an urban environment. We can have users identify locations of objects relative to maps or to each other. We could have users retrieve objects from the environment. The metrics we plan to use will reflect the cognition required. Distance and response time will remain interesting measures, but now the absolute distance will become more important. We will be able to add directional measures as well, concomitant with the increased complexity of the task for a mobile user. Since our application is designed for a military context, we intend to design our cognitive-level tests in conjunction with military domain experts and have at least some of the subjects in our studies be active members of the military. This introduces the opportunity to measure system performance by comparing against current performance of dismounted warfighters in these tasks. This combined design and evaluation methodology will enable us to evaluate the Battlefield Augmented Reality System and its users.

References

[1] T. P. Caudell and D. W. Mizell. Augmented reality: An application of heads up display technology to manual manufacturing processes. In *Proceedings of Hawaii International Conference on System Sciences*, volume II, pages 659–669. IEEE Computer Society Press, Jan. 1992.

[2] Concepts Division, Marine Corps Combat Development Command. A concept for future military operations on urbanized terrain, July 1997.

[3] J. E. Cutting. How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, and Computers*, 29(1):29–36, 1997.

[4] P. Edwards, D. Hawkes, D. Hill, D. Jewell, R. Spink, A. Strong, and M. Gleeson. Augmented reality in the stereo operating microscope for otolaryngology and neurological guidance. In *Medical Robotics and Computer Assisted Surgery*, Sept. 1995.

[5] S. R. Ellis and B. M. Menges. Localization of object in the near visual field. *Human Factors*, 40(3):415–431, Sept. 1998.

[6] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In *International Symposium on Wearable Computing (ISWC)*, pages 74–81, Oct. 1997.

[7] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62, July 1993.

[8] C. Furmanski, R. Azuma, and M. Daily. Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, pages 215–224, Sept. 2002.

[9] J. L. Gabbard, J. E. Swan II, D. Hix, M. Lanzagorta, M. A. Livingston, D. Brown, and S. Julier. Usability engineering: Domain analysis activities for augmented reality systems. In *Proceedings of SPIE (International Society for Optical Engineering)*, The Engineering Reality of Virtual Reality 2002, Jan. 2002.

[10] M. A. Livingston, J. L. G. J. Edward Swan II, T. H. Höllerer, D. Hix, S. J. Julier, Y. Baillot, and D. Brown. Resolving multiple occluded layers in augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR2003)*, pages 56–65. IEEE, Oct. 2003.

[11] M. A. Livingston, L. J. Rosenblum, S. J. Julier, D. Brown, Y. Baillot, J. E. Swan II, J. L. Gabbard, and D. Hix. An augmented reality system for military operations in urban terrain. In *Interservice/Industry Training, Simulation, and Education Conference*, page 89, Dec. 2002.

[12] U. Neumann and A. Majoros. Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 4–11, 1998.

[13] J. P. Rolland and H. Fuchs. Optical versus video see-through head-mounted displays in medical visualization. *Presence: Teleoperators and Virtual Environments*, 9(3):287–309, June 2000.

[14] J. P. Rolland, C. Meyer, K. Arthur, and E. Rinalducci. Method of adjustments versus method of constant stimuli in the quantification of accuracy and precision of rendered depth in head-mounted displays. *Presence: Teleoperators and Virtual Environments*, 11(6):610–625, Dec. 2002.

[15] M. Rosenthal, A. State, J. Lee, G. Hirota, J. Ackerman, K. Keller, E. D. P. MD, M. Jiroutek, K. Muller, and H. Fuchs. Augmented reality guidance for needle biopsies: A randomized, controlled trial in phantoms. In *Lecture Notes in Computer Science: Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, volume 2208, pages 240–248, Oct. 2001.

[16] A. State, D. T. Chen, C. Tector, A. Brandt, H. Chen, R. Ohbuchi, M. Bajura, and H. Fuchs. Case study: Observing a volume-rendered fetus within a pregnant patient. In *Proceedings of IEEE Visualization '94*, pages 364–368, 1994.

[17] A. State, M. A. Livingston, G. Hirota, W. F. Garrett, M. C. Whitton, E. D. Pisano MD, and H. Fuchs. Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 439–446. ACM SIGGRAPH, Addison Wesley, Aug. 1996.

[18] R. T. Surdick, E. T. Davis, R. A. King, and L. F. Hodges. The perception of distance in simulated visual displays: A comparison of the effectiveness and accuracy of multiple depth cues across viewing distances. *Presence: Teleoperators and Virtual Environments*, 6(5):513–531, Oct. 1997.

[19] B. Thomas, B. Close, J. Donoghue, J. Squires, P. D. Bondi, M. Morris, and W. Piekarski. ARQuake: An outdoor/indoor augmented reality first person application. In *International Symposium on Wearable Computers*, pages 139–146, Oct. 2000.

[20] A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger. Augmented reality in architectural construction, inspection, and renovation. In *Proceedings of the Third ASCE Congress for Computing in Civil Engineering*, June 1996.

Measuring Cooperative Robotic Systems Using Simulation-Based Virtual Environment

Xiaolin Hu
Computer Science Department
Georgia State University, Atlanta GA, USA 30303

Bernard P. Zeigler
Arizona Center for Integrative Modeling and Simulation
University of Arizona, Tucson AZ, USA 85721

ABSTRACT

Simulation-based study plays an important role in experimenting, understanding, and evaluating intelligent robotic systems. While robot models can be created and studied in a simulated environment, replacing some of the robot models with their real robot counterparts brings simulation-based study one step closer to the reality. It also provides the flexibility to allow real robots to be experimented within a virtual environment. This capability of robot-in-the-loop simulation is especially useful for large-scale cooperative robotic systems whose complexity and scalability severely limit the possibility for study and evaluation in a physical environment with real robots. This paper presents a simulation-based approach that allows a cooperative robotic system to be effectively evaluated in a virtual environment with combined real and virtual robots. This capability adds to conventional simulation-based study to form an integrated measuring process. An example of robotic convoy system is presented together with metrics to measure the formation coherence of cooperative robotic system. Some preliminary simulation results are presented.

KEYWORDS: *Cooperative Robotic System, Virtual Environment, Robot-in-the-Loop Simulation, Robotic Convoy System*

1. INTRODUCTION

Cooperative robotic systems couple computational intelligence to the physical world. These systems consist of multiple homogenous or heterogeneous robots that perceive the environment, make decisions, and carry out commands to affect the environment. Communication and cooperation is important for theses systems since their robots work as a collective team to finish common tasks. Several taxonomies and metrics have been defined to classify these systems. For example, Dudek, *etc.* [1] classifies robotic collectives along seven dimensions: size of the collective, communication range, communication topology, communication bandwidth, collective reconfigurability, processing ability of each collective unit, and collective composition. Balch [2] classifies the performance metric of multirobot tasks based on time, subject of action, resource limits, group movement, platform capabilities, *etc*.

The increasing complexity of collective robotic systems calls for systematic methods as well as supporting environments to experiment, understand, and evaluate these systems. To serve this purpose, modeling and simulation technologies are frequently applied. With simulation-based methods, models of robots can be built and simulated. Different configurations can be easily applied to experiment and measure the performance of the system under development. To allow simulation of robotic systems that actively interact with an external environment, an environment model needs to be created. This environment model serves as a "virtual" environment to provide sensory input to robot models and to response to robots' actuation. For example, a virtual environment for mobile robots simulation can have virtual obstacles that can be sensed by robot models, and it responds to robots' movements by updating new sensory information to robot models.

While robot models can be created and studied in a simulated environment, replacing some of the robot models with their real robot counterparts will bring simulation-based study one step closer to the reality and provides the flexibility to allows real robots to be experimented in a virtual environment. This capability of robot-in-the-loop simulation is especially useful for large-scale cooperative robotic systems whose complexity and scalability severely limit experimentation in a physical environment using all real robots. This paper presents an approach that allows a cooperative robotic system to be effectively evaluated in a virtual environment with combined real and virtual robots. This research is an extension to our previous work on a simulation-based software development methodology for cooperative robotic systems [3, 4]. This methodology supports "model continuity" so the simulation models in the design stage can be directly mapped to real robots to control the real robots in execution. It greatly eases the transition from simulation-based study to real robot implementation and

increases the confidence that the final system implements the behavior as been developed. This research is based on the Discrete Event System Specification (DEVS) modeling and simulation framework [5].

The concept of virtual environment has been largely used by the technology of virtual reality (VR), which has been applied to various areas such as simulation of manufacturing plants, the planning of robotic workcells, and robot teleoperation systems. While the research of VR mainly deals with the interaction with human operators, our work focuses on the interaction between robots and the virtual environment. The following research work is related to our research from this perspective. Komoriya and Tani [6] developed a virtual environment that allows a single real robot to be experimented in a virtual environment. Wang [7] proposed a simulation environment that allow real and virtual robot to work together. The work of RAVE [8] developed a simulation environment that supports multiple mobile robotic systems. Our research extends these works by developing a well-defined architecture, an incremental development process, and by integrating experimental frames to measure cooperative robotic systems with combined real and virtual robots in a systematic way.

This paper is organized as follows. Section 2 presents the virtual measuring environment from three aspects: the architecture, the measuring process, and the relationship to experimental frames. Section 3 describes a robotic convoy system as an illustrative example. The models of this system are first described, several metrics are then presented, and some preliminary simulation data is given. Section 4 concludes this work and provides future research directions.

2. A VIRTUAL EVALUATION ENVIRONMENT FOR ROBOTIC SYSTEMS

The effectiveness of this simulation-based virtual evaluation environment is supported by a well-defined architecture, an incremental measuring process, and by integrating experimental frames to specify metrics for performance measurement. Next we present these three aspects respectively.

2.1 Architecture of the Virtual Evaluation Environment

In this research we view robotic systems as a particular form of real-time systems that monitor, respond to, or control, an external environment. This environment is connected to the computer system through sensors, actuators, and other input-output interfaces [9]. A robotic system from this point of view consists of sensors, actuators and the decision-making unit. A cooperative robotic system is composed of a collection of robots that communicate with each other and interact with an environment.

This above description suggests the basic structure for a simulation-based virtual environment for robotic systems: an environment model, and a collection of robot models that include a decision making model, sensors, and actuators. The environment model represents the real environment within which the robotic system will be executed. It may include virtual obstacles, virtual robots, or any other entities that are useful for simulation-based study. It forms a virtual environment for the robots. The robot model represents the control software that governs the robot's behavior. It also includes sensor and actuator interfaces to bridge the decision-making model and the simulation-based virtual environment. In our research, we clearly separate a robot's decision-making unit, which is modeled as a DEVS atomic or coupled model, from the sensors and actuators that are modeled as DEVS Activities. Couplings can be added between DEVS Activities and the environment model thus messages can be passed between the decision-making model and the environment model through sensor/actuator Activities.

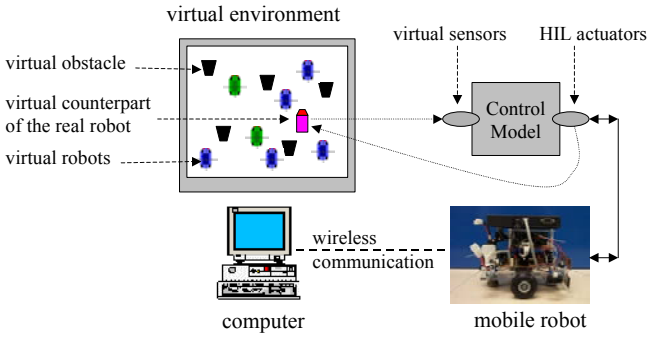


Figure 1: Architecture of robot-in-the-loop simulation

The clear separation between robots' decision-making model and sensor/actuator interfaces brings several advantages. First, it separates a robot's decision-making from hardware interaction, thus making it easier for the designer to focus on the decision-making model, which is the main design interest. Secondly, the existence of a sensor/actuator interface layer makes it possible for the decision-making model to interact with different types of sensors/actuators, as long as the interface functions between them are maintained the same. Thus depending on different experimental and measuring objectives, a robot model can be equipped with different sensors/actuators to be experimented and measured. Our previous work [10] has taken advantage of these features to allow direct transferring of the decision-making models from simulation to real robots execution – a capability referred as model continuity. During simulation, the decision-making model interacts with a virtual environment through virtual sensors/actuators; during real execution, the real robots' decision-making models interacts with a real environment through real sensor/actuator interfaces. An intermediate stage can also be developed to allow the decision-making model on a real robot to interact with the simulation-based virtual environment. We call this stage robot-in-the-loop simulation. It is achieved by configuring a real robot to use a combination of virtual and real

sensors/actuators. For example, Figure 1 shows an experimental setup where one real mobile robot works together with a virtual environment. In this example, the mobile robot uses its virtual sensor to get sensory input from the virtual environment and uses its real motor interface to move the robot. As a result, this real robot moves in a physical field based the sensory input from a virtual environment. Within this virtual environment, the robot can "see" virtual obstacles and other virtual robots that are simulated by computers. This capability of robot-in-the-loop simulation brings simulation-based study one step closer to the reality. It also makes it possible to study and measure several real robots within a large robotic system that may include hundreds of robots. In this case, the rest of robots can be provided by the simulation-based virtual environment.

One important issue for the robot-in-the-loop simulation is the synchronization between the real robots and the virtual environment. For example, in Figure 1, when the decision-making model issues a moving command, the real robot will move a distance in the physical environment. This change of position should also be updated by the virtual environment. For this purpose, each real robot has a virtual counterpart in the virtual environment. When a real robot moves, the position of its virtual counterpart will be updated. Thus the synchronization between the real robot and the virtual environment is actually the synchronization between the real robot and its virtual counterpart. Ideally, an independent monitoring system is needed to track the movement of the real robots and then inform the virtual environment to synchronize the distance and time of robots' movements. In our current implementation, a set of HIL (hardware-in-the-loop) sensors/actuators has been developed. These HIL sensors/actuators drive the real sensor/actuators, while in the meantime are coupled to the virtual environment thus messages can be sent to it. For the example shown in Figure 1, the HIL motor drives the motor of the robots. In the meantime it catches the *moveComplete* signal returned from the motor and then sends a message to the virtual environment to update the position of its virtual counterpart.

2.2 From Robot Model To Real Robot – An Incremental Measuring Process

Based on this virtual measuring environment, an incremental measuring process is developed. This process includes three steps and supports smooth transitions between them. These steps are measuring based on conventional simulation, measuring based on robot-in-the loop simulation, and measuring based on real robot execution. Figure 2 gives an example with two robots to illustrate this process.

The first step is conventional simulation, where all components are models and simulated by fast-mode or real-time simulators in one computer. As shown in Figure 2(a), both robot models are equipped with virtual sensors and actuators to interact with the virtual environment. Couplings between two robots can also be added so they can send messages to each other. We note that this is the same setup as the simulations that most robotic research uses. It has the most flexibility as all components are models and different configurations can be easily applied to measure the system under development.

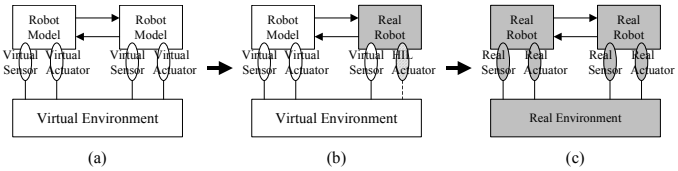


Figure 2: An incremental measuring process

The second step is robot-in-the-loop simulation where one or more real robots are measured and experimented within a virtual environment together with other virtual robots (robot models) that are simulated by computer. In this step, the virtual robots still use virtual sensors/actuators. However, depending on the measuring objectives, the real robots may have a combination of virtual and HIL sensors/actuators. For example, the real robot shown in Figure 2(b) uses a virtual sensor and a HIL actuator. The couplings between the two robots are maintained the same so the real and virtual robots can interact with each other in the same way as in the first step. However, the real commutation in this step happens across a wireless network, which is transparent to the robots. Since real robots are involved in the simulation, robot-in-the-loop simulation has to run in a real-time fashion.

The final step is the real system measurement, where real robots are measured in a real physical environment. These robots use real sensors and actuators. They communicate in the same way as the first two steps since the couplings between them are not changed through the process. The measurement of this step is from the reality, thus having the most fidelity. However, it is also most costly and time consuming among the three steps.

This incremental measuring process brings simulation-based study closer and closer to the reality. As the process proceeds, the flexibility (easy to experiment different configurations) and productivity (time saving and cost saving) of the measurement decreases and the fidelity (loyal to the reality) of the measurement increases.

2.3. Specify Measuring Metrics Using Experimental Frame

An experimental frame is a specification of the conditions within which the system is observed or experimented [5]. In DEVS-based modeling and simulation framework, an experimental frame is realized as a system that interacts with the source system, or System Under Test (SUT), to obtain the data of interest under specified conditions. It consists of four major subsections:

- *input stimuli*: specification of the class of admissible input time-dependent stimuli. This is the class from

which individual samples will be drawn and injected into the model or system under test for particular experiments.

- *control*: specification of the conditions under which the model or system will be initialized, continued under examination, and terminated.
- *metrics*: specification of the data summarization functions and the measures to be employed to provide quantitative or qualitative measures of the input/output behavior of the model. Examples of such metrics are performance indices, goodness-of-fit criteria, and error accuracy bound.
- *analysis*: specification of means by which the results of data collection in the frame will be analyzed to arrive at final conclusions. The data collected in a frame consists of pairs of input/output time functions.

When an experimental frame is realized as a system to interact with the SUT (or its model), the four specifications become components of the driving system. For example, a generator of output time functions implements the class of input stimuli.

Integrate experimental frames into the virtual measuring environment brings the advantage that measuring metrics can be formally specified. More research is on the way to integrate them in a structured way. In the meantime a set of measuring metrics is also under development for cooperative robotic systems.

3. ROBOT CONVOY: A CASE STUDY EXAMPLE

The presented virtual measuring environment has supported the development of a robotic convoy system. Below we briefly describe the model of this system, its measuring metrics, and some preliminary results that are collected from simulation-based study. We note that most results presented in this paper are collected from simulations that do not involve real robots. But in the next step we plan to measure the system using robot-in-the-loop simulation and expect to reach more interesting results. For example, we plan to use one real robot to run robot-in-the-loop simulation to check the convoy speed of this real robot and compare it with the data collected from the conventional simulation. Another result that we plan to check is to use two real robots neighboring to each other and then check the back robot's position errors based on the position and direction of its front robot in the physical environment.

3.1 System Description and System Model

This robot convoy system consists of an indefinite number of robots, saying N robots (N>1). These robots are in a line formation where each robot (except the leader and the ender) has a front neighbor and a back neighbor. The robots used in this system are car type mobile robots with wireless communication capability. They can move forward/backward and rotate around the center, and have whisker sensors and infrared sensors. [11].

One of the basic goals of this convoy system is to maintain the coherence of the line formation and to synchronize robots' movements. Synchronization means a robot cannot move forward if its "front" robot doesn't move, and it has to wait if its "back" robot doesn't catch up. To serve this purpose, synchronization messages are passed between a robot and its neighbors. To achieve coherence of the line formation, the moving parameters of a "front" robot are passed back. This allows the back robot to plan its own movement accordingly based on its front robot's movement. The system has no global communication and coordination since we want to study how global behavior can be achieved using localized sensing and communication.

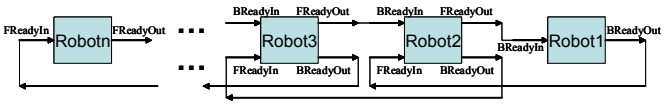


Figure 3: System model of the robotic convoy system

Figure 3 shows the model of this system. As we can see, this model includes N models (each of them is a DEVS coupled model), which are corresponding to the N robots in the system. Each intermediate robot model has two input ports: *FReadyIn*, *BReadyIn* and two output ports: *FReadyOut*, *BReadyOut*. These ports are used to send and/or receive synchronization messages between robots and to pass moving parameters from a "front" robot to the "back" robot. The couplings between them are shown in Figure 3.

During the convoy, the *leader* robot (*Robot1* in Figure 3) decides the path of convoy. Meanwhile, it will turn around if its infrared sensors indicate that there are obstacles ahead. All other robots conduct movement based on their sensory input and the moving parameters passed back from their front robots. Specifically, a robot will "predict" where its front robot is and turn to that direction. It then moves forward or backward to "catch" its front robot. After that it may go through an "adjust" process to make sure that it does not lose its front robot. This adjust process is necessary because noise and variance exist during a movement so a robot will not reach the desired position and direction after the movement. During adjustment, a robot "scans" around until it finds its front robot. Then it sends out a synchronization message to "inform" its front and back neighbors. Thus robots actually go through a basic "turn—move—adjust—inform" routine. For example, a robot R_{i-1} will turn angle α_{i-1} to the direction of its front robot R_{i-2}, move distance d_{i-1} to "catch" its front robot,

and then adjust itself with angle β_{i-1} to make sure it "sees" its front robot R_{i-2}. Figure 4 shows these moving parameters.

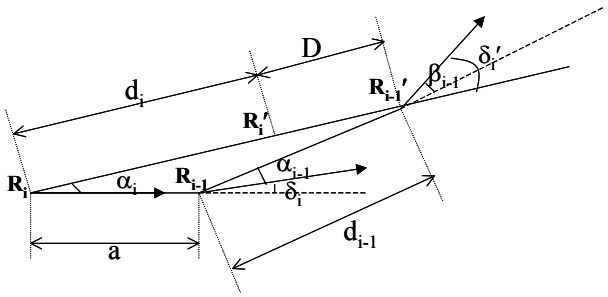


Figure 4: Moving parameters for robots' convoy

After the adjustment, R_{i-1} sends out a synchronization message to its neighbors. This synchronization message contains information of α_{i-1}, d_{i-1}, and β_{i-1}. Based on this information and its sensory data, R_i plans its movement. This is shown by Figure 4 and formulated by formula (1)-(3). Among these formulas, δ_i is the angle (direction) difference between R_i and R_{i-1}; a is the distance between robot R_i and R_{i-1} and can be calculated from the robot's infrared sensor data and the size of the robot. Specifically, the turning angle α_i of R_i is calculated by formula (1); the moving distance d_i can be calculated from formula (2), where D is the desired distance between R_i and R_{i-1}. Then the new angle difference δ_i' between R_i and R_{i-1} is updated by formula (3), where β_i is the adjusting angle for R_i. We note that due to noise and variance, the δ_i' calculated from formula (3) will not be the exact angle difference between R_i and R_{i-1}. However, it seems that this error does not accumulate as time proceeds.

$$tg\alpha_i = \frac{d_{i-1} * \sin(\alpha_{i-1} + \delta_i)}{a + d_{i-1} * \cos(\alpha_{i-1} + \delta_i)} \qquad (1)$$

$$d_i = \frac{d_{i-1} * \sin(\alpha_{i-1} + \delta_i)}{\sin \alpha_i} - D \qquad (2)$$

$$\delta_i' = \delta_i + \alpha_{i-1} + \beta_{i-1} - \alpha_i - \beta_i \qquad (3)$$

The model of each robot is developed based on the subsumption architecture [12]. It has the *Avoid* model to avoid collisions with any objects; the *Convoy* model to control robot's movement based on the rules as described above. It also has DEVE Activities to represent the sensor/actuator interfaces of the robot. A detailed description of a similar model can be found at [3].

Figure 5 shows the *Environment* model that we used for this example. This *Environment* model includes *TimeManager* models and the *SpaceManager* model. For each robot, there is a *TimeManager* corresponding to it. This *TimeManager* models the time for a robot to conduct a movement. The *SpaceManager* models the moving space, including the dimension, shape and location of the field and the objects inside the field. It also keeps track of robots' *(x,y)* positions and moving directions during simulation. Such tracking is needed to supply robots with the correct sensory data. To account for variability in the real motion, a random number generator provides a source of additive noise. Note that in this example we have ignored the dynamics of a movement as we treat each movement as an atomic action so the positions and directions of robots are updated discretely.

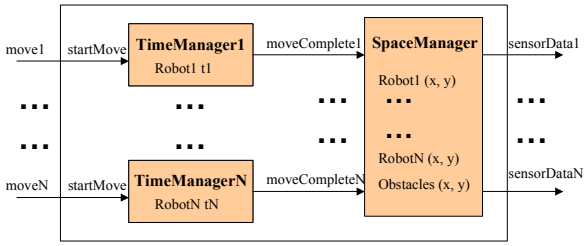
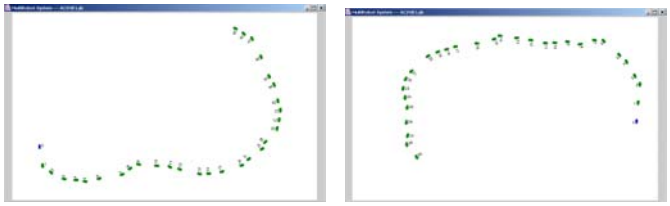


Figure 5: Environment model

With all these models, simulation was run and a graphic user interface was developed to show robots' movements. Figure 6 shows two snapshots of a robotic convoy system with 30 robots within a field surrounded by walls. As can be seen in this system, robots will not follow the exact track of the leader robot. However, they are able to follow their immediate front robots closely, thus forming a coherent team from a global point of view. Note that obstacles can also be easily added within the field.



(1) (2)

Figure 6: Snapshot of robots in motion

3.2 Measuring Metrics and Simulation Results

A good set of measuring metrics is very important to study this robotic convoy system. This section describes several metrics that we have developed. These metrics are neither final nor complete. However, they serve as a starting point to analyze and measure this system. Some preliminary simulation results based on these metrics are also presented and analyzed.

Convoy Speed and Number of Adjustment

The convoy speed of the team and the number of adjustment for each robot are among the most obvious results that can be obtained from simulation-based study. Both them can be viewed as metrics for the system's performance. In fact, these two metrics are correlated to each other: the larger the number of adjustment, the slower the convoy speed. Since

robots move in a coordinated way, we define the convoy speed as the speed of the leader robot. This can be calculated by dividing the moving distance by the logic time of simulation. The number of adjustment can be obtained directly from each robot.

Formation Coherence

Due to noise and variance in reality, there exists difference between a robot's real position, direction (angle) and its desired position and direction. This difference is affected by the variance of movement in real execution, which is modeled by adding noise into robot's movement in simulation. On the other hand, even though variance exists, this system can still conduct the convoy with some level of formation coherence. This is because an "adjust process" has been implemented that allows robots to adjust their positions/directions based on the feedback from its infrared sensors. Apparently the level of formation coherence is affected by the variance of movement. If this variance is large enough, even though a adjust process exists, the system will eventually fail to maintain its formation coherence.

To study this problem, we calculate each robot's position errors under the effect of distance noise factor (DNF) and angle noise factor (ANF). These two factors are the ratio of the maximum distance variance and maximum angle variance as compared to the robot's moving distance respectively. For example, if the angle noise factor is 0.1 and a robot moves forward 60, after its movement the robot will have maximum 6 degrees variance from its desired direction. Once each robot's position error is known, the average position error of the team can be derived. This average is an indicator for the convoy system's formation coherence: the smaller the error is, the more coherent the convoy system is. Formula (4) – (7) shows how the average position error can be calculated. In these formulas, D is the desired distance between robots and N is the total number of robot. In case the formation coherence is broken, saying robot R_i lose itself, $E_i(t)$ will increase continuously, making the average error $E(t)$ increase too. Note that the desired position ($x_{i\text{-}desired}$, $y_{i\text{-}desired}$) of R_i is calculated from its front robot R_{i-1}'s position, not related to any specific formations. Thus systems with different line formation shapes may have the same position errors.

$$E_i(t) = \sqrt{(x_i(t) - x_{i-desired}(t))^2 + (y_i(t) - y_{i-desired}(t))^2} \quad (4)$$

$$x_{i-desired}(t) = x_{i-1}(t) - D * \cos\theta_{i-1}(t) \quad (5)$$

$$y_{i-desired}(t) = y_{i-1}(t) - D * \sin\theta_{i-1}(t) \quad (6)$$

$$E(t) = \sum E_i(t) \Big/ N \quad (7)$$

Figure 7 shows the average poison error for a system with 30 robots, DNF=0.1, and ANF=0.08. The system starts with all robots at their desired positions. Thus as simulation proceeds, the position error increases from 0. It then reaches a "stable" stage where the position error oscillates around an average value (35.7 In this example). As we can see, in this system the position error does not accumulate over time. Thus we say that this system's formation coherence is maintained.

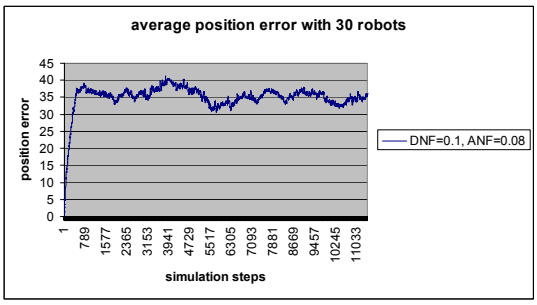


Figure 7: Average position error with 30 robots

Sensitivity

Since the formation coherence is affected by the noise factors, sensitivity analysis is useful to study if the system is robust to noise factors. To conduct sensitivity analysis, we run simulations with different noise factors and calculate the position errors. Figure 8 shows a system with 30 robots' average position errors under the effect of three sets of DNF and ANF: set 1 has DNF = 0.04, ANF = 0.04; set 2 has DNF =0.1, ANF = 0.08; set 3 has DNF = 0.2, ANF = 0.1. For analysis purpose, we omit the "transient " stage when the simulations start.

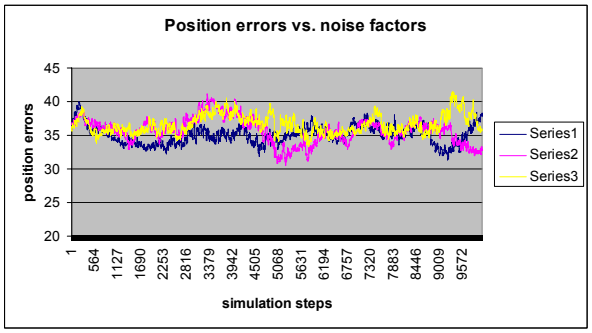


Figure 8: Average position errors vs. noise factors

Figure 8 shows that different noise factors result in different error patterns. However, for this system, all three errors are still maintained within a boundary (they do not accumulate as time increases). By calculating the average of them, we have average1 = 35.1, average2 =35.7, and average3 =36.6. From these data we can see that as the noise factor increases, the position error increases too. However, this change is insignificant as compared to change of the noise factors. Although more analysis is needed to reach any quantitative conclusion, we can say that this system is insensitive to the noise factors as long as these factors are within a safe boundary. This is because the system impalements an adjust process that allows robots to adjust themselves based on the feedback from their IR sensors.

Scalability

Scalability refers to the ability of a system to maintain its quality as the scale of the system increases. To study scalability, we change the number of robots and run simulation to see how that affects system's average position error (average over number of robots and over time). Figure 9 shows the position errors for the number of robots to be 10, 20, 30, and 40 with DNF =0.1 and ANF =0.08. It shows that the average position error increases as the number of robot increases. If this trend holds true with more robots, the system is not scalable in the sense that it will eventually break as more robots are added into the system.

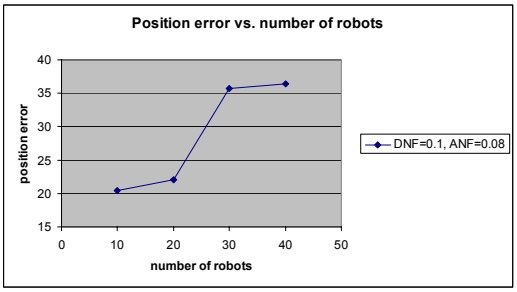


Figure 9: Average position errors vs. number of robots

4. CONCLUSION

This paper presents a simulation-based virtual evaluation environment for cooperative robotic systems. This virtual environment allows a combination of real and virtual robots to work together for a system-wide study and measurement. An incremental measuring process is developed to transition simulation-based study closer to reality as the process proceeds. Based on this virtual environment, a robotic convoy system was developed and presented in this paper as an illustrative example. Coherence metrics for this system were defined and preliminary simulation results were discussed.

We note that most results presented in this paper are collected from simulations that do not involve real robots. But in the next step we plan to measure the system using robot-in-the-loop simulation and expect to gather more interesting results. In the meantime, a set of more complete evaluation metrics is also under development for the robotic convoy systems presented in this paper.

5. ACKNOWLEDGEMENT

This research was supported by NSF grant DMI- 0122227, "DEVS as a Formal Framework for Scalable Enterprise Systems."

6. REFERENCE

[1] Dudek, G., Jenkin, M., and Milios, E., "A Taxonomy of Multirobot Systems", *Robot Teams*, Edited by Balch, T., and Parker L.E., A K Peters, 2002

[2] Balch, T., "Taxonomies of Multirobot Task and Reward", *Robot Teams*, Edited by Balch, T., and Parker L.E., A K Peters, 2002

[3] Hu, X., and Zeigler, B. P., "Model Continuity to Support Software Development for Distributed Robotic Systems: a Team Formation Example", *Journal of Intelligent & Robotic Systems, Theory & Application, Special Issue: Multiple and Distributed Cooperating Robots*, pp. 71-87, January, 2004

[4] Hu, X., and Zeigler, B.P., "A Simulation-based Software Development Methodology for Cooperative Real-time Intelligent Systems", to appear in *Annual of Complex Systems and Intelligence Science*, World Scientific Publishing Co., 2004

[5] Zeigler, B.P., Kim, T.G., *et al*.. Theory of Modeling and Simulation. New York, NY, Academic Press, 2000.

[6] Komoriya, K.; Tani, K., "Utilization of the virtual environment system for autonomous control of mobile robots", *Intelligent Motion Control, 1990. Proceedings of the IEEE International Workshop on*, Volume: 2, 20-22 August 1990

[7] Wang. J.: Methodology and design principles for a generic simulation platform for distributed robotic system experimentation and development. *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, Volume: 2, 1997 Page(s): 1245 -1250 vol.2

[8] Dixon, K.; Dolan, J.; Wesley Huang; Paredis, C.; Khosla, P., "RAVE: a real and virtual environment for multiple mobile robot systems", *Intelligent Robots and Systems, 1999. IROS '99. Proceedings*. 1999 IEEE/RSJ International Conference on , Volume: 3 , 17-21 Oct. 1999

[9] Shaw, S. C., Real-time Systems and Software, John Wiley & Sons, 2001

[10] Hu, X., *A Simulation-based Software Development Methodology for Distributed Real-time Systems*, Dissertation, University of Arizona, 2003

[11] Peipelman. J., N. Alvarez, K. Galinet, R. Olmos.: 498 A & B Technical Report. Department of Electrical and Computer Engineering, University of Arizona, 2002

[12] Brooks, R. A., "A Robust Layered Control System For A Mobile Robot", IEEE Journal Of Robotics And Automation, RA-2, April. pp. 14-23, March 1986

Behavioral Entropy in Human-Robot Interaction

M. A. Goodrich[1], E. R. Boer[2], J. W. Crandall[1], R. W. Ricks[1], and M. L. Quigley[1]

[1] Computer Science Department
Brigham Young University
Provo, UT, USA

[2] Erwin R. Boer Consulting
San Diego, CA, USA

ABSTRACT

The ability to quickly and accurately measure how various design decisions affect human workload is an important need in human-robot interaction (HRI) and other HMI domains. Although various techniques allow workload to be estimated, it is important to develop methods for obtaining workload estimates objectively and in real-time without interfering with the normal operation of human. In this paper, we develop *behavioral entropy* as a technique for estimating human workload in HRI domains. We develop relevant theory and present case studies that help validate the power of behavioral entropy.

1 Introduction

In a recent article on useful metrics in human-robot interaction (HRI), Fong et al. identified the need to find "nonintrusive measures of workload that can characterize operator stress in real-time" [3]. The importance of having a real-time estimate requires an objective (rather than subjective) measure of workload that is reliable and applicable to many interfaces. The purpose of this paper is to present a technique, called *behavioral entropy* [6] that measures human workload in HRI domains. This metric efficiently utilizes operator activity to estimate human workload.

A real-time measure of workload in HRI has several possible applications.

- **Design of adjustable autonomy systems.** Intelligent interfaces could be used to identify high workload situations, and the resulting information could be used to adjust robot autonomy or alert other humans to support the operator. This facilitates design of more efficient mixed-initiative systems [1] that follow principles of situation-adaptive autonomy [4].
- **Comparison of interfaces and autonomy modes.** Various HRI systems, including various interfaces and robot autonomy modes, could be compared over time. This ability to compare designs over time allows not only comparison of average workload, but also comparisons of peak workload, minimum workload, and workload patterns.
- **Diagnosis of causes of high workload.** External events that trigger high workload could be identified and diagnosed. By associating a real-time estimate of workload with external events, those events that cause workload spikes could be identified. These events might include environmental contingencies, robot failures, interface issues, and so on.
- **Design of Adaptive Systems.** Interfaces or robots that learn to support human activity could be improved. Most HRI learning systems either learn by direct teaching or learn by observing a human teleoperating a robot. These systems could be augmented to include implicit human cues, such as identifying robot behaviors that cause workload spikes, and thereby improve interaction efficiency through interface adaption.

The idea of behavioral entropy was developed in for use in estimating driver workload in an automobile driving context. This first application restricted attention to human activity as recorded in the steering wheel of a vehicle and was called "steering entropy." Subsequently, Boer generalized this concept to general human activity, and denoted the concept as behavioral entropy [6].

Behavioral entropy differs in a number of ways from the three other primary methods for evaluating workload: physiological measurements, secondary task studies, and *post hoc* workload measurements (such as NASATLX). Physiological measurements exploit the strong correlation between human effort and the body's physical response. Such measures are objective and near to real-time, but much work needs to be done to understand the precise nature of the correlation between effort and response; this work includes developing signal processing techniques that rapidly and correctly separate signal from noise. Secondary task studies allow diagnosis of human workload by measuring how performance declines as other work is added. However, such measures are invasive and change the way the primary task

is performed. *Post hoc* measurements exploit a human's ability to express their perceived workload after the fact. Such measures are important because they allow a human to be able to state how they perceived their experiences, but they are subject to many psychological biases, such as recency effects. Moreover, they are not real-time.

Behavioral entropy exploits patterns observed in human activity within an HRI context. Generally speaking, when intelligent operators perform a practiced skill under conditions of good information, they use an anticipatory control strategy. This means that they are able to predict the consequences of their actions or inactions, and select efficient behaviors that alter these consequences. When human operators are under conditions of high workload or other form of degraded performance, they anticipate less and react more and, as a result, their action selection tends to be more exaggerated. Anticipatory behaviors tend to be more smooth with less dramatic magnitudes and less frequent changes than reactive behaviors. Behavioral entropy is sensitive to this difference between reactive and anticipatory behaviors.

This paper is organized as follows. We first review and develop the key concepts associated with behavioral entropy. We then present three case studies that utilize behavioral entropy in HRI-related domains. The first two case studies help establish the hypothesis that average behavioral entropy is a useful and objective metric for comparing design decisions. The third case study helps illustrate that behavioral entropy can be used in real-time. We conclude by presenting future work with an emphasis on work needed to allow behavioral entropy to be used in broad-reaching HRI studies.

2 Behavioral Entropy

Behavioral entropy estimates workload by first observing patterns of human activity under normal conditions, and then noting deviations from these patterns. Consider, for example, how a human might teleoperate a robot via a joystick under laboratory conditions (good communications, alert operator, etc.). Under these ideal conditions, joystick activity follows observable patterns.

Such patterns of activity can be captured in a model of activity. A well-known phenomena associated with modelling is that simple models often explain most activity, but extending these models to explain all activity often makes the models grow exponentially in their complexity. This is true in human-robot interaction domains as well. For example, much of what is done with the joystick under teleoperation can be described with simple ARMA models [5], but modelling all joystick activity requires very sophisticated models.

Norbert Wiener once said, "It is my thesis that the physical functioning of the living individual and the operation of some of the newer communication machines are precisely parallel in their analogous attempts to control entropy through feedback" [8]. Through repeated interactions with robot or interface systems, humans build an understanding of various effects and relationships. Perhaps most importantly, they build an understanding of (a) the effect of their actions on the systems and (b) the dynamics of the environment.

Such an understanding translates into an efficient interaction. To paraphrase Wiener, people work to reduce entropy so skilled behavior minimizes entropy. This manifests itself in human behavior that is anticipatory, of the lowest possible bandwidth, and of the lowest possible magnitude. Such behavior lends itself to modelling and prediction.

2.1 Modelling

Suppose that we identify a simple model that describes how the operator uses the input device to a human-robot interface. (Such input devices can include a joystick, mouse, stylus, etc.) Formally, let x_t denote the state of the world at time t and let a_t denote operator activity at time t. A model M, denoted by,

$$M : X_t \times X_{t-1} \times \ldots \times X_0 \times A_t \times A_{t-1} \times \ldots \times A_0 \rightarrow A_{t+1}$$

can be used to predict operator activity at time $t+1$,

$$\hat{a}_{t+1} = M(x_t, x_{t-1}, \ldots, x_0; a_t, a_{t-1}, \ldots, a_0),$$

where the $\hat{a}$ indicates a prediction. Given this model we can generate a prediction of what we think the operator will do next.

If we adopt Wiener's hypothesis that people work to control entropy, then we can believe the hypothesis that people's behavior patterns have lower magnitude, have lower bandwidth, and are anticipatory when good information is present and the task is well practiced. If so, then low frequency components of their observed activities represent the anticipatory aspects of their behavior. Consequently, we should be able to identify a model of this behavior.

Their are several possible choices for these models. We could use general linear models, such as ARMA or state-space models, but in the interest of simplicity we restrict attention to only one type of model in this paper[1]: a Taylor series expansion. The Taylor series expansion supposes that behavior is a smooth function of past activities, and then uses the first derivatives to model the key elements of this function.

[1]Note that it might sometimes be better to use a sample and hold model to predict joystick movement because joystick operation, under some conditions, tends to be "bang-bang." This is left as an area for future work.

2.2 Model Errors

Clearly, a model will not correctly predict all operator activity. Let $e_t = \hat{a}_t - a_t$ denote the error in this prediction. The statistical properties of this error are useful in estimating operator workload. To illustrate this, suppose that the prediction error sequence, e_t, has been observed for $0 \leq t \leq N$. Given this sequence, $\{e_0, e_1, \ldots, e_N\}$, we can create a histogram of prediction errors. By normalizing this histogram, we create a probability mass function that is a non-parametric estimate of the prediction error density function. Let $p_E(e;t)$ denote this estimate of the prediction error density function.

The key idea behind using behavioral entropy is to look at the type of information that exists in the prediction error density function. More precisely, we will look at the information present in the prediction error density functions under two conditions. If one condition is produced under circumstances that allow better anticipatory control than the second condition, then operator activity under the first condition should be more predictable. In other words, there will be less information in the prediction error density function. Since good interfaces and autonomy modes support operators in their desire to minimize entropy, good designs should have more predictable behaviors. To better understand how to describe the information available in the prediction error density function, it is useful to review the relationship between creating a model and the notion of information.

2.3 Models and Information

One way to interpret a model of a phenomena is as a mechanism that gives you *information* about the phenomena. In this sense, we use the term "information" in the information theoretic sense as the number of bits required to describe the phenomena. If the model is very good, then deviations from the model predictions likely arise from randomness; if the model is poor, then deviations from the model predictions likely arise from structured aspects of the phenomena that are not captured in the model. For example, consider a phenomena where two variables are related to each other by a cosine function. If we create a linear model for this sinusoidal relationship, then deviations from the model predictions arise from the fact that the underlying phenomena is a sinusoid and not a line. If, by contrast, we create a sinusoid model for this relationship, then deviations from model predictions arise from random perturbations in the relationship.

We can use this relationship between model predictions and information to create a mechanism that identifies when activity is no longer ascribed to the phenomena encoded in the model. In other words, we can use the prediction error density function to detect when things are different from what we predict and therefore detect when the phenomena is behaving oddly. Since predictions are subject to random error, we are actually going to use the prediction error density function to detect when things are *different enough* to conclude that the observed phenomena is not consistent with what was predicted.

Consider the amount of information available in the prediction error density. Under ideal conditions (e.g., laboratory setting, alert human, no interruptions) the prediction error density, $p_E(e;t)$, has a certain amount of information in it. This information is attributable to random noise and to small unmodelled aspects of the pattern of human activity. If conditions of high workload occur, then the pattern of human activity changes and so does the resulting prediction error density. By comparing the amount of information present in the prediction error density function under ideal conditions to the information present under loaded conditions, we can detect when these loaded conditions have occurred.

For example, consider the problem of teleoperating a robot via a joystick. We can create a simple model for how the joystick moves under ideal conditions and measure the information in the corresponding prediction error density. When the task suddenly becomes more difficult, operator activity tends to become more erratic and more pronounced. Instead of seeing small changes in the joystick position made relatively infrequently, large and rapid changes in joystick position are more frequently observed. If we were to compare the prediction error density under ideal conditions with the density under the loaded conditions, we would see that the density under loaded conditions is much more spread out; this is illustrated in Figure 1. This increased density spread indicates that there is information in the system not captured by the model; it indicates that the operator is doing more than we predicted. Such things can occur, for example, when an operator overcompensates after hav-

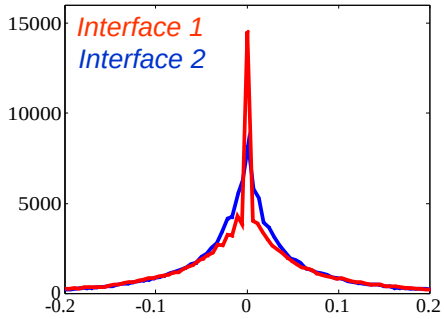


Figure 1: Prediction error histograms under two workload conditions: nominal and loaded.

ing attention diverted or when an operator is confused because information is presented poorly.

2.4 Model Information and Prediction Error Entropy

To create a metric that represents this change in prediction error density, we return to the information theoretic interpretation of the model. We use *entropy*, $H(E;t)$, defined as $H(E;t) = -\sum_{e \in E} p_E(e;t) \log p_E(e;t)$, as the measure of information available in the prediction error density. If we identify baseline entropy using ideal conditions then we can detect periods of high workload by comparing $H(E;t)$ against the baseline entropy. Similarly, if we can identify entropy under one HRI system design, then we can compare this entropy with another design to help determine which design better supports the human.

We refer to $H(E;t)$ as behavioral entropy, indicating that it is the amount of information present in a human's behavior that was not captured by a model. Experiments in automobile driving indicate that this objective measure of entropy correlates well with subjective measures of workload [6].

2.5 Segue

In the remainder of this paper, we present three case studies that use behavioral entropy to perform various HRI-related tasks.In the case studies, we will first present the goal of the experiment, describe what the operator was asked to do, discuss characteristics of the environment and the interface, present the model used to predict operator activity, and present what we use as a baseline. The first two studies use average behavioral entropy and lend support to the thesis that behavioral entropy discriminates between good and bad operating conditions. The third case study uses a real-time version of behavioral entropy to learn proper force feedback; this case study uses a reinforcement learning technique to show that real-time estimates of behavioral entropy are informative.

3 Case Study 1: Comparing Usability of Two Teleoperation Schemes

In the first case study, behavioral entropy was used to compare two different robot autonomy modes to determine which autonomy mode was easier for humans to use. The hypothesis is that differences in behavioral entropy correlate well with other measures of performance and are therefore useful in comparing different robot autonomy modes. We compute the prediction error density function using a prediction error sequence from the entire experiment, and compare the entropy of this density function with other performance measures under two robot autonomy modes.

Subjects were asked to drive a robot around the top floor of the Computer Science Department at Brigham Young University using two different autonomy modes: manual teleoperation and shared-control teleoperation [2]. In addition to driving the robot with their right hand (with a joystick), the users were asked to answer multiple choice (two-digit) addition and subtraction problems with their left hand. This experiment setup is illustrated in Figure 2. Subjects were told to guide the robot through the hallways as quickly as possible while answering as many math questions as possible. The video feed from the robot's onboard camera was displayed on the same screen as the math problems. In this case study, entropy calculations were taken of joystick movements. Only the angle (not the magnitude) from the joystick input was used to calculate entropy.

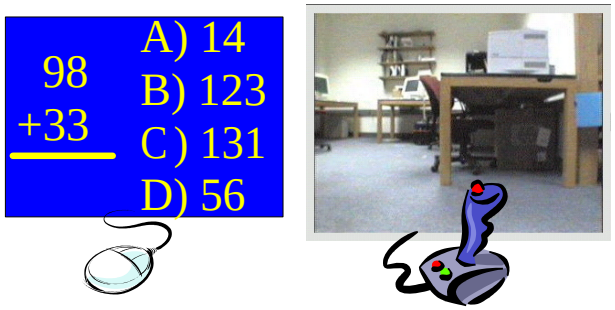


Figure 2: Interface used to compare the two autonomy modes.

3.1 Methods

A second-order Taylor series model of operator behavior was used. This means that the operator activity at time t, a_t, was determined using observations of activity at times $t-3$ through $t-1$ (i.e., using a_{t-1}, a_{t-2}, and a_{t-3}). In this experiment, only joystick angle was used, and it is a reasonable assumption that if the operator is using angle a at times $t-3$ through $t-1$ then they will likely use this same angle at time t.

An important aspect computing entropy is selecting how to reliably create a discretized probability mass function from histogram data. In this experiment, a single operator guided the robot through the maze using the shared control autonomy mode without performing the secondary task. The history of joystick angles was

recorded, and the prediction error histogram was created. This histogram was discretized into 9 unequally spaced bins.

The bins were created using the following procedure. Using the resulting baseline prediction error density, we identify the parameter, α, which encapsulates 90% of the data, $Pr(-\alpha <= \text{error} <= \alpha) = 0.90$. This value of α is used to classify each angle, or the error from the predicted angle, into nine bins,

$$\{(-\infty, -5\alpha), [-5\alpha, -2.5\alpha), [-2.5\alpha, -\alpha), \ldots$$
$$[-\alpha, -0.5\alpha), [-0.5\alpha, 0.5\alpha], (5\alpha, \infty)\}.$$

Since the bins were created from a single operator, this implementation of behavioral entropy is not sensitive enough to allow comparisons between individuals. Simply put, these values will be slightly different for each individual under ideal circumstances so the entropy computed from these values will differ under loaded conditions. As a result, entropy calculations should not be used to compare two individuals. However, since the same model and binning scheme were used under the two experimental conditions (with shared control and with direct control), it is possible to compare entropy for a given individual on the two different tasks.

3.2 Results

This experiment was performed in the real world and in a simulated world. Various results are shown for these case studies. Table 1 and 2 show the results from experiments in real and simulated worlds, respectively. In the tables, high values are good and low values are bad, with the exception of the entropy measurement which is reversed. In the tables, *Neglect* indicates the percentage of time that the operator spent doing arithmetic problems, *Performance* indicates how efficiently the primary task was completed as a percentage of the maximum possible performance, *# per min* indicate the number of arithmetic problems that were attempted per minute, and *% Correct* indicates what percentage of the attempted arithmetic problems were answered correctly by the subject.

For all measurements in both tables, subjects tended to do better using shared control than using direct control. Behavioral entropy is consistent with these other measurements since the highest entropy measure for shared control is lower than the lowest entropy measure for manual control. Also, entropy is highly correlated with performance (lower entropy corresponds to higher performance) and the amount of time the human "neglected" (i.e., did math problems) the robot (lower entropy means, generally, more neglect). There also appears to be correlation between secondary task proficiency and entropy.

Shared-Control Results

Participant	A	B	C	D	Ave.
% Neglect	51%	67%	46%	63%	57%
% Performance	77%	96%	81%	86%	85%
# per min.	9.5	18.9	8.9	10.6	12.0
% Correct	74%	98%	94%	66%	83%
Entropy	0.56	0.42	0.51	0.35	0.46

Direct-Control Results

Participant	A	B	C	D	Ave.
% Neglect	36%	31%	22%	62%	38%
% Performance	57%	76%	58%	60%	63%
# per min.	6.4	9.1	3.9	9.8	7.3
% Correct	72%	85%	79%	61%	74%
Entropy	0.72	0.79	0.67	0.63	0.70

Table 1: Results from the experiment in the real world.

The key to understanding how this data supports the use of entropy as a measure of workload lies in the dual task nature of the experiment. Adopting a limited resource model for cognitive information processing [7], we can assume that motivated subjects spend most of their cognitive effort either guiding the robot or solving math problems. This assumption is supported by the observation that the shared control autonomy mode was easier to use and freed subjects to spend more time solving math problems.

In the absence of a secondary task, it is reasonable to assume that performances using the two autonomy modes would have been closer. The presence of the secondary task provided stronger evidence that the shared control autonomy mode was easier to use, but this secondary task also changed the nature of the task that the operator was asked to perform.

Behavioral entropy data was consistent with the conclusion that direct control required more work. Since behavioral entropy only required observations of operator activity (and did not require an intrusive secondary task), we could have used behavioral entropy without the secondary task to conclude that the shared control autonomy mode was easier to use than the direct control autonomy mode.

In summary, since higher entropy values occurred under direct control, the evidence supports the hypothesis that entropy allows us to identify which autonomy mode imposes higher human workload.

4 Case Study 2: Comparing the Usability of Two Interfaces

In the previous case study, we used behavioral entropy to measure the differences between two autonomy modes. In this case study, we determine whether entropy is a reliable method for determining which of two interfaces provides better support for robot teleoperation.

Shared-Control Results

Participant	A	B	C	D	E	F	G	Ave.
% Neglect	74%	72%	77%	61%	73%	72%	74%	72%
% Performance	97%	88%	94%	98%	85%	92%	97%	93%
# per min.	12.0	12.4	10.3	12.1	13.8	16.3	15.8	13.2
% Correct	71%	63%	39%	94%	85%	88%	78%	74%
Entropy	0.37	0.49	0.45	0.32	0.39	0.55	0.29	0.41

Direct-Control Results

Participant	A	B	C	D	E	F	G	Ave.
% Neglect	65%	70%	70%	34%	70%	68%	73%	64%
% Performance	83%	74%	96%	96%	88%	75%	81%	84%
# per min.	10.2	12.5	9.8	6.4	11.5	12.7	13.4	10.9
% Correct	57%	63%	38%	79%	71%	88%	77%	67%
Entropy	0.68	0.77	0.69	0.57	0.66	0.72	0.67	0.68

Table 2: Results from the simulated world.

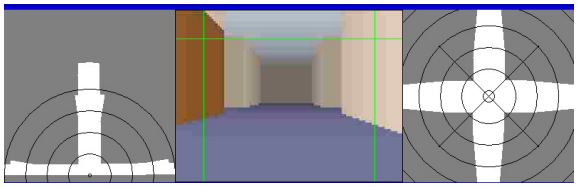


Figure 3: Interface that displays sensor readings side by side.

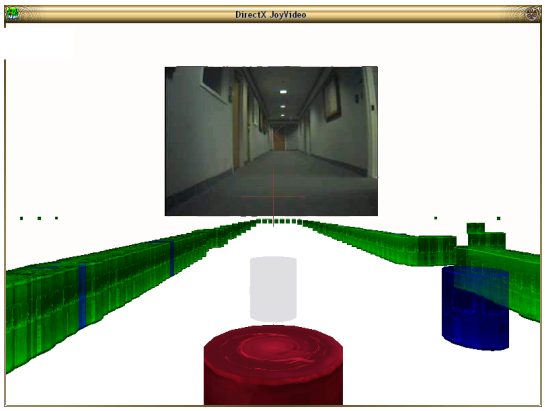


Figure 4: Interface that integrates sensor readings into perspective view.

The two interfaces are shown in Figures 3-4. The first interface displays, from left to right, laser range finger readings, video, and sonar in a side by side format. The second interface integrates these three sensor readings in a pseudo-perspective view, with a representation of the robot displayed in this view.

We conducted a series of experiments to compare the two interfaces. In a balanced experiment design with a randomized schedule, subjects teleoperate a simulated robot through three mazes while performing a memory task where they must remember five images. After completing the maze, subjects complete a memory test by selecting the images they saw before from a list and putting the images in order.

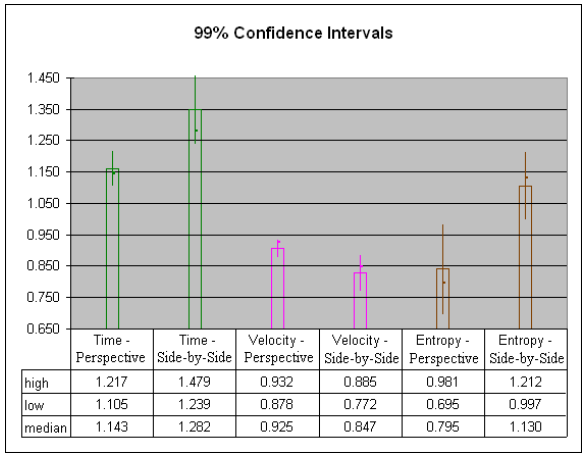
4.1 Methods

As in the previous case study, the model of joystick angles was based on a Taylor series. Given studies on human control characteristics, we used a sample interval of 150ms and averaged all joystick angles within a 150ms window as our sample. Given the series of joystick angles, we created the prediction error density using the difference between the predicted value and the observed value. From the set of prediction error densities (one for each maze and for each interface), it is necessary to identify a baseline density from which bins are created. We did this by having each subject guide the robot through one maze without performing the memory task using the side by side interface. Prediction errors from this entire data set were then used to create the bins used to determine entropy using the technique described in the previous case study.

4.2 Results

The following data were collected for 32 subjects: time to guide a robot through a maze, behavioral entropy, average velocity, number of collisions, and performance on a memory task. (Most likely, the memory tasks were not hard enough because about 70% of the test subjects aced the memory task.) With the exception of the memory task, for which we did not get any meaningful data, all of these measures demonstrate that the new interface is effective for helping people control a robot.

Figure 5 summarizes the data, and shows that the side by side interface is inferior to the perspective interface for each measurement. These findings support the conclusion that behavioral entropy is a useful measure for determining when one interface is more difficult to use than another. Moreover, the data is strongly supported



	Time - Perspective	Time - Side-by-Side	Velocity - Perspective	Velocity - Side-by-Side	Entropy - Perspective	Entropy - Side-by-Side
high	1.217	1.479	0.932	0.885	0.981	1.212
low	1.105	1.239	0.878	0.772	0.695	0.997
median	1.143	1.282	0.925	0.847	0.795	1.130

Figure 5: A comparison of the performance metrics averaged over all subjects and all test worlds.

by the number of collisions experienced; the number of collisions using the side by side display were more than doubled the number of collisions using the perspective display.

The perspective interface tends to be easier to use because it helps people predict where the robot will be heading and updates this information frequently. The side by side interface requires people to do their own prediction and only updates sensor values when new information is received. Using the side by side interface caused people to change their control input when new information was received from the robot. This causes the position of the joystick to be somewhat erratic and jump from position to position. People driving the perspective interface often make more frequent but less dramatic corrections. This can be attributed to lower workloads or finer control.

5 Case Study 3: Using Behavioral Entropy to Build an Interface

In this section, we present a case study that uses a real-time estimate of behavioral entropy as a major factor in constructing an estimate of driver workload. This workload estimate is then used to learn haptic control policies for an accelerator pedal that increase the safety of the driver without significantly increasing workload. We use the ability of reinforcement learning to detect patterns in stochastic reinforcers to support the conclusion that the real-time estimate of behavioral entropy contains useful information about when people have workload spikes.

5.1 Methods

In the experiment, subjects followed an erratic lead vehicle with and without the learned force profile. During the experiment, subjects solved two-digit arithmetic problems that appeared on the simulator by pushing buttons on the steering wheel.

We trained an artificial agent using satisficing Q-learning, a dual attribute version of the standard Q-learning algorithm, to minimize workload while preserving safety. This was done by creating the following dichotomous goals: Goal #1: Don't allow the vehicle to experience a crash or a near-crash. Goal #2: Reduce driver workload as much as possible. Clearly, these two goals are in conflict with each other whenever the vehicle is in a non-trivial situation. Goal #1 was realized by penalizing policies that lead to a collision or near collision. Goal #2 was realized by only rewarding actions that induced a low user workload. Both behavioral entropy and impedance (i.e. the extent to which interface actions directly opposed driver actions) were used to estimate driver workload and determine whether actions produced a workload low enough to be rewarded.

5.2 Results

A control policy for a force-feedback gas pedal was learned using the methodology described above. Entropy of the accelerator pedal position was calculated in real-time and combined with instantaneous impedance to form an estimate of driver workload. This estimate of driver workload was compared against baseline driving and empirically chosen thresholds to determine whether an action induced too much workload to be rewarded. The learning algorithm was trained during ten minutes of exploratory driving by a single operator who allowed several rear-end collisions to occur in order to propagate penalty data throughout the state space. The agent learned to balance driver workload with expected risk, applying forces to the pedal only in states where experience demonstrated it to be useful.

Test subjects responded enthusiastically to this haptic support. Pedal entropy remained similar to drivers that were in unassisted trials, but the overall safety (as measured by time spent with time to contact less than 0.7 seconds) was reduced by 45%. Using high entropy to prevent rewarding an action during the training period was very helpful in this context, as the agent learned a control policy that informed the user of danger without significantly increasing overall entropy. This evidence supports the conclusion that the online estimate of behavioral entropy contained useful information about the workload experienced by a distracted driver with and without force feedback support. This evidence is bolstered by plotting the prediction error density functions

and noting that the density corresponding to no forces is shorter and fatter than the other; these densities were shown in Figure 1.

6 Discussion and Future Work

In this paper, we presented three case studies that demonstrated how behavioral entropy can be used in HRI studies. These case studies showed that behavioral entropy reliably predicted workload and correlated well with other measures of human performance. The third case study also demonstrated how a real-time estimate of behavioral entropy provided useful information to a machine learning algorithm; this algorithm decreased the number of near collisions in a driving simulator without increasing subjective workload.

Two areas of future work need to be explored before entropy can be used widely. First, a guide for selecting parameters in the entropy computation algorithm need to be identified. These parameters include what models should be chosen, how model parameters should be chosen, how binning should be performed, and how a window size for real-time entropy estimates should be selected.

Second, the relationship between entropy and other human factors measures should be better established. This includes researching how average entropy or its variations (e.g., peak entropy, minimum entropy) correlate with, for example, trust, neglect tolerance, interface efficiency, and so on.

Acknowledgements

The authors would like to thank DARPA (under contract #NBCH1020013) and Nissan Motor Company for their partial support of this work.

References

[1] David J. Bruemmer, Julie L. Marble, David D. Dudenhoeffer, Matthew O. Anderson, and Mark D. McKay. Mixed-initiative control for remote characterization of hazardous environments. In *HICSS*, 2003.

[2] J. W. Crandall and M. A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *Proceedings of the 2002 IEEE /RSJ International Conference on Intelligent Robots and Systems*, Luasanne, Switzerland, 2002.

[3] T. Fong, D. Kaber, M. Lewis, J. Sholtz, A. Schultz, and A. Steinfeld. Common metrics for human-robot interaction. Submitted to IROS 2004, 2004.

[4] T. Inagaki and K. Inoue. Adaptive choice of a safety management scheme upon an alarm under supervisory control of a large-complex system. *Reliability Engineering and System Safety*, 39:81–87, 1993.

[5] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1987.

[6] O. Nakayama, T. Futami, T. Nakamura, and E. R. Boer. Development of a steering entropy method for evaluating driver workload. In *SAE Technical Paper Series: #1999-01-0892: Presented at the International Congress and Exposition*, Detroit, Michigan, March 1-4, 1999.

[7] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, third edition, 2000.

[8] N. Wiener. *The Human Use of Human Beings: Cybernetics and Society*. Free Association Books, London, 1989.

Evaluation of Human-robot Interaction in the NIST Reference Search and Rescue Test Arenas

Jean Scholtz, Brian Antonishek, and Jeff Young
National Institute of Standards and Technology

Abstract

In this paper we discuss studies we have been conducting on human-robot interaction (HRI) during the Urban Search and Rescue (USAR) competitions in the NIST Reference Test Arena. We discuss some of the analyses we have already done on the data we have collected and present the guidelines we have produced based on these studies. We discuss future plans for augmenting USAR competitions to specifically compare different methods of HRI.

Introduction

The ultimate evaluation of how humans and robots interact is the measure of their combined performance. In search and rescue the human-robot team has two goals: to locate victims and to provide accurate information about their location and their alertness state to human rescuers. These goals need to be achieved under a number of constraints. Teams need to operate for extended periods of time; the number of personnel used in the operation should be limited due to the dangerous nature of the operation; and the tasks need to be accomplished quickly to maximize the lives that can be saved [2]. Many human-robot search and rescue teams have participated in Urban Search and Rescue (USAR) competitions in the NIST test arenas [7,8]. The overall scoring for these competitions emphasizes these goals and constraints. Although scoring varies from year to year, the teams are rewarded for locating victims in a timely fashion, accurately assessing their condition, and providing good maps for rescue workers. Teams are penalized for causing further damage to the collapsed structure. Teams requiring multiple operators for individual robots are also penalized.

Good human-robot interaction (HRI) contributes heavily to a team's overall score. However, there are a number of other contributing factors as well, including the mobility of the robot, the skill of the operator, the robustness of the hardware, software, and communications, and the sensory perception provided. We are interested in evaluating the various user interfaces to determine what information and information presentation contributes to the overall performance of the system.

Pros and Cons of Using the USAR Competitions for HRI Evaluation

The primary benefit of using these competitions for studying HRI evaluation is that the competitors provide many more ideas for user interfaces than we, as researchers, could possibly prototype and test.

The limitations are that we can only study the operator role [11]. The operators in the competitions are expert users, i.e., robotics researchers. We are not allowed to interfere with the competition environment which means that we cannot collect think-aloud or talk-aloud protocols [5,6] from the operators. It is difficult to interview the operators after their runs as they are busy getting ready for their next round. Moreover, the teams come from allover the world and there are language barriers to overcome. The user interfaces and the robots are dynamic. The teams make changes during the competitions. Different robots are used; different sensors are used; different teammates take turns at being the operator.

In spite of the limitations, these competitions provide a rich source of data in a reasonable USAR simulation.

Data Collection

We have collected data at six major competitions starting in 2002. We collect video data of the user interface, the operator, and the robot as it moves through the arena. In addition we collect information about the robot's path and coverage of the arenas. We also have access to the overall performance scores including penalties occurred.

We typically tap into the video output of the operator control unit (OCU) and direct this to a scan converter which sends the converted output to a video recorder for later analysis. As the setup time for teams to get ready for their rounds is between 10 and 15 minutes, data collection setup has to be quick and flawless. Prior to the initial rounds, we test out the data collection equipment with each team who agrees to participate in our study. We make sure that all the video is time stamped so that we can easily move between the operator view of the user interface and ground truth as represented by the robot moving in the arena during analysis. It is difficult to tape the movement of the robot in the arena, as portions of the arenas are covered. Debris and multiple levels in the arenas make it difficult to see the robot at all times without being physically in the arena. We try to capture data from outside of the arena as our presence can cause the sensors on the robot to mistakenly identify us as victims or unintentionally point out possible paths through the debris. Figure 1 shows three different sections of the Robocup 2004 NIST test arenas.

Figure 1: NIST Test Arenas at the Robocup USAR 2004 Competition

Analysis of Data

We have completed analysis of two sets of data at this point in time. Our initial data analysis was completed on data collected at the 2002 USAR competition at the American Association of Artificial Intelligence [14]. We collected data from all the teams in the competitions but we coded only data from the four top ranking teams. We used the data from the semifinals and finals. We were interested in looking at how the overall performance correlated with a finer analysis of performance. We looked at the video tapes and coded the amount of time each team spent in navigation or monitoring navigation, in identifying victims, in logistics, and in failures. Table 1 contains the definitions of these terms.

Table 1: Definitions of Coded Activities

Activity Coded	Definition
Navigation or monitoring navigation	This activity was coded when operators were teleoperating a robot, or in the case of semi-autonomous robots, when the operator was issuing navigation commands and watching the user interface to assess how the robot was moving.
Victim identification	We coded this activity when the operator thought he had sensed a victim and moved closer or used other sensors to assess the status of the victim.
Logistics	Activities such as starting up another robot were coded as logistics.
Failures	Hardware, software, and communications dropouts were coded as failures.

Table 2 shows the percentage of times the four teams spent in these activities. Note that we were only able to code two of the three runs due to issues with the data collection mechanism. The total time is given in minutes. Each team was allocated 15 minutes for their runs. It was difficult to actually coordinate with the competition officials to know the actual start and stop times and, in one case, we lost some time due to a data collection issue. Note that the percentages do always add up to 100%. This is basically due to rounding areas in calculating times.

Table 2: How teams spent their time

	Run	Total Time (min)	% Time			
			Navigation/ Monitoring Navigation	Victim ID	Failure	Logistics
Team A	1	10:39	46	51	0	3
	3	14:45	62	18	19	1
Team B	1	14:33	81	19	0	0
	3	16:42	77	23	0	0
Team C	1	13:26	59	23	17	0
	3	14:39	69	12	18	0
Team D	1	15:12	55	32	0	12
	3	13:30	87	4	0	9

We found that teams using some sort of automatic mapping were more successful in navigating the arenas. Operators who had to keep maps in their heads became confused about where they were at times. We looked at the penalties incurred by the teams and found instances where the operators were unaware of the surroundings of the robots or the status of the robots. In particular, few robots in this particular competition had a view of what was behind them. In situations where the operator was forced to back up or to make a series of tight turns this resulted in penalties for bumping into walls or victims.

In our analysis of a second set of data collected at Robocup 2003, we looked at issues of awareness [9]. Burke [1] identified situation awareness (SA) as a major component needed for effective human-robot performance. Scholtz[10] has modified Endsley's SAGAT [4] methodology for measuring the SA provided by supervisory interfaces for semi-automonous driving vehicles. Scholtz also analyzed the time needed for operator acquisition of SA in two types of terrains [12, 13]. In this analysis we used a modification of awareness tailored to HRI [3].

If we consider teams consisting of humans and robots, we can define 5 types of awareness:

- Human-robot awareness
- Robot – human awareness
- Human-human awareness
- Robot – robot awareness
- Humans' overall mission awareness

In the majority of teams competing in the USAR Test Arenas, we are able to evaluate only human-robot awareness. That is, does the human have knowledge of the location, status, and behavior of the robot? We find few teams that have multiple robots with any collaboration capabilities (robot-robot awareness) or use multiple operators (human-human awareness). Moreover, the current generation of robots in these competitions has no awareness of the operators' status (robot-human awareness).

We used an indirect means of assessing human-robot awareness as we are not able to intervene and ask the operator to verbally describe any given situation. We coded critical incidents observed in the video tapes of the robot moving in the arena. Critical incidents are defined as a situation where the robot was in a position that could potentially be harmful to the robot, the environment, a victim, or the mission. Originally, we had intended to code critical incidents that were "avoided", such as when the robot was able to move through an extremely tight space without causing any damage. However, we found that we were unable to do this consistently. We were able to consistently locate and code critical incidents that had a negative outcome, e.g. the robot bumped into a wall. We classified the critical awareness incidents into one of five categories: global navigation, local navigation, victim identification, obstacle extraction, vehicle state. Obstacle encounter was coded when the robot had actually run into an obstacle and had to perform maneuvers to free itself. Vehicle state awareness was coded when the operator did not realize that the robot was in other than a normal state, e.g. tipped over. In the runs we coded, we found evidence of critical incidents only in the categories of local navigation, obstacle extraction, and vehicle state. We did see evidence of the other types of critical incidents but these were not in the actual runs selected for coding (the semifinals and finals). Table 3 shows the numbers of critical incidents occurring for the three teams analyzed. These three teams were selected for analysis as they placed in the final round of the competition.

Table 3: Analysis of Critical Incidents by Team

	Local navigation	Obstacle encounter	Vehicle state
Team A	4	6	5
Team B	1	9	2
Team C	10	11	5
Total	15	26	12

Obstacle encounters were the most prevalent types of critical incidents. Robots became entangled in loose debris in the arenas and it was difficult for the operators to know that.

In the most recent competition, Robocup 2004, we noted that teams typically had one of two sources of situation awareness information implemented. A number of teams used some sort of overhead cameras to provide a frame of reference for the robot in relationship to the environment. Other teams had implemented some sort of automatic mapping software, using a variety of sensors, including sonar and ladar. At this point we have not had time to do a full analysis, but an early analysis looks at the five teams who were in the final runs. Table 4 shows the penalties by team.

Table 4: Penalties by type of situation awareness

Penalties for teams using automatic mapping		Penalties for teams using overhead cameras	
Team A	0	Team P	80
Team D	5	Team S2	5
Team S1	40	(note Team S2 had only 3 runs completed as they had to end one run prematurely due to a problem with the robot)	

These penalties are all local navigation penalties. That is, the robot either bumped into the walls of the arena or into a victim. While these results should be viewed as very preliminary, our impression is that the automatic mapping is more helpful in providing situation awareness. This is not surprising, as the video information, while helpful, still requires considerable interpretation by the operator. Also, if there happens to be any sort of communication interference, the video is extremely difficult to view. The teams that we analyzed were the top scoring teams, which implies that they had reasonable coverage of the area and located a number of victims. Low scoring teams may have few penalties due primarily to an inability to move very far into the USAR arena.

The majority of teams we have analyzed have been teleoperated, using autonomy only for such things as mapping. While we have had some fully autonomous teams in the competitions, they have not been successful in navigating the difficult environment in the test arena. In our first analysis, two of the teams operated in semi-autonomous modes. The operators were responsible for overall navigation, but left the local navigation (obstacle avoidance, waypoint navigation) to the robots in many instances. We intend to analyze future teams to determine how critical incidents change based on the level of autonomy.

Discussion

Based on the analyses we have completed to date, we have been able to provide some guidelines for human-robot interaction design. These are summarized below.

- Information for effective situation awareness should include:
    - a frame of reference to determine the position of the robot relative to the surrounding environment
    - indicators of vehicle state, such as pitch, roll, traction indicators, indicators of sensor status, and camera positions relative to the robot body.
    - a map to provide global navigation information
- Minimize the number of windows provided to the operator.
- Provide a fused view of sensor information.
- Support multiple robot operators in a single display.
- Provide help from the robot in determining what mode of autonomy is most useful.

To date, we have been able to analyze data collected during USAR competitions to provide some guidelines for the design of effective user interfaces for USAR robots. We are encouraged that our work is making a difference as the situation awareness offered in the user interfaces deployed in current competitions is certainly increasing. The downside of our work is that the analysis takes considerable time and by and large the results are consumed by human-computer interaction researchers, not robotics researchers.

Future Plans

We are interested in providing feedback about HRI designs in a more timely fashion and to the robotics community more directly. In the final rounds of Robocup 2004, robots were placed in an internal spot in one of the arenas. The operator had to first assess where the robot was and then devise a strategy for moving out into the arena to locate victims. We are currently working on devising extensions to this, similar to the compulsories in figure skating competitions. This would help us assess during the competition how well the operator is able to gain situation awareness based on the user interface. While the NIST Reference Arenas provide a standard area in which the robots have to perform, there is no guarantee that robots encounter the same obstacles. Moreover, due to variations in size and mobility, we cannot expect robots to do equally well in navigating the same environment.

It is important for effective search and rescue that teams are in control. This means having good SA at all time about where team members, including robots, are and what they are doing. Situation awareness could be demonstrated by placing robots in specific situations (such as close to obstacles, on different types of surfaces and grades, or near negative obstacles) and measuring the time and accuracy of the SA by the

operator –robot team. We are working on a user interface design for our own robotics platform. It would be possible to consider our performance as a baseline that the teams should try to best.

References

1. Burke, J., R. Murphy, and M. Coovert. (2004). Moonlight in Miami: An Ethnographic Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise. *Human Computer Interaction.*

2. Casper, J. (2002). *Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at the World Trade Center*. MS Thesis, University of South Florida Department of Computer Science and Engineering.

3. Drury, J. L., J. Scholtz, and H. A. Yanco. Awareness in human-robot interactions. *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, Washington, DC, October 2003.

4. Endsley, M. 2000. Theoretical Underpinning of Situation Awareness: Critical Review (2000) in Mica R. Endsley and Daniel J. Garland (Eds.) *Situation Awareness Analysis and Measurement.* Lawrence Erlbaum Associates, Mahwah, New Jersey. 3-32.

5. Ericsson, K. A. and H. A. Simon (1980). "Verbal reports as data." *Psychological Review*, Vol. 87, 215 - 251.

6. Ericsson, K. A., & Simon, H. A. (1993). *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: The MIT Press.

7. Jacoff, A., E. Messina, and J. Evans (2001). "A reference test course for autonomous mobile robots." In *Proceedings of the SPIE-AeroSense Conference*, Orlando, FL, April.

8. Jacoff, A., E. Messina, and J. Evans (2000). "A standard test course for urban search and rescue robots." In *Proceedings of the Performance Metrics for Intelligent Systems Workshop*, August.

9. Scholtz, J., J. Young, J. Drury, and H. Yanco. (2004). "Evaluation of Human-Robot Interaction Awareness in Search and Rescue." ICRA 2004.

10. Scholtz, J., B. Antonishek, and J. Young. (2004). Evaluating human-robot interfaces: development of a situational awareness assessment methodology. *Hawaii International Conference on System Science 37 (HICSS 37)*. Hawaii, Jan. 2004.

11. Scholtz, J. (2003). Theory and Evaluation of Human Robot Interactions, *Hawaii International Conference on System Science 36 (HICSS 36)*. Hawaii, Jan., 2003.

12. Scholtz, J., B. Antonishek, and J. Young. (2003). Evaluation of Operator Interventions in Autonomous Off-Road Driving. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems, (PERMIS).* Gaithersburg, MD Sept. 2003.

13. Scholtz, J., B. Antonishek, and J. Young (2004). Operator Interventions in Autonomous Off-road Driving: Effects of Terrain. To appear in *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, The Hague, NL October 2004.

14. Yanco, H., J.L. Drury, and J. Scholtz. (2004). Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition.. *Human-Computer Interaction, 2004.*

Task Performance Metrics in Human-Robot Interaction: Taking a Systems Approach

Jennifer L. Burke, Robin R. Murphy, Dawn R. Riddle & Thomas Fincannon
Center for Robot-Assisted Search and Rescue
University of South Florida
Tampa, FL 33620

ABSTRACT

Performance metrics for human-robot interaction in urban search and rescue (USAR) are just beginning to appear in the literature as researchers try to establish a way of describing and evaluating human-robot task performance in this high-risk, time-critical domain. In this paper we propose that human-robot interaction metrics should focus on the work system as a whole, examining the robot's effects on human task performance within the overarching context of human work. Moreover, these effects should be examined within the context of real-time human performance in field settings, rather than in simulation or experimental environments. This position stems from a basic assumption that we are interested in measuring human-robot interaction in USAR because we want to see how it affects and aids human performance in this time and safety-critical environment. We present a methodology for collecting data in the field and subsequent analysis using the Robot-Assisted Search and Rescue Coding System (RASAR-CS), specifically developed for this domain. The RASAR-CS allows us to capture 1)basic verbal and non verbal communications describing the task and how it is accomplished (what is being said, by who to whom); 2)situation awareness information requirements (from the robot and other sources) - for developing and maintaining situation awareness, including the ability to capture changing requirements over time; 3)team processes enabling coordinated activities, efficient communication and strategy planning; and 4) human-robot interaction in terms of: robot-operator initiated robot activities, and physical interaction with robot.

KEYWORDS: *human-robot interaction, performance metrics, field methodologies*

1. INTRODUCTION

Human-robot interaction in the Urban Search and Rescue (USAR) domain is a field of study that has drawn increasing interest in light of the use of robots at the World Trade Center [5] and its designation as a benchmark domain in the seminal DARPA-NSF study on Human-Robot Interaction conducted in 2001 [2]. Performance metrics for human-robot interaction in USAR are just beginning to appear in the literature as researchers try to establish a way of describing and evaluating human-robot task performance in this high-risk, time-critical domain. In the afore-mentioned DARPA-NSF study, simple base measures were proposed: the ratio of persons to robots (h-r ratio), spatial relationships (commander, peer, teleoperator, developer) and authority relationships (supervisor, operator, peer and bystander). Some of the metrics proposed subsequently focus on aspects of the robot system exclusively, (e.g., the interface) or on aspects of human performance solely in relation to working with the robot [8, 9, 14]. In this paper we take a more human-centric position: human-robot interaction metrics should focus on the work system as a whole, examining the robot's effects on human task performance within the overarching context of human work. This position stems from a basic assumption that we are interested in measuring human-robot interaction in USAR because we want to see how it affects and aids human performance (ultimately, that is the goal for measuring human-robot interaction in any work-related field or application).

1.1 *Field Studies in USAR*

Field studies conducted with rescue workers offer the most valid setting in which to study human-robot interaction. USAR is an established work environment offering opportunities to study the effects of introducing robotic technology into a workplace and occupation with existing goals, tasks and processes. It is arguably one of the first workplace applications where robots work in the same spaces with people whose jobs do not normally involve robotics to perform a task (Industrial robots are usually separated from humans, and are not mobile). Moreover, robots have been used in real disaster responses, and are gradually becoming incorporated into USAR training both nationally and internationally. Real-time high fidelity training exercises are conducted regularly in order for USAR task force members to attain or maintain certification; these exercises offer a double advantage for studying HRI in that the targeted end-users may be observed performing in realistic work environments. USAR task forces can be characterized as extreme teams [11] who function in dynamic, high risk, time critical environments. Team members must function in conditions which are often physically, mentally and emotionally taxing. Field studies with participants who are truly representative of this user group for whom the technology is being optimized offer the most power in terms of generalizability.

1.2 *Focusing on Human Performance*

Measures of human-robot interaction in USAR must focus on human performance. The current state of the practice in robot-assisted search and rescue is teleoperation. Though autonomous and semi-autonomous robots may soon be entering the workplace, they will still be machines designed to perform tasks as determined by a person. Robots are not conscious, they have no projects of their own other than those assigned to them. Clancey [7] points this out to illustrate that it's too soon to talk about human-robot cooperation or collaboration: instead, robots serve as assistants to people working toward a project goal. Therefore the measure of a robot's usefulness, efficiency and functionality is based solely on whether it contributes to helping a person (or team) accomplish a goal by making that person's or team's task performance more efficient, effective, or easy in some way. This means measuring human performance (aided by robots) is the key. This is different from the position taken in Drury et al. [9] that usability requirements, which focus primarily on the robotic system, are the most appropriate way to measure human-robot interaction. We believe human-robot systems must be examined and measured in terms of their effect on human performance, since that is what they are designed to augment or improve.

What are the criteria for measures of robot-assisted human performance in USAR? In this domain there are established goals: search, rescue (extrication), structural evaluation, medical assessment & treatment, information transfer, command & control, and logistics. Blitch [1] pointed out the potential applications of robots in tunnel and confined space search: now, it is evident that there are many more tasks in which robots may play a part in USAR, e.g. medical reachback, shoring, communications & information transfer, and safety monitoring. Past experience shows that new technologies evolve when they reach the workplace, and many times end up performing tasks or serving purposes for which they were not originally intended. What we can do is identify tasks as they emerge, study the human-robot interaction processes and determine optimal task allocation and roles, understanding that this is an iterative process that will change as the technology advances. Based on these tasks, we can measure human-robot performance both individually (one person operating a robot) and in teams (more than one person operating a robot or robots).

Our field research has shown that situation awareness and team processes are two constructs which relate to human performance when working with robots [3, 5, 6]. Situation awareness (SA) as defined by Endsley [10] is "…the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" (p. 97). Our studies have shown situation awareness to be related to performance and that most of the operators' time is spent gathering/maintaining situation awareness [3, 4]. Operators with high situation awareness ratings were better performers in our study of 28 robot operators [4]. Team processes are also related to operator performance; operators who talked more with their teammates about goal-directed aspects of the task had higher situation awareness ratings and found the victim more often. There is an interactive affect between situation awareness and team process—suggesting operators who talk more with their tether manager or teammate are better at building a mental model of the robot as it functions in the void space, and also are better at building a shared mental model of the search. Research on teams and mental models has suggested that having a shared mental model of the problem space can increase situation awareness and team performance [11, 16]. Effective planning and communication strategies were found to increase team shared mental models and correspondingly team performance. Therefore, human-robot interaction in USAR needs to be measured not only at the individual level, but also at the team level.

2. METHODOLOGY

In this section we present our methods of field data collection and data analysis, including a description of the Robot-Assisted Search and Rescue Coding System (RASAR-CS).

2.1 *Data Collection in the Field*

Data collection is an observational procedure, where the researcher is present during the user-robot interaction, though not an active participant. We tape the interaction using 2-4 cameras depending on the environment (Table 1). Minimally, one camera records the robot's eye view directly from the operator control unit (OCU), and a second camera records the operator (making sure to have a clear view of the face) as she works with the robot. When environmental conditions permit, we set up a third camera on a tripod to record the operator's hands manipulating the OCU, and a fourth camera to record an external view of the robot when available. Depending on the environment and the number of personnel available for the data collection process, some cameras may be fixed on tripods; however, in USAR conditions, most of the time views 2-4 must be handheld due to lack of level spots for setting up a fixed camera. Video recordings of the operators manipulating the robot, the robot's eye view, and the available external views are edited and synchronized to create tapes with 2 views side-by-side. These videotapes are then used to code statements and gestures made by both the operators and surrounding personnel, and robot movements. Trained raters code the videotapes using the Noldus Observer Video-Pro [13] observational coding software.

Camera No.	View	Setup
1	Robot's eye view	Attached to OCU
2	Operator view	Tripod or handheld
3	Operator-OCU view	Tripod or handheld
4	Robot-external view	Tripod or handheld

Table 1. Camera views for human-robot interaction field research in USAR.

2.2 *Video-based Interaction Analysis*

What, then, are some appropriate measures and metrics for USAR human-robot interaction? Primary human performance outcome measures for search at the most basic level include: Was a victim found, how long did it take, were any victims missed, were important cues noticed (heat, color, objects, information synthesis with knowledge about the event & environment)? Other measures that are related to these primary outcomes specifically measure situation awareness and team processes. These measures are gathered via a video-based Interaction Analysis technique for investigating HRI in rescue robotics. Interaction Analysis (IA) is an interdisciplinary approach to studying the interaction of humans with each other and with objects in their environment. Jordan & Henderson [12] assert, "Video-based Interaction Analysis is a powerful tool in the investigation of human activity that is particularly effective in complex, multi-actor, technology-mediated work settings … It is currently undergoing a period of rapid development, driven, in part, by researchers' dissatisfaction with conventional methods, and in part by the ubiquity of video equipment." (p.44)

The goal of Interaction Analysis is to identify regularities in the ways in which participants utilize the resources of the complex social and material world of actors and objects within which they operate. To do this we must examine two components of IA, which are intertwined, but distinct as well: human-human interaction, and human-object interaction. Interaction Analysis assumes that knowledge and action are fundamentally social in origin, organization, and use. Knowledge is seen as located in the interactions between people engaged with the material objects in their surroundings; therefore communication analysis plays an important role in Interaction Analysis as a means of analyzing human-human interactions.

Although variety of approaches to examining communication, we chose the FAA's Controller-to-Controller Communication and Coordination Taxonomy (C^4T) [15] framework as the starting point for the development of our communication analysis system designed to assess HRI in rescue robotics. The C^4T uses verbal information to assess team member interaction from communication exchanges in an air traffic control environment. We used the C^4T model because it captures the "how" and "what" of team communication by coding form, content and mode of communication. Our goal, however, is two-fold, not only to capture the "how" and "what" of USAR robot operator teams, but also the "who", and to capture observable indicators of robot operator situation awareness. In addition, in order to adhere to the tenets of IA, the framework must be extended to include examination of physical interactions with the robot system(s) in the environment.

2.3 *Robot-Assisted Search and Rescue Coding System (RASAR-CS)*

A methodology to capture and assess robot assisted task performance in rescue robotics must consider both human team member interactions (robot operator and other team members), and human – robot interactions. To meet the goals of a methodology capable of defining robot assisted tasks, and examining SA and teamwork defined earlier, we developed the Robot-Assisted Search and Rescue Coding System (RASAR-CS). The RASAR-CS captures *Basic verbal and non verbal communications* describing the task and how it is accomplished (what is being said, by who to whom); *Situation Awareness Information requirements* (from the robot and other sources) - for developing and maintaining situation awareness, including the ability to capture changing requirements over time; *Team processes* enabling coordinated activities, efficient communication and strategy planning; and Human-Robot interaction in terms of: *Robot-operator initiated robot activities, and Physical interaction with robot.*

Following the Interaction Analysis approach, the RASAR-CS consists of four main coding components enabling analysis of SA and team factors through human-human interaction and human robot interaction. These components include verbal communication, communication medium, nonverbal interaction and robot movements.

2.3.1 Human-Human Verbal Communication

The verbal communication analysis codes team member statements across four categories: 1) Speaker-recipient dyad - who is speaking to whom, 2) Content or topic of the communication 3) Statement form or grammatical structure of the communication, 4) Function or intent of the communication (Table 2). By examining dyad, content, and form, we can examine task procedures and team coordination. Similarly, content and function provide indicators of operator situation awareness.

Speaker-recipient dyad. Based on review of the search task videotapes, potential conversants included the operator, tether manager, team member, the group, and the robot specialist/researcher. Dyad codes indicate the speaker, followed by the recipient. For example, "operator-tether manager" indicates a statement was made by the operator

Category	Subcategories	Definitions
Human-Human Verbal Communication		
Sender/Recipient Dyad	Operator-tether manager	Operator: individual teleoperating the robot
	Tether manager-operator	Tether manager: individual manipulating the tether and assisting operator with robot
	Team member-operator	Team member: one other than the tether manager who is assisting the operator (usually interpreting)
	Operator- team member	
	Researcher-operator	Researcher: individual acting as scientist or robot specialist
	Operator-researcher	
	Other-operator	Other: individual (not tether manager, team member, or researcher) interacting with the operator
	Operator-other	
	Operator-group	Group -set of individuals interacting with the operator
Content	Robot state	Robot functions, parts, errors, capabilities, etc.
	Environment	Characteristics, conditions or events in the search environment
	Information synthesis	Connections between current observation and prior observations or knowledge
	Robot situatedness	Robot's location and spatial orientation in the environment; position
	Victim	Pertaining to a victim or possible victim
	Navigation	Direction of movement or route
	Search strategy	Search task plans, procedures or decisions
	Off task	Unrelated or extraneous subject
Statement Form	Question	Request for information
	Instruction	Direction for task performance
	Comment	General statement, initiated or responsive, that is not a question, instruction or answer
	Answer	Response to a question or an instruction
Function	Non-operator	Default for statements made by individuals other than the operator
	Seek information	Asking for information from someone
	Report	Sharing observations about the robot, environment, or victim
	Clarify	Making a previous statement or observation more precise
	Confirm	Affirming a previous statement or observation
	Convey uncertainty	Expressing doubt, disorientation, or loss of confidence in a state or observation
	Plan	Projecting future goals or steps to goals
	Provide information	Sharing information (other than *report*) in response to a question or offering unsolicited information
Team Communication	Coordination	Team members coordinate actions to synchronize specific proximal task activities
	Planning	Planned strategies for future goal accomplishment
Source of Information used in discussion	Audio	Verbal information or information from previous dialog
	Visual image	Robot image or information from image provides the basis for statement
	Sensor	Sensor or information from sensor provides the basis for statement
Human-Robot Interaction (Nonverbal interaction via the robot)		
Physical orientation	Ear to robot	Ear is directed toward the robot
	Eye to robot	Turning so that the human looks at the robot
	No verbal communication	No verbal communication with the operator
Gestures	Come forward	Motioning toward the robot to move forward
	Thumbs up	Closing the fist with the thumb extended upward
	Stop	Holding up a hand with the palm toward the r
	Pointing	Using fingers to point in a particular direction or at a specific object
	"OK" sign	Closing the thumb and forefinger in a circle indicating the "OK" sign
	Other	Other gestures (usually conversational with no intended message)
Interaction with Robot	Clean lens	Cleaning the robot camera lens
	Move/shift	Altering the position of the robot
	Pick up	Lifting the robot off the surface upon which it is moving
	Other	Other physical contact with the robot
Robot Movement	Moving	Forward or backward locomotion
	Stationary	No movement at all
	Panning	Rotating side to side without forward movement, or manipulating the camera lens up/down

Table 2. RASAR-CS (for USAR search task)

and directed toward the tether manager (Note: the code "tether manager – operator" indicates the tether manager initiated the communication with the operator).

Content. Seven elements representing the content were generated: 1- Statements related to robot functions, parts, errors, or capabilities (Robot state), 2- Statements describing characteristics, conditions or events in the search environment (Environment), 3- Statements reflecting associations between current observations and prior observations or knowledge (Information synthesis), 4- Statements surrounding the robot's location, spatial orientation in the environment, or position (Robot situatedness), 5- Indicators of direction of movement or route, (Navigation), 6- Statements reflecting search task plans, procedures or decisions (Search Strategy), and finally 7- Statements unrelated to the task (Off Task). The first four content elements are relevant to building and maintaining SA in search operations, while the elements of navigation and search strategy require SA.

Form. Similar to the C^4T taxonomy, the form category contains the elements: 1- Question (request for information), 2- Instruction (direction for task performance), 3- Comment (general statement, initiated or responsive, that is not a question, instruction or answer) and 4- Answer (response to a question or an instruction).

Function. Function refers to the intent of the communication - elements include: 1- Seek information (asking for information from someone), 2- Report (sharing observations about the robot or environment), 3- Clarify (making a previous statement or observation more precise), 4- Confirm (affirming a previous statement or observation) 5- Convey uncertainty (expressing doubt, disorientation, or loss of confidence in a state or observation), 6- Plan (projecting future goals or steps to goals), 7- Provide information (sharing information other than that described in report, either in response to a question, or offering unsolicited information).

The function elements of reporting and providing information merit explanation, as they appear very similar. Reporting involves perception and comprehension of the robot's state or situatedness, the environment or information synthesis. Any other information shared by an operator, in answer to a question or on his own, is classified as providing information (for example search strategy or navigation). Indicators of SA are captured in the function category primarily through the elements reporting and planning. When operator shares information (reports) based on the robot's eye view, we can infer the first two levels of SA, perception and comprehension, have taken place. The third SA level, planning and projection, is captured in the function category as the element "plan."

2.3.2 Team

Communication. Team communication offers insights into how goals are accomplished. Categories include: 1- Coordinating activities (to synchronize specific proximal task activities) and 2- Planning (for future goal accomplishment).

Medium. Team communication is also coded according the medium used to in conveying information: 1- Visual (visual image provided the foundation for the communication), 2- Auditory (verbal information provided the foundation for the communication), and 3- Sensor (sensor provided the foundation for the communication).

2.3.3 Human – Robot Interaction

Nonverbal interaction with robot. Nonverbal HRI includes nonverbal communication between humans via the robot camera, and physical interaction of humans with the robot. When robots are co-located with humans, humans physically orient to the robot and use gestures when communicating with the operator in control of the robot. Additionally, humans have the ability to physically touch or interact with the robot to cooperatively accomplish goals. The three main nonverbal categories include: physical orientation, gestures, and physical interaction with co-located robot.

Physical orientation. Physical orientation includes positioning the body during communication with the robot operator so that the 1- Ear is directed toward the robot (ear to robot), and 2- Turning so that the human looks at the robot (eye to robot).

Gestures. Again, while communicating with the robot operator, gesture can be used to convey meaning to the operator via the robot camera. Gestures include: 1- Come forward (motioning toward the robot to move forward), 2- Pointing (using fingers to point in a particular direction or at a specific object), 3- Thumbs up (closing the fist with the thumb extended upward), 4- Stop (holding up a hand with the palm toward the robot), and 5- OK (closing the thumb and forefinger in a circle indicating the "OK" sign).

Physical Interaction with Robot. Physical interaction codes include: 1- Clean lens (cleaning the robot camera lens), 2- Move/shift (altering the position of the robot), and 3- Pick up (lifting the robot off the surface upon which it is moving).

Robot Movement. The three major robot movement coding categories of the RASAR-CS include: 1- Moving (traveling forward or back), 2- Stationary (no movement at all) and 3- Panning (turning from side to side without forward or backward movement).

3. CONCLUSIONS

We have presented a field methodology for examining human-robot interaction in USAR which focuses on robot-assisted human performance. Using a video-based Interaction Analysis technique, we examine both human-

human interaction and human-robot interaction with measures designed to capture performance of human-robot systems. The Robot-Assisted Search and Rescue Coding Scheme enables us to

- *Examine archival videotaped data*. Video data involving users provides a richness of information that we previously had no established means of harvesting.
- *Decompose novel robot assisted tasks.* Understanding how USAR personnel use robots to accomplish tasks provides the foundation for developing a model of robot assisted task performance, which can be used for defining best practices and generating field training.
- *Identify task specific SA requirements* and effective *modalities for information transfer* among team members for use in system design (e.g., operator control unit interfaces, and web pages for remote team members).
- *Evaluate requirements for team performance* such as shared mental models, coordination of activities, and patterns of cooperative behavior.
- Obtain *quantifiable SA and team data* for evaluating effective performance.
- *Adapt and respond to changing task and technology requirements*. The RASAR-CS can be reconfigured to meet needs of various tasks and to be responsive to changes in technology as advances in robotics occur.

The RASAR-CS allows researchers to decompose both human-robot and human-human interaction in a meaningful way to define robot assisted task performance including task procedures, situation awareness requirements, and team process and coordination. The system can be applied across tasks and domains by utilizing the procedures outlined for modifying the relevant codes. In assessing complex environments it is important to use multiple methods of assessment. The RASAR-CS is an effective methodology to add to researchers' HRI toolkit for analysis of archival videotapes of field data, or used as a complement to other techniques, e.g. onsite expert ratings of situation awareness and team process, self ratings of situation awareness and team process, and user ratings of traditional evaluative components (usefulness, ease of use, effectiveness, satisfaction) for using the robot.

4. References

[1] Blitch, J. G. (2002). Robot intelligence for tunneling and confined space search and rescue. In *Performance Metrics for Intelligent Systems Workshop*, 2002.

[2] Burke, J., Murphy, R.R., Rogers, E., Scholtz, J., and Lumelsky, V. (2004). Final report for the DARPA/NSF interdisciplinary study on human-robot interaction. *IEEE Systems, Man and Cybernetics Part C, 34*(2), 103-112.

[3] Burke, J., Murphy, R.R., Coovert, M., Riddle, D. (2004). Moonlight in Miami: An ethno-graphic study of human-robot interaction in USAR. *Human-Computer Interaction, special issue on Human-Robot Interaction, Volume 19, Nos. 1-2,* 85-116.

[4] Burke, J. & Murphy, R.R. (2004). Report on the 2002 Connecticut Field Exercise. Center for Robot-Assisted Search and Rescue Technical Report.

[5] Casper, J. & Murphy, R. (2002). Workflow study on human-robot interaction in USAR. *Proceedings of the 2002 International Conference on Robotics and Automation* (pp. 1997-2003). Piscataway, NJ: IEEE Press.

[6] Casper, J. & Murphy, R. (2003). Human-robot interactions during the robot-assisted search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 33*(3), 367 -385.

[7] Clancey, W.J. (2004). Roles for agent assistants in field science: Understanding personal projects and collaboration. *IEEE Systems, Man and Cybernetics Part C, 34*(2), 125-137.

[8] Drury, J., Riek, L., Christiansen, A., Eyler-Walker, Z., Maggi, A., Smith, D., (2003). Evaluating human-robot interaction in a search-and-rescue context. In *Performance Metrics for Intelligent Systems Workshop*, 2003.

[9] Drury, J.L., Scholtz, J. & Yanco, H.A. (2003). Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction, special issue on Human-Robot Interaction, Volume 19, Nos. 1-2.*

[10] Endsley, M. (1988). Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society 32nd Annual Meeting, 1,* (pp. 97-101). Santa Monica, CA: Human Factors Society.

[11] Jones, H. & Hinds, P. (2002). Extreme work groups: Using SWAT teams as a model for coordinating distributed robots. *Proceedings of the CSCW 2002 Conference on Computer Supported Cooperative Work* (pp.372-381). New York: ACM.

[12] Jordan, B. & Henderson, A. (1995). Interaction analysis: Foundations and practice. Journal of the Learning Sciences, 4(1), 39-103.

[13] Noldus, L., Trienes, R., Hendriksen, A., Jansen, H. & Jansen, R. (2000). The Observer Video-Pro: new software for the collection, management, and presentation of time-structured data from videotapes and digital media files. *Behavior Research Methods, Instruments & Computers, 32*, 197-206.

[14] Olsen, D.R. & Goodrich, M.A. (2003). Metrics for evaluating human-robot interactions. In *Performance Metrics for Intelligent Systems Workshop*, 2003.

[15] Peterson, L., Bailey, L., & Willems, B. (2001). *Controller-to-controller communication and coordination taxonomy (C^4T).* (DOT/FAA/AM-01/19). Department of Transportation, Federal Aviation Administration, Office of Aerospace Medicine, Washington, D.C.

[16] Prince, C. & Salas. E. (2000). Team situation awareness, errors, and crew resource management: Research integration for training guidance. In M. Endsley & D. Garland (Eds.), *Situation Awareness Analysis and Measurement*, (pp. 325-347). Mahwah, NJ: Erlbaum.

Reduction of User Interaction by Autonomy

Arin Morfopoulos, Michael McHenry, and Larry Matthies

Jet Propulsion Laboratory
4800 Oak Grove Drive, Pasadena, CA 91109
{arin.morfopoulos, michael.mchenry, larry.matthies}@jpl.nasa.gov

Abstract

This paper describes experiments that quantify the improvement that autonomous behaviors enable in the amount of user interaction required to navigate a robot in urban environments. Many papers have discussed various ways to measure the absolute level of autonomy of a system; we measured the relative improvement of autonomous behaviors over teleoperation across multiple traverses of the same course. We performed four runs each on an "easy" course and a "hard" course, where half the runs were teleoperated and half used more autonomous behaviors. Statistics show 40-70% reductions in the amount of time the user interacts with the control station; however, with the behaviors tested, user attention remained on the control station even when he was not interacting. Reducing the need for attention will require better obstacle detection and avoidance and better absolute position estimation.

Keywords: User interaction, attention, autonomy, robot.

1. Introduction

The usefulness of mobile robots is a strong inverse function of the amount of user interaction required to control them; hence, the value of autonomous behaviors is partly a function of how much they reduce the amount of user interaction required. Thus, quantifying user interaction as a function of the available robot control modes is necessary for measuring progress. Quantitative user interaction experiments can also reveal where existing robot behaviors have trouble dealing with the environment, and therefore are useful for prioritizing further development.

In this paper, we describe results of user interaction experiments with a Packbot robot equipped with stereo cameras, a single axis scanning LADAR, and a variety of behaviors ranging from teleoperation to waypoint following with obstacle avoidance. We counted user button clicks and mouse drags in a series of trials employing different behaviors over two courses, an "easy" course and a "hard" course. Two trials on each course used just teleoperation, two used more autonomous behaviors. We did not measure the amount of time the user was watching the control station (ie. user attention) when he was not physically interacting with it. This is an important additional step needed in future work. As we will show, however, current autonomous capabilities still require considerable user attention even though user interaction *per se* may be small, because robots are likely to get in trouble if the user is not watching an image stream from the robot.

The rest of this paper is organized as follows. First, section 2 reviews related prior work. Then we describe our experimental setup and how we collected the data (Section 3). The actual data collection follows this with all the attendant graphs and tables (Section 4). We use these results to highlight key areas where more development is needed to reduce both user interaction and user attention. The most significant areas are position estimation and path planning capabilities that enable autonomous traverses beyond line of sight from the robot (Section 5) and obstacle detection and avoidance capabilities that can cope with negative obstacles and moving objects, such as cars and people (Section 6).

2. Related Work

Goodrich, Crandall et al [1,6,7,8] have done a series of user interaction and attention studies. Their data lies chiefly in robotic simulation, which allowed them to force the user's attention to another task and to quantify the effect of user inattention upon performance. They call this "neglect tolerance". We could not afford to quantify this, because real robots currently face too much risk of damage from unseen obstacles and moving traffic when the user is not attending to downlinked imagery. Goodrich, Crandall et al have also defined a general model of teleoperation, waypoints, and scripted waypoints in terms of interaction vs. performance. This is a theoretical model that closely describes the behaviors we use. Teleoperation requires constant interaction whereas waypoints require more work to initiate, but once started can be left alone for a much longer period.

Tejada [3] discussed a model for a 3-D graphical user interface to use with urban search and rescue teams. Our experiment did not quantify the effect of different Operator Control Units, since our focus was on the effect of different behaviors in the robot.

Frost [4] discussed the difficulties of pure teleoperation, but only as general observations. No measured times to complete a course or accomplish an objective are given. This paper is particularly relevant because it describes the same robotic platform with which we conducted our tests. The autonomy used was entirely different, but the chassis and thus mechanical ability was the same.

Bruemmer [5] used a robotic system to work in a nuclear disposal facility. Their work described the difficulties of a teleoperated system using only visual feedback for control information. They were forced to place cameras inside each room to have enough data to allow safe navigation because the view from the robot's cameras was not sufficient. As a result of these difficulties they devised an autonomous system to assist in control and to prevent the user from endangering the robot or the environment. This is an excellent example of where it would be valuable to quantify the benefit of new behaviors. They tried teleoperation, found it lacking, created autonomy to assist, but have not yet measured the improvement enabled by that autonomy.

3. Experiment Setup

The robotic platform in our experiments ("Urbie") was an iRobot Packbot chassis with an electronic payload developed by JPL under the Tactical Mobile Robotics (TMR) and Mobile Autonomous Robot Software (MARS) programs sponsored by DARPA. Urbie's sensors included a black and white stereo camera pair, a SICK LADAR, 3-axis accels and gyros, a magnetometer with pan/roll/tilt axes, and track encoder feedback. Robot state was maintained by a Kalman filter which estimated the orientation (roll/pitch/yaw) of the robot. The KF orientation estimate was combined with the track encoder data to produce an estimate of robot x/y position via dead reckoning.

Behaviors used included visually designated waypoint following, stairclimbing, and teleoperation. An Obstacle Detection and Avoidance (ODOA) module could be dynamically incorporated via an arbiter system to make any of the above behaviors become Safeguarded. To use "visually designated waypoint following" a user selected a heading from the robot's field of view and defined a desired distance; the robot would then attempt to hew to that line as closely as possible (while avoiding obstacles should ODOA be enabled). Stairclimbing allowed autonomous traverses of single flights of stairs.

A single experienced user controlled the robot over each run. The user had feedback from one of the stereo cameras allowing a keyhole view of the environment from the robot's perspective. Location of the robot's current position relative to the starting position was recorded but not viewed during the run.

Two courses were used: one fairly easy run taking place entirely on road surfaces and one comparatively hard run which involved driving on grass, sidewalks, and stairs as well as roads. Data was collected for four runs over each of the two courses. For each course two of the four runs used only teleoperation and the remaining two used autonomous behaviors. The hard course consisted of positive and negative obstacles, two flights of stairs and narrow pathways between positive and negative obstacles. The easy course was twice the distance but involved only paved roads with few obstacles. The user was familiar with each course and had walked them beforehand. Each course was designed as a navigational course rather than an exploration effort, in the sense that the user knew where he wanted to go and roughly what obstacles he would face.

During each run all mouse clicks and drags were recorded and time stamped along with position. This allowed us to relate user interaction with the robot's location.

4. Data and results

For each course we will show an overhead photo as well as graphs of a teleoperation run and an autonomous run. The graphs relate user interaction to the location at which that interaction occurred. The path of the robot is drawn in blue, and any user interactions are denoted by an 'X' superimposed on the path where the interaction took place. The robot's location was recorded for every 20 cm traveled, and the size of the 'X' reflects the number of seconds the operator spent interacting with the control unit during that 20 cm segment. Specifically, marker size is proportional to the cube root of the interaction seconds per segment. Thus a point drawn 3 times the size of another point represents 9 times as much interaction.

The space between X's indicates no user interaction. This indicates distance traveled using waypoints or stairclimbing, where the robot was entirely autonomous.

After discussing each course we will show combined statistics gathered over all 8 runs.

4.1 Easy Course

Fig. 1 is an overhead image of the Easy Course with the robot path drawn in red. Note that the entire path is along paved roads.

Fig 1. Overhead image of the Easy Course

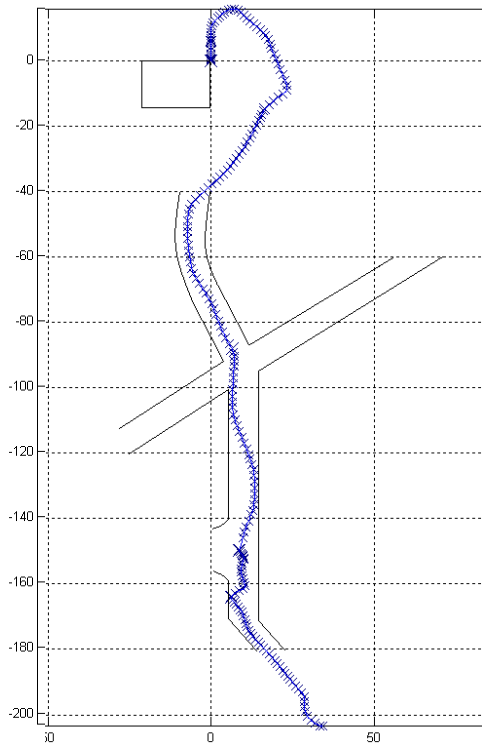


Fig 2.Teleoperation over easy course

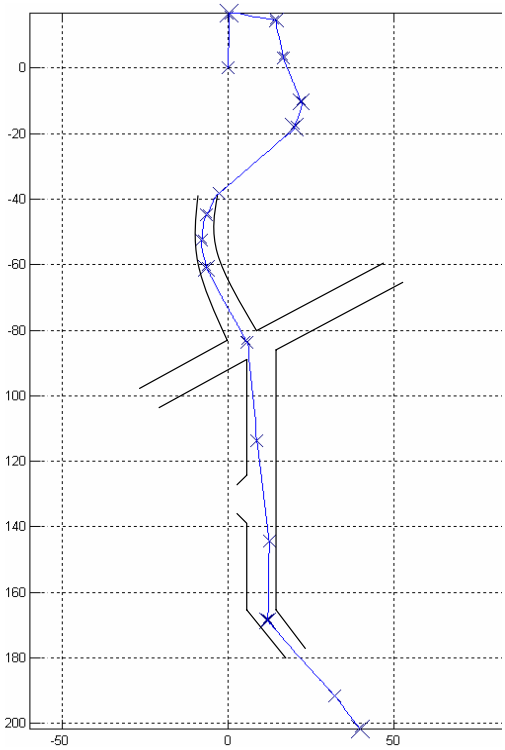


Fig 3. Waypoint over easy course

Fig. 2 and Fig. 3 show the graphs of user interaction as a function of position. Streets are overlain as hand-drawn lines.

During the teleoperation run the user was interacting for the entire run, constantly commanding a direction. During the waypoint run the user could interact for brief intervals to set up a new waypoint and then wait.

Each waypoint was set to 30 meters and could only be selected with the current image from the robot. This forced the user to intervene in the curved portion of the first half of the path to change directions, whereas in the second half the user could wait for the 30 meters to be reached before needing to interact again.

4.2 Hard Course

Fig. 4 displays the overhead image of the Hard Course with the chief obstacles labeled. Unlike the Easy Course, the path is over grass, sidewalks, streets, and up two flights of stairs. Negative obstacles forced the user to pay close attention during teleoperation and waypoints. Cars on the street required special concern and attention. Several situations were too complex for the autonomy on board and the user had to manually switch to teleoperation and guide the robot over the difficult sections.

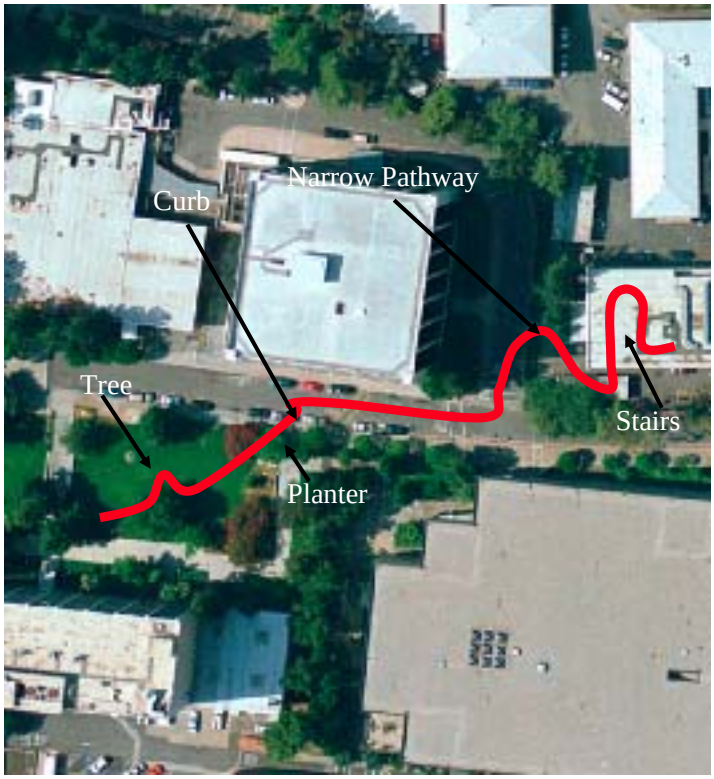


Fig 4. Overhead image of the Hard Course

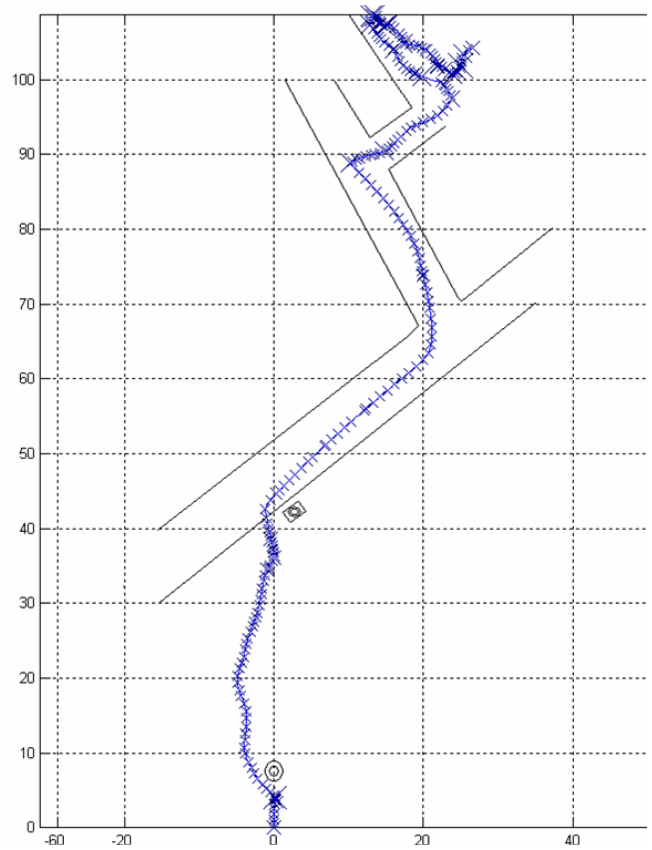


Fig 5. Teleoperation over Hard Course.

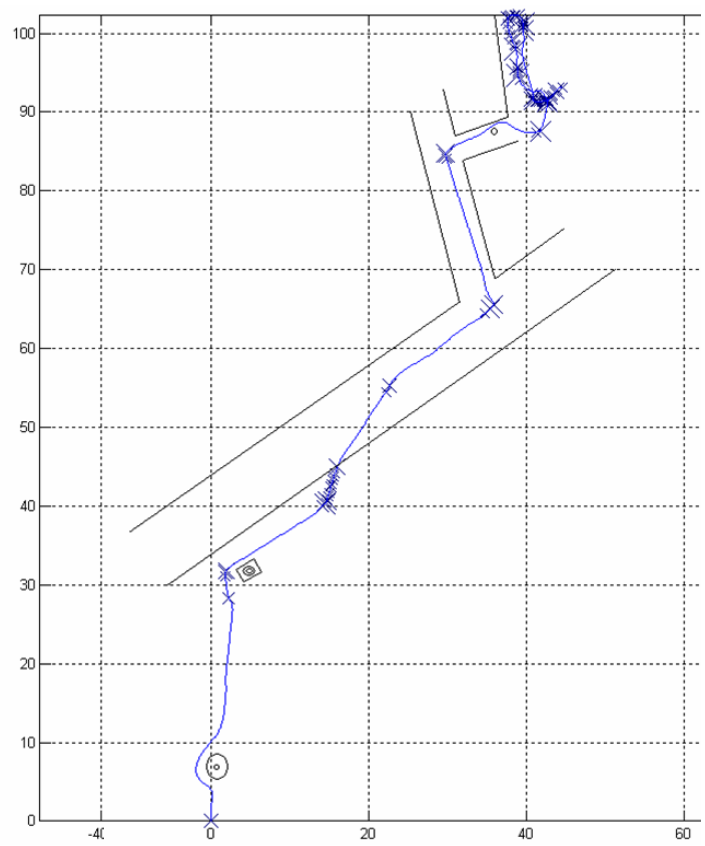


Fig 6. Autonomous and Teleoperation behaviors

Obstacles along the path (in the order they appear) are a

- Tree
- Negative obstacle (planter on the right shown as a small square).
- Negative obstacle requiring traversal (curb).
- Moving obstacles (cars)
- Narrow pathway with a pole in the middle
- Stairs

Fig 5. shows a sample run with only teleoperation. Notice that the points of interaction become denser in the difficult areas near the curb, stairs and narrow pathways. Hand drawn on the image are two obstacles (the tree and planter) as well as the street edges.

Fig 6. illustrates the mix of autonomous behaviors and teleoperation used for the Hard Course.
A few areas should be emphasized:

- The tree was avoided autonomously using ODOA.
- The planter was avoided by manually aborting a waypoint and selecting a new waypoint. The ODOA capability on Urbie would not have avoided this negative obstacle.
- The robot could not go down the curb because the path was blocked by a car (not shown in the figure); we followed the curb, bounded by negative obstacles on either side.
- Once past the car, the user was forced to teleop down the curb, as waypoint mode drives too fast to be safe.
- The user could not do a line of sight waypoint directly to the base of the stairwell; instead he used 4 waypoints, one after the other.
- The robot avoided the pole autonomously (shown as small circle in the middle of the narrow pathway).

Recall that there was no interaction on any path segment where no X's appear. Any smooth, unblemished line was traveled autonomously.

In the Fig. 6 the stairwell took so much interaction over such a small distance that it appears no autonomy was used. In fact, waypoints were used to the base of the stairwell and autonomous stairclimbing was used up both flights of stairs. The landings required manual teleoperation to maneuver in such tight quarters. Fig 7. illustrates this with a closer view (the robot enters from the bottom of the graph, finishes the run to the right).

	Easy Course Metrics				Hard Course Metrics			
	Teleop		Waypoint		Teleop		Waypoint	
Length (meters)	275	272	270	274	169	159	155	154
Time of Run (seconds)	264	380	355	370	423	459	448	488
Average Speed (m/sec)	**1.24**	**0.72**	**0.76**	**0.74**	**0.40**	**0.35**	**0.35**	**0.32**
Percent of time user interacting	**99%**	**92%**	**30%**	**26%**	**81%**	**79%**	**51%**	**45%**
User Interaction per meter (sec/m)	**0.79**	**1.29**	**0.40**	**0.35**	**2.04**	**2.27**	**1.46**	**1.44**
Percent of Distance Traveled Autonomously	**0**	**0**	**99%**	**99%**	**0**	**0**	**80%**	**78%**
Ave time of Autonomous Drive (sec)	**0.00**	**0.00**	**14.65**	**18.93**	**0.00**	**0.00**	**9.55**	**13.11**

Table 1. Interaction data over all 8 runs

A quick aside is necessary to explain Fig 7 properly. Only the first half of the first flight of stairs was traveled autonomously because a power fluctuation caused by the extreme driving conditions caused the communication link to be dropped. The user did not resume the stairclimbing behavior but simply teleoperated to the next flight.

A second oddity is that the second flight of stairs appears longer than the first, which is a result of extreme track slippage. Recall that the robot's orientation estimate was combined with the track encoder data (which is unreliable on stairs) to produce an estimate of robot x/y position via dead reckoning.

At the top of the second flight of stairs the robot traveled through a doorway inside the building using teleoperation.

4.3 Quantitative Measures

Table 1 the sum of all quantitative measurements over the 8 runs (with 4 runs per course). Attention should be directed to five areas in particular.

1. Percent of time user interacting is not 100% during teleop runs because the user had to pause occasionally to observe the surroundings, during which time he is not actually issuing any commands to the robot.
2. During the first teleop run on the Easy Course the user drove very aggressively and a little unsafely. The speed during this run is higher than the other 3 over the Easy Course. The user also never stopped to get his bearings during that run.
3. The user was forced to use teleoperation over the Hard Course during a few difficult segments. The Percent of Distance Traveled Autonomously was below 100% during those runs as a result.
4. The average time of autonomous segments on the Easy Course was longer than on the Hard Course because the path was generally straighter on the Easy Course.
5. The time to complete a course did not improve as a result of autonomy, even though the speed during autonomy (not shown) was higher than during teleoperation.

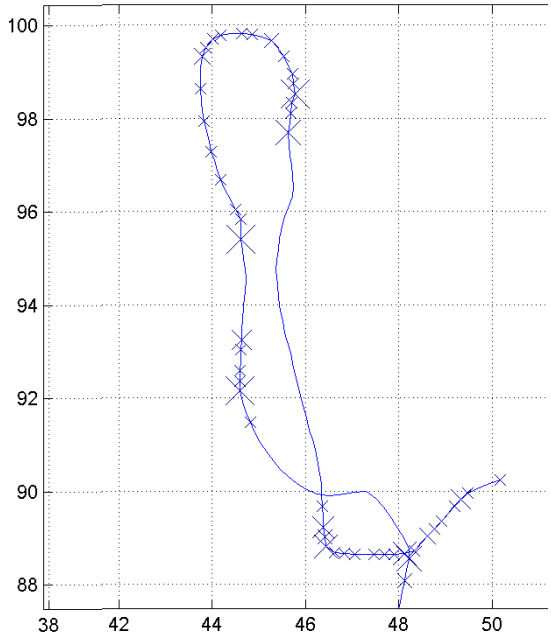


Fig 7. Close-up of the stairwell.

5. Autonomous Traverses Beyond Line of Sight

One interesting result is that autonomy did not reduce the overall time of the run even though the velocity during the autonomous waypoints was usually higher than a human driver would have been comfortable with in teleoperation. The setup time and piece by piece selection of waypoints took up the remaining time; each waypoint could only be selected from the robot's current field of view, which only allows short traverses before being occluded by obstacles.

The overall interaction time could have been much reduced if the user was able to select a series of waypoints at a time instead of waiting for the robot to finish each leg before commanding the next. Alternatively, the user could have selected a single goal, far in the distance, and trusted the robot get there autonomously. We call the first ability Scripted Waypoints and the second ability Single Point Goal Selection.

5.1 Scripted Waypoints

The challenge with scripted waypoints is knowing the absolute position of the robot in a global reference frame. The problem is really twofold: a) how does the user know where he wants to go before the robot can see it and b) translating this point to a coordinate frame the robot can use to navigate to.

The first part of the problem can be solved by providing the user with an overhead image or map of the terrain. In theory the user would be able to select points on the overhead image on the fly and have the robot drive from one to the next.

The second part of the problem is translating the points that the user clicks on in an overhead image to a reference frame the robot can use to autonomously navigate to. The predominant global localization method is using GPS with a known relation between GPS coordinates and the overhead image.

In general, obtaining GPS data in a dense urban setting can be difficult, at times impossible. The buildings to either side of the robot prevent the acquisition of enough satellites to provide meaningful data. In other types of terrain with significant sky coverage, GPS is a viable option and not to be discounted, but in many urban environments GPS availability can not be assumed.

An attractive alternative is to use the buildings themselves as landmarks to help localize the robot. Recent work at JPL has shown that it is possible to use vision algorithms to extract features from onboard imagery, to match those features to known buildings derived from the overhead image, and thereby to estimate the position of the robot in the global reference frame. This option is under ongoing development in the MARS program and would be a great boon to autonomous capabilities for terrain types where GPS is unavailable.

This option has the additional benefit of being viable within buildings, where GPS is nearly guaranteed to fail. The feature template could be a rough blueprint of the site or a sequence of pictures taken by hand during some reconnaissance run a priori. In either case the robot could continue to calculate its position in an absolute reference frame, allowing better state estimation and bounding absolute error.

5.2 Single Point Goal Selection

An autonomous behavior above and beyond scripted waypoints would be a single waypoint, very distant, with enough path planning ability on board to get there without further advice. The two greatest challenges to do this are a sophisticated path planner and a very accurate state estimator.

The path planner would have to be capable of escaping cul-de-sacs, navigating narrow path ways and capable of broad goal definitions.

Even with a perfect path planner, accurate position in the global reference frame is still needed. GPS could provide this global frame, but as we discussed it may not be available. Without a global position sensor, any state estimator, such as the Kalman Filter we used on our robot, will drift over time. Although usually very accurate, our filter did not tolerate climbing stairs or driving off of curbs very well. The sudden acceleration and deceleration, as well as extreme track slippage and skidding, brought about a slight error in pose every time. Compounded over another 50 meters of travel this slight error would grow to several meters of error in the pose estimate. As discussed in 5.1, matching vision sensor data to a prior map obtained from overhead imagery is a promising approach to this problem.

6. User Attention

The user looked at the robot feedback during the entire course of each run. He never looked directly at the robot, but neither could he look away from the screen to do anything else. In any real application, this is a critical flaw. The main reason that constant attention was needed was that the robot's current ODOA capability was not adequate to keep it out of trouble that could damage the robot or terminate the mission. The main types of trouble that could be encountered were negative obstacles and moving objects.

6.1 Negative Obstacles

Urban settings are rife with negative obstacles: planters, stairs, curbs, and walkway edges. Rails that prevent humans from walking off pathways are often placed too high to be spotted by ODOA and allow the robot to travel without resistance into a crevasse. Some negative obstacles such as curbs are traversable, but only at low speeds. Although negative obstacle detection has been addressed for years for cross-country navigation, even there the problem is not completely solved, and we have not yet integrated such capabilities into Urbie. Moreover, the urban domain has

unique characteristics that could profit from algorithms designed specifically for it.

The inability of our autonomous system to guard against negative obstacles forced a great amount of vigilance on the part of the user. Referencing back to the Hard Course, the site of greatest interaction besides stair climbing was driving off the curb slow enough to be safe. The autonomous behaviors could not identify the ledge as dangerous, so the user was forced to manually switch to teleoperation and drive off slowly. Only when safely down could the user switch back into an autonomous mode.

6.2 Moving Obstacles

Since much of the navigation was on roads, the user needed to keep an eye on the traffic during autonomous drives. The robot could drive directly into the path of a car, not being able to calculate that the car was in fact in motion. There has been work on moving object detection with LADAR and with stereo vision [9,10]; incorporating this into obstacle avoidance algorithms is a critical next step. Street driving is the easiest of all autonomous traverses: relatively straight, on a solid surface, with few holes or cliff edges, and the few obstacles that exist are large (cars, trucks). This is where autonomy can shine, but without some form of moving object ODOA, the best autonomy will be able to aspire to is fewer user interactions, with no hope of reducing user attention.

7. Conclusions

We did a series of experiments to quantify the effect of our system's autonomy upon the amount of user interaction required. We did this to gain insight into where the system would need to improve to reduce the strain on the user.

We found that great amounts of improvement could be achieved with the ability to command traverses effectively beyond the current field of view. On simple terrain this could lead to almost zero interaction necessary for hundreds of meters at time-- although for complicated terrain this would require a very clever path planner and state estimation accuracy beyond what is currently available. Reducing attention on top of interaction would require a robust negative and moving obstacle detection behavior.

8. References

[1] D. R. Olsen and Michael A. Goodrich. "Metrics for evaluating human-robot interactions". *Performance Metrics for Intelligent Systems Workshop(PerMIS)*, 2003.

[2] M. Tambe P. Scerri, and D. V. Pynadath. "Towards a theory of adjustable autonomy". *Journal of Artificial Intelligence Research*, volume 17,pages 171–228, 2002.

[3] S.Tejada, E. Normand, and S. Sharma. "Virtual Synergy An Interface for Human-Robot-Agent Interaction". *Workshop on Humans and Multi-Agent Systems*, 2003,

[4] T. Frost, C. Norman, S. Pratt, and B. Yamauchi. "Derived Performance Metrics and Measurements Compared to Field Experience for the PackBot". *Performance Metrics for Intelligent Systems Workshop(PerMIS)*, 2002

[5] D. Bruemmer, J. Marble, D. Dudenhoeffer, M. Anderson, and M. McKay. "Intelligent Robots for Use in Hazardous DOE Environments". *Performance Metrics for Intelligent Systems Workshop(PerMIS)*, 2002

[6] M. A. Goodrich and E. R. Boer. "Model-based human-centered task automation: A case study in ACC system design". *IEEE Transactions on Intelligent Transportation*, part A: Systems and Humans, Vol. 33, No. 3, May 2003

[7] M. A. Goodrich and D. R. Olsen, Jr. "Seven Principles of Efficient Interaction". *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, October 5-8, 2003, pp. 3943-3948.

[8] J. W. Crandall. *Towards Developing Effective Human-Robot Systems*. M.S. Thesis, Brigham Young University, 2003.

[9] E. Prassler, J. Scholz, M. Schuster, D. Schwammkrug. "Tracking a large number of moving object in a crowded environment", *IEEE Workshop on Perception for Mobile Agents*, Santa Barbara, CA, June 1998.

[10] A. Talukder and L. Matthies. "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow", *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, Sendai, Japan, October 2004.

A Mixed Initiative Human-Robots Team Performance Assessment System For Use in Operational and Training Environments

A. Freedy, J. McDonough, R. Jacobs, E. Freedy, S. Thayer and G. Weltman
Perceptronics Solutions Inc., Sherman Oaks CA 91423
M. Kalphat and D. Palmer
U.S. Army RDECOM-STTC, Orlando FL

ABSTRACT

Military forces of the future will use mixed manned and unmanned forces for a broad variety of functions. Measurement of overall effectiveness in these mixed initiative systems will be essential in order to achieve optimal system performance levels. Behavioral measures of both human and unmanned performance obtained in system simulations or in live exercises will be used to continuously diagnose performance and identify required areas of training requirements. Likewise, specialized training will be necessary in order to leverage the complementary cognitive functions of human and machine to forge fighting entities and units with capabilities superior to those of humans or machines in isolation. Our team is currently developing a Mixed Initiative Team Performance Assessment System (MITPAS) consisting of a methodology, tools and procedures to measure the performance of mixed manned and unmanned teams in both training and real world operational environments. The work is being performed under SBIR Phase I and II contracts administered by RDECOM/STTC, Orlando, FL. Our objective is to provide a scalable turnkey MITPAS software system integrated with simulation and training environments, utilizing COTS HLA data logging tools and containing protocols for evaluation of various manned/unmanned team configurations in selected event-based scenarios. This paper describes our in-progress development of a underlying Multi-Dimensional Performance Model, our preliminary MITPAS architecture and our Use Case Scenario based experimental and evaluation plan, as well as our ideas for future applications of the completed MITPAS.

KEYWORDS: *mixed initiative teams, human-robot performance assessment, robotic training systems*

1. INTRODUCTION

Mixed initiative introduces a new and unique aspect to the psychology of team performance: the interaction of two cognitive systems -- human and autonomous unmanned robot. In addition to the critical performance factors associated with human teams -- which include information exchange, communication, supporting behavior and team leadership -- the mixed manned/unmanned team adds a number of challenging new dimensions. Foremost among these is the ability of the human team to manage, predict, collaborate and develop trust with unmanned systems that may sometimes exhibit fuzzy responses in unstructured and unpredictable environments [1] [2] [3] [4] [8] [9].

The critical challenge in our work has been to develop system-specific measures of behavior on which to base assessment of the mixed initiative team performance. Such measures must be unique to the information and decision environment associated with human-robotic teams and to directly link together behavioral processes important to mixed manned/unmanned tactical outcomes. The measures need to provide feedback for skill improvement in collaboration as well as adaptation to stress and workload, and they should help define the training needs themselves.

2. PERFORMANCE MEASURES

Our work on the definition of relevant performance measures began with the realization that future unmanned platforms will have some capability to operate autonomously within the scope of their mission tasking, but will be continuously "commanded" by human operators who will each direct the activities of a number of robots. As more is learned about modeling human behavior, increased sophistication in autonomous operations by robotic systems can be expected to reduce dependence on human supervisory controllers. At today's and the near-future state of understanding, however, certain functions are not well supported by automation and can be performed at a much higher level of competence by human beings in collaboration with the robotic entities.

Accordingly, at the performance level there are new human factors issues that require new types of skills and training. These emerge from the nature of the robots as decision-making systems operating in uncertain, unpredictable and unstructured environments. The key new performance issues include:

- Performing supervisory control of robots

- Adapting to variations in the level autonomy the robots exhibit in response to environmental and task variables
- Varying task allocation to exploit the distinct advantages of the human and robotic component (e.g., a robot can endure long mission duration, survive better but may have only about 80% of human cognitive capabilities)
- Monitoring of robots' decisions and actions to maintain to achieve transparency of robot actions
- Overriding robot decisions and actions when necessary
- Helping to solve problems and handle contingencies

Research performed to date on measurement of team training performance has focused on both the individual and team levels [5] [6]. It is recognized that while both process and outcome measures are essential, training feedback mainly comes from process measures. The guiding principles are: (1) measurement and remediation must emphasize processes that are linked to outcomes; and (2) Individual and team levels deficiencies must be distinguished to support the instructional process. In our view these principles are directly applicable to the manned/unmanned team with the addition of another level in the team structure, which we term as the Collective Manned/Unmanned (CMU) level, and which represents the major new dimension that is added to the team task characteristics and structure. Our selection of measuring instruments and speci-fication of associated measurement methodologies thus extends the individual-team matrix of Cannon–Brower [1] to include the present case of collaborative manned/unmanned teams.

3. PERFORMANCE MODEL

The basis of our MITPAS approach has been to develop a Manned/Unmanned Team Multi-Dimensional Performance Model that captures the critical performance attributes of the distinct human and robotic decision and control environment. Figure 1 below provides an overview of the hierarchical structure of the Model's performance dimensions.

The Performance Model we are developing draws on four separate research areas that have been pursued independently in the past but which are being integrated in this project to establish meaningful criteria of overall performance. These research areas are:

- **Psychology of Team Performance** - Human team performance measurement in C3 information environments, performance variables, training evaluation and measuring team related expertise, management of workload and stress.
- **Unmanned Systems** - Principles of establishing performance metrics for autonomous systems
- **Mixed Initiative Systems** – Research and findings on the critical variables which affect human decision and control of autonomous systems
- **War Fighting Behavior** – Observations and measurements of combat team performance in war fighting tasks C3 tasks

We have integrated and adapted theories and concepts in these areas to processes associated with manned/unmanned team

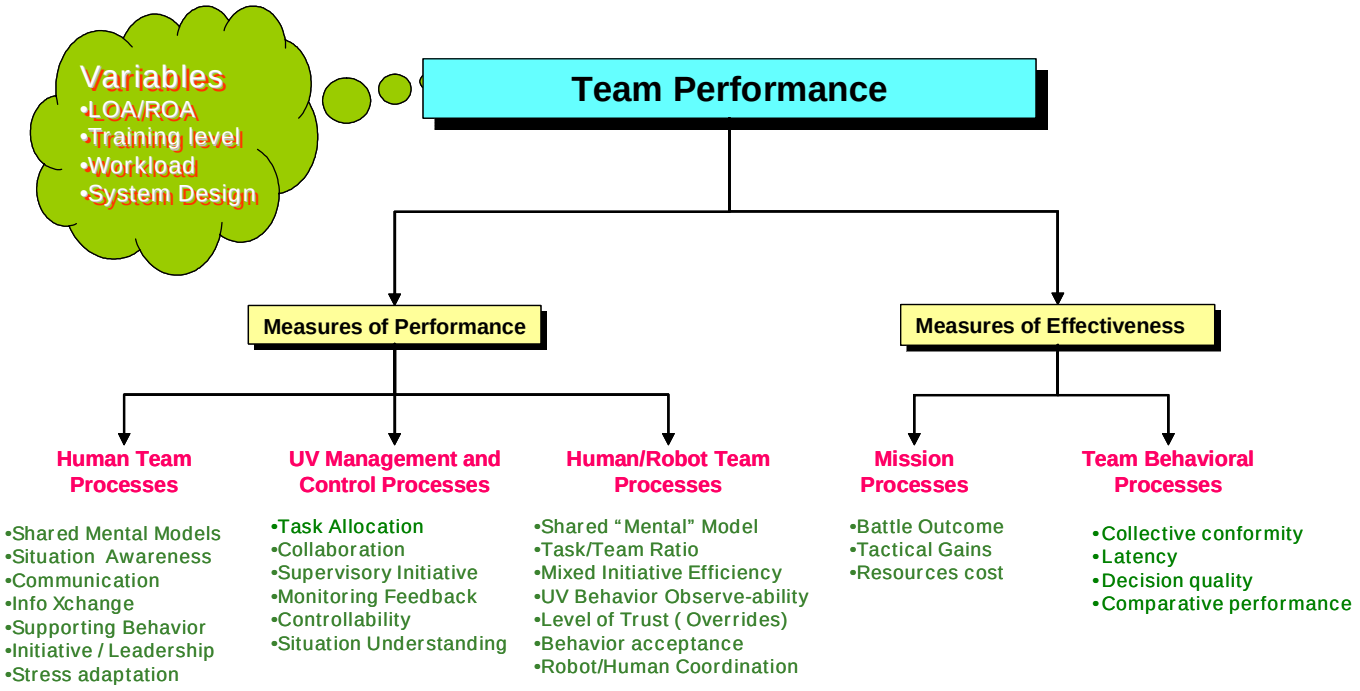


Figure 1 System Performance Model

performance and training. Most critical were variables related to the decision making behavior of the unmanned

systems, such as behavior transparency to the human collaborators, human trust in robot decisions and human abilities to synergize the autonomy of robots so as to add to the capability of the total team. Issues such as behavior prediction, level of autonomy and acceptance of robots actions have also been examined and identified for possible high impact variable on total system performance.

In accord with this approach, we have created a preliminary System Performance Model which captures the critical performance attributes of the distinct process of behavior composition environment. Our objective was to identify the dimensions of performance which contribute to effective outcomes of collaborative manned-unmanned tasks and, in particular, to formulate measures to evaluate training in processes that are unique to the collective team of humans and robots. Accordingly, we have built a taxonomy of specific processes which can be decomposed into explicit behavioral objectives side-by-side with measures of effectiveness based on actual outcomes. Our focus is on process measures that are closely linked to outcomes, because it is these measures that will provide the feedback necessary for training. The three levels of team processes critical to training evaluation and remediation are: (1) individual human; (2) team human; and (3) collective human/robot team.

We decomposed the processes into these three levels and developed taxonomy of measures for each level. We narrowed the performance measures to the simplest factor structure that adequately cover the dimension of teamwork as was found in previous investigators [2]. The actual Performance Model will consist of a multi-dimensional task process performance schema which will (1) aggregate the performance measures at each level, (2) provide for training feedback at each level, and (3) provide a multi-attribute discriminate function to determine an overall level of proficiency as well as a "pass-fail' score. The weights of the attributes will be established in simulations in which the linkage between specific task performance measures and outcomes can be estimated. There are two main types of measures: Measures of Performance (MOP) and Measures of Effectiveness (MOE); these are defined separately below.

1. Measures of Performance (MOP)

These are observable and derived measures of the operators' task skills, strategies, steps or procedures used to accomplish the task. They consist of the cognitive and interactive processes of the individual and team in collaborating together and controlling the robotic entities in a coordinate manner. MOP evaluates the human factor involved in a complex system. MOP was divided into 3 distinct classes of processes dimensions:

- **Human Team Processes** - These processes represent the dimensions of the human team interaction
- **UV Management and Control Processes** - These processes represent the tasks associated with real time control and monitoring of the autonomous entities
- **Human/Robot Team Processes** - These processes represent the dimensions of the human interaction with the robotic elements

2. Measures of Effectiveness (MOE)

These measure the "goodness' of the composed behavior in quality and the execution of war-fighting tasks. MOEs are influenced by much more than human performance. These measures also contain variance accounted for by system design, the surrounding environment and luck [6]. The measure consists of the following dimensions

- **Mission Effectiveness** - Observable measures of the success of the mission as determined by objective military criteria.
- **Behavioral Effectiveness** - Measures of the dimension of behavioral effectiveness of the system in the battlefield

We anticipate that only a relevant and/or application-specific subset of all possible performance measures will be used in the turnkey MITPAS because: (1) some of the measures may be correlated; and (2) the selected ones will require assurance of high diagnostic value, which is referred to as *discrimination validity*, in the particular situation. In our future laboratory tests we plan to reduce the possible set of measures to a manageable subset.

4. MITPAS FUNCTIONAL REQUIREMENTS

Our plan is to implement MITPAS as a turnkey software package incorporating three major capabilities:

- Tools to set identify and specify key events that must be included in an exercise in order to stimulate execution of actions by participants that are the targets of performance measurements;
- Tools to capture data during the conduct of the exercise, including automated extraction from data loggers and formats for observational inputs from observers and controllers;
- Analytical tools to combine the data collected and produce quantitative measures of the performance and effectiveness of the human-robotic team(s) being studied;
- Report generation tools to allow researchers and trainers to produce diagnostic and prescriptive arrays of the analytic products.

We will also build initial tactical and technical databases, using proposed FCS tables of organization and equipment and similar documents from other UV programs, databases.

Figure 2 below diagrams the MITPAS system and its place within the training and evaluation environments. The Figure focuses on MITPAS as an adjunct to the existing distributed interactive training environment, specifically the OneSAF Test Bed (OTB), in which it will be developed and initially evaluated. Figure 2 also expands on the normal context diagram conventions to include the internal components of the system as well, highlighting which components interact with which outside entities.

5. MITPAS ARCHITECTURE

In our planned future efforts we will complete and implement the MITPAS software architecture, developing the interfaces with external systems and user interfaces to support identification of scenario requirements, selection of measures, monitoring and data collection, and post-exercise review and analysis. We will also develop the analytical engine within

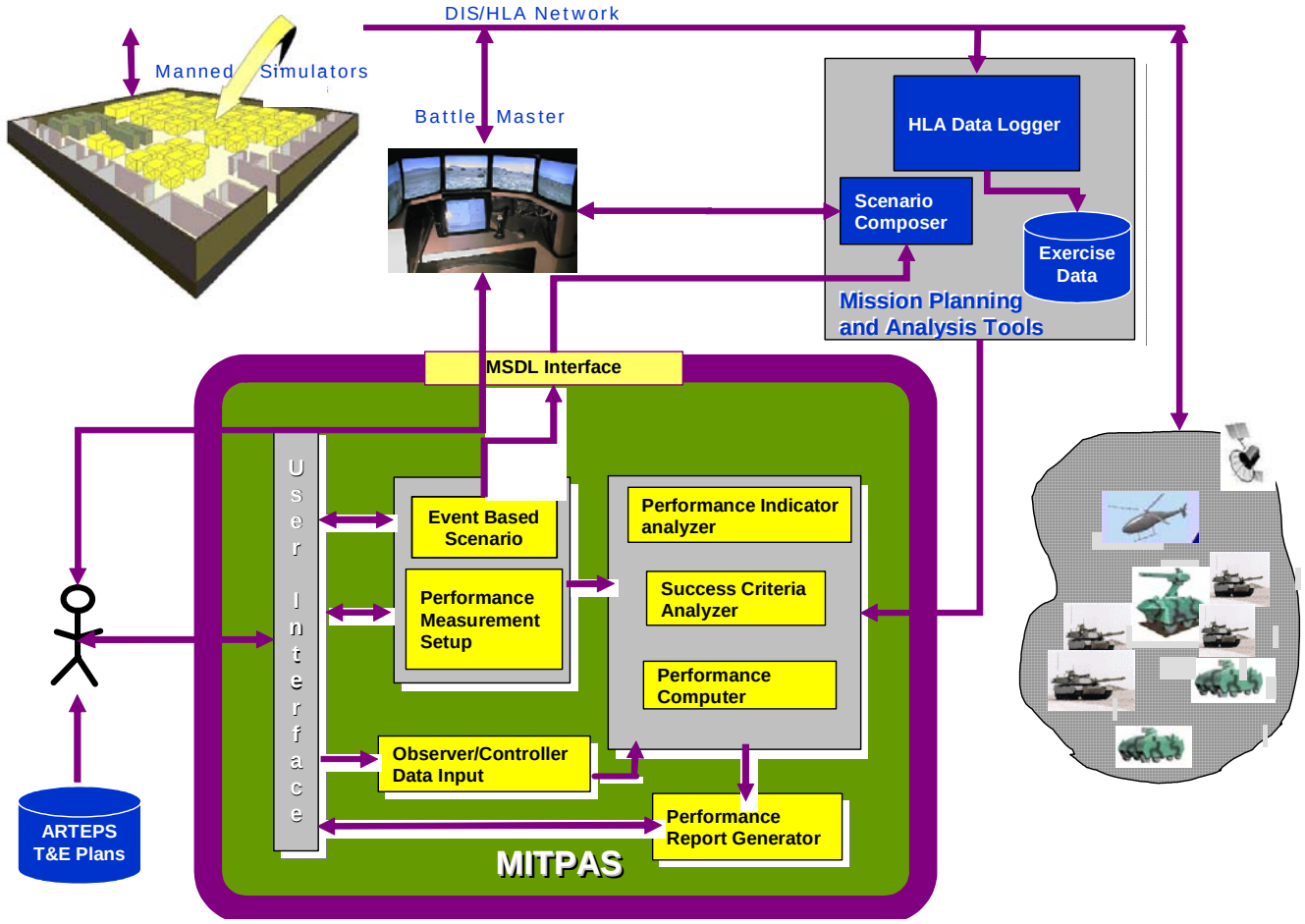


Figure 2 MITPAS Components and Context

In its initial implementation the system will also serve as the environment in which candidate measures and metrics are tested against actual exercise performance in experiments to identify and validate those measures that are most correlated with and predictive of successful tactical performance and battle outcome. We will define the high-level system functions in terms of Use Case Scenarios and Interaction Diagrams for the various types of users as well as for interactions between MITPAS and external systems, such as:

- Military Instructors and systems performance evaluators
- Unit commanders who assign and monitor mission status
- System Designers and Planners

the software, and as the performance measurement algorithms are developed they will be embedded in that component. The development of components will be done iteratively, in a spiral development process, providing an early initial capability for experimentation, and evolving as experiments yield more data about performance and system requirements. In brief, we will implement a MITPAS Prototype System that will:

- Provide a Core Infrastructure for measuring the performance of Mixed-Initiative exercises. The core infrastructure is designed to facilitate the rapid implementation of performance measurement and analysis algorithms as well as to enable integration

with multiple heterogeneous simulation and test environments.

- Implement the specific performance measurement and analysis detailed for the scenario described in this proposal using the Core Infrastructure

Careful consideration will be given to allow the system to be scalable and provide extensive integration capabilities to meet evolving performance assessment requirement over the system life-cycle. Critical to achieving these goals is the use of a modular component-based software architecture which extensively leverages open standards and de-facto standard best practices in distributed system development.

Furthermore, the system will leverage established tools and components which have emerged from prior DoD investment in modeling and simulation as well as independently developed tools for collecting and analyzing data for DIS and HLA. Additional consideration will be given to developing and emerging standards in the training and simulation communities. In particular, the MITPAS Core Infrastructure will be designed to support the Test and Training Enabling Architecture (TENA) under development for PEO STRI as a product of the Foundation Initiative 2010. TENA provides significant improvements on HLA and is designed to be used with embedded training systems and in training ranges.

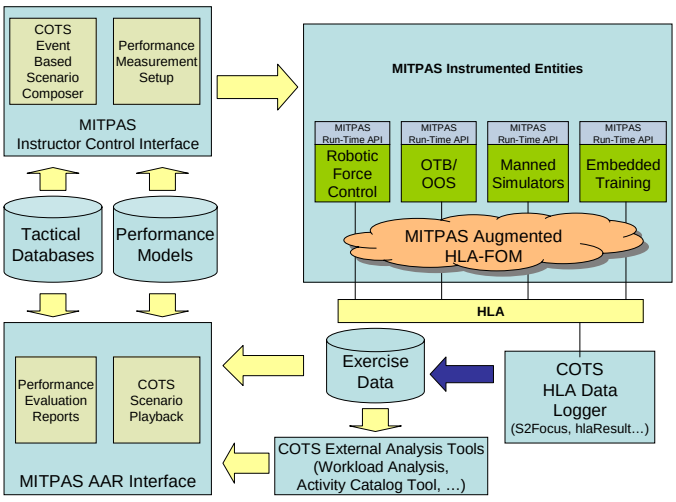


Figure 3 MITPAS Architecture

Figure 3 shows the main MITPAS system architecture. The system is comprised of the following core components:

- MITPAS Instructor Console – An application to set parameters for a given Mixed-Initiative exercise as well as construct a scenario
- MITPAS Instrumentation Run-Time API – A middleware toolkit with APIs in C and Java to enable rapid instrumentation of entities including C4I Systems, simulation systems, and embedded training systems
- HLA Data Logger Interface – A connection to an existing data capture mechanism for capturing and managing data from an HLA data-stream
- MITPAS AAR Interface - An application which implements the analysis and reporting capabilities of the system as well as invocation of Scenario playback
- MITPAS Augmented HLA FOM – Supports capture of additional data such as human interaction events, MITPAS will require augmenting of a particular HLA Federation Object Model to add the additional classes and interactions.

6. Use Case Scenario

We will use scenario-based training trials as the experimental paradigm to identify, refine and validate MITPAS measures. Scenario-based training relies on controlled exercises, or vignettes, in which the target training audience is presented with cues that are similar to those found in the actual task environment and then given performance feedback. In mature training environments such scenarios are developed using training and doctrinal materials such as ARTEPS and Mission Training Plans along with validated performance measures. In the MITPAS project, however, the goal is to identify and validate measures for a type of unit that does not yet exist and for whom no training documents have been developed. Accordingly, we have developed a baseline scenario based on:

- Examination of candidate performance measures
- Study of the Future Combat System 2015 Unit of Action Design
- Sponsor focus on countermine capabilities

Our current MITPAS use-case scenario focuses on a platoon of a Reconnaissance Troop, reinforced with Engineers, which is escorting a convoy in an Iraq-like environment. The platoon employs UGVs, SUGVs, UAVs (Type 3), MULES and an ACRV, which allows for representation of a wide range of robotic capabilities and supports experiments focusing on soldiers controlling individual robots, on those controlling multiple homogeneous or heterogeneous robots, or on a leader controlling mixed human and robotic elements. Our current scenario requires subjects to deal with an improvised explosive device, a traditional minefield, small unit enemy action, casualties, and maintaining communications.

We are able to identify a set of critical control events within the MITPAS scenario that exemplify the type of mixed initiative performance we are trying to assess. In the future these critical events will be further refined in cooperation with our RDECOM PMs. In addition, the final scenario events and candidate performance measures will then mapped to each other to ensure that scenario execution will elicit the actions

that the measures require. Table 1 below shows an initial stage in the process, in which measures are mapped into scenario events based on the current findings. The purpose here is to demonstrate the methodological approach, rather than provide an exhaustive listing, which will form part of the planned future effort.

Table 1 Scenario Events vs. Performance Measures

MEASURE		EVENT																									
		1	1.a	1.b	2	2.a	2.a.1	2.b	2.b.1	3	3.a	3.a.1	3.a.1.a	3.a.2	3.a.2.a	3.b	4	4.a	4.a.1	5	5.a	5.b	6	6.a	6.a.1	6.b	6.b.1
Human Team Processes	Shared Mental Models	X	X	X		X		X	X		X	X	X	X		X	X	X		X	X	X	X	X		X	X
	Situation Awareness	X	X	X		X		X	X		X	X	X	X		X	X	X		X	X	X	X	X		X	X
	Communication	X	X	X		X	X	X	X	X	X	X	X	X		X	X		X	X	X	X	X	X	X	X	X
	Information Exchange	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X
	Supporting Behavior	X	X	X		X		X	X		X	X	X	X		X	X	X		X	X	X	X	X		X	X
	Initiative/ Leadership	X	X	X		X		X	X		X	X	X	X		X	X	X		X	X			X		X	X
	Stress Adaptation	X	X	X		X		X	X		X	X	X	X		X	X	X		X	X			X		X	X
UV Mgmt and Control Processes	Task Allocation	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X
	Collaboration	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X
	Supervisory Initiative	X	X	X		X		X	X		X	X		X		X	X	X		X	X			X		X	X
	Monitoring Feedback	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	Controllability	X	X	X		X	X	X	X		X	X		X	X	X	X	X		X	X	X		X		X	X
	Situation Understanding	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

7. Criteria for Success

Our approach to establishing criteria of success will follow the concepts of the Army Training and Evaluation Program (ARTEP), which is the cornerstone program of unit training.

Each ARTEP consists of defined tactical tasks to be performed under specified conditions to a criterion or standard. To determine if the standard is reached, the ARTEP provides evaluators with a list of Task Steps and Performance Measures scored Go, No Go or Not Evaluated. The ratio of subjective Go to No Go marks and the significance of each determine whether the performance standard has been met. While the Rates have evolved over decades to capture virtually all-relevant measures of performance with regard to human collectives, collectives of humans and robots will demand the exercise of additional skills by the human elements. The robots' decisions will not always be transparent to the humans. Human acceptance of these decisions will depend on understanding the robots' capabilities and anticipate robot behavior. The approach was proposed in our Phase I proposal and furtherer analysis validated its applicability and effectiveness.

We will aggregate the individual performance measures into a scoring criterion by starting with selected ARTEPS that can be adapted to human-robotic collectives (using FCS training studies as a guide) and adding additional measures such as the ones discussed above. The single-score-for-a-single-task methodology of ARTEP will be expanded to provide a single score for a collective patterns of tasks We propose a multi-dimensional criterion of performance success, P, that combines the direct performance measures across the various experimental (robot system) variables, as described below:

1. Let x be the pattern of performance measures $x_j = (x_{1j}, x_{2j}, x_{3j}, \ldots, x_{nj})$ under the various conditions, i.e. level of automation, stress, etc marked by the subscript j

The multi-attribute performance score for condition j is:

$$g(x_j) = w_1 x_{1j} + w_1 x_{2j} + w_1 x_{3j} + \ldots + w_n x_{nj},$$

thus

$$g(x_j) = \sum_{i=1}^{n} w_i x_{ij}$$

2. To get a total score cross all conditions the combined score is

$$\sum_{j=1}^{m} g(x_j)$$

3. The combined aggregated score for all performance measures and condition will then be:

$$\text{Total Score } P = \sum_{j=1}^{m} \sum_{i=1}^{n} w_i x_{ij}$$

4. The weights $(w_1, w_2, \ldots, w_n)$ will be determined by testing experiments and expert judgment using a parameter estimation protocol of the type used in trainable pattern recognizers.

We have developed a schema employing a factor analytic approach to reducing and refining the set of measures to reflect underlying orthogonal performance dimensions [7]. This strategy will be employed using a virtual battlespace to collect data for analysis.

The scenarios, candidate measures and algorithms, and the OTB V2.0 virtual testbed provide a framework for a multi-stage data collection effort within which soldiers with representative background, experience, training, and skill levels will be asked to execute FCS missions as part of a human-MULE robot team. After a verification and validation effort to ensure that the test software produces the intended data products, mission trials will be conducted in which soldiers will team with robots to perform specific assignments within the exercise scenarios. The simulation, instrumented with the selected data extraction and analysis tool, will produce measure data for each of the candidate measures constituting the independent variables.

Dependent variable data will come from a different source. The Objective Force combat development community will be asked to provide subject matter experts to observe the trials and to provide subjective evaluations of the execution of the human-robot team. Accepting the expert judgment to be the reference standard for performance evaluation, the factor analysis process will be employed to examine the value of the component and composite linear factor combinations of the candidate measures in accounting for observed performance. The intent is to seek to identify a reduced set of orthogonal underlying composite measures to which a practically substantial proportion of the measure variance (in relation to expert subject judgment) can be allocated. Conceptually, the process can be thought of as a rotation of the principal variable axes within the data space to identify a new coordinate set that minimizes the data variance. The rotated axes are linear combinations of the original set, and correspond to underlying variable factors suggested by the distribution of the data in the variable space. Factor analysts often look upon this as "first-stage solution" and will typically follow this with further non-orthogonal rotations to achieve what they call a "simple structure". For our purposes however, this will not be advisable, as non-orthogonal rotation has implications to the independence, transformation, and scaling of the data.

8. Experimental Plan

Our planned experimental test program is structured in four parts. Following is a preliminary description of each phase; the detailed test design will be produced during the requirements development effort.

1. Laboratory System Pilot Runs

In the first phase, the test environment will be set up and validated. Pilot runs will confirm that the measurement algorithms are functioning correctly, that the scenario is properly simulated, that the participating virtual platforms and behaviors representations are valid, and that the human operator interface is fully functional. Pilot runs will be conducted to confirm that the design is fully responsive to the requirements of the program.

2. Model Validation and Tuning

The second phase will be devoted to collecting data across the spectrum of operations in the scenario, expert observation and evaluation, and reduction of the measure set through factor analysis. The focus will be on the simplest form of human-robot team, a single operator supervising the activities of one or two robots. The scenario will be executed in the context of FCS embedded individual training with an emphasis on what might become ARTEP/Drill tasks for the human-robot team.

3. Battle Operations in Simulation

We will validate the reduced measure set by applying it to a more complex set of activities representative of FCS battlespace operations. The scenario will involve sequences of the types of tasks that formed the focus for phase two, and it will be executed by a small team consisting of two or more human operators and several virtual robots. This will introduce the dimension of collaboration and allocation of responsibilities to the scenario execution.

4. Field Operation with Live UVs

As an option, we propose in a fourth phase to demonstrate the operation of the performance measurement system in a live

simulated environment using instrumented UVs operating on a tactical range.

9. Conclusions

The key challenge being addressed in this project is the fact that autonomous vehicles, or agents, will need to interact and coordinate with each other and with human systems. Measurement of overall effectiveness in these mixed initiative systems will be essential in order to achieve optimal system performance levels. Behavioral measures of both human and unmanned combat system (UCS) performance obtained in system simulations or in live exercises will be used to continuously diagnose performance and identify required areas of training requirements [3].

Likewise, specialized training will be necessary in order to leverage the complementary cognitive functions of human and machine to forge fighting entities and units with capabilities superior to those of humans or machines in isolation. Embedded training is also projected to be an important part of the Future Combat System (FCS) to assure that performance levels remain high during all operational phases. Overall, a clear and definite need exists for methods and mechanism to assess and determine criteria for successful performance of unmanned systems and manned/unmanned teams in both training environments and the real warfighting situations.

We believe that meeting this need will also lead to significant commercial product opportunities in the large and rapidly expanding military and non-military markets for robotic systems. The focus of our SBIR commercialization strategy will be transformation of the MITPAS prototype into a suite of software modules for use in a variety of mixed initiative and mobile agent applications. The software product will be optimized to meet military and non-military market requirements. It will be sold and/or licensed to DoD and Homeland Defense agencies and prime contractors, to civil organizations that employ remote human controlled robotic agents and unmanned vehicles in hostile environments and for counter terrorism activities and local law enforcement, and also to companies manufacturing and distributing industrial and personal robots. In addition, we plan to explore in Phase the application of the MITPAS as a commercial tool for helping military and non-military emergency response teams determine when and how to use mixed initiative teams on a particular type of mission, e.g., in a bomb disposal situation.

10. References

[1] Fong, T., Thorpe, C. and Baur, C., "Collaboration, Dialogue and Human Robot Interaction." *Proc. 10th International Symposium on Robotic Research*, Spinger Veralng, 2002

[2] Fong , T., Kaber, D, Scholtz, M.L. , Shultz, A. and Steinfeld, A., "Common Metrics for Human-Robot Interaction" *IEEE International Conference on Intelligent Robots and Systems,* Sendai, Japan, 2004

[3] Kalphat, H. M., and Stahl, J., "STRICOM's Advanced Robotics Simulation STO: The Army Solution to Robotics M&S," *Proceedings of the Eleventh Conference on Computer Generated Forces & Behavioral Representation*, Orlando, FL, May 2002.

[4] Cannon-Bower, J.A. and Salas, E., "Individual and Team Decisoio Making Under Stress: Theoretical Underpinning" in *Making Decisions Under Stress*, (Ed) Cannon-Bowers, J. and Salas E., American Psych. Assoc., WDC, 1998

[5] Singh, P.N, Thayer, M.S., "ARMS: Autonomous Robots for Military Systems, A Survey of Collaborative Robotic Core Technologies and Their Applications," CMU–RI-TR-16, DARPA Project Report, 2001

[6] Smith-Jentsch, K.A., Johnson, J.H. and Payne, S.C., "Measuring Team Related Expertise in Complex Environments", in *Making Decisions Under Stress* (Ed) Cannon-Bowers, J. and Salas E., American Psych. Assoc., WDC, 1998

[7] Tatsuoka, M.M., *Multivariate Analysis, Techniques for Educational and Psychological Research*, John Wiley & Sons, 1971

[8] Albus, J.S., "Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles" *PerMIS Proceedings*, 2002

[9] Clough, B.T., "Metrics, Schmetrics! How the Heck Do You Determine an UAV's Autonomy Anyway?" *PerMIS Proceedings*, 2002

11. Acknowledgements

1. The work reported here was funded under Contract No. N61339-04-0020 administered by US Army RDECOM-STTC, Orlando, FL

2. Dr. Robert Jacobs serves as a staff member of the Institute for Defense Analysis, Alexandria, VA
3. Dr. Scott Thayer serves as Systems Scientist in the Robotic Institute at Carnegie Mellon University, Pittsburg, PA
4. Michelle Kalphat serves as Chief Engineer, Intelligent Behaviors for Autonomous Systems & Simulation Technologies, US Army RDECOM-STTC, Orlando, FL
5. Donnie Palmer serves as Computer Scientist, Intelligent Behaviors for Autonomous Systems & Simulation Technologies, US Army RDECOM-STTC, Orlando, FL

BIOLOGIZING CONTROL SYSTEMS

Paul Wang
Duke University
Durham, NC
ppw@ee.duke.edu

Abstract

One of the most important performance metrics for a control system is the reliability. To accomplish this objective, some intelligent controller must be developed. Relevant theory on this line has been von Newmann's cellular automata theory.

However, von Neumann's theory of cellular automata is not general enough to realize the 'biologizing" of control systems. Furthermore, the non-autonomous responses must be studied in order to realizing the truly reliable system via self-organizing schemes. This paper presents both new features.

ENTROPY AS A PHILOSOPHY

George N. Saridis PhD

Dedicated to Ilya Prigogine

PREFACE

The ancient Greeks considered philosophy as the mother of sciences engulfing the essence of all human achievements. It is therefore, natural to associate her with elements of universal scientific developments. The intention of this article is to present the current point of view of the situations governing our planet, our societies, our environment, and attempt to foresee their future developments**.** Entropy, the thermodynamic measure, is an analytic tool that may serve to evaluate phenomena that dominate our lives, and therefore belongs to the realm of philosophy. Since the trends show a universal trend for equalization, like the concepts of codification and globalization, an increase in global entropy is appearing as an obvious result. Is this going to create an "equivalent thermal death" to our planet? Is there a cure to this phenomenon? The recently developed theory of Chaos, introduced by Ilya Prigogine gives us some hopes for a possible redemption of this catastrophe. This will be discussed scientifically in the sequel, by considering the various elements of our present situations.

1. INTRODUCTION

The creation of the Universe and the mystery of life have always been problems puzzling the philosophers. Technological progress and the increasing knowledge of the elements of the environment have not considerably improved their beliefs. With the years a feeling of uncertainty about the Universe and the human existence has been generated and recently has become more pronounced.

Young people ask questions about God, our past and future existence and in general, how much control we do have on our lives and our environment. The world exists the way we see it and the models we have produced are inadequate to generate any positive answers. This creates a chaotic situation in the minds of the thinkers, desperately trying to create a viable model of the world that explains the past and predicts the future.

Since the early days, when human thought was developed, many attempts were made to systematically formalize the human activities in the Universe. Biological, Societal, Religious, Scientific, and Legalistic theories, to mention a few, were created independently; to model and represent the world we live in. Most of those were dealing philosophically with believes, that were created by the people in the form of irreversible phenomena.

Philosophy was always the main source and foundation of progressive theories. Ancient Greeks considered it as the mother of all sciences. It was always considered as a source of global energy. The historian Thucydides said "□□□□□□□□□□□□ □□□ □□□□ □□□ □□□□□□□□□□□□ □□□□□ □□□□□□□□. " It is the ground of development of clear modern thought both scientific and literary. It is in this aspect that will be used in the present document to describe our world.

In our days with the development of computer science scientists from different areas were able to communicate with each other end establish analytic models that approximate the behavior of various phenomena to the degree that we presently understand them. Therefore we hear biologists talk about models of DNA, ecologists about models of the environment, sociologists about globalization, thermodynamisists and mechanists coordinating their efforts, and so on and so forth. Analytic approaches to describe human activities appeared also in the world of literature. Such models, still at a primitive stage, are considered around the equilibrium of each process, and obey certain laws that are emanating mainly from physics, the science that started it all.

There is a notion of pessimism in this approach generated by the second axiom of thermodynamics where entropy invokes the **thermal death**, for all natural activities. Is this argument valid or it is assumed due to the closeness to the natural equilibriums? May be when we move away we may

encounter a regeneration the way that Prigogine proposes with his theory of **Chaos**. There is still hope for our world.

However, little has been devoted to the concept of energy as the source of generation of such activities. Entropy, an irreversible source of energy, that like life itself, looks like a strong candidate to describe the phenomenon of the Universe and belongs to the philosophical framework. This claim is also justified by the conditions of disorder and tastelessness that exist in the modern world. This of the entropy formulation will be elaborated in the sequel.

2. WHAT IS ENTROPY

In the modern times one may extend the concept of philosophy to cover entropy, the low level energy, which results as a terminal by-product of any kind of effort created in most cultures. **Entropy** has been defined as the residual irreversible energy generated in thermodynamics (Boltzmann 1872), and has been extended by Prigogine (1980,1996), Saridis (1995) et al (1988,1957), to denote the energy of the probabilistic view of the world. Such a concept is very popular in modern science, and expresses the uncertainty of the model of the world we have created. Since the idea of a probabilistic world is generic, with an "arrow of time pointing forward" (Coveney and Highfield 1990), it may cover most cultural activities it may also be interpreted by a **Global Philosophy**.

For centuries scientists theologians and sociologists were arguing that the world is deterministic. "**There is always a beginning and an end of the world**" they said, in an anthropomorphic way, since life has a beginning and an end. This includes the modern theory of the "Big Bang" as the creator of the universe. However, life and the world in general, represent irreversible phenomena. Without trying to interpret dogmatically metaphysical phenomena, scientists tried to build analytic models of the existing world and found only crude approximations to match reality. The alternative was the assumption of a probabilistic description that fitted best the data.

The physicist Clausius discovered this low quality energy, which appeared in the second law of thermodynamics and was named Entropy. According to this law the production of work is followed by the production of residual energy that irreversibly increases the total level of the lower quality energy of the world. This phenomenon would lead to the exhaustion of the useful energy, preventing the creation of new useful work, thus creating a thermal death of the world. Boltzmann used entropy to study the behavior of gases (Boltzmann 1872, Prigogine 1996).

Entropy was given a different interpretation of by Claude Shannon (1963), as a measure of uncertainty in information theory related to telecommunication systems. This interpretation was used by Saridis (1995), to introduce a theory presenting Automatic Control as a generalization of the theory of entropy, based on the designer's uncertainty to obtain the optimal solution. This concept is hereby extended to cover subjects related to the environment, finances, pollution and other problems that puzzle our present society.

3. FOUNDATION OF AN ENTROPY MODEL

The contribution of this work is the introduction of uncertainty to the Universe, and to effectively restrain the growth of the Global Entropy, created by the human intervention with the environment. Since the text is addressed to the unspecialized reader, an attempt will be made to introduce first the probabilistic concepts of our cultural systems.

It is not the intention of this effort to challenge the question if the world is either deterministic or random. Instead in my opinion it recognizes that the models we use to represent it are uncertain, being only an approximation of the real world. It is well known that "Mother Nature does not read mathematics".

The model of the world I propose is based on the idea that the world lives in an uncertain space of approximations, where every point is assigned a probability of success based on some performance criterion. The selection of the appropriate model is based on the point in space of highest probability.

Modern technology, has greatly contributed to the improvement of the quality of human life, and to explain various phenomena of our cultural environment. At the same time, it has increased the production of waste, traffic congestion, biological pollution and in general environmental decay, which can be measured by the increase of the Global Entropy of our planet, an energy that tends to deteriorate the quality of our environment (Brooks and Wiley 1988, Prigogine 1980, Rifkin 1989). This entropy, in agreement with the arrow of time, is modeled according to the second law of thermodynamics, which is an irreversible phenomenon of generation of low quality energy.

The original idea of using entropy to describe the deterioration of the quality of energy, when useful

work is produced, is due to Boltzmann with his monumental work in statistical thermodynamics (Boltzmann 1872). Shannon {1963) followed with the introduction of entropy in his information theory. Accordingly, Brooks and Wiley (1988) recommended entropy as the unifying theory for Biological Evolution, and Rifkin (1989) suggested the use of entropy to measure the production of waste in environmental systems. This concept is utilized in this paper in order to manage its component due to human functions.

Saridis, using Janes' principle of Maximum Entropy (1957), formulated the engineering design problem as a problem with uncertainty, since the designer does not know a-priori the outcome of his design (Saridis 1995). Entropy was used as the measure of the energy associated with the assumption of irreversibility of the process. This way the optimal control problem was recast as an entropy minimization problem and the known expressions were reproduced.

In addition Saridis, working on the problem of reducing analytically the increase of entropy generated by human intervention in global systems, added an extra term to the equation of work producing systems, and used optimal control to minimize the effect of the resulting entropy. The combination of the two approaches produces an analytic method of reducing the part of global entropy due to human intervention with the environment.

4. THE PROBABILISTIC VIEW OF THE WORLD

A formal presentation will be made, of the probabilistic approach developed from an entropy point of view, and thus present a method to minimize its effect to our cultural systems. This theory has in addition to the practical applications, a philosophical foundation that has implications to the quality of life and the future of our planet. Experimental results are due after the collection of data from environmental systems.

Global Entropy, associated with irreversible phenomena, appears when we consume energy in order to accomplish some work in our environment we simultaneously create a low quality residual energy, that irreversibly reduces the quality of the environment and leads to a chaotic situation. An infinite number of paradigms exist in our environment, starting with the pollution of the air and the water resources, increase of the waste areas, traffic congestion, financial disasters, unemployment with the resulting crime, and in general the decay of the life-sustaining resources of mankind (Prigogine 1980).

Entropy in our cultural environment has been introduced through our modern technologies, as an energy producing work, like the latest major improvements in the average quality of life. These are producing major increases in the production of waste, traffic congestion, biological pollution and in general environmental decay, which can be interpreted as the increase of the Global Entropy of our planet, an energy that tends to deteriorate the quality of our present environment. According to the second law of thermodynamics this is an irreversible phenomenon, and nothing can be done to eliminate it. Brooks and Wiley suggested entropy in Biological Evolution (1988), while Rifkin used entropy as the measure of decay (1989), and Faber et. al. (1995) propose Entropy for Economic systems. However, other major cultural areas of thought, like societal systems, religion, legal and governmental theories have not kept up with the technological achievements, and therefore are missing in benefits due to the lack of appropriate models for their study. The theory of Entropy, if introduced properly, in addition to its practical applications, has a philosophical foundation that has implications to the quality of life and the future of our planet.

There exists a huge literature with analytic formulations of the problems that concern modern societies, like ecology, environment, biochemical systems, econometrics, and other applications. A good summary of those systems can be found in Singh's **Systems and Control Encyclopedia** (Singh 1987) which served as a source for the development of the material associated with the application of Optimal Control for the reduction of the Global Entropy generated by the work produced by humans in order to improve the quality of life.

The analytic models considered here are for;

Art
Biological Systems
Biochemical Systems
Ecosystems
Engineering Systems and Manufacturing
Environment and Pollution
Phenomena in Modern Science: The Universe
Religion
Socioeconometric and Political Systems
Societal Systems

In most of these cases, the models have been modified to introduce the human effort as a random control variable to be optimized. Such an approach presents as

a solution the possible improvement of the problem of decay. These models may not be the most general or the most popular ones, since controversies exist among the various researchers, however they are representative to demonstrate the idea of improving the quality of our world by reducing the entropy generated by the work produced.

5. "CHAOS" AND THE WORLD DYNAMICS AWAY FROM THE EQUILIBRIUM.

Assuming the analytic model for human activities, the theory of **Chaos**, or the dynamic behavior of nonlinear dynamic systems away from the equilibrium, developed by Pigogine for biosystems originally, came to assist modern scientists to expand the analytic concepts to other cultural phenomena.

From their primitive years, humans have tried to understand and formalize the world around them, through the sensors that nature provided them. To do this they have used various models to represent "approximations" of the functions of the world. They separated those functions into two categories governed by:

1. The Physical Laws of Nature describing deterministic physical phenomena, and
2. The Behavioral Laws describing non-deterministic organic, environmental and societal phenomena.

For the first class, linear mathematics proved to be a concise methodology to approximately describe the time-reversible results of physical experiments near their equilibrium. Logical (Aristotelian) analysis and statistical exhaustive search, were the methodologies used to classify and study the evolutionary behavior of living organisms, environmental and ecological changes and societal phenomena that demonstrated time-irreversibility known as the arrow of time (Brooks and Wiley 1988, Prigogine 1989).

In the recent years, with the progress of the mathematical science and the development of digital computers, probabilistic and stochastic methods and analytic logic are replacing statistical aggregation and classical logical analysis in the realm of mathematics to describe the world's phenomena. Linear analytic models were assumed to be sufficiently accurate to represent useful models of this world, as viewed by human senses. "Reductionism" that has been a powerful tool to analyze and predict physical phenomena, was promptly extended to cover natural phenomena for description and prediction of their behavior. The deterministic model evolved on the principles of the Newtonian mechanics.

However, there were cracks in this wonderful and supposedly airtight, reasoning system. Physical discrepancies and analytic paradoxes marred the perfect models that the world thought infallible. One of the major difficulties encountered was the gap between Newtonian mechanics and thermodynamics. Scientists discovered that heat was produced by the collision of millions of particles in a perfect gas, generating irreversibly entropy, a lower level of energy. However, Poincare showed, that it is practically impossible to study the motion of more than three bodies and thus understand the process. Boltzmann(1872) bridged this gap by introducing statistical methods to describe kinetic phenomena and equate their average kinetic energy with entropy. This pioneer work showed a way to model uncertain and complex physical phenomena in continuous time and connected them to irreversible evolutionary models described by Darwin (Brooks and Wiley 1988), and Shannon (1963) followed with his celebrated **information theory**.

Prigogine (1989) after observations on biochemical phenomena, studied the behavior of dynamic systems away from their equilibrium, a procedure which led to their nonlinear behavior. Jump phenomena led to possible nonunique alternate situations that would converge to a static equilibrium with linear behavior. Therefore, it explained the unrealistic transition from order to disorder as interpreted by the disciples of determinism. This originated the theory of **Chaos** proposed by Prigogine.

Saridis (1995), in the meanwhile, using Jaynes' principle (1957), formulated the engineering design problem as a problem with uncertainty, since the designer does not know a priori the outcome of his design. Entropy was used as the measure of the energy expressing the cost of the irreversible associated process. Considering control as the work and entropy as its cost, the optimal control problem was recast as an entropy minimization problem and the known expressions were reproduced. The cost of the reliability of the design was also expressed as entropy, and was considered as a natural extension of the proposed theoretical development.

Major problems regarding the completeness, consistency and decidability of a statement in a discrete event space, arose with Gödel's theorem of undecidable statements that limited the use of digital computers for the analytic solution of complex problems. Such problems existed with the Diophantine equations and other paradoxes but they were swept

under the rug, so that they would not challenge the power of the computer. Such problems were remedied by introducing new quasi-statistical engines like artificial intelligence, fuzzy set theory and other such techniques. All those problems were blamed to the inadequacy of linear models, and the complexity of systems operating away from the equilibrium point like most of the biological, environmental and societal systems do. Thus, the theory of Chaos was introduced by Prigogine (1996) to study and analyze such cases. The benefit of these discoveries was that complexity and undecidability brought all these problems together and global formulation of their solution was sought. Uncertainty, which is indirectly associated with time irreversibility, was the common element representing the doubt of the outcome of such systems, and stochastic approaches were introduced which have entropy as a common measure.

Thus, the uncertainty of modeling of complex systems is the reason of introducing entropy, in Shannon's sense, as a measure of quality of large complex continuous or discrete event systems. In essence, since entropy is energy irreversibly accumulated when work is performed, and originally introduced in thermodynamics, it is generalized to any kind of dynamic system appearing in nature. Therefore, entropy measures the waste produced when work is done for the improvement of the quality of human life, the struggle of the species in an ecosystem, the biological reactions of a living organism, even the politics of in a societal system. Entropy assumes a stochastic model with uncertain outcome, which is suitable to describe the new complex model of the world.

The theory also gives us a hope that there may be a way out of the accumulation of entropy which may lead us to the **thermal death**.

The question now is if and how this model and the underlining systems, can be improved, by reducing the waste of energy represented by entropy, using analytic methods. Concepts from control theory, used the introduction of a control term in the analytic model, has been proposed to solve this problem.

6. APPLICATIONS

6.1 Biochemical Systems

Prigogine(1980) in his work "**from Being to becoming**" identifies chaotic phenomena in certain biochemical reactions away from their equilibrium. Since these reactions are irreversible he characterized them with lower level Entropy. Experiments like the Belousov Zhabotinskii have established the value of chaotic behavior of chemical reactions.

6.2. Biological Systems

Biological systems produce work through their lifetime. This generates Entropy expressed by the energy of decay of the system. Aging and the deterioration of the human body, diseases and organic decay are typical examples of various forms of Entropy. Medicine has being struggling to reduce their effect by optimizing the duration of human life.

Scientists in Biology stared understanding the models of DNA and are simulating them on digital computers. The resulting models, although primitive are attempting to construct and explain the structure of living organisms. DNA is a map of human heredity. Cloning, with the use of DNA has been used for biological preservation. Transplants are typical examples of an optimal control process. They may represent alternate solutions related to Chaos. This gives us hope of reduction of the biological Entropy.

6.3 Ecosystems

The energy produced by the work of living organisms generates Entropy in the form of decline of reproduction of species and the decay of their environment. This may be measured analytically, by the entropy which expresses the irreversible residue of low level energy. The phenomenon of such devastation is obvious in the environment which is full of waste dumps and barren land where various endangered species used to live. It is really heartbreaking to look at lands, which were full of life and energy to appear barren and desolate. The plants are replaced by waste, the living creatures are gone forever and the ponds have turned into swamps. Wetlands are filled with poisonous waste, while floods take place due to constructions in ravines. There is of course recycling as a control measure of the environment. However, this is only a temporary
solution since the total entropy of the system still increases even at a slower rate.

As mentioned above, Saridis (1998) working on the problem of reducing analytically the increase of entropy generated by human intervention in ecological systems, added an extra term to the equation of ecosystems and used entropy formulation of optimal control to minimize its effect. However, this is only a temporary solution and not an answer to the question.

The problem is devastating since, as they used to say, there is no more room "to go West" to conquer new

land. There is of course the outer space to go and pollute, but this technology is still many years away.

6.4 Entropy and Religion

Religious fanaticism is a scheme that produces pointless work. As a result it generates material destruction, mental agony, pros elitism and contempt to human life all variations of mental Entropy. This phenomenon has existed throughout the centuries as a by-product of human weakness to face the realities of the world.

There is a revived effort by the clergy of all persuasions to attract and proselytize more faithfuls with after life promised rewards. Absolute brainwashing with dogmatic overtones has been demonstrated all over the world, especially in the Moslem religion. Kamikaze bombers have created religious terrorism. The value of human life is thus reduced by the promise of sanctitude. "Religion is the opium of the world" said Karl Marx.

The problem is not a privilege of Islam. Other religions, including Christianity, are encouraging low level emotions. Masses are drawn to hysteria in order to chase away the scare of death. Buddhism is an exception to such a craze.

Unsuccessful attempts to equalize various religions, e.g. Christianity, help to increase the entropy of belief. Could the theory of Chaos if applied, relieve humanity of such a disaster?

6.5 Entropy and the Society.

Modern society is getting more structured and more dynamic. On the other hand, ethical and moral deterioration is getting more pronounced in our present days. A formal presentation of an entropy theory to describe the ethical and moral decay of our times is considered. It is developed from an entropy point of view that relates optimal control theory to the Global Entropy, and thus represents a method to minimize its effect to our society. This theory has in addition to the practical applications, a philosophical foundation that has implications to the quality of life and the future of our planet. Experimental results substantiate this theory.

Since the urbanization has dominated our social dynamics the lowering of the quality of our societies has begun. Migration of the members of the poorer societies to the richer ones have been the cause of this deterioration by introducing lower quality life in congested neighborhoods, and cheaper workforces which lower the Entropy of the system.

The decay of the family ties is characteristic of the modern society, due to busy parents. Their lack of generates loneliness and anxieties to both the parents and their children. Stress appears as a result which leads to drug addiction and suicide of the young people. The multimedia contribute more to the decay of family life.

The newly introduced concept of **globalization**, which intends to decrease the gap between rich and poor societies, may be viewed as a method of equalizing resources of the world thus increasing its total Entropy. Further more it exploits the working class by monopolizing the marketed products.

6.6 Entropy and the Art.

Entropy measures the residue energy of generation of the atrocious modern art of our times. The age of cheap and bad taste for painting, sculpture, literature, clothing and other forms of art has been dominant.

Paintings like the ones produced by a well known modern Greek painter, the 2004 Olympic mascots, sculptures like the moving part ones or made of scrap metals and literature like the trashy memoirs of the so called celebrities of the year, are tasteless products of our times.

Music like "rap" which replaced jazz in the black culture, the screechy singers and the "meowing" bimbos on the stage are another example of the lowered quality of art. Meaningless songs are filling the air. Baggy trashy clothes that the modern artists of the stage and fashion show models are wearing are all cheap and tasteless.

Their low quality taste, are **Kitsch** which represents the new form of Entropy in modern Art.

6.7 Environmental and Pollution

The energy of accumulation of pollutants in the environment and its decay is next in the list of entropy generating agents. Poisonous chemicals, non-biodegradable plastic products, nuclear waste, are accumulating on our lands and seas that kill life producing organisms and are emasculating Mother Nature.

Modern technology has greatly contributed to the improvement of the quality of human life. At the same time, it has increased the production of waste, traffic

congestion, biological pollution and in general environmental decay, which can be measured by the increase of the Global Entropy of our planet, an energy that tends to deteriorate the quality of our environment. Accordingly, Rifkin (1989) suggested the use of entropy to measure the production of waste in environmental systems. This concept is utilized in this work in order to manage its component due to human activities.

Entropy was used as the measure of the energy associated with the assumption of irreversibility of the process. The methodology of using Entropy with Automatic Control, produces an analytic method of reducing the part of global entropy due to human intervention with the environment. Again Saridis' approach to minimize their effect through analytic optimization methods is not sufficient to eliminate the problem. As in the ecosystem description one has to think of moving to other planets to find clean environments.

This approach contributes to the introduction of the entropy approach of optimal control theory, to environmental systems, to effectively restrain the growth of the Global Entropy, created by the human intervention with the environment.

6.8. Manufacturing and Engineering Systems

The evolution of the digital computer in the last thirty years has made possible to develop fully automated systems that successfully perform human dominated functions in industrial space, generating waste interpreted as entropy with automation as a major factor in modern technological developments. It is aimed at replacing human labor in

a. Hazardous environments,
b. Tedious jobs,
c. Inaccessible remote locations and
d. Unfriendly environments.

Automation possesses the following merits in our technological society: reliability, reproducibility, precision, independence of human fatigue and labor laws, and reduced cost of high production (Valavanis, Saridis 1992), with minimal human supervision, leaving humans to perform higher level jobs.

Manufacturing on the other hand, dedicated to make or process a finished product through a large scale industrial operation. is an integral part of the industrial process. In order to improve profitability, modern manufacturing, which is still a disciplined art, involves some kind of automation. Going all the way and fully automating manufacturing is the dream of every industrial engineer. However, as a work producing process, it is generating Entropy producing environmental pollution, loss of manual jobs, and marketing.

The National Research Council reacted to these problems by proposing among other items a new discipline called: **Intelligent Manufacturing** (The Comprehensive Edge 1989).

Intelligent Manufacturing is the process that utilizes Intelligent Control, with entropy as a measure, in order to accomplish reduction of entropy. It possesses several degrees of autonomy, by demonstrating (machine) intelligence to make crucial decisions during the process. Such decisions involve scheduling, prioritization, machine selection, product flow optimization, etc., in order to expedite production and improve profitability and creating non-recyclable products which contain entropy. A case study of Intelligent Manufacturing dealing with a nuclear plant may be found in (Valavanis, Saridis 1992).

At the present time the application of such technology, even though cost-effective in competitive manufacturing, is faced with significant barriers due to (The Comprehensive Edge 1989);

a. Inflexible organizations
b. Inadequate available technology
c. Lack of appreciation and
d. Inappropriate performance measures

However, globalization and international competition, and the need for more reliable precisely reproducible products is directing modern manufacturing towards more sophistication with the generation of more irreversible energy.

Automated multiple product scheduling is needed when the factory produces more than one product on the same set of stations and the ordering of production must be set as a minimum operating cost scheduling problem. The problem is mathematically formulated to set the order of production using entropy as a measure in the Intelligent Control's three level structure (Varvatsoulakis, Saridis, and Paraskevopoulos 1999). The complete system is able to issue high-level task plans and use them to control the stations of the factory in the lower level of the hierarchy. The system includes a learning algorithm designed to obtain asymptotically optimal task plans for production control in uncertain environments.

6.9 Socio-economic and Political Systems, Globalization

Our societies have developed econometric (analytic) systems to measure the conditions of the economy. A typical document is the book by Faber, Niemes and Stephan (1995) that gives analytic examples of the various economic behaviors of markets using entropy as a measure. This approach indicates the accumulation of entropy that leads to disastrous global equalization of the economy.

The modern attitude of global equalization of human resources named **globalization** converges to an equilibrium point of maximum entropy, where no further progress in our planet is possible. All the nations will have the same future without any chance for growth or societal improvement. There are certain advantages in globalization, like equally sharing the wealth the food and the technological achievements of the world. However, total equalization will lead to a lack of progress and global boredom as a result of the accumulated entropy, to say the least.

Governmental corruption, favoritism, and the resulting absolutist behavior create a devastating worldwide situation. Tastelessness in the Arts and Mass Media, Globalization, and Politicalisation are main characteristics.

A characteristic phenomenon of increase of global entropy, similar to the effects of globalization, is the so called "theory of anarchy" and its followers. Their no order theories lead to a disastrous situation of global maximum entropy and the methods used to implement it are catastrophic. In theory these situations represent a total equalization of the world that gradually increases its entropy content.

In modern political ideology many systems are decaying tending to a common denominator. An example is "anarchy" which symbolizes the absolute equality and resembles the entopic thermal death.

6.10 The Sciences and the Universe.

Phenomena in Modern Physics are prominent. Is nuclear energy reversing the Entropy phenomena?

It may represent the energy of deterioration of spatial bodies, or the death of stars according to "The "Big Bang" theory.

The idea of irreversibility as the objective uncertainty of a-priori solution was introduced in the designer's problem; it was reintroduced when considering that control produces useful work on a system, which generates the cost of performance as entropy (Saridis 1985). This irreversibility is interpreted by considering that, when the cost of performance is paid, the system cannot be recovered. This property can be witnessed by visiting the junkyards of old automobiles: new cost must be paid to recover the metal of the old wrecks or in the construction of Space Stations, where the energy of space transportation cannot be recovered.

7. SHOULD WE BE AFRAID OF "CHAOS"

Entropy, as a philosophical device, gives a very pessimistic outlook for the future of our world. It represents energy and should be considered as such. **Thermal death** in all its aspects, including total equalization of our society, ecology, economy and technology, tastelessness in art, and global boredom are its characteristic predictions that the arrow of time points at.

Should there be a possible cure of the problem? Is death the end of the line? Should we hope that that a supernatural deity will give continuity to the live of our universe? Even though the arrow of time points forward, the theory of chaos provides new situations, which gives hope for alternatives than the thermal death and the end of the world.

Life has always been based on a differential among its elements and it is necessary for its existence. Chaos, which considers points away from the equilibrium, represents changes in behavior and therefore a differential in activities. Therefore, it is a device against the equilibrium and the thermal death. It gives hopes for survival.

A typical example is Darwin's theory of evolution of the species (Brooks and Wiley, 1988), where biological bifurcation of the genetic chains may serve as a case of the theory of Chaos. Another example is the perpetual energy emanating from the sun due to nuclear reactions. They both represent the defeat of thermal death and a hope for the continuity of the future of our world.

Is there a conflict of the concept of entropy with God? I do not believe so since it provides alternate solutions that we witness in life. Chaos expresses an optimistic answer to this question.

8. CONCLUSIONS

The concept of Entropy creates a pessimistic view for the future of our universe. The equalization of all kinds of sources of activities is leading to the equivalent of

thermal death and universal boredom of our world. This, according to modern thoughts (Prigogine 1996), may be due that sciences were recently considering world phenomena only close to the equilibrium. An excursion away from it, which has been developed currently (Saridis 1995) promises changes of this image. The theory of Chaos creates some hopes to reverse the catastrophe. The possibility of colonization of other planets may be the needed answer. The first steps in that direction have already been taken.

9. REFERENCES

L Boltzmann, (1872) "Further Studies on Thermal Equilibrium between Gas Molecules" *Wien Ber.,* **66**, p. 275.

W. L. Brogan, (1974) ***Modern Control Theory*** Quantum Publishers New York N.Y.

D. R. Brooks and E. O. Wiley, (1988) ***Evolution as Entropy*** University of Chicago Press, Chicago Il.

P. Coveney, and R. Highfield, (1990) ***The Arrow of Time***, Fawcett Colombine New York N.Y.

M. Faber, H. Niemes, G. Stephan, (1995), ***Entropy, Environment and Resources*** Springer Verlag, Berlin Germany.

Jaynes, (1957) "Information Theory and Statistical Mechanics", ***Phys. Review*****, Vol.** 4, p.106.

Ilya Prigogine,(1980), ***From Being to Becoming*** , Freeman and Company, San Francisco.

Ilya Prigogine,(1996), ***La Fin des Certitudes*** Editions Odile Jacob, Paris France.

Jeremy Rifkin (1989) ***Entropy into the Greenhouse World*** Bantam Books New York.

G. N. Saridis, (1995) ***Stochastic Processeses, Simulation, and Control the Entropy Approach,*** John Wiley and Sons, ew York.

Saridis, G. N., (1996), "Architectures for Intelligent Controls" *Chapter 6*, in ***Intelligent Control Systems***, M. M. Gupta, N. K. Singh (eds) IEEE Press New York NY.

G. N. Saridis (2001) ***Entropy in Control Engineering***, World Scientific Publishing Co. Singapore.

G. N. Saridis (2001) ***Hierarchical Intelligent Machines,*** World Scientific Publishing Co. Singapore.

Claude Shannon, W. Weaver (1963) ***The Mathematical Theory of Communications***, Aeolian Books, Urbana Ill.

Madan G. Singh, (editor), (1987) ***Systems and Control Encyclopaedia; Theory, Technology, Applications***, Vol. 1-8, Pergamon Press, Oxford UK.

Varvatsoulakis M., Saridis G. N., and Paraskevopoulos P., (1998) "A Model for the Organization Level of Intelligent Machines" ***Proceedings of 1998 International Conference on Robotics and Automation***, Leuven Belgium May 15.

The Comprehensive Edge: Research Priorities for U.S. Manufacturing (1989) Report of the National Research Council on U.S. Manufacturing, National Academy Press.

ISSUES IN STOCHASTIC SEARCH AND OPTIMIZATION

PerMIS 2004
NIST

James C. Spall

The Johns Hopkins University

Applied Physics Laboratory (JHU/APL)

james.spall@jhuapl.edu

Performance Metrics and Optimization

- How are performance metrics used?
  - Sensitivity studies
  - System design
  - Decision aid for strategic planning
  - Adapting system over time
  - Detecting instability; avoiding unstable performance
  - Evaluating system reliability
  - Design of experiments
  - Mathematical modeling and parameter estimation
  - And ***on and on***….
- Most of above involve optimization
- **Claim:** Impossible to have a performance metrics conference w/o ***seriously*** considering optimization!

Search and Optimization Algorithms as Part of Problem Solving

- There exist many deterministic and stochastic algorithms
- Algorithms are ***part*** of the broader solution
- Need clear understanding of problem structure, constraints, data characteristics, political and social context, limits of algorithms, etc.
- "Imagine how much money could be saved if truly appropriate techniques were applied that go beyond simple linear programming." (Z. Michalewicz and D. Fogel, 2000)
  - Deeper understanding required to provide truly appropriate solutions; COTS usually not enough!
- Many (most?) real-world implementations involve stochastic effects

Potpourri of Problems Using Stochastic Search and Optimization

- Minimize the costs of shipping from production facilities to warehouses
- Maximize the probability of detecting an incoming warhead (vs. decoy) in a missile defense system
- Place sensors in manner to maximize useful information
- Determine the times to administer a sequence of drugs for maximum therapeutic effect
- Find the best red-yellow-green signal timings in an urban traffic network
- Determine the best schedule for use of laboratory facilities to serve an organization's overall interests

Two Fundamental Problems of Interest

- Let Θ be the domain of allowable values for a vector θ
- θ represents a vector of "adjustables"
  - θ may be continuous or discrete (or both)
- Two fundamental problems of interest:

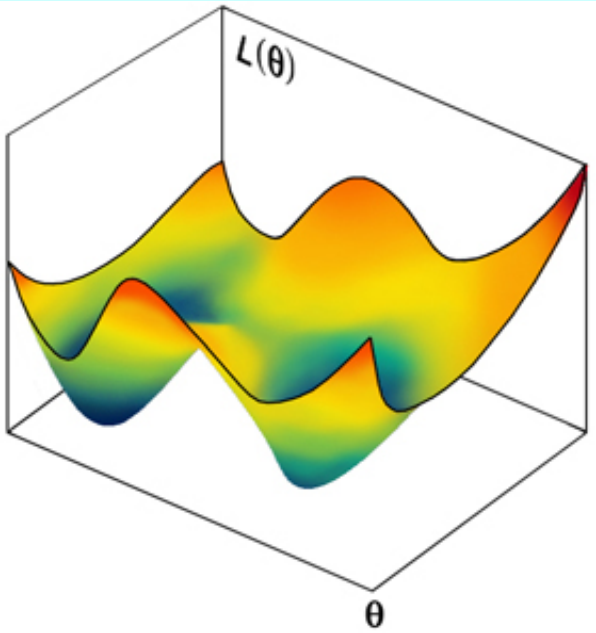
Problem 1. Find the value(s) of a vector $\theta \in \Theta$ that minimize a scalar-valued *loss function* $L(\theta)$

— or —

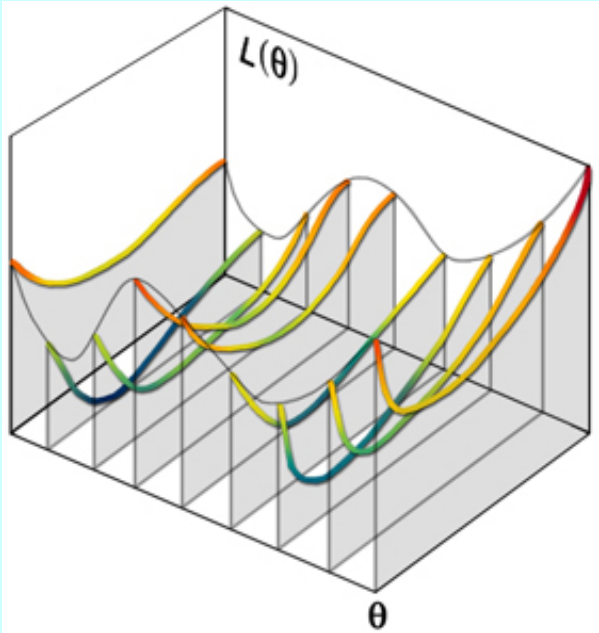
Problem 2. Find the value(s) of $\theta \in \Theta$ that solve the equation $\boldsymbol{g}(\theta) = \mathbf{0}$ for some vector-valued function $\boldsymbol{g}(\theta)$

- Frequently (but not necessarily) $\boldsymbol{g}(\theta) = \partial L(\theta)/\partial\theta$

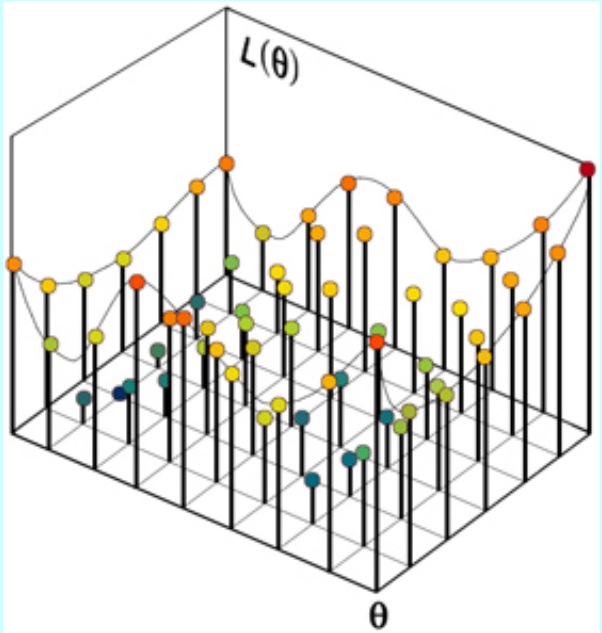
Three Common Types of Loss Functions



Continuous



Discrete/
Continuous



Discrete

Stochastic Search and Optimization

- Focus here is on ***stochastic*** search and optimization:

A. Random noise in input information (e.g., noisy measurements of $L(\theta)$)

— and/or —

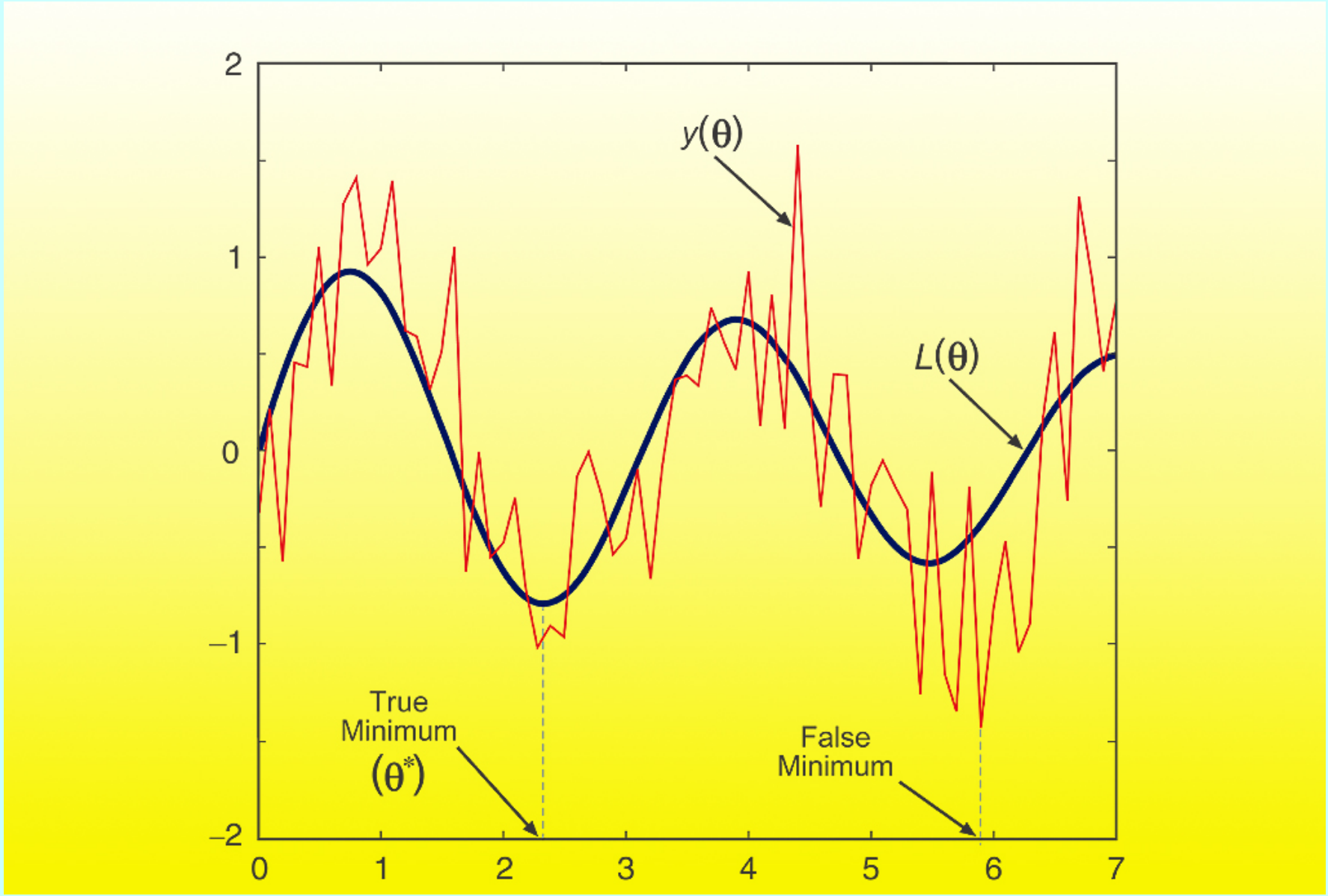
B. Injected randomness (Monte Carlo) in choice of algorithm iteration magnitude/direction

- Contrasts with deterministic methods
  - E.g., steepest descent, Newton-Raphson, etc.
  - Assume perfect information about $L(\theta)$ (and its gradients)
  - Search magnitude/direction deterministic at each iteration
- Injected randomness (B) in search magnitude/direction can offer benefits in efficiency and robustness
  - E.g., Capabilities for global (vs. local) optimization

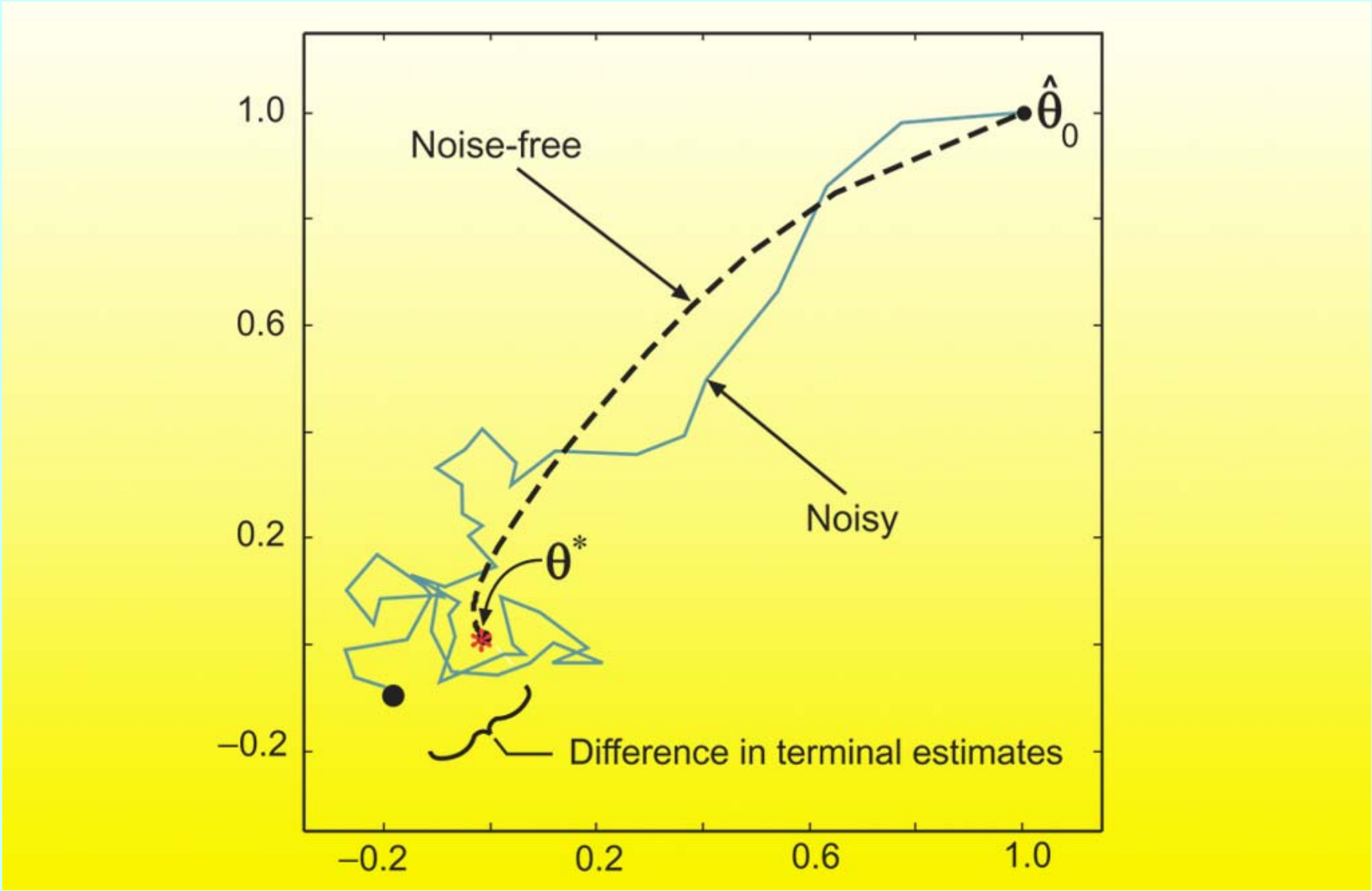
Some Popular Stochastic Search and Optimization Techniques

- Random search
- Stochastic approximation
  - Robbins-Monro and Kiefer-Wolfowitz
  - SPSA
  - NN backpropagation
  - Infinitesimal perturbation analysis
  - Recursive least squares
  - Many others
- Simulated annealing
- Genetic algorithms
- Evolutionary programs and strategies
- Reinforcement learning
- Markov chain Monte Carlo (MCMC)
- Etc.

Effects of Noise on Simple Optimization Problem



Example Search Path (2 variables): Steepest Descent with Noisy and Noise-Free Input



Example of Noisy Loss Measurements: Tracking Problem

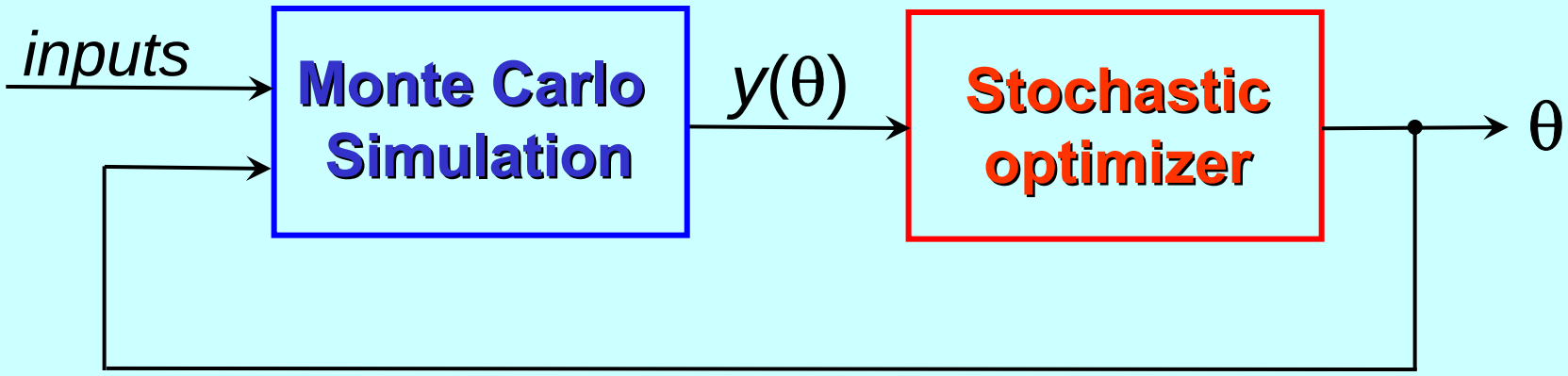
- Consider tracking problem where controller and/or system depend on design parameters θ
  - E.g.: Missile guidance, robot arm manipulation, attaining macroeconomic target values, etc.
- Aim is to pick θ to minimize mean-squared error (MSE):

$$L(\theta) = E\left(\left\|\text{actual output} - \text{desired output}\right\|^2\right)$$

- In general nonlinear and/or non-Gaussian systems, ***not possible*** to compute $L(\theta)$
- Get ***observed*** squared error $y(\theta) \equiv \|\cdot\|^2$ by running system
- Note that $y(\theta) = \|\cdot\|^2 = L(\theta) + \text{noise}$
  - Values of $y(\theta)$, not $L(\theta)$, used in optimization of θ

Example of Noisy Loss Measurements: Simulation-Based Optimization

- Have credible Monte Carlo simulation of real system
- Parameters θ in simulation have physical meaning in system
  - E.g.: θ is machine locations in plant layout, timing settings in traffic control, resource allocation in military operations, etc.
- Run simulation to determine best θ for use in ***real system***
- Want to minimize ***average*** measure of performance $L(\theta)$
  - Let $y(\theta)$ represent ***one*** simulation output ($y(\theta) = L(\theta)$ + noise)



Some Key Properties in Implementation and Evaluation of Stochastic Algorithms

- Algorithm comparisons via number of evaluations of $L(\theta)$ or $\boldsymbol{g}(\theta)$ (not iterations)
  - Function evaluations typically represent major cost
- Curse of dimensionality
  - E.g.: If $\dim(\theta) = 10$, each element of θ can take on 10 values. Take 10,000 random samples: Prob(finding one of 500 best θ) = 0.0005
  - Above example would be even ***much harder*** with only noisy function measurements
- Constraints
- Limits of numerical comparisons
  - Avoid broad claims based on numerical studies
  - Best to combine theory ***and*** numerical analysis

Global vs. Local Solutions

- Global methods ***tend*** to have following characteristics:
  - Inefficient, especially for high-dimensional θ
  - Relatively difficult to use (e.g., require very careful selection of algorithm coefficients)
  - Shaky theoretical foundation for global convergence
- Much "hype" with many methods (genetic algorithm [GA] software advertisements):
  - **"…can handle the most complex problems, including problems unsolvable by any other method."**
  - **"…uses GAs to solve *any* optimization problem!"**
- But there are ***some*** mathematically sound methods
  - E.g., ***restricted settings*** for GAs, simulated annealing, and SPSA

No Free Lunch Theorems

- Wolpert and Macready (1997) establish several “No Free Lunch” (NFL) Theorems for optimization
- NFL Theorems apply to settings where parameter set Λ and set of loss function values are finite, discrete sets
  - Relevant for continuous θ problem when considering digital computer implementation
  - Results are valid for deterministic and stochastic settings
- Number of optimization problems—mappings from Λ to set of loss values—is finite
- NFL Theorems state, in essence, that no one search algorithm is “best” for all problems

No Free Lunch Theorems—Basic Formulation

- Suppose that

N_θ = number of values of θ

N_L = number of values of loss function

- Then

$(N_L)^{N_\theta}$ = number of loss functions

- There is a finite (but possibly huge) number of loss functions
- Basic form of NFL considers average performance over all loss functions

Illustration of No Free Lunch Theorems
(Example 1.7 in *ISSO*)

- Three values of θ, two outcomes for noise free loss *L*
  - Eight possible mappings, hence eight optimization problems
- Mean loss across all problems is same regardless of θ; entries 1 or 2 in table below represent two possible *L* outcomes

Map / θ	1	2	3	4	5	6	7	8
θ_1	1	1	1	2	2	2	1	2
θ_2	1	1	2	1	1	2	2	2
θ_3	1	2	2	1	2	1	1	2

No Free Lunch Theorems (cont'd)

- NFL Theorems state, in essence:

> ***Averaging (uniformly) over all possible problems (loss functions L), all algorithms perform equally well***

- In particular, if algorithm 1 performs better than algorithm 2 over some set of problems, then algorithm 2 performs better than algorithm 1 on another set of problems

> ***Overall relative efficiency of two algorithms cannot be inferred from a few sample problems***

- NFL theorems say nothing about ***specific*** algorithms on ***specific*** problems

Relative Convergence Rates of Deterministic and Stochastic Optimization

- Theoretical analysis based on convergence rates of iterates $\hat{\theta}_k$, where k is iteration counter
- Let θ^* represent optimal value of θ
- For ***deterministic*** optimization, a standard rate result is:

$$\left\|\hat{\theta}_k - \theta^*\right\| = O(c^k),\ 0 < c < 1$$

- Corresponding rate with ***noisy measurements***

$$\left\|\hat{\theta}_k - \theta^*\right\| = O\left(\frac{1}{k^\lambda}\right),\ 0 < \lambda \leq 1/2$$

- Stochastic rate inherently slower in theory and practice

Concluding Remarks

- Stochastic search and optimization very widely used
  - Handles noise in function evaluations
  - Generally better for global optimization
  - Broader applicability to “non-nice” problems (robustness)
- Some challenges in practical problems
  - Noise dramatically affects convergence
  - Distinguishing global from local minima not generally easy
  - Curse of dimensionality
  - Choosing algorithm “tuning coefficients”
- Rarely sufficient to use theory for standard deterministic methods to characterize stochastic methods
- “No free lunch” theorems are barrier to exaggerated claims of power and efficiency of any specific algorithm
- Algorithms should be implemented in context: **“Better a rough answer to the right question than an exact answer to the wrong one” (Lord Kelvin)**

Selected References on Stochastic Optimization

- Fogel, D. B. (2000), *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (2nd ed.), IEEE Press, Piscataway, NJ.
- Fu, M. C. (2002), "Optimization for Simulation: Theory vs. Practice" (with discussion by S. Andradóttir, P. Glynn, and J. P. Kelly), *INFORMS Journal on Computing*, vol. 14, pp. 192–227.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- Gosavi, A. (2003), *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer, Boston.
- Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.
- Kushner, H. J. and Yin, G. G. (2003), *Stochastic Approximation and Recursive Algorithms and Applications* (2nd ed.), Springer-Verlag, New York.
- Michalewicz, Z. and Fogel, D. B. (2000), *How to Solve It: Modern Heuristics*, Springer-Verlag, New York.
- Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, Hoboken, NJ.
- Zhigljavsky, A. A. (1991), *Theory of Global Random Search*, Kluwer Academic, Boston.

Contact Info. and Related Web Sites

- ***james.spall@jhuapl.edu***
- ***www.jhuapl.edu/SPSA*** (Web site on stochastic approximation algorithm)
- ***www.jhuapl.edu/ISSO*** (Web site on book *Introduction to Stochastic Search and Optimization*)

Comparing Algorithms: Rules of Thumb and an Example

S. Balakirsky, T. R. Kramer

National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

This paper provides twenty rules of thumb for comparing algorithms having the same function, one of which is to be selected for use in an intelligent system. The rules are illuminated by a specific example, the comparison of three algorithms for dealing with the collection of open nodes that is at the heart of the Dijkstra graph search method. For each rule, a description is given of how the rule was applied in building the example.

KEYWORDS: *algorithm, binary tree, compare, Dijkstra, graph, jump search, list, run time, search*

1. INTRODUCTION

To produce intelligent behavior, most intelligent systems include computer implementations of one or more complex algorithms. Often a comparison must be done to support making a choice among algorithms performing the same function.

This paper[1] provides a set of rules of thumb for comparing algorithms having the same function. The rules given here are elementary. Most readers will already know many or all of them. The paper also presents a specific example of comparing algorithms. The example is intended to highlight the utility of following the rules. Our purposes are to help builders of intelligent systems make better decisions among competing algorithms and to encourage caution and thoroughness in comparing algorithms.

Section 2 gives our rules of thumb for comparing algorithms for use in an intelligent system. Section 3 presents the example, comparing algorithms for dealing with open nodes in Dijkstra graph search. Section 4 discusses how the rules were used in the example. Section 5 gives the paper's conclusion.

2. RULES OF THUMB

RULE OF THUMB 1: Compare algorithms both in theory and in practice. Algorithms may be compared using theory (by examining how they work) or using experimentation (by examining the behavior of a system running an implementation of the algorithm). Both methods should be used. Theory directs experimentation, but reality is sufficiently more complex than theory that it is naive to rely on theory alone.

2.1 Comparing Algorithms in Theory

RULE OF THUMB 2: Do a big O analysis. Theoretical analysis of the running time of algorithms is a mainstream activity of computer science. The well-known idea [7], [11] is to identify the "big **O**" order of the time a program will take as a function of the size of the problem being solved. The difficulty of doing this ranges from fairly easy to impossible. A theoretical analysis can never guarantee an algorithm will work well, but if it indicates the algorithm should work badly for large problems (by running in order $\mathbf{2^N}$ time, for example, where **N** quantifies the size of the problem under consideration), one can be quite sure it will work badly. Depending on the application, it may be appropriate to do a worst-case analysis, an average behavior analysis, or both.

If the order of time taken, **T**, is $\mathbf{O(}\boldsymbol{F}\mathbf{(N))}$ for some function **F**, this means, $\mathbf{T = K(}\boldsymbol{F}\mathbf{(N))}$ for some constant **K**. The constant of proportionality, **K**, is constant only for a given computer, compiler, and operating system. If the usual assumption of a one-tier memory model (only RAM) is made, **K** may still fail to be constant. As pointed out in [8], modern computers generally have a multi-level memory hierarchy, including at least cache, RAM, and disk. Cache memory may be up to ten times as fast as RAM, and disk memory may be 1000 times as slow. Thus, **K** will actually be a constant only if the characteristics of memory handling do not change over the range of **N** being tested.

RULE OF THUMB 3: Do a functional analysis. A functional analysis is an analysis of how the algorithm works — what the main routine does, what the subroutines are and what they do, etc. This is a harder, one-off activity, usually requiring understanding of a large body of source code or pseudocode and the language in which it is written. Here again, only negative guarantees are available; if it does not work in theory, it won't work in practice.

RULE OF THUMB 4: Identify dimensions of the domain space. Typically, the problem domain an algorithm addresses will have several independent aspects that affect the functioning of the algorithm. These aspects can be used as the dimensions of a domain space. Some aspects, such as problem size, may be effectively continuous, while others may have continuous segments divided by discontinuities or may be discrete.

RULE OF THUMB 5: Identify regions of interest in the domain space. Also typically, only a limited part of a domain space will be of interest. The portion of interest may be

1. Certain commercial equipment, instruments, or materials are identified in this paper in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

bounded either naturally (by discontinuities in the dimensions of the space), or synthetically (by picking limits along continuous dimensions). It is important to identify the regions of interest because algorithms may, and almost always do, perform qualitatively differently in different parts of the space. In this paper, we use the term "sea change" to mean a qualitative change in the functioning of an algorithm in different parts of the domain space.

RULE OF THUMB 6: In the functional analysis, consider all regions of interest in the domain space. A functional analysis will not be complete unless it identifies (i) what is qualitatively different in different parts of the domain space where the algorithm is intended to work, and (ii) how qualitative differences in the part of the domain space affect the performance of the algorithm.

RULE OF THUMB 7: Get the source code. While natural language and pseudocode versions of algorithms can be analyzed, the analysis is likely to miss key points if it is not done on the source code, the first key point being: does the code implement the algorithm correctly. Without having the code in hand, it cannot be debugged, modified, or recompiled.

2.2 Comparing Algorithms in Practice

Where theoretical analysis is not well-developed, only experimentation is available to compare algorithms. Even where a theoretical analysis of algorithm performance is available, it cannot reveal how an implementation will perform. An abstract algorithm does not run on an abstract machine. An implementation of the algorithm in a specific computer language, compiled by a specific compiler (or interpreted by a specific interpreter) for a specific machine architecture is what runs. And it runs on a specific piece of computer hardware, under control of a specific operating system, possibly depending on a specific file server connected by a specific communications system. Each of these items can affect the performance of an algorithm [6], [8].

RULE OF THUMB 8: Determine what is to be optimized, and test that. This is obvious but ranges from easy to nearly impossible in practice. Algorithms that return an answer guaranteed to be optimal for a well-defined single measure (least cost, for example) provide the easy cases. In hard cases (computer vision has them), it is difficult even to characterize what one is trying to optimize. There may be trade-offs among optimizing several aspects of performance: average performance vs. worst-case performance, minimizing bad results vs. maximizing good ones, minimizing running time vs. maximizing answer quality, falling off a cliff at domain region boundaries vs. sliding down gradually, etc.

RULE OF THUMB 9: In experiments, keep the number of variable factors to a minimum. This is a main tenet of standard experimental procedure, which should be followed insofar as possible. Keep all conditions but one the same between tests, changing only one factor at a time. In particular, when comparing two algorithms, to the extent possible:

1. Implement them in the same computer language.
2. Compile them on the same compiler, using the same compiler settings.
3. Run them on the same computer.
4. Test them under similar conditions of computer usage.

If it is known what computer the algorithm will be running on as part of the complete intelligent system, that is the computer that should be used for testing. Otherwise, the same tests should be repeated on each computer that might be used in the full system.

RULE OF THUMB 10: Conduct one or more series of tests that are ordered by size but have the same values in other dimensions. If problem size is variable in an application, it usually varies continuously. The range of sizes should match the range presented by the regions of interest in the domain space, if that is known. When such a series of tests is conducted, if theory indicates there are no sea changes in system behavior across the range of sizes, test results (such as time) should lie on a smooth curve of some sort when the results are plotted on a graph showing problem size on one axis and results on the other axis. If there is a kink in the curve where theory does not predict a sea change, double-check the theoretical analysis and/or run the same tests on a different computer. If the kink is not found on a second computer, the cause of the kink lies somewhere in the first computer.

RULE OF THUMB 11: Conduct identical tests on substantially different computers. If the same tests can be run on substantially different computers, and the ratio of the times taken by the two computers is nearly constant across tests, this is (mushy) evidence that the system operating regimes are not undergoing sea changes between tests. If one of the computers behaves qualitatively differently between two tests, it is unlikely that the second computer will have a sea change at the same point.

RULE OF THUMB 12: Conduct tests with the computer lightly loaded. The two most significant measures of load on the computer are memory usage and CPU usage.

If RAM memory is not lightly loaded, the operating system will use secondary memory with the effect, according to [8], that "memory access times can vary by factors as large as a million" in the worst case. The same paper observes that because of memory effects, "[predicted] running times that are off by three orders of magnitude are not unusual."

Reduce the number of processes competing for time as far as possible while testing. For example, do not run two tests simultaneously. Ideally, while many processes will inevitably be resident on the test computer, only the process being tested should use any significant amount of CPU time. The Unix "time" command, for example, shows the percentage of CPU time used by the process that was timed. If this command is used for timing, use results only for those runs for which CPU usage is near 100%.

RULE OF THUMB 13: Monitor computer use during testing. Memory usage per process on Sun computers[1] and other computers using unix-like operating systems may be checked with **top** or **ps**. **Top** also shows total memory usage.

1. In this paper, "Sun computer" means a Sun computer running the Solaris operating system (which is unix-like).

Memory usage per process and in total on PC's[1] may be checked using the Windows Task Manager.

CPU usage for a process may be checked on computers using unix-like operating systems with the **top** and **time** commands. **Top** also shows total CPU usage. On PC's, a performance meter is available inside the Windows Task Manager (and in a stand-alone process) that will show total CPU usage. On PC's the Windows Task Manager will show CPU usage per process.

Monitoring processes are designed to use minimal computer resources, so having them running while testing will probably not interfere with testing. Some monitoring processes (**top**, for example) show what resources they themselves are using. If it is suspected that a monitoring process is using significant resources, run a timing test with the monitor on, then run the same test with the monitor off and compare results.

RULE OF THUMB 14: Conduct tests with the computer realistically loaded. Conducting tests with the computer lightly loaded should always be done, but a light load may not be possible in the full system. If the load on a computer running the full system is known, conduct tests under those load conditions. Where performance changes dramatically between lightly loaded and realistically loaded conditions, consider reconfiguring the full system.

RULE OF THUMB 15: Understand the effects of the test harness and compensate for them. The algorithm being tested will have interfaces to the rest of the full system and may require data structures to exist. To test without the full system, a test harness is built, typically in the form of computer code for a driver including a main routine and routines to set up data. If the full system normally builds data structures while it runs, as opposed to building them by reading a file, it may be necessary to define a file format and have file-reading code in the driver. When tests are run using the harness, some time will be used by the driver code. If time to do the same functions is not required by the full system, that time should be diluted or subtracted in analyzing test results.

Typically, a test harness will consume a significant amount of time primarily when it starts up (for file reading, for example), and possibly when it shuts down.

RULE OF THUMB 16: Use representative test cases. The test cases used should be typical of the region of domain space for which an algorithm is being tested. If possible, harvest test cases from data acquired during intelligent system operation. It is very rare for a test case generator to exist that is guaranteed to produce test cases that satisfy some metric for representativeness. Producing representative test cases typically requires both establishing criteria for representativeness and conducting secondary experiments on candidate test cases to see how well they meet the criteria.

RULE OF THUMB 17: Use standard test case sets, if available and appropriate. In some domains, standard sets of test cases may be available. Even where these are available, care is needed in deciding if they are really in the proper domain region and, if so, whether they are adequately representative.

1. In this paper, "PC" means a PC running an MS Windows operating system.

A common pitfall with test cases is that an algorithm is tuned for a specific set of test cases that are not adequately representative of the domain region. When further testing is done or the full system is built and run, performance is significantly lower than expected. Where no good metric for representativeness exists, the only solution for this problem may be to use very large sets of test cases.

RULE OF THUMB 18: Collect secondary data showing what the algorithm is doing internally. This is useful for:

1. verifying that data which should be the same between two algorithms or implementations is the same.
2. verifying that an algorithm is doing in practice what it should be doing in theory.
3. tuning the algorithm for better performance.
4. understanding where the algorithm is spending the majority of its time.

This is at the border between testing and debugging. If an algorithm is not doing what the tester thinks it should be doing, either the tester is confused or there is a bug in the code or the algorithm itself.

RULE OF THUMB 19: Deal with measurement error. Another obvious rule. This is like "check the hull for leaks" in that checking any one spot is easy, but there may be a lot to check, and if you miss one spot, the boat sinks. Deciding between algorithms does not usually require high precision or high accuracy; getting measurements within 10% to 20% overall is probably good enough. Where components of error are additive, of course, the error in an individual component needs to be lower.

Quick checks should be applied to measuring tools such as the **time** or **top** command or a performance monitor. Things that measure time can be double-checked against clocks or watches. Where more than one tool is available, use both and compare. For example, CPU usage on a Sun computer is given both by **time** and **top**.

Much of the data taken by computer tools such as **top** and performance monitors is an average over some time period, and one should be aware of this when using those tools. For example, performance monitor plots may show CPU usage ramping up over a few seconds, staying level for a long time, and then ramping down again. The ramps are almost certainly not real. What is actually happening is a quick jump from low to high when a process starts and a jump back to low when the process ends. The ramps (which help the eye follow the curve) are artifacts of using a rolling time average. The real life span of the process is probably from the beginning of the ramp up to the beginning of the ramp down.

Performing repeatability tests is very useful. The total variation in a repeated test contains at least three components, each of which has random and systematic parts:

1. variation caused by differences in the internal computer environment between tests.
2. errors in the tools used to take the measurements.
3. errors in reading the measurements provided by the tools.

Although equal and opposite random errors are possible in theory, they are unlikely to occur repeatedly in practice. If the variation between repeated measurements is small in every

case, it is nearly certain that all three components of random error are small. If possible, adjust test procedures so that variations in repeated tests are small compared with the quantity being measured.

RULE OF THUMB 20: At widely separated times, repeat tests performed earlier. Repeating old tests will help catch systematic errors that vary slowly over time.

3. Example

Many systems may use a graph search algorithm. In building planning systems for domains as diverse as autonomous vehicles [3] and automated atom assembly, researchers at NIST have been using Dijkstra graph search [5]. This finds a least cost path (if there is any path) between any two nodes in a directed graph. In previously reported work [4], we compared three implementations of the Dijkstra algorithm. We were aware that even the best of these three (which we will call **List**) could be improved by implementing faster methods of using the collection of open nodes that is at the heart of Dijkstra graph search. **List** maintains the open nodes in a linked list arranged in increasing cost order. **List** uses linear search for removing and reinserting nodes whose cost changes. We implemented two algorithms embodying more efficient methods of dealing with the open nodes. The first of these also keeps the open nodes in a list but is a form of jump search [10] that overlays the list with more structure; the system that uses it we call **Tabs**. The second uses a type of binary tree for the open nodes [1], [2], [12], and the system using it we call **AVL**. We ran a series of tests on **List**, **Tabs**, and **AVL** and compared them. This paper uses that comparison as an example. Test data and descriptions of the tests are given here. Details of the algorithms are given in a separate, not yet published paper.

The three implementations all produce the same results in theory, and the results are guaranteed to be optimal (measured by least cost). Theoretical analyses of average time for removal and insertion operations on the collection of open nodes are straightforward. They show average times of **O(M)** for **List**, **O(sqrt(M))** for **Tabs**, and **O(log(M))** for **AVL**, where **M** is the size of the open nodes collection. As discussed in [9], however, the theoretical average time of graph search is generally computable (with difficulty if at all) only for well-characterized graphs over which the average is to be taken. In order to apply the average time equations just given, we need to know how **M** varies during operation, on the average, for a given number of nodes **N**, and this depends heavily on the characteristics of the test graphs. The test graphs we used are well-characterized as follows, but we do not have average time equations for node-to-node searches in graphs with these characteristics.

The sets of test graphs have the following characteristics:

1. In each set, each graph has twice as many nodes as the preceding graph.
2. In each set, the number of arcs leaving a node is fixed; one set has 16 arcs from each node, the other 2 arcs from each node.
3. The cost of each arc is a randomly chosen positive integer less than 50.
4. The node at the end of the each arc is randomly chosen, except that the node at the end must differ from the node at the beginning, and for a given beginning node, the end nodes must all differ.
5. Each graph is not necessarily completely connected.

Although we do not have big **O** equations for expected behavior, we knew from our functional analysis of the algorithm that nearly all of the processing time is spent in removing nodes from the open nodes collection and inserting nodes back into this collection. It was also clear that the average **M** gets bigger as **N** gets bigger. Thus, since **M/sqrt(M)** and **M/log(M)** increase rapidly with **M**, we expected **Tabs** and **AVL** to outperform **List** by increasing margins as **N** increases. Since **sqrt(M)/log(M)** increases as **M** increases, we also expected that for sufficiently large **N**, **AVL** would be faster than **Tabs**.

We ran tests first on a Sun Ultra 60 and then on a Pentium 4 class PC. Both computers have 512 megabytes of RAM memory. The PC is at least 5 years newer, so we expected it both to run faster and to do more active memory management.

The results for the Sun are shown for branching factor 16 in Figure 1, and for branching factor 2 in Figure 2. Rather than showing times for the three algorithms, the graphs show the ratio of the time taken by **List** to the time taken by **AVL** and the ratio of the time taken by **List** to the time taken by **Tabs**. The basic reason for using the ratio is because we are comparing algorithms, and using the ratio washes out the effects of using a specific problem, leaving only the effects of the algorithms. Further discussion is given in Section 4.

As shown in the figures, our expectations were correct. **AVL** and **Tabs** both outperform **List** by increasing margins as **N** increases and **AVL** outperforms **Tabs** by an increasing margin when **N** is more than 8000. Smooth curves fit the data very closely, implying that for both sets of test cases, there was no sea change in the behavior of the Sun over the range of graph sizes we used.

Figure 3 shows a comparison of speed on the sun versus speed on the PC. Since the same source code was used on the Sun and the PC, we expected that, in the absence of a sea change in behavior of the PC at some point, the speed ratios would be roughly constant over the range of problem sizes and that the ratio would be about the same for all three systems. Figure 3 shows that this was the case for **AVL** and **Tabs**, but not for **List**. Further discussion is given in Section 4.

From the functional analyses of the three systems, we expected the search time taken by each implementation to be almost proportional to the number of comparisons performed. This is because, except for the main loop, most of the code consists of loops or recursive function calls in which the number of repetitions depends on a comparison, and the other operations that execute during a repetition are always the same for the code segment performing the repetition. **List** has the fewest other operations per comparison, while **AVL** has the most, so we expect comparisons per second to decrease from **List** to **Tabs** to **AVL**. As shown in Table 1 below, the data for branching factor 16 largely bear this out. For each of the three systems and four test cases, the table shows the time taken on the Sun computer to find the answer and the number of comparisons made while finding the answer. The times taken on the PC differed, but the numbers of comparisons did not.

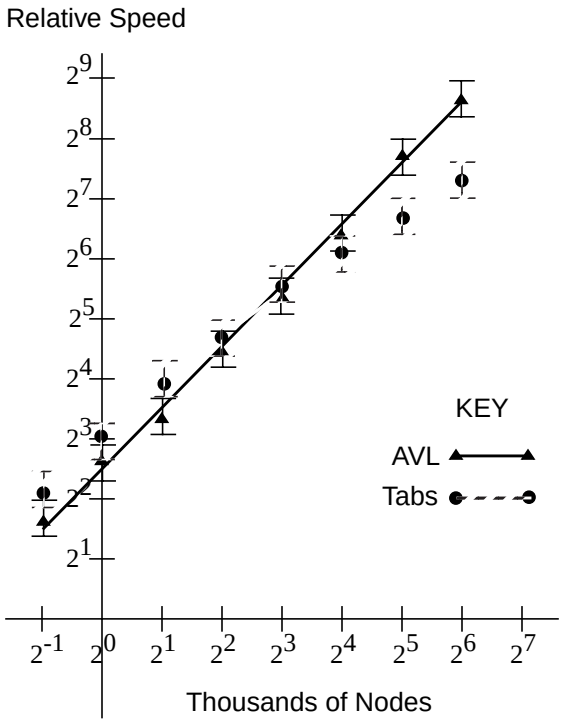


Figure 1. Speed Relative to List for Branching Factor 16, Sun

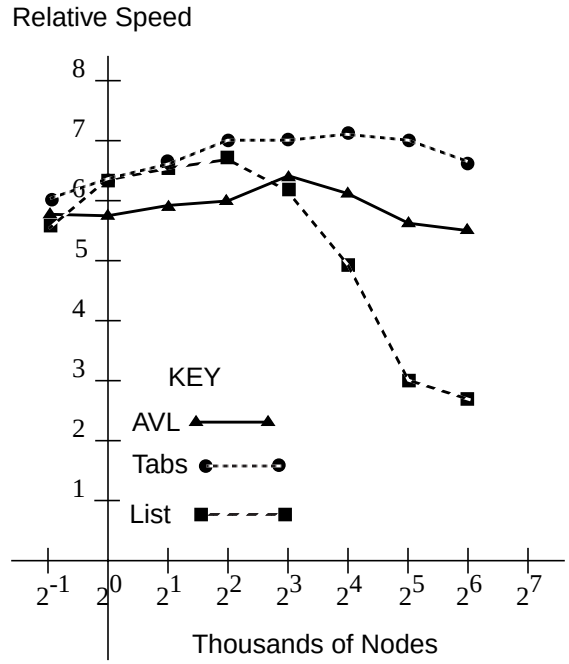


Figure 3. Speed on PC Relative to Speed on Sun, for Branching Factor 16

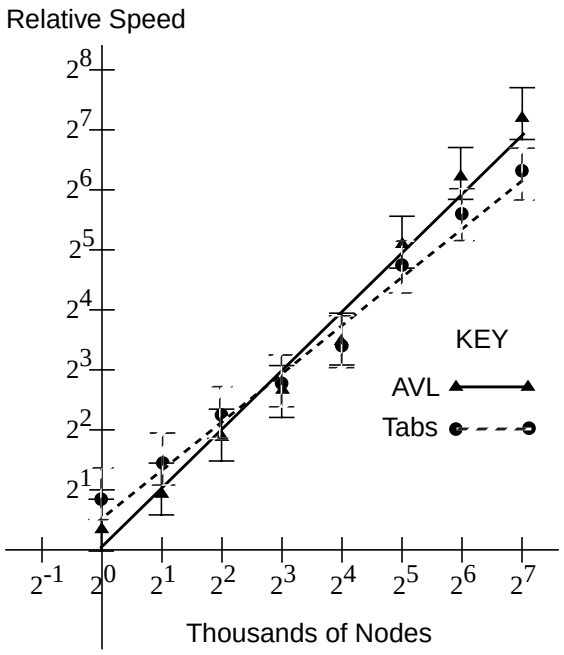


Figure 2. Speed Relative to List for Branching Factor 2, Sun Computer

Table 1: Data for Branching Factor 16, Sun

type↓	nodes →	1000	4000	16000	64000
AVL	seconds	0.0111	0.0605	0.268	1.64
	comps	20,167	108,812	458,585	2,392,759
Tabs	seconds	0.00836	0.0515	0.334	4.56
	comps	34,841	247,648	1,899,736	22,214,824
List	seconds	0.0694	1.31	22.7	693
	comps	523,530	9,338,357	130,440,329	2,417,260,900

4. How the Example Used the Rules

RULE OF THUMB 1: Compare algorithms both in theory and in practice. This was done, as described in Section 3.

RULE OF THUMB 2: Do a big O analysis. This was attempted as described in Section 3.

RULE OF THUMB 3: Do a functional analysis. We studied the functioning of the algorithms for handling the collection of open nodes. The functional analysis did not reveal any characteristics of the algorithms that would be expected to produce sea changes in the regions of graph space of interest.

RULE OF THUMB 4: Identify dimensions of the domain space. We were not focused on a specific application for this work, so we looked at the characteristics of abstract directed graphs (what the Dijkstra algorithm deals with). Following Rule of Thumb 10, we used problem size as one dimension.

In our earlier paper, [4] on testing implementations of the Dijkstra algorithm, the functional analysis indicated that versions that keep the list of open nodes in cost order will differ from versions that do not keep the list in order. This difference appeared when the branching factor of the graph (how many arcs leave each node) was used as a dimension of the domain space. We continued to use the branching factor as a dimension in the tests reported in this paper, even though we did not expect to see significantly different behavior on this dimension.

RULE OF THUMB 5: Identify regions of interest in the domain space. In automated atom assembly, we have dealt with graphs having 62 to 437,582 nodes using Dijkstra search. Other applications have not had as large an upper bound. The limits of our testing were within a somewhat narrower range: 500 to 128,000 nodes.

RULE OF THUMB 6: In the functional analysis, consider all regions of interest in the domain space. As already described, the region of interest was in graph space and was all one piece. It did not have qualitatively different parts.

RULE OF THUMB 7: Get the source code. This was easy, since we wrote the code.

RULE OF THUMB 8: Determine what is to be optimized, and test that. The Dijkstra algorithm does not halt until an optimum (least cost) answer is found. We accepted optimizing cost, as provided by the algorithm in its usual form. Since there is no way to predict the cost of the answer before the search is conducted, it is not possible to stop the search when a 120% optimum cost (or whatever fraction or margin) solution is found. The Dijkstra algorithm could readily be modified either to stop at some point between finding the first answer and finding an optimal answer or to keep going after finding the first optimal answer until all equally optimal answers have been found, but we did not do this.

RULE OF THUMB 9: In experiments, keep the number of variable factors to a minimum. Our tests included three variable factors: the computer (Sun or PC), branching factor of the graph (2 or 16), and problem size (500 to 128,000 nodes). In the range 1000 to 64,000 nodes, the 3-dimensional test space has all data points lying on lines for which two of the three variables are constant. For example, as shown in Figure 1, the computer and the branching factor were held constant while problem size was varied.

To reduce variable factors in the source code, we remodularized the code. We placed the code defining a graph, a node, and a Dijkstra search node in the file dijk.hh and used this file in building each of the three systems. We built the test harnesses for **AVL** and **Tabs** by copying the harness for **List** and changing a very few lines.

RULE OF THUMB 10: Conduct one or more series of tests that are ordered by size but have the same values in other dimensions. Two series of tests in increasing size order are shown in Figure 1 and Figure 2 for the Sun Computer. The same tests were run on a PC and used for building Figure 3.

RULE OF THUMB 11: Conduct identical tests on substantially different computers. We ran the same test cases on both a Sun and a PC. These are known to be substantially different.

Figure 3 shows that for **Tabs** and **AVL**, the ratio of the speed of the PC to the speed of the Sun is almost constant over the range of sizes tested for branching factor 16. **Tabs** runs about 6.6 times as fast on the PC. **AVL** runs about 6.0 times as fast on the PC. For **List**, the ratio is not constant, decreasing by a factor of about 2 over the same range. This appears to indicate a sea change in the behavior of the PC over this range. Since we expected more active manipulation of memory on the PC, but we determined that secondary memory is not being used, we hypothesize that the sea change is caused by a difference in the use of cache memory between the smaller and larger problems. We have not tried to verify this.

If figures similar to Figure 1 and Figure 2 were drawn for the PC, they would not have the nice appearance of those figures because the algorithm comparison would be confounded by the sea change in computer behavior. If we had tested only on the PC, we would have had a very hard time determining whether the algorithm or the computer was responsible.

RULE OF THUMB 12: Conduct tests with the computer lightly loaded. For both Sun and PC, we made sure no processes were running that would compete significantly for system resources. While any test was in progress we did not move the mouse or touch the keyboard. The following observations establish that both computers were always lightly loaded during tests.

RULE OF THUMB 13: Monitor computer use during testing. On the Sun, we monitored with both time and top that the test process was using almost all of the CPU. When testing on the Sun, we recorded results only when CPU usage was at least 98% according to the results provided by the **time** command.

On the PC we used the Task Manager and/or the performance monitor to monitor CPU usage. In all cases, CPU usage was at zero both before and after each test. During tests run on the PC, CPU usage reached a very flat plateau at 50%, as opposed to just under 100% on the Sun. This may indicate that the PC operating system's CPU allocation policy does not allow any process to have more than half the CPU, even if the other half is available. We did not attempt to investigate this further.

The Sun and the PC each had 512 megabytes of RAM memory. On the Sun, the largest any search process became was 13 megabytes. There were always over 100 megabytes of free RAM and no CPU time was spent on swapping. On the PC, the largest any search process became was 22 megabytes. There were always over 200 megabytes of free RAM and no CPU time was spent on swapping.

We checked that none of the monitors uses a significant portion of the CPU.

RULE OF THUMB 14: Conduct tests with the computer realistically loaded. Our tests were not conducted using a complete intelligent system. We did not know what a realistic load would be, so we did not do this.

RULE OF THUMB 15: Understand the effects of the test harness and compensate for them. The test harness for Dijkstra search was a main routine taking four arguments (i) the name of a graph file, (ii) the node number of the start node, (iii) the node number of the goal node, (iv) the number of times to repeat the search. The number of times to repeat the search was selected so as to satisfy the requirement that the total time taken by the test be at least 30 seconds. Preliminary testing of each test case was done to determine a number of repetitions that would meet this requirement but not need more than a few minutes for each test.

A file format for an abstract directed graph was devised, and a file reader was built into the test harness that would read the file and build a directed graph structure. Testing revealed that the smaller files were read in a second or less, but the larger ones required up to 7 seconds on the Sun and up to 18 seconds (for the same file) on the PC. We did not attempt to determine why the PC was slower. The reading time was calculated by running the same test with two different numbers of repetitions and using the equation TotalTime = (ReadTime + (RunTime x repetitions)). The reading time was also observed on the PC performance meter. Where reading times were more than a second, the run times were calculated using the equation just given.

Reading times are out of the scope of comparing the algorithms we were comparing. They are effects of the test harness, and we compensated for them.

The other effect of the test harness we identified was the time taken by the "for" loop that repeats the test. Each time around that loop, each node in the graph is marked UNOPENED, since Dijkstra search requires that marking when it starts up. This is a very small amount of processing compared with the processing done on nodes in Dijkstra search, so we did not compensate for it.

RULE OF THUMB 16: Use representative test cases. The graphs we used were constructed (by a graph-building program we wrote) to produce graphs with the characteristics described earlier. The user of the program specifies number of nodes, the branching factor, and an upper bound on arc cost. In real problems, a constant branching factor seems unlikely. We doubt that the algorithms would have compared differently if we had used variable branching factors, but we have neither an analysis nor experimental data to substantiate that. Since our graph construction methods were not selected to produce graphs similar to those found in any specific application, we cannot claim that the test graphs are representative.

An equally severe problem was picking the start and goal nodes for each test case. Clearly, in each graph we constructed, some pairs of nodes would be connected by a short cheap path while other pairs would be connected only by more expensive longer paths. Finding the least expensive path would be relatively easy for the short cheap paths and relatively hard for the long expensive paths. To get average results, in each graph, we could have randomly selected a set of pairs, repeated the tests for all pairs in the set, and averaged the results. This would have required a much larger amount of testing than we were prepared to do. Instead, for each graph we randomly selected a set of 7 pairs, timed all 7 pairs using **AVL** and only a few repetitions to identify the pair with median time, and used that one pair as the representative for the graph in testing with **List** and **Tabs** and retesting with **AVL**.

This selection procedure still left substantial differences in the relative difficulty of the test cases. When run time was plotted against problem size for the algorithms being tested, the resulting lines were rather jagged. But the three curves zigged and zagged together, implying the zigs and zags were effects of problem difficulty, not the algorithms being tested. This suggested factoring out the difficulty of the representative problem by plotting the ratios of the times, not the actual times, and that is what we did.

For a specific instance of a test case, the time taken can be viewed as the relative difficulty of the test case (compared with the average difficulty) multiplied by the average time taken. If the average time taken is given by $\mathbf{T = \mathit{F}(N)}$, then letting $\mathbf{D_{Case}}$ be the relative difficulty, the equations for time taken for a specific test case are:

$\mathbf{T_{ListCase}} = \mathbf{D_{Case}} \times \mathbf{\mathit{F}_{List}(N)}$,

$\mathbf{T_{TabsCase}} = \mathbf{D_{Case}} \times \mathbf{\mathit{F}_{Tabs}(N)}$, and

$\mathbf{T_{AVLCase}} = \mathbf{D_{Case}} \times \mathbf{\mathit{F}_{AVL}(N)}$.

When the ratio of any two times is taken, the $\mathbf{D_{case}}$ in the numerator cancels with the $\mathbf{D_{case}}$ in the denominator, removing the effect of the difficulty of the case.

RULE OF THUMB 17: Use standard test case sets, if available and appropriate. We are not aware of any standard test case sets for pure graph search using the range of sizes and branching factors we have used.

RULE OF THUMB 18: Collect secondary data showing what the algorithm is doing internally. We collected secondary data for three purposes: (i) to be sure **List**, **AVL**, and **Tabs** were behaving identically where they were supposed to be doing so, (ii) to determine how the Dijkstra algorithm behaves on problems with different sizes and branching factors, (iii) to measure those things that analysis indicated the three algorithms would do differently.

We collected secondary data by adding conditionally compiled code to the source code files for the three systems. Timing tests were conducted with versions of the executable systems compiled without data collection. A subset of the same tests were repeated with versions of the systems compiled with data collection.

To be sure the systems were behaving identically where they should be identical, two files were generated: one listing the node numbers in the order in which they were opened and giving the total number of nodes opened, and the other listing the total number of nodes open each time around the node processing loop of the Dijkstra algorithm. For each test case, files generated on the Sun by the three systems were compared and the data items just mentioned were found to be identical. For a few test cases, similar files were also generated on the PC. All data that should be identical were found to be identical, even the maximum 2,417,260,900 comparisons.

RULE OF THUMB 19: Deal with measurement error. When testing on the PC, a digital watch reading in seconds was used to time the tests. Since this automatically introduces a random error of one second, all tests on the PC were adjusted to run for at least 30 seconds, so that the random error from reading the watch would not be more than about 3% of the time being measured. Timing for the PC was spot-checked using the PC's performance monitor. Timing on the Sun was done with the **time** command; spot checks were done with a digital watch and an analog clock.

RULE OF THUMB 20: At widely separated times, repeat tests performed earlier. This rule was followed throughout testing. No significant increases in variability occurred.

5. Conclusion

This paper has presented 20 rules of thumb to follow in comparing algorithms performing the same function that might be used in an intelligent system. An example was provided of how the rules were applied in comparing three algorithms for maintaining the collection of open nodes in Dijkstra search. For some rules, the example described a pitfall that was avoided by having followed the rule. For other rules, the example showed how difficult it can be to follow the rules.

The rules of thumb presented here can certainly be improved, and surely there are other rules it would be helpful to add. The intent of these rules is to help builders of intelligent systems make better decisions among competing algorithms. We hope they will be useful for that purpose.

6. References

[1] Adelson-Velskii, G. M. and Landis, E.M., "An Algorithm for the Organization of Information", *Doklady Akademiia Nauk SSSR*, Vol. 146, No. 2, 1962, pp. 263-266.

[2] Baeza-Yates, R., "Height Balance Distribution of Search Trees", *Information Processing Letters*, Vol. 39, No. 6, 1991, pp. 317-324.

[3] Balakirsky, Stephen B., Lacaze, Alberto, "Value-Driven Behavior Generation for an Autonomous Mobile Ground Robot", in Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, April 1-5, 2002.

[4] Balakirsky, Stephen B., and Kramer, Thomas R., "NOT (Faster Implementation ==> Better Algorithm), A Case Study", in Proceedings of the PerMIS '03 Workshop, NIST Special Publication 1014, 2003.

[5] Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische Mathematik*, Vol. 1, 1959, pp. 269-271.

[6] Hooker, J. N., "Needed: An Empirical Science of Algorithms", *Operations Research*, Vol. 42, No. 2, March-April 1994, pp. 201-212.

[7] Knuth, Donald E., *The Art of Computer Programming, Volume 3 Sorting and Searching*, Addison Wesley, Reading, Massachusetts, 1973,

[8] McGeoch, C. C., "Experimental Analysis of Algorithms", Notices of the AMS, Vol 48, No. 3, March 2001, pp. 304-311.

[9] Priebe, V., *Average-case complexity of shortest-paths problems*, Ph.D. dissertation, University of Saarland, 2001, available at http://www.mpi-sb.mpg.de/~priebe/diss.pdf.

[10] Schneiderman, Ben, "Jump Searching: A Fast Sequential Search Technique", *Communications of the ACM*, Vol. 21, No. 10, 1978, pp. 831-834.

[11] Sedgewick. R., *Algorithms*, Addison Wesley, Reading, Massachusetts, 1983, pp. 12-16.

[12] Tarjan, R. E., *Updating a Balanced Search Tree in O(1) Rotations*, Information Processing Letters, Vol. 16, No. 5, 1983, pp. 253-257.

Theoretical Framework for Comparing Several Stochastic Optimization Approaches

James C. Spall, Stacy D. Hill and David R. Stark

The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099 U.S.A.
E-mail: james.spall@jhuapl.edu

Abstract

This paper establishes a framework for formal comparisons of several leading optimization algorithms, establishing guidance to practitioners for when to use or not use a particular method. The focus in this paper is five general algorithm forms: random search, simultaneous perturbation stochastic approximation, simulated annealing, evolutionary strategies, and genetic algorithms. We summarize the available theoretical results on rates of convergence for the five algorithm forms and then use the theoretical results to draw some preliminary conclusions on the relative efficiency. Our aim is to sort out some of the competing claims of efficiency and to suggest a structure for comparison that is more general and transferable than the usual problem-specific numerical studies.

Keywords: Stochastic optimization; rate of convergence; random search; simultaneous perturbation stochastic approximation (SPSA); simulated annealing; evolutionary computation; genetic algorithms.

1. INTRODUCTION

To address the shortcomings of classical deterministic algorithms, a number of powerful optimization algorithms with embedded randomness have been developed. The population-based methods of evolutionary computation are only one class among many of these available *stochastic* optimization algorithms. Hence, a user facing a challenging optimization problem for which a stochastic optimization method is appropriate meets the daunting task of determining which algorithm is appropriate for a given problem. This choice is made more difficult by some dubious claims that have been made about some popular algorithms. An inappropriate approach may lead to a large waste of resources, both from the view of wasted efforts in implementation and from the view of the resulting suboptimal solution to the optimization problem of interest.

Hence, there is a need for objective analysis of the relative merits and shortcomings of leading approaches to stochastic optimization. This need has certainly been recognized by others, as illustrated in recent conferences on evolutionary computation, where numerous sessions are devoted to comparing algorithms. Nevertheless, virtually all comparisons have been numerical tests on specific problems. Although sometimes enlightening, such comparisons are severely limited in the *general* insight they provide. Some comparisons for *noisy* evaluations of a simple spherical loss function are given in Arnold (2002, Chap. 6); however, some of the competitors were implemented in non-standard forms, making the results difficult to interpret for an analyst using a more conventional implementation. Spall (2003) also has a number of comparisons (theoretical and numerical) for the cases of noise-free and noisy loss evaluations. On the other end of the spectrum are the "No Free Lunch Theorems" (Wolpert and McReady, 1997), which simultaneously consider all possible loss functions and thereby draw conclusions that have limited practical utility since one always has at least *some* knowledge of the nature of the loss function being minimized.

Our aim in this paper is to lay a framework for a *theoretical* comparison of efficiency applicable to a broad class of practical problems where some (incomplete) knowledge is available about the nature of the loss function. We will consider five basic algorithm forms—random search, simultaneous perturbation stochastic approximation (SPSA), simulated annealing (SAN), and two forms of evolutionary computation (evolution strategy and genetic algorithms). The basic optimization problem corresponds to finding an optimal point θ^*:

$$\theta^* = \arg\min_{\theta \in \Theta} L(\theta),$$

This work was partially supported by the JHU/APL IRAD Program and U.S. Navy Contract N00024-98-D-8124. An expanded version of this paper is available upon request.

where $L(\theta)$ is the loss function to be minimized, Θ is the domain over which the search will occur, and θ is a p-dimensional (say) vector of parameters. We are mainly interested in the case where θ^* is a *unique* global minimum.

Although stochastic optimization approaches other than the five above exist, we are restricting ourselves to the five general forms in order to be able to make tangible progress (note that there are various specific implementations of each of these general algorithm forms). These five algorithms are general-purpose optimizers with powerful capabilities for serious multivariate optimization problems. Further, they have in common the requirement that they only need measurements of the objective function, not requiring derivative information (gradient or Hessian) for the loss function.

One might ask whether questions of relative efficiency are relevant in light of the "no free lunch (NFL)" theorems of Wolpert and Macready (1997) and others. The NFL theorems state, in essence, that the expected performance of any pair of optimization algorithms across all possible problems is identical. In practice, of course, one is not interested in solving "all possible problems," as there is usually some prior information about the problems of interest and this prior information will affect the algorithm implementation. Hence, the NFL results may not adequately reflect the performance of candidate algorithms as they are actually applied. In other words, some algorithms *do* work better than others on problems of interest. Nevertheless, the NFL results are an important backdrop against which to view the results here, providing limits on the extent to which one algorithm can be claimed as "better" than another.

2. SIMPLE GLOBAL RANDOM SEARCH

We first establish a rate of convergence result for the simplest random search method where we repeatedly sample over the domain of interest, $\Theta \subseteq \mathbb{R}^p$. This can be done in recursive form or in "batch" (non-recursive) form by simply laying down a number of points in Θ and taking as our estimate of θ^* that value of θ yielding the lowest L value.

To evaluate the rate, let us specify a "satisfactory region" $S(\theta^*)$ representing some neighborhood of θ^* providing acceptable accuracy in our solution (e.g., $S(\theta^*)$ might represent a hypercube about θ^* with the length of each side representing a tolerable error in each coordinate of θ). An expression related to the rate of convergence of the above simple random search algorithm is then given by

$$P(\hat{\theta}_k \in S(\theta^*)) = 1 - [1 - P(\theta_{\text{new}}(k) \in S(\theta^*)]^k \quad (2.1)$$

We will use this expression in Section 7 to derive a convenient formula for comparison of efficiency with other algorithms.

3. SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

The next algorithm we consider is SPSA. This algorithm is designed for continuous variable optimization problems. Unlike the other algorithms here, SPSA is fundamentally oriented to the case of *noisy* function measurements and most of the theory is in that framework. This will make for a difficult comparison with the other algorithms, but Section 7 will attempt a comparison nonetheless. The SPSA algorithm works by iterating from an initial guess of the optimal θ, where the iteration process depends on a highly efficient "simultaneous perturbation" approximation to the gradient $g(\theta) \equiv \partial L(\theta)/\partial\theta$.

Assume that measurements $y(\theta)$ of the loss function are available at any value of θ:

$$y(\theta) = L(\theta) + \textit{noise} .$$

For example, in a Monte Carlo simulation-based optimization context, $L(\theta)$ may represent the mean response with input parameters θ, and $y(\theta)$ may represent the outcome of one simulation experiment at θ. In some problems, exact loss function measurements will be available; this corresponds to the *noise* = 0 setting (and in the simulation example, would correspond to a deterministic—non-Monte Carlo—simulation). Note that no direct measurements (with or without noise) of the gradient of $L(\theta)$ are assumed available.

The SPSA procedure is in the general recursive SA form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) , \quad (3.1)$$

where $\hat{g}_k(\hat{\theta}_k)$ is the estimate of the gradient $g(\theta)$ at the iterate $\hat{\theta}_k$ based on the above-mentioned measurements of the loss function and $a_k > 0$ is a "gain" sequence. This iterate can be shown to converge under reasonable conditions (e.g., Spall, 1992, and Dippon and Renz, 1997, for local convergence; Maryak and Chin, 2001, for global convergence). The essential basis for efficiency of SPSA in multivariate problems is due to the gradient approximation, where only two measurements of the loss function are needed to estimate the p-dimensional gradient vector for any p; this contrasts with the standard finite difference method of gradient approximation, which requires $2p$ measurements.

Most relevant to the comparative analysis goals of this paper is the asymptotic distribution of the iterate. This was derived in Spall (1992), with further developments in Chin (1997), Dippon and Renz (1997), and Spall (2000). Essentially, it is known that under appropriate conditions,

$$k^{\beta/2}(\hat{\theta}_k - \theta^*) \xrightarrow{\text{dist}} N(\mu, \Sigma) \text{ as } k \to \infty , \quad (3.2)$$

where $\beta > 0$ depends on the choice of gain sequences (a_k and c_k), μ depends on both the Hessian and the third derivatives of $L(\theta)$ at θ^* (note that in general, $\mu \neq 0$ in contrast to many well-known asymptotic normality results in estimation), and Σ depends on the Hessian matrix at θ^* and the variance of the noise in the loss measurements. Given the restrictions on the gain sequences to ensure convergence and asymptotic normality, the fastest allowable value for the rate of convergence of $\hat{\theta}_k$ to θ^* is $k^{-1/3}$. This contrasts with the fastest allowable rate of $k^{-1/2}$ for gradient-based algorithms such as Robbins-Monro SA.

Unfortunately, (3.2) is not directly usable in our comparative studies here since the other three algorithms being considered here appear to have convergence rate results only for the case of *noise-free* loss measurements. The authors are unaware of any general asymptotic distribution result for the noise-free case (note that it is *not* appropriate to simply let the noise level go to zero in (3.2) in deriving a result for the noise-free case; it is likely that the rate factor β will also change if an asymptotic distribution exists). Some partial results, however, are available that are related to the rate of convergence. Gerencsér (1999) established that the moments $\left[E\left(\left\| \hat{\theta}_k - \theta^* \right\|^q \right) \right]^{1/q}$ converge to zero at a rate of $k^{-1/2}$ for any $q > 0$, when a_k has the standard $1/k$ decay rate. More recently, Gerencsér and Vágó (2000) established that the noise-free SPSA algorithm has a geometric rate of convergence when *constant* gains $a_k = a$ are used. In particular, for functions having bounded third derivatives, they show for sufficiently small a,

$$\limsup_{k \to \infty} \frac{\left\| \hat{\theta}_k - \theta^* \right\|}{\eta^k} = 1 \text{ a.s.}$$

for some $0 < \eta < 1$. Gerencsér and Vágó (2000) go further for quadratic loss functions by specifying η in terms of a and the Hessian matrix of L. Unfortunately, even in the quadratic case, η is not fully specified in terms of quantities associated with L and the algorithm itself (i.e., η depends on unknown constants).

4. SIMULATED ANNEALING ALGORITHMS

The SAN method (Metropolis et al., 1953; Kirkpatrick et al., 1983) was originally developed for optimization over discrete finite sets. The Metropolis SAN method produces a sequence that converges in probability to the set of global minima of the loss function as T_k, the *temperature*, converges to zero.

Gelfand and Mitter (1993) present a SAN method for continuous parameter optimization. They obtained discrete-time recursions (which are similar to a stochastic approximation algorithm) for Metropolis-type SAN algorithms that, in the limit, optimize continuous parameter loss functions.

Furthermore, like SPSA, SAN has an asymptotic normality result (but unlike SPSA, this result applies in the noise-free case). Let $H(\theta^*)$ denote the Hessian of $L(\theta)$ evaluated at θ^* and let I_p denote the $p \times p$ identity matrix. Yin (1999) showed that for $b_k = (b/(k^\gamma \log (k^{1-\gamma} + B_0))^{1/2}$,

$$[\log (k^{1-\gamma} + B_0)]^{1/2} (\hat{\theta}_k - \theta^*) \to N(0, \Sigma) \text{ in distribution,}$$

where $\Sigma H + H^T \Sigma + (b/a) I = 0$.

5. EVOLUTIONARY COMPUTATION: EVOLUTIONARY STRATEGIES

There are three general approaches in evolutionary computation (EC), namely Evolutionary Programming (EP), Evolutionary Strategies (ES) and Genetic Algorithms (GA). All three approaches work with a population of candidate solutions and randomly alter the solutions over a sequence of generations according to evolutionary operations of competitive selection, mutation and sometimes recombination (reproduction). The fitness of each population element to survive into the next generation is determined by a selection scheme based on evaluating the loss function for each element of the population. The selection scheme is such that the most favorable elements of the population tend to survive into the next generation while the unfavorable elements tend to perish.

The principle differences in the three approaches are the selection of evolutionary operators used to perform the search and the computer representation of the candidate solutions. EP uses selection and mutation only to generate new solutions. While both ES and GA use selection, recombination and mutation, recombination is used more extensively in GA. A GA traditionally performs evolutionary operations using binary encoding of the solution space, while EP and ES perform the operations using real-coded solutions. The GA also has a real-coded form and there is some indication that the real-coded GA may be more efficient and provide greater precision than the binary-coded GA. The distinction among the three approaches has begun to blur as new hybrid versions of EC algorithms have arisen.

Global convergence results can be given for a broad class of problems, but the same cannot be said for convergence *rates*. Both Beyer (1995) and Rudolph (1997a) examine ES algorithms that include selection, mutation and recombination. The function analyzed in both cases is the classic spherical fitness function $L(\theta) = \|\theta\|^2$ whose exact solution is of course known. Convergence rates based on the spherical fitness function are somewhat useful, if it is assumed that the sphere approximates a local basin of attraction. A number of other convergence rate results are also available for that fitness function, for example Qi and Palmeiri (1994) for real-valued GA. The most practically useful convergence rates for EC algorithms seem to be for the class of strongly convex fitness

functions. The following theorem due to Rudolph (1997b) is an application of a more general result by Rappl (1989). The theorem will be the starting place for the specific convergence rate result that will be used for comparison in Section 7.

An EC algorithm has a *geometric rate of convergence* if and only if $E[\, L_k^* - L(\theta^*)] = O(\eta^k)$ where $\eta \in (0, 1)$ is called the *convergence rate*. Under conditions, the convergence rate result for a $(1, \lambda)$-ES using only selection and mutation on a (K, Q)-strongly convex fitness function is geometric with a rate of convergence

$$\eta = (1 - M_{\lambda,p}^2 Q^2)$$

where Q is a constant, $M_{\lambda,p} = E[B_{\lambda:\lambda}] > 0$, and where $B_{\lambda:\lambda}$ denotes the maximum of λ independent identically distributed Beta random variables. The computation of $M_{\lambda,p}$ is complicated since it depends on both the number of offspring λ and the problem dimension p. Asymptotic approximations are available and will be shown next. Assuming p is fixed and $\lambda \to \infty$ then $M_{\lambda,p} \approx (2\, p^{-1} \log \lambda)^{1/2}$. To extend this convergence rate from a $(1, \lambda)$-ES to a $(N_{\text{pop}}, \lambda)$-ES, note that each of the N_{pop} parents generate λ / N_{pop} offspring. Then the convergence rate for the $(N_{\text{pop}}, \lambda)$-ES where offspring are only obtained by mutation is

$$\eta \le [1 - (2p^{-1} \log(\lambda / N_{\text{pop}})) / Q^2]$$

for (K, Q)-strongly convex functions.

6. EVOLUTIONARY COMPUTATION: GENETIC ALGORITHMS

As discussed in Stark and Spall (2001), it is possible to cast the GA in the framework of Markov chains. This allows for a rate of convergence analysis. Consider a GA with a population size of N. Further, suppose that each population element is a binary string of length b bits. Hence, there are 2^b possible strings for an *individual* population element. Then the total number of possible populations is given by

$$N_{\text{pop}} \equiv \frac{(N + 2^b - 1)!}{(2^b - 1)!\, N!}.$$

It is possible to construct a Markov transition matrix Π that provides the probability of transitioning from one population of size N to another population of the same size. This transition matrix is $N_{\text{pop}} \times N_{\text{pop}}$. An individual element in the transition matrix can be computed according to the formulas in Stark and Spall (2001) (see also Suzuki, 1995). These elements depend in a non-trivial way on the population size, crossover rate, mutation rate, and number of elements considered "elite."

Of primary interest in analyzing the performance of GA algorithms using Markov chains is the probability of obtaining a population that contains the optimum θ^*. Let π_k be an $N \times 1$ vector having j^{th} component, $\pi_k(j)$, equal to the probability that the k^{th} generation will result in population j. From basic Markov chain theory,

$$\pi_k^T = \pi_{k-1}^T \Pi = \pi_0^T \Pi^k$$

where π_0 is an initial probability distribution.

The stationary distribution of the GA is then given by

$$\bar{\pi}^T \equiv \lim_{k \to \infty} \pi_k^T = \lim_{k \to \infty} \pi_0^T \Pi^k .$$

Further, under standard ergodicity assumptions for Markov chains, $\bar{\pi}$ satisfies $\bar{\pi}^T = \bar{\pi}^T \Pi$. This equation provides a mechanism for solving directly for the stationary distribution (e.g., Iosifescu, 1980, pp. 123–124).

Unfortunately, from a practical view, the Markov chain approach has a significant deficiency. The dimension N grows very rapidly with increases in the number of bits b and/or the population size N. A perhaps more intuitive estimate of the size of N_{pop} can be obtained by Stirling's Approximation as follows:

$$N_{\text{pop}} \approx \sqrt{2\pi} \left(1 + \frac{2^b - 1}{N}\right)^N \left(1 + \frac{N}{2^b - 1}\right)^{2^b - 1} \left(\frac{1}{2^b - 1} + \frac{1}{N}\right)^{1/2}$$

Thus far, our analysis using the above approach has been restricted to scalar θ systems (requiring fewer bits b than a multivariate system) and low N_{pop}.

7. COMPARATIVE ANALYSIS

7.1 Problem Statement and Summary of Efficiency Theory for the Five Algorithms

This section uses the specific algorithm results in Sections 2 to 6 above in drawing conclusions on the relative performance of the five algorithms. There are obviously many ways one can express the rate of convergence, but it is expected that, to the extent they are based on the theory outlined above, the various ways will lead to broadly similar conclusions. We will address the rate of convergence by focusing on the question:

> *With some high probability 1 − ρ (ρ a small number), how many $L(\cdot)$ function evaluations, say n, are needed to achieve a solution lying in some "satisfactory set" $S(\theta^*)$ containing θ^*?*

With the random search algorithm in Section 2, we have a closed form solution for use in questions of this sort while with the SPSA, SAN, and EC algorithms of Sections 3 through 5, we must apply the existing asymptotic results, assuming that they apply to the finite-sample question above. For the GA, there is a finite sample solution using the Markov chain approach. For each of the five algorithms, we will outline below an analytical expression

useful in addressing the question. After we have discussed the analytical expressions, we present a comparative analysis in a simple problem setting for varying *p*.

Random Search

We can use (2.1) to answer the question above. Setting the left-hand side of (2.1) to 1 – ρ and supposing that there is a constant sampling probability $P^* = P(\theta_{new}(k) \in S(\theta^*)) \; \forall \; k$, we have

$$n = \frac{\log \rho}{\log(1 - P^*)}. \qquad (7.1)$$

Simultaneous Perturbation Stochastic Approximation

From the fact that SPSA uses two $L(\theta^*)$ evaluations per iteration, the value *n* to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is then

$$n = 2\left(\frac{2d(p)\sigma}{\delta s}\right)^3$$

where from standard $N(0, 1)$ distribution tables, there exists a displacement factor, say $d(p)$, such that the probability contained within $\pm\, d(p)$ units contains probability amount $(1 - \rho)^{1/p}$. We are interested in the *k* such that $2d(p)\sigma/k^{1/3} = \delta s = s_i^+ - s_i^-$ (the common length of a side in a *p*-fold hypercube).

Simulated Annealing

The value *n* to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is

$$\log n = \frac{1}{1-\gamma}\left(\frac{2d(p)\sigma}{\delta s}\right)^2.$$

Evolutionary Strategy

As discussed in Section 6, the rate-of-convergence results for algorithms of the evolutionary computation type are not as well developed as for the other three algorithms of this paper. Theorem 6.1 gives a general bound on $E[L(\hat{\theta}_k) - L(\theta^*)]$ for application of a (N, λ)-ES form of EC algorithm to strongly convex functions. A more explicit form of the bound is available for the (1, λ)-ES. Unfortunately, even in the optimistic case of an explicit numerical bound on $E[L(\hat{\theta}_k) - L(\theta^*)]$, we cannot readily translate the bound into a probability calculation for $\hat{\theta}_k \in S(\theta^*)$, as used above (and, conversely, the asymptotic normality result on $\hat{\theta}_k$ for SPSA and SAN cannot be readily translated into one on $L(\hat{\theta}_k)$ since $\partial L/\partial\theta = 0$ at θ^*—see, e.g., Serfling, 1980, pp. 122–124—although Lehmann, 1983, pp. 338–339 suggests a possible means of coping with this problem via higher-order expansions). So, in order to make some reasonable comparison, let us suppose that we can associate a set $S(\theta^*)$ with a given deviation from $L(\theta^*)$, i.e., $S(\theta^*) = S(\theta^*, \varepsilon) = \{\theta: L(\hat{\theta}_k) - L(\theta^*) \le \varepsilon\}$ for some prespecified tolerance $\varepsilon > 0$. As presented in Rudolph (1997b), $E[L(\hat{\theta}_k) - L(\theta)] \le c^k$ for sufficiently large *k*, where *c* is the convergence rate in Section 6. Then by Markov's inequality,

$$1 - P(\hat{\theta}_k \in S(\theta^*)) \le \frac{E[L(\hat{\theta}_k) - L(\theta^*)]}{\varepsilon} \le \frac{c^k}{\varepsilon}, \qquad (7.2)$$

indicating that $P(\hat{\theta}_k \in S(\theta^*))$ is bounded below by the ES bounds mentioned in Section 5.

The full version of the paper employs Markov's inequality and the bound in Rudolph (1977b) to show that there are λ evaluations of the fitness function for each generation *k* so that $n = \lambda k$, where

$$k = \frac{\log \rho - \log(1/\varepsilon)}{\log\left[1 - \frac{2}{pQ^2}\log(\lambda/N)\right]}.$$

Genetic Algorithm

As mentioned in Section 6, while the GA has a relatively clean theory that applies in both finite and asymptotic samples, there are significant challenges in computing the elements of the Markov transition matrix Π. The number of possible states—corresponding to the number *N* of possible populations—grows extremely rapidly with the number of population elements *N* or the number of bits *b*. The computation of the $N_{pop} \times N_{pop}$ transition matrix Π quickly overwhelms even the most powerful current personal computers.

Nevertheless, in principle, the Markov structure is convenient for establishing a convergence rate for the GA. The full version of the paper provides value for *n*.

7.2 Application of Convergence Rate Expressions for Varying *p*

We now apply the results above to demonstrate relative efficiency for varying *p*. Because the GA result is computationally explosive as *p* gets larger (requiring a larger bit string length and/or population size), we restrict the comparison here to the four algorithms: random search, SPSA, SAN and ES. Let $\Theta = [0, 1]^p$ (the *p*-dimensional hypercube with minimum and maximum θ values of 0 and 1 for each component). We want to guarantee with probability 0.90 that each element of θ is within 0.04 units of the optimal. Let the (unknown) true θ, θ^*, lie in $(0.04, 0.96)^p$. The individual components of θ^* are θ_i^*. Hence,

$$S(\theta^*) = [\theta_1^* - 0.04, \theta_1^* + 0.04] \times [\theta_2^* - 0.04,\ \theta_2^* + 0.04] \times \ldots$$

$$\times [\theta_p^* - 0.04,\ \theta_p^* + 0.04] \subset \Theta .$$

Table 7.1 is a summary of relative efficiency for the setting above for p = 2, 5, and 10; the efficiency was normalized so that all algorithms performed equally at p = 1, as described below. The numbers in Table 7.1 are the ratios of the number of loss measurements for the given algorithm over the number for the best algorithm at the specified p; the highlighted values 1.0 indicate the best algorithm for each of the values of p. To establish a fair basis for comparison, we fixed the various parameters in the expressions above (e.g., σ in SPSA and SAN, ρ for the ES, etc.) so that the algorithms produced identical efficiency results for p = 1 (requiring n = 28 measurements to achieve the objective outlined above). These parameters do not explicitly depend on p. We then use these parameter settings as p increases.

Table 7.1. Ratios of loss measurements needed relative to best algorithm at each p for $1 \le p \le 10$

	p = 1	p = 2	p = 5	p = 10
Rand. Search	1.0	11.6	8970	2.0×10^9
SPSA	1.0	1.5	1.0	1.0
SAN	1.0	1.0	2.2	4.1
ES	1.0	1.9	1.9	2.8

Table 7.1 illustrates the explosive growth in the relative (and absolute) number of loss evaluations needed as p increases for the random search algorithm. The other algorithms perform more comparably, but there are still some non-negligible differences. For example, at p = 5, SAN will take 2.2 times more loss measurements than SPSA to achieve the objective of having $\hat{\theta}_k$ inside $S(\theta^*)$ with probability 0.90. Of course, as p increases, all algorithms take more measurements; the table only shows relative numbers of function evaluations (considered more reliable than absolute numbers).

This large improvement of SPSA and SAN relative to random search may partly result from the more restrictive regularity conditions of SPSA and SAN (i.e., for formal convergence, SPSA assumes a several-times-differentiable loss function) and partly from the fact that SPSA and SAN work with *implicit* gradient information via gradient approximations. The performance for ES is quite good. The restriction to strongly convex fitness functions, however, gives the ES in this setting a strong structure not available to the other algorithms. It remains unclear what practical theoretical conclusions can be drawn on a broader class of problems.

REFERENCES

Arnold, D. V. (2002), *Noisy Optimization with Evolution Strategies*, Kluwer, Boston.

Beyer, H.-G. (1995), "Toward a Theory of Evolution Strategies: On the Benefits of Sex—the (μ/μ,λ) Theory," *Evolutionary Computation*, vol. 3, pp. 81–111.

Chin, D. C. (1997), "Comparative Study of Stochastic Algorithms for System Optimization Based On Gradient Approximations," *IEEE Transactions on Systems, Man, and Cybernetics—B*, vol. 27, pp. 244–249.

Dippon, J. and Renz, J. (1997), "Weighted Means in Stochastic Approximation of Minima," *SIAM Journal on Control and Optimization*, vol. 35, pp. 1811–1827.

Gelfand, S. and Mitter, S.K. (1993), "Metropolis-Type Annealing Algorithms for Global Optimization in R^d," *SIAM Journal of Control and Optimization*, vol. 31, pp. 111–131.

Gerencsér, L. (1999), "Convergence Rate of Moments in Stochastic Approximation with Simultaneous Perturbation Gradient Approximation and Resetting," *IEEE Transactions on Automatic Control*, vol. 44, pp. 894–905.

Gerencsér, L. and Vágó, Z. (2000), "SPSA in Noise-Free Optimization," in *Proceedings of the American Control Conference*, pp. 3284–3288.

Iosifescu, M. (1980), Finite Markov Processes and Their Applications, Wiley, New York.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983), "Optimization by Simulated Annealing," *Science*, vol. 220,
pp. 671–680.

Lehmann, E.L. (1983), *Theory of Point Estimation*, Wiley, New York.

Maryak, J.L. and Chin, D.C. (2001), "Global Random Optimization by Simultaneous Perturbation Stochastic Approximation," in *Proceedings of the American Control Conference*, pp. 756–762.

Metropolis, N., Rosenbluth, A., Rosenbluth, M. Teller, A. and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092.

Qi, X. and Palmeiri, F. (1994), "Theoretical Analysis of Evolutionary Algorithms with Infinite Population Size in Continuous Space, Part I: Basic Properties," *IEEE Transactions on Neural Networks*, vol. 5, pp. 102–119.

Rappl, G. (1989), "On Linear Convergence of a Class of Random Search Algorithms," *Zeitschrift für angewandt Mathematik und Mechanik (ZAMM)*, vol. 69, pp. 37–45.

Rudolph, G. (1997a), Convergence Properties of Evolutionary Algorithms, Kovac, Hamburg.

Rudolph, G. (1997b), "Convergence Rates of Evolutionary Algorithms for a Class of Convex Objective Functions," *Control and Cybernetics*, vol. 26, pp. 375–390.

Rudolph, G. (1998), "Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon," *Fundamenta Informaticae*, vol. 34, pp. 1–22.

Serfling, R.J. (1980), Approximation Theorems of Mathematical Statistics, Wiley, New York.

Spall, J. C. (1992), "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control*, vol. 37, pp. 332–341.

Spall, J.C. (2000), "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *IEEE Transactions on Automatic Control*, vol. 45, pp. 1839–1853.

Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, Hoboken, NJ.

Stark, D. R. and Spall, J. C. (2001), "Computable Bounds on the Rate of Convergence in Evolutionary Computation," in *Proceedings of the American Control Conference*, pp. 918–922.

Suzuki, J. (1995), "A Markov Chain Analysis on Simple Genetic Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics—B*, vol. 25, pp. 655–659.

Wolpert, D. H. and Macready, W. G. (1997), "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67–82.

Yin, G. G. (1999), "Rates of Convergence for a Class of Global Stochastic Optimization Algorithms," *SIAM Journal on Optimization*, vol. 10, pp. 99–120.

Measuring the Performance of Automated Planning Systems

Dana Nau
Department of Computer Science
and Institute for Systems Research
University of Maryland,
College Park, MD 20742, USA
email: nau@cs.umd.edu

Malik Ghallab
LAAS-CNRS
7, Avenue du Colonel Roche
31077, Toulouse, cedex, France
email: Malik.Ghallab@laas.fr

ABSTRACT

In this paper, we describe existing performance measures for automated planning algorithms, and discuss the limitations and biases inherent in those performance measures. We point out the importance of developing a performance measure that explicitly the restrictive assumptions on which a planning algorithm depends, and we propose a composite performance measure based on three factors:

- the scope of the planning algorithm: which set of restrictive assumption are needed and which can be lifted,
- the control knowledge and tuning required for each planning domain,
- the size of the problems that can be solve in a reasonable amount of time in each area of its scope (i.e., for each combination of relaxed assumptions it can handle).

KEYWORDS: automated planning, AI planning, performance measurement

1. INTRODUCTION

Great strides have been made in automated planning during the past few years, and the technology is becoming mature enough to be useful in a variety of demanding applications, ranging from controlling space vehicles such as Deep Space 1 [6] to playing the game of bridge [31]. Successes such as these are creating a great potential for synergy between theory and practice: observing what works well in practice can lead to better theories of planning, and better theories can lead to better performance in practical applications.

Despite this potential, there currently is a substantial gap between theoretical and application-oriented work. The theoretical work tends to be rather narrow in scope, focusing on highly restricted cases such as *classical planning*, with the most common performance measure being the speed of the planner's combinatorial search. The application-oriented work generally depends on *ad hoc* application-specific programming efforts, search techniques, and measures of performance.

For most planning systems, presentations of the planning algorithm may discuss some of the assumptions and restrictions explicitly—but usually the algorithm will also depend on additional assumptions and restrictions that are tacit in the representation rather than explicit. As a consequence, it is often very difficult to judge whether a planning algorithm can be useful for real-world problem solving, and it is often even more difficult to tell whether an application-specific planning algorithm can be generalized to work in anything other than the specific application for which the algorithm has been written. *Better ways are needed to judge the scope and generalizability of planning algorithms and techniques.*

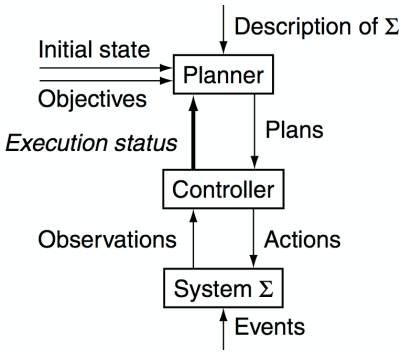


Figure 1: A simple conceptual model for planning. Σ is a state-transition system, as described in the text.

As a step toward meeting that need, we describe a general conceptual model for planning, and use it to classify and discuss the kinds of restrictive assumptions that are often made in automated planning research. We believe that with suitable refinement, such a classification will provide a useful performance measure for automated planning algorithms, by providing a way to give a clearer account of what restrictions a planning algorithm requires.

2. CONCEPTUAL MODEL FOR PLANNING

Since planning is concerned with choosing and organizing actions for changing the state of a system, a conceptual model for planning requires a general model for a dynamic system. This model, shown in Figure 1, includes three components:

- A *state-transition system* Σ that evolves as specified by its state-transition function γ, according to the events and ac-

Experimental Studies of Integrated Cognitive Systems

Pat Langley
Computational Learning Laboratory
Center for the Study of Language & Information
Stanford University, Stanford, CA 94305 USA
langley@csli.stanford.edu

Elena Messina
National Institute of Standards & Technology
100 Bureau Drive, Mail Stop 8230
Gaithersburg, MD 20899 USA
elena.messina@nist.gov

Abstract— In this paper, we examine the issues that arise in the experimental study of integrated cognitive systems. We review the reasons why such artifacts are difficult to evaluate, then consider some dependent measures that can be used to characterize their behavior. Next we discuss independent variables that can influence this behavior, in particular features of the domain and characteristics of the system, including its knowledge and experience. We then turn to domains and testbeds that support experiments with such systems, giving examples of some promising candidates. We conclude with a discussion of the scientific goals of experimentation, which involve understanding the mapping from domain and system characteristics onto behavior.

I. Introduction and Motivation

For more than a decade, research in artificial intelligence has relied on experimentation as a key element in evaluation. Machine learning was perhaps the first subdiscipline to adopt systematic experiments (e.g., Kibler & Langley, 1988), but their use has spread throughout the broader community (e.g., Cohen, 1995). Today, experiments are the primary means by which AI researchers evaluate their methods, and the experimental techniques as mature and well understood.

However, the experimental study of integrated cognitive systems is less well established and clearly needs more development. The reasons should be clear from the phrase itself, which reflects the nature of the intelligent artifacts being constructed. First, it is inherently more difficult to evaluate systems than component algorithms, since they are harder to construct and analyze. Second, it is more challenging to run experiments with cognitive systems, since they rely on complex, multi-step reasoning rather than simple classification or reactive control. Finally, evaluating claims about integrated systems is problematic because it involves the examination of interactions among their components. Together, these distinctive factors have slowed the development of an experimental method for such complex entities.

In this paper, we propose an experimental framework that is appropriate for the study of integrated cognitive systems. In the next section, we discuss basic and higher-order dependent measures that can arise in such experiments. After this, we consider three main classes of independent factors that can influence system behavior, then turn to domains and testbeds that would support the experimental evaluation of such systems. In closing, we discuss the broader scientific goals of experimentation, which aim not to show superiority but to identify reasons for observed behaviors.

II. Dependent Measures of System Behavior

As scientists, we are concerned with understanding the behavior of integrated cognitive systems, which in turn means that we require ways to observe and characterize this behavior. In this context, it is important to distinguish between between metrics and dependent measures. These terms are closely related, but the first is typically associated with prescriptive benchmarks that are used to determine one system's superiority other another, whereas the second is generally associated with systematic experiments that aim at scientific understanding. The comments that follow are relevant to both approaches to evaluation, but our focus here is on the latter, which we think is far more appropriate for the current stage of the field. We organize our treatment into three broad categories: basic measures, averaged metrics, and higher-order variables.

A. Basic Measures of System Behavior

The existing literature reports a variety of basic measures that are relevant to integrated cognitive systems. These provide the simplest ways to describe the observed behavior of an intelligent construct. We should clarify that behavior always occurs in the context of some task, whether provided externally or generated by the agent itself, and some situation, whether it involves the agent's physical environment or its mental state. We will refer to this context informally as the problem that the agent is attempting to solve.

Perhaps the most straightforward behavioral measure concerns whether the agent succeeds or fails at handling a given problem. For example, a cognitive system may prove or fail to prove a geometry theorem, it may or may not solve a novel puzzle, it may or may not deliver a package to a specified address, and it may win or lose a given game. This measure offers only one bit of information, but it may still be valuable when combined with other results, as we will see shortly.

However, other problem-related measures provide more detail. One such metric is the efficiency or speed with which the cognitive system handles a given problem.

For instance, one can count the number of states in a problem space considered during a geometry proof, the time it takes a driver to deliver a package, and the number of moves until checkmate in a chess game. Such a dependent variable gives information about the cognitive or physical efficiency with which the agent handles a particular problem.

Of course, some paths to success are more desirable than others, so we may also want to measure the quality of the cognitive system's solution to a problem. For example, a geometry proof may have few or many steps and thus be more or less elegant, a package deliverer may drive safely and politely or dangerously and impolitely on his way to an address, and a chess player may lose only a few unimportant pieces or many important ones in defeating an opponent. Metrics of this sort offer details about the desirability of the cognitive agent's behavior in accomplishing a given task.

B. Combined Measures of Behavior

The field of statistics tells us we should not draw conclusions from individual cases, but rather that we should rely on multiple samples. We can then combine the results from these samples and calculate a more robust dependent variable. Taking the average of sampled measurements is the most common and obvious combination scheme, but calculating cumulative scores is another possibility. The important thing is that, by combining measures for different samples, we can partly cancel out variation due to unknown or unavailable factors, and thus increase the chance of meaningful results.

Naturally, this approach requires some population from which to draw samples, typically different problems from within a single domain, although sampling from across domains is also possible. For instance, we might present the cognitive system with different geometry theorems to prove, ask it to deliver packages to distinct addresses or even in different cities, and confront it with different chess opponents or even chess-like games with alternative rules. The population from which one draws samples determines the generality of one's conclusions about the cognitive system's behavior. We may suspect that the agent can prove theorems not only in geometry but also in algebra, but sampling from the former domain provides no evidence for the latter. An empirical study should state clearly the population being sampled, ideally in formal terms but always in enough detail that others can replicate the sampling process.

We should note that combined measures of behavior offer more than guards against unknown factors and random noise. This approach also lets one convert qualitative measures, such as success or failure on a problem, into quantitative ones, such as the percentage or total number of problems solved. This makes them especially useful for researchers who want to make claims about new functionality, which at first glance appear to involve only qualitative evidence, but which can be handled in quantitative terms with averaged, cumulative, or other combined measurements of system performance.

C. Higher-Order Measures of Behavior

Although combined measures guard against unknown influences and offer quantitative variables, they still present only a small window into often complex behavior. Metrics that average across domains improve the situation, since they provide information about a cognitive system's broader generality, but more sophisticated responses are certainly possible.

For instance, we might plot the dependent measure for a novel system against the same measure for a baseline or control system, with each point summarizing the two systems' behaviors on a distinct problem. We can then use regression to fit a line to the points, which gives both a slope and an intercept as higher-order measures. A positive intercept means the novel system does better than the control even on easy problems, whereas a slope greater than one means it scales to difficulty better than the baseline system.

Another example, which we will discuss more later, involves learning curves, in which one plots a behavioral measure like efficiency or quality against the number of training cases a learning system has encountered. Such curves typically have either an exponential or sigmoid shape, so that linear regression is not appropriate, but we can fit them with other parametric forms. These produce higher-order measures for the system's performance at the outset, its rate of improvement as a function of experience, and its asymptotic performance.

Both of these examples involve some form of variation, though this need not be systematic. In general, whenever one collects simple measures of a cognitive system's behavior under a number of distinct conditions, these can be used to calculate higher-order measures that summarize its behavioral characteristics across the conditions from which the samples were taken.

III. Influences on System Behavior

A scientific experiment should do more than measure a system's behavior under one or more condition. The goal of experimentation is to understand the factors that influence the behavior, which means one should measure the dependent variables in multiple situations that differ along some dimension. Such a factor is often referred to as an independent variable, since one can typically vary it independently of others. As with dependent measures, different independent variables can reveal different facets of the system under study. In this section, we examine three broad classes of controllable factors that are appropriate for the experimental evaluation of integrated cognitive systems.

A. Characteristics of the Task and Domain

One important type of independent variable concerns aspects of the problem domain and the tasks which occur within it. The simplest version of this idea involves collecting multiple samples for an experimental condition, which we have already discussed above. For studies with an intelligent system, this means running the system multiple times on different problems from a domain, and then combining the results in some fashion. For this purpose, one draws sample tasks from some distribution over the problem domain. This may involve specifying a fixed set of problems or tasks, but another strategy involves creating a generator that can produce sample problems. In either case, one should state the relation between these samples and the broader class of problems over which one hopes to generalize.

An important variation on this idea involves running the system on problems from different domains to ensure its generality. If we are interested in this central issue, then it is essential to demonstrate successful behavior not only across different tasks within the same domain, but across a variety of distinct domains. For instance, most AI work on game playing has focused on a single game like chess, which Pell (1996) argues has produced systems that are optimized for that domain but do not demonstrate general intelligence. Instead, he defined an entire class of chess-like games and developed a system that plays reasonably when given information about their board, pieces, and rules.

Such studies ensure generality, but they do not by themselves reveal the reasons for variations in system behavior. For this, we must examine the relation between problem difficulty and response. We can order problems by the results they produce on some behavioral measure like problems solved or efficiency of solutions, but this does not provide much insight. Ideally, one should vary experimentally the problem difficulty and examine its effects on system behavior. This in turn requires an analysis of the domain that suggests what factors influence the difficulty of problems.

Kibler and Langley (1998) provide an early domain analysis for machine learning. They propose a number of factors that affect the difficulty of induction tasks, including the complexity of the target concept, the number of irrelevant features, and the amount of noise in the training data. Their analysis focused on classification, but they mention analogous difficulty factors for other areas, such as the regularity of problem spaces and the structure of target grammars. One factor they overlooked was the rate of environmental change, which can pose a challenge for any learning system.

Studies that vary problem difficulty typically rely on synthetic domains to control this factor, but Langley (1996) warns against their casual use. Synthetic problems give one fine-grained control over domain characteristics, which can let one determine how these factors influence behavior. But one must be careful to ensure that these problems are sufficiently similar to ones which arise in natural domains that they remain relevant. Nor should one utilize synthetic problems except to support the systematic variation of domain features. In general, a well-balanced experimental program includes studies with both synthetic domains, to provide insight, and natural ones, to ensure relevance.[1]

B. Characteristics of the System

If we want to understand why a cognitive system behaves well or poorly, then we must vary characteristics of that system. The simplest version of this idea involves replacing the entire system with another, as typically occurs in competitions. Unfortunately, even when one system behaves uniformly better than another, which seldom happens, such comparisons provide no insight into the reasons for their behavioral differences.

One form of finer-grained study involves varying the parameters associated with the cognitive system and measuring the effect on its behavior. For instance, one might alter the depth to which search occurs in a system that proves geometry theorems, the utility function used to guide a driving system's choices, and the relative values of pieces in a chess player. Such experiments can lead to conclusions about the importance of a parameter to system behavior, which may be unchanged across a wide range of parameter values, change slowly as the parameter varies, or produce sudden shifts at certain threshold values. Parametric studies may also detect interactions among settings that indicate nonlinear effects.

Another experimental approach compares the basic system's behavior with that when one or more of its modules has been removed. For example, one might compare a driving agent with and without a component for planning routes. Similarly, one might examine a geometry theorem prover with and without a module that learns from previous proofs or a chess player that can or cannot analyze its opponent's strategy. Such lesion studies let one draw conclusions about the contribution of the removed components to the system's overall behavior. They can be especially useful in understanding integrated cognitive systems, since they can reveal interactions among modules. For instance, inclusion of planning and learning abilities in a driving system may provide benefits greater than their sum when used alone.

C. Knowledge and Experience of the System

Cognitive systems rely centrally on knowledge about a domain to make inferences and generate candidate solutions to the problems they encounter. Knowledge is just as important a determinant of behavior as the domain and system characteristics. However, the precise impact of

[1] Unfortunately, this mixture is quite rare in the literature, presumably because it requires extra effort from experimenters, but this does not reduce its importance for the study of intelligent systems.

knowledge on a specific intelligent system is an open issue that can be studied experimentally.

The methodology of lesion studies, which we discussed above in the context of system components, can be adapted easily to knowledge. We can run a geometry theorem prover with and without access to lemmas, we can ask a driver to deliver packages with and without a cognitive map of the city, and we can provide or not provide a chess player with a library of opening moves. In some cases, such lesion studies are equivalent to experiments with system modules, since certain components may be included only to utilize a specific type of knowledge. But the modules of many cognitive systems have more general abilities, so that running them with and without access to knowledge can uncover its importance independent of the component processes themselves.

Of course, the knowledge utilized by a cognitive system does not usually come in large packages, but rather in small, modular knowledge elements. As a result, one can also vary systematically the amount of knowledge available to the agent of a given type. For instance, a theorem prover may have access to many or few lemmas, a driver responsible for delivering packages may have a more or less complete cognitive map, and a chess player may know about different numbers of opening moves. Experiments that treat knowledge in this manner produce graphs that plot behavioral measures like efficiency and quality against knowledge. These can also provide higher-order metrics that describe the rate of improvement per knowledge element, as we discussed earlier.

For cognitive systems that learn, we can examine the effects of experience in a similar manner. Here one relates the number of problems solved, the time spent by the agent, or other measures of experience to the standard behavioral variables. For example, one can graph the percentage of geometry theorems proved as a function of the number of previous efforts, the efficiency of package delivery against the number of earlier trips, and the number of chess pieces lost against the number of games played. As mentioned earlier, such learning curves also provide higher-order information about the rate of improvement and asymptotic behavior.

IV. Repositories for Cognitive Systems

As we have noted, experimental studies of intelligent systems require some class of problems on which to measure behavior, but developing such tasks can be time consuming and expensive. The natural response is to develop a common repository of domains and problems for use by the research community. The earliest example was the UCI Machine Learning Repository (Blake & Merz, 1998), launched by David Aha in the late 1980s. This provided a variety of well-documented data sets for the evaluation of supervised learning systems, and within a few years it became so popular that most papers on machine learning utilized it in their experimental studies. Another model came from computational linguistics, where the annual TREC competitions came to drive many research efforts and has been imitated by other fields, such as the AI planning community.

Unfortunately, despite their advantages, repositories and competitions also have negative aspects. Their very ease of use can encourage a community to focus only on the technical issues they represent. For example, the UCI repository encouraged increased learning research on classification domains at the expense of work on problem-solving tasks. Moreover, many learning researchers have adopted a 'bake-off' mentality that is concerned only with improving performance scores over earlier systems, and competitions like TREC have much the same effect. To the extent that the contents of repositories come to be viewed as benchmark problems, they lose their usefulness for genuine scientific studies.

A. Desirable Characteristics of Testbeds

Nevertheless, a common repository is an obvious means to encourage and support research on integrated cognitive systems, so we should consider what characteristics would make it most useful. Like the UCI repository, it must include a variety of distinct domains to ensure the generality of experimental results. Moreover, its contents must be well documented and it must be easy for researchers to use, with a standardized format or interface to simplify interaction with different cognitive systems. These are key characteristics of existing repositories that are well worth replicating in new ones.

However, the repository should support experiments with integrated cognitive systems in ways that previous ones have not. For example, it should not contain data sets like the UCI site or the TREC competitions, or even sets of problems, like the planning competitions. Instead, it should provide the community with environments or testbeds in which researchers can evaluate their creations. Unlike many component AI algorithms, a cognitive system exists over time and requires some environment in which to operate. This environment need not be a physical one, but embodied cognitive systems are perhaps the most interesting variety, so the repository should contain some testbeds that support the study of physical agents.

A testbed provides supporting or enabling infrastructure for work on a given problem domain. Each testbed must include a definition of the tasks or missions that arise in its domain, stated in terms of initial situations and the desired states or objectives. Each domain should support a range of such tasks and, ideally, come with a problem generator that researchers can use to produce novel ones. A testbed provides infrastructure that facilitates experimentation by the community and thus can lead to insights about alternative approaches. Examples of infrastructural support include: external databases, such as geographic information systems, and the means to connecting to

these resources; the controlled capture, replay, halting, and restart of scenarios; and methods for capturing relevant performance measures via application programming interfaces, access to variables and parameters, and external physical instrumentation.

A well-designed testbed for cognitive systems eases their experimental evaluation, which follows naturally from certain desirable attributes of the infrastructure and problem set. To assist researchers in evaluating high-level behavior, it should provide an environment that has little or no dependence on actuation or sensor processing. In addition, the infrastructure and problem domain should offer a rich operating environment, with the ability to model and control various entities. The testbed should let researchers vary, in quantifiable ways, the difficulty or complexity of the environment or mission. Moreover, although the study of integrated systems is crucial, a testbed should also support evaluation of component subsystems, such as reasoning and learning methods, through parametric and lesion studies.

For domains that involve an external setting, one can certainly create a physical testbed to support evaluation, but another option is to develop a realistic simulated environment that can be used by many more research groups at much lower cost. For example, Jacoff, Messina, and Evans (2001) describe a physical testbed for evaluating robot search and rescue, whereas Balakirsky and Messina (2002) report a simulated environment to support research on the same problem. Simulated testbeds have an additional advantage in that they allow easy variation of domain parameters, ranging from details of the environmental layout to noise in the agent's sensors. Moreover, they let one record detailed traces of the intelligent system's physical behavior and its mapping onto cognitive state, which in turn supports detailed analyses and replay starting from any point along the agent's behavioral trajectory.

However, as we noted above, testbeds that rely on synthetic domains also come with the danger of irrelevance. Whenever possible, they should be based closely on a physical testbed and provide simulations of sufficiently high fidelity. Wang (2003) describes one such simulated domain that incorporates models, based on a gaming engine that supports kinematics and dynamics, of the physical NIST arenas for urban search and rescue. To further ensure relevance for intelligent systems that sense their environment, a testbed may provide data sets collected from real sensors in analogous locations (e.g., Shneier, 2003). Such additions can help retain the advantages of physical environments while offering the affordability and ease of simulated ones.

B. Promising Domains and Testbeds

We can clarify the desirable features of testbeds with some examples. We have already mentioned the search and rescue domain, for which NIST has developed both physical and simulated testbeds. The primary task involves searching for survivors in an urban area after an earthquake or similar disaster. This domain requires the combination of sensing, planning, and action in an integrated cognitive system that can recognize humans, find routes through dangerous areas, and execute its plans successfully. The testbeds have been in place for a number of years and have been used effectively in a number of international competitions.

Another candidate domain involves flying a simulated aircraft in a military setting. Keeping an airplane aloft can be a challenging control task, but by itself this does not require much cognitive activity or integration of different capabilities. However, Jones et al. (1999) report a complex environment in which an agent must fly a jet fighter, distinguish friendly from enemy aircraft, respond according to established doctrine, and communicate with other pilots. Their intelligent agent operated within the ModSAF environment, which was populated by other aircraft, some controlled by programs and others by humans. A related set of problems would involve flying an unmanned reconnaissance vehicle over enemy territory to gather information while avoiding dangerous areas.

A third challenging domain involves in-city driving. This raises few problems at the control level, since keeping a car upright, on the road, and within its lane does not require much intelligence. But the presence of buildings, sidewalks, traffic signs and signals, moving and parked vehicles, and pedestrians make for a very rich environment that requires the allocation of perceptual attention and other resources. Moreover, driving can support many distinct high-level tasks, such as delivering packages, tailing another car unobtrusively, and pulling over vehicles for moving violations. These all require the integration of cognitive, perceptual, and motor components in a complex dynamical setting.

There already exist many simulated driving environments, but few have been developed with the intention of evaluating intelligent systems. Moriarty and Langley (1998) report a simulator for highway driving, but this environment had low fidelity and agents had limited options. More recently, Choi et al. (2004) describe an in-city driving environment, which they have used to evaluate a cognitive driving agent, that includes many more objects and a broader range of activities. Balakirsky, Scrapper, and Messina (in press) are developing another infrastructure, Mobility Open Architecture Simulation and Tools, that provides well-defined interfaces to the various driving subsystems and rich visualization at various levels of resolution. Several organizations are using this system to test subsystems for vehicle control, but it remains to be seen whether the environment meets all the requirements for evaluating an integrated cognitive system.

Both driving and flying involve control of an individual agent, but an equally important class of domains involve managing a large set of other agents. Commanding troops in a battlefield scenario is one example that requires capa-

bilities like monitoring, situation assessment, planning and scheduling of activities, and allocation of resources. However, interactive strategy games like Civilization have similar characteristics and complexity, and they are familiar to more people. Aha and Molineaux (2004) are constructing a framework that simplifies the interface to such games, and thus will provide a set of related testbeds for the experimental study of integrated cognitive systems. Michael Genesereth (personal communication, 2004) is developing a different infrastructure to support an annual competition in generalized game playing (http://games.stanford.edu/), with the intent of fostering research efforts on flexible approaches to intelligent behavior.

V. Concluding Remarks

In the preceding pages, we have considered the dependent measures and independent factors that arise in studying integrated cognitive systems, along with characteristics of repositories and testbeds to support such experiments. Before closing, we should situate these comments in the broader context of scientific experimentation. As in other fields, the aim of systematic experiments is not to show that one approach is superior to another but rather to increase our understanding of complex systems. Such understanding may also lead to improved artifacts, but the overriding goal is to produce replicable and interpretable results that add to our scientific knowledge about intelligent behavior.

To this end, researchers should not carry out unmotivated comparisons between different systems or environments. In most cases, one should have a clear question in mind or a specific hypothesis that one wants to test, and the experimental design should reflect this intention. Simple demonstrations of functionality and generality are reasonable when one first develops a cognitive system, but they should quickly give way to scaling studies that reveal its ability to handle complexity and to lesion studies that identify the roles that its components play in determining overall behavior.

Whenever possible, experimental results should be utilized to test such hypotheses. Because most studies involve averaging across samples, one should be careful about drawing conclusions. Statistical tests can be useful for this purpose, but they are overrated, in that one can sometimes obtain 'significant' differences between experimental conditions even when they are not substantial. Nor are statistical tests required when differences are large, although reporting confidence intervals is crucial for conditions with high variance.

Results that agree with an hypothesis lend it evidence, though they do not 'confirm' it; science can never draw final conclusions about any situation. Results that diverge from one's expectations count as evidence against a claim, and thus require additional explanation. Negative results need not imply failure, since they can lead one to alter assumptions about system behavior and suggest new ways to test them. The iterative loop of hypothesize and test is as central the study of intelligent systems as to other experimental disciplines.

Nevertheless, integrated cognitive systems pose special challenges that require creative adaptation of standard experimental methods. We must develop testbeds that exercise the full capabilities of such systems, rather than emphasizing tasks that can be handled by simple classification or reactive control. We must study behavior at the system level, rather than focusing on component algorithms. Finally, we must design experiments that illuminate the manner in which the modules of such systems interact to produce flexible and robust behavior. Taken together, these steps should let us transform the study of integrated cognitive systems into a dynamic and well-balanced experimental science.

Acknowledgements

This research was funded in part by Grant HR0011-04-1-0008 from Rome Laboratories. Discussions with David Aha, Michael Genesereth, and Barney Pell contributed to the ideas presented in this paper.

References

Aha, D. W., & Molineaux, M. (2004). Integrating learning in interactive gaming simulators. Proceedings of the AAAI-2004 Workshop on Challenges of Game AI. San Jose, CA: AAAI Press.

Balakirsky, S., & Messina, E. (2002). A simulation framework for evaluating mobile robots. Proceedings of the Performance Metrics for Intelligent Systems Workshop. Gaithersburg, MD.

Balakirsky, S., Scrapper, C., & Messina, E. (in press). Mobility Open Architecture Simulation and Tools Environment. Proceedings of the International Conference Integration of Knowledge Intensive Multi-Agent Systems. Boston.

Blake, C. L. & Merz, C. J. (1998). UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Choi, D., Kaufman, M., Langley, P., Nejati, N., & Shapiro, D. (2004). An architecture for persistent reactive behavior. Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (pp. 988–995). New York: ACM Press.

Cohen, P. R. (1995). Empirical methods for artificial intelligence. Cambridge, MA: The MIT Press.

Jacoff, A., Messina, E., & Evans, J. (2001). Experiences in deploying test arenas for autonomous mobile robots. Proceedings of the Performance Metrics for Intelligent Systems Workshop. Mexico City, Mexico.

Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P. G., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. AI Magazine, 20, 27–41.

Kibler, D., & Langley, P. (1988). Machine learning as an experimental science. Proceedings of the Third European Working Session on Learning (pp. 81–92). Glasgow, Scotland: Pittman.

Langley, P. (October, 1996). Relevance and insight in experimental studies. IEEE Expert, 11–12.

Moriarty, D., & Langley, P. (1998). Learning cooperative lane selection strategies for highways. Proceedings of the Fifteenth National Conference on Artificial Intelligence (pp. 684–691). Madison, WI: AAAI Press.

Pell, B. (1996). A strategic Metagame player for general chess-like games. Computational Intelligence, 12, 177–198.

Shneier, M., Chang, T., Hong, T.H., Cheok, G., Scott, H., Legowik, S., & Lytle, A. (2003). A repository of sensor data for autonomous driving research. Proceedings of the SPIE Aerosense Conference. Orlando, FL.

Wang, J., Lewis, M., & Gennari, J. (2003). A game engine based simulation of the NIST USAR arenas. Proceedings of the 2003 Winter Simulation Conference (pp. 1039–1045). New Orleans, LA.

tions that it receives. Σ includes a set S of states, a set A of actions, a set E of events, and a state-transition function $\gamma : S \times A \times E \rightarrow 2^S$.

- A *controller*. Given as input the state s of the system (or more generally, some observations that give partial knowledge of the current state), the controller provides as output an action a according to some plan.
  $\eta : S \rightarrow O$ that maps S into some discrete set $O = \{o_1, o_2, \ldots\}$ of possible observations. The input to the controller is then the observation
- A *planner*: given as input a description of the system Σ, an initial situation and some objective, it synthesizes a plan for the controller in order to achieve the objective.

The planner's objective can be specified in several different ways.

1. The simplest specification consists of a *goal state* s_g or a set of goal states S_g; the objective is achieved by any sequence of state transitions that ends at one of the goal states.

2. More generally, the objective is to satisfy some condition over the sequence of states followed by the system; for example, one might want to require states to be avoided, states that the system should reach at some point, and states that it should stay in.

3. An alternative specification is through a utility function attached to states, with penalties and rewards, the goal being to optimize some compound function of these utilities, e.g. sum or maximum, over the sequence of states followed by the system.

4. Another alternative is to specify the objective as tasks that the system should perform. These tasks can be defined recursively, as sets of actions and other tasks.

3. RESTRICTIVE ASSUMPTIONS

The conceptual model in the last section was deliberately quite general, in order to provide a starting point for describing a number of restrictive assumptions:

- **Assumption A0 (Finite Σ).** The system Σ has a finite set of states.
- **Assumption A1 (Fully Observable Σ).** The system Σ is *fully observable*, i.e., one has complete knowledge about the state of Σ; in this case the observation function η is the identity function.
- **Assumption A2 (Deterministic Σ).** The system Σ is *deterministic*, i.e., for every state s and event or action u, $|\gamma(s, u)| \leq 1$. If an action is applicable to a state, its application brings a deterministic system to a single other state. Similarly for the occurrence of a possible event.
- **Assumption A3 (Static Σ).** The system Σ is *static*, i.e., the set of events E is empty. Σ has no internal dynamics; it stays in the same state until the controller applies some action.[1]
- **Assumption A4 (Attainment Goals).** The only kind of goal is an *attainment goal*, which is specified as an explicit goal state s_g or a set of goal states S_g. The objective is to find any sequence of state transitions that ends at one of the goal states. This assumption excludes, for example, states to be avoided, constraints on state trajectories, and utility functions.
- **Assumption A5 (Sequential Plans).** A solution plan to a planning problem is a linearly ordered finite sequence of actions.
- **Assumption A6 (Implicit Time).** Actions and events have no duration, they are instantaneous state transitions. This assumption is embedded in state-transition systems, a model that does not represent time explicitly.
- **Assumption A7 (Off-line Planning).** The planner is not concerned with any change that may occur in Σ *while* it is planning; it plans for the given initial and goal states regardless of the current dynamics, if any.

The simplest case, *classical planning*, combines all eight restrictive assumptions: complete knowledge about a deterministic, static, finite system with restricted goals and implicit time. Here planning reduces to the following problem:

> Given $\Sigma = (S, A, \gamma)$, an initial state s_0 and a subset of goal states S_g, find a sequence of actions $\langle a_1, a_2, \ldots, a_k \rangle$ corresponding to a sequence of state transitions $(s_0, s_1, \ldots, s_k)$ such that $s_1 \in \gamma(s_0, a_1)$, $s_2 \in \gamma(s_1, a_2)$, ..., $s_k \in \gamma(s_{k-1}, a_k)$, and $s_k \in S_g$.

Since the system is deterministic, if γ is applicable to s then $\gamma(s, a)$ contains one state s'. To simplify the notation, we will say $\gamma(s, a) = s'$ rather than $\gamma(s, a) = \{s'\}$. For this kind of system, a plan is a sequence $\langle a_1, a_2, \ldots, a_k \rangle$ such that $\gamma(\gamma(\ldots\gamma(\gamma(s_0, a_1), a_2), \ldots, a_{k-1}), a_k)$ is a goal state.

The assumption about complete knowledge is needed only at the initial state s_0, because the deterministic model allows all of the other states to be predicted with certainty. The plan is unconditional, and the controller executing the plan is an *open-loop controller*, i.e., it does not get any feedback about the state of the system.

Classical planning may appear trivial: planning is simply searching for a path in a graph, which is a well understood problem. Indeed, if we are given the graph Σ explicitly then there is not much more to say about planning for this restricted case. However, it can be shown [14] that even in very simple problems, the number of states in Σ can be many orders of magnitude greater than the number of particles in the universe! Thus

[1] The name of this assumption is inaccurate, because the plan is intended precisely to change the state of the system. What the name means is that the system remains static *unless controlled transitions take place.*

it is impossible in any practical sense to list all of Σ's states explicitly. This establishes the need for powerful *implicit* representations that can describe useful subsets of S in a way that both is compact and can easily be searched.

The simplest representation for classical planning is a *set-theoretic* one: a state s is represented as a collection of propositions, the set of goal states S_g is represented by specifying a collection of propositions that all states in S_g must satisfy, and an action a is represented by giving three lists of propositions: preconditions to be met in a state s for an action a to be applicable in s, propositions to assert and propositions to retract from s in order to get the resulting state $\gamma(s, a)$. A plan is any sequence of actions, and the plan solves the planning problem if, starting at s_0, the sequence of actions are executable, producing a sequence of states whose final state is in S_g.

A more expressive representation is the *classical representation*:[2] starting with a function-free first-order language L, a state s is a collection of ground atoms, and the set of goal states S_g is represented by an existentially closed collection of atoms that all states must satisfy. An operator is represented by giving two lists of ground or unground literals: preconditions and effects. An action is a ground instance of an operator. A plan is any sequence of actions, and the plan solves the planning problem if, starting at s_0, the sequence of actions are executable, producing a sequence of states whose final state satisfies in S_g. The *de facto* standard for classical planning is to use some variant of this representation.

4. CLASSICAL PLANNING VERSUS PLANNING APPLICATIONS

For nearly the entire time that automated planning has existed, it has been dominated by research on classical planning. For a while, the dominance was so complete that the term "domain-independent planning system" was used to refer to planning systems whose scope was that of classical planning, as if classical planning were capable of representing all possible planning domains.

In reality, it can be proved [14, Chapters 1–3] that classical planning systems are restricted to a very narrow class of planning domains. This class excludes most problems of practical interest, because most practical planning problems do not satisfy the restrictions of classical planning. Here are a few examples:

- **Process planning for machined parts.** Process planning is an important manufacturing task, and many millions of R&D dollars have been spent to try to automate it [23]. The state space consists of the possible states of the workpiece, including the workpiece geometry and various other parameters. The action space consists of the possible ways to modify the workpiece using machining operations. Both spaces are effectively infinite [17]. The actions have nondeterministic outcomes due to random variations—but in process planning the outcomes usually are approximated deterministically by the use of machining tolerances [9]. The planner must consult with CAD modelers to reason about the workpiece geometry, and must query databases to obtain information about the available machines, tooling, fixturing, and process parameters. With the exception of a few specialized process-planning tasks such as sheet-metal bending [16] and NC toolpath generation [28], generative process planning tools do not work very well and have not achieved significant industrial use. By far the most widely used process-planning tools are those that provide information to help expert humans do the process planning. Other approaches, e.g., [3, 8], illustrate the same trend for planning in other manufacturing applications.
- **Planning declarer play in bridge.** At the beginning of play in a bridge hand, the declarer (the player who chose the trump suit) needs to develop a plan for how to play the hand. The outcomes of the declarer's actions are uncertain, due both to uncertainty about how the opponents will respond and uncertainty about how they *might be able* to respond (since the declarer does not know which opponent holds which cards). A game tree containing all of the possibilities would have about 2.3×10^{24} leaf nodes on the average and about 5.6×10^{44} in the worst case [30, p. 226]. Since most bridge games are over in just a few minutes, it would not be feasible to explore any significant fraction of such a game tree. Instead, techniques have been developed that use various combinations of game-tree search, Monte Carlo simulation, and reasoning about possible strategies [12, 15, 31]. The resulting programs can play better than the average human bridge player, but not as good as the best human players.
- **Ship-movement planning.** Planning the movements of ships is important both commercially and militarily [11]. The state space and action space are effectively infinite: states include positions and velocities of ships, and actions correspond to movements of the ships along various routes. Since movements of different ships may occur concurrently, it is important to make sure they do not interfere with each other. The outcomes and durations of the actions cannot be known with certainty, because of factors such as weather, currents, and the behavior of the ships' operators. Elaborate simulation tools are available to aid in planning ship movements but the planning is still done manually [1]. Similarly, other transportation-planning applications, such as for railways [2], have focused on interactive approaches for planning.

Many other examples could easily be cited; see for example the PLANET repository's "Real-World Planning and Scheduling page" at ⟨http://vitalstatistix.nicve.salford.ac.uk/planet2⟩.

[2] This has also been called *STRIPS-style* representation), after an early planning system [27] that used a similar representation scheme.

5. EXISTING PERFORMANCE MEASURES

In this section, we do a quick survey of existing performance measures, and draw several conclusions about the limitations of those measures.

5.1. Survey

Performance measures for classical planners. The existence of a standard representation scheme for classical planning has made it relatively easy to develop large collections of planning problems on which different planning algorithms can be compared. In the three international planning competitions that have occurred so far [24, 4, 22], many hundreds of classical planning problems have been generated, from about fifteen different planning domains.[3] The most common performance measures have been *success rate*, *speed*, and *solution size*, i.e., the fraction of problems solved, the CPU time needed to solve them, and the size of the solution found (the latter two are normally measured as a function of the problem size). From these measures, one can get a rough idea of the size of the problems that a planner can solve in a reasonable amount of time.

A partial generalization. The 2002 International Planning Competition [22] included several collections of planning problems that did not satisfy all of the restrictions of classical planning. In these problems, Restrictions A0, A4, and A6 were weakened, by generalizing the planning language to include numeric computations and optimization goals.[4]

Although these generalizations may seem rather modest, they demonstrated some interesting things about the nature of classical planning, as discussed below.

For each of the planners in the competition, the planning engine was problem-independent, and the input for each planning problem included the initial state, the goal or objective to be achieved, and the set of operators for the problem domain. However, the planners varied in terms of how much additional knowledge was made available to them about how to solve problems in the planning domain. The planners in the competition can be classified into three categories:

- **Non-tunable planners.** In these planning systems, the problem input consists solely of the information specified above: initial state, goal or objective, and operators. In the competition, the planners in this class included most, but not all, of the ones that Long and Fox [22] have called "fully automated" planners.
- **Tunable planners.** Although these planning systems have usually been classified as "fully automated," there are ways to tune them for better performance in a given planning domain. In the 2002 competition, the planners in this class included LPG [13] and FF [18].[5] For LPG, one of the inputs was a setting to optimize its performance for speed, quality, or something in between, and LPG was run with all three settings during the competition. For FF, there were two different versions, both of which were entered in the competition.
- **Domain-configurable planners.** These are planning systems whose input includes detailed information about how to solve problems in the relevant problem domain. Such planners have sometimes been called "hand-tailored" planners [22], but that term is not accurate since the planning engine is domain-independent. They have also been described as "hand-tailorable" [26] or "control-intensive" [5] planners. In the competition, the planners of this type included SHOP2 [26], TLPlan [5], and TALplanner [19].

Performance measures for application-specific planners. For application-specific planning systems, usually the performance measures and the ways of testing them are also application-specific. For example, manufacturing-planning systems are tested on collections of manufacturing-planning problems that are specific to the particular domain in which the planning is done (e.g., see [29]); and in computer bridge [31], there are annual competitions in which performance is measured by playing the programs against each other on a set of bridge hands, using the normal rules for a bridge tournament. These kinds of measures are useful for the application domain at hand, but they are not directly generalizable to other domains.

5.2. Observations

From the survey in the previous section, we can make the following observations.

Observation 1: *There is a tradeoff between the amount of work needed to configure a planner for a domain, and planner's speed and coverage of that domain once it has been so configured.* Here are several examples:

- In the planning competitions, the non-tunable planners were the ones that had the highest running time and solved the fewest planning problems—but configuring a non-tunable planner requires no workwhatsoever, provided that the planner is capable of representing the planning domain.
- In the planning competitions, the tunable planners were faster than the fully automated ones. However, some experimentation may be required to find the settings that give the best overall performance.

[3] In classical planning, a *domain* is basically a set of planning operators. For each domain it is possible to produce an unlimited number of randomly generated problems by specifying initial and goal states.

[4] In the 2004 International Planning Competition, which was in progress at the time that we wrote this paper, some of the restrictions have been weakened further. For details, see ⟨http://www-rcf.usc.edu/~skoenig/icaps/icaps04/planningcompetition.html⟩.

[5] Some of the other planners in the competition may also be capable of being tuned, but LPG and FF were the only ones for which results were submitted using more than one setting or version.

- In the planning competitions, the domain-configurable planners solved planning problems several orders of magnitude faster than the others, and solved many problems that were too large for the other planners to solve. However, the domain-configurable planners require a significant amount of up-front work to formulate the domain-specific knowledge that enables them to run so quickly, and this work must be redone each time one switches to a new domain.
- In order to get top-level performance in a specific application domain, it may be necessary to develop a domain-specific planner.[6] However, developing and tuning such planners may require years of work. The resulting planning system may be quite good for its particular application domain, but cannot be used to solve problems in any other domain.

Observation 2: *Performance in classical planning domains does not predict performance in other planning-competition domains.* For example:

- Some of the planning systems were designed, sometimes consciously and sometimes tacitly, with classical planning in mind. These planners did well on classical domains, but on non-classical domains they did not perform very well (if they could be used at all).
- On the other hand, some of the planning systems were designed, from the ground up, to work on non-classical planning domains. These systems generally performed well on both the classical and non-classical domains.

Observation 3: *Performance in planning-competition domains does not predict performance in real-world application.* For example:

- Most of the planning systems in the competition, including both good and bad performers, would not be directly usable in real-world applications, because of restrictions on the kinds of planning problems that they can solve.
- A planner that performed poorly in the 2002 planning competition, IxTetT [20], is used quite successfully for the application of robot motion planning [21], a domain which most of the systems in the competition would be unable to address.
- One of the best performers in the 2002 planning competition, SHOP2 [26], is also proving useful in several application areas. It is developing a user base that includes universities, companies such as Sony, Lockheed Martin, and SIFT, and government laboratories such as NIST and NRL.

From the above observations, we conclude that it is not adequate merely to measure running time and percentage of problems solved. Such figures are not meaningful unless one also knows the class of planning problems over which such performance can be achieved, and how much the performance will be degraded on broader classes of planning problems.

[6] Some examples of such systems include Bridge Baron for computer bridge [31], the Intelligent Bending Workstation for sheet-metal bending [16], and RAX for autonomous spacecraft control [25].

6. A PROPOSED PERFORMANCE MEASURE

In this section, we discuss three different aspects of a planning system's performance that we believe are important to measure: the scope of the problems that the planner can solve, the amount and kind of control knowledge that must be given to the planning system, and the size of the problems that the planning system can reasonably solve.

6.1. Problem Scope

We believe that any useful measure of performance for a planning system needs to include the *scope* of the problems that the corresponding planning algorithm is capable of solving. The set of restrictive assumptions in Section 2 can be used as a basis for defining what this scope is. More specifically:

- **Relaxing Assumption A0 (Finite Σ).** An enumerable, possibly infinite set of states may be needed, for example, to describe actions that construct or bring new objects in the world, or to handle numerical state variables. This brings in some theoretical issues about decidability and termination.
- **Relaxing Assumption A1 (Fully Observable Σ).** If we allow a static, deterministic system to be partially observable, then the observations of Σ will not fully disambiguate which state Σ is in. For each observation o, there may be more than one state s such that $\eta(s) = o$. Without knowing which state in $\eta^{-1}(o)$ is the current state, it is no longer possible to predict with certainty whether an action is applicable and what state Σ will be in after each action.
- **Relaxing Assumption A2 (Deterministic Σ).** In a static but nondeterministic system, each action can lead to different possible states, so the planner may have to consider alternatives. Usually nondeterminism requires relaxing Assumption A5 as well. A plan must encode ways for dealing with alternatives, e.g., *conditional* constructs of the form "do a and, depending on its result, do either b or c", and *iterative* constructs, like "do a until a given result is obtained." Notice that the controller has to observe the state s: here we are planning for a *closed-loop control*.
  If the complete knowledge assumption (Assumption A1) is also relaxed, this leads to another difficulty: the controller does not know exactly the current state s of the system at run-time. A limiting case is *null observability*, where no observations at all can be done at run-time. This leads to a particular case of planning for open-loop control called *conformant planning*.
  Some ways of dealing with nondeterminism are extensions of techniques used in classical planning (such as Graph-based or SAT-based planning), while others are designed specifically to deal with nondeterminism, such as planning based

on Markov Decision Processes (MDPs) [7, 14] or model-checking techniques [10, 14].

- **Relaxing Assumption A3 (Static Σ).** We can easily deal with a dynamic system Σ if it is deterministic and fully observable, and if we further assume that for every state s there is at most one contingent event e for which $\gamma(s, e)$ is not empty, and that e will necessarily occur in s. Such a system can be mapped into the restricted model: one redefines the transition for an action a as $\gamma(\gamma(s, a), e)$, where e is the event that occurs in the state $\gamma(s, a)$.
  In the general model of possible events that may or may not occur in a state and "compete" with actions, a dynamic system is nondeterministic from the view point of the planner even if $|\gamma(s, u)| \leq 1$, u being either an action or an event. Deciding to apply action a in s does not focus the planner's prediction to a single state-transition. Here again, a conditional plan will be needed.
- **Relaxing Assumption A4 (Restricted Goals).** Controlling a system may require more complex objectives than reaching a given state. One would like to be able to specify to the planner an *extended goal* with requirements not only on the final state but also on the states traversed, e.g., critical states to be avoided, states that the system should go through, states it should stay in and other constraints on its trajectories. It may also be desirable to have utility functions to be optimized, e.g., to model a system that must function continuously over an indefinite period of time.
- **Relaxing Assumption A5 (Sequential Plans).** Here, a plan may be a mathematical structure that can be richer than a simple sequence of actions. As examples, one may consider a plan to be a partially ordered set, a sequence of sets, a conditional plan that forces alternate routes depending on the outcome and current context of execution, a "universal plan" or a "policy" that maps states to appropriate actions, or a deterministic or nondeterministic automaton that determines what action to execute depending on the previous history of execution. Relaxing Assumption A5 is often required when other assumptions are relaxed, as we have seen in the case of nondeterministic systems (Assumption A3) or when relaxing Assumptions A1, A3, A4 and A6. Plans as partially ordered sets, or as sequences of sets of actions, are more easily handled than conditional plans and policies.
- **Relaxing Assumption A6 (Implicit Time).** In many planning domains, action duration and concurrency have to be taken into account. Time can also be needed for expressing temporally constrained goals and occurrence of events with respect to an absolute time reference. However, time is abstracted away in the state-transition model.[7] This conceptual model considers actions or events as instantaneous transitions: at each clock tick, the controller synchronously reads the observation for the current state (if needed) and applies the planned action.
- **Relaxing Assumption A7 (Offline Planning).** The control problem of driving a system towards some objectives has to be handled online with the dynamics of that system. While a planner may not have to worry about all the details of the actual dynamics, it cannot ignore completely how the system will evolve. At the least, it needs to check, online, whether a solution plan remains valid, and, if needed, to revise it or replan. Other approaches consider planning as a process that modifies the controller online.

For a detailed presentation of techniques for solving planning problems with various combinations of these restrictions, see [14].

6.2. Control Knowledge

Another important aspect of a planning system's performance is what kind of additional control knowledge (other than just the problem definition) will need to be given to the planning system in order for it to address practical problems. This includes, for example, whether the planner needs such knowledge, how precise and specific to a problem the knowledge needs to be, whether the planner needs to be fine-tuned for different planning domains, and how easily this knowledge can be acquired and formalized. It would be quite difficult to express this feature in precise quantified measurements, but a qualitative assessment of this feature can be made, on the basis of a small set of predefined classes ranging from planners that require no control knowledge to those that require the domain author to do some highly demanding algorithm development.

6.3. Problem Size

A third important aspect of performance is what size of problem a planning system can reasonably solve. For this performance aspect, the traditional measures have been numeric ones, along the lines of "this planner can solve problems of size n in time t" for various values of n and t. This has typically been measured by running the planner on a randomly generated set of planning problems.

Such a performance measure has an obvious appeal, but as we concluded in the preceding section, it also has an important limitation: it is highly biased by theset of benchmark problems on which the planner is tested. If a planning system can solve "toy problems" in which the solution plans contain hundreds or even thousands of actions, this does not necessarily say anything about how well—or even whether—the system can solve more useful classes of planning problems.

A more useful way of measuring performance would be to use several classes of problems, ranging in scope from toy problems to very demanding applications, and measure performance in each class.

[7] Other formalisms, such as *timed automata*, extend state-transition systems by incorporating an explicit representation of time.

6. CONCLUSION

In this paper, we have described existing performance measures for automated planning algorithms, and have discussed the limitations and biases inherent in those performance measures. We have pointed out the importance of developing a performance measure that explicitly the restrictive assumptions on which a planning algorithm depends—and as initial step toward such a performance measure, we have defined and discussed a list of restrictive assumptions that are common to most automated planning systems. We believe that this list provides an initial step toward developing a *taxonomy of restrictions* that can be used to measure the scope of planning algorithms.

Based on the above considerations, we have proposed a composite performance measure based on three factors:

- the scope of the planning algorithm: which set of restrictive assumption are needed and which can be lifted,
- the control knowledge and tuning required for each planning domain,
- the size of the problems that can be solve in a reasonable amount of time in each area of its scope (i.e., for each combination of relaxed assumptions it can handle).

Several aspects of this performance measure are not yet (or not yet fully) developed, and we hope that this paper will encourage researchers to make the effort needed to develop them.

ACKNOWLEDGEMENT

This work was supported in part by the CNRS, France, and in the US by AFRL contract F30602-00-2-0505 and NSF grant IIS0412812. The opinions expressed in this paper are those of authors and do not necessarily reflect the opinions of the funders.

REFERENCES

[1] D. W. Aha. Plan deconfliction, repair, and authoring in EDSS. Technical report, Naval Research Laboratory, 2002. Progress report.

[2] J. Allen and G. Ferguson. Human-machine collaborative planning. In *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space*, 2002.

[3] R. Aylett, J. Soutter, G. Petley, P. W. H. Chung, and A. Rushton. Ai planning in a chemical plant domain. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 1998.

[4] F. Bacchus. The AIPS '00 planning competition. *AI Magazine*, 22(1):47–56, 2001.

[5] F. Bacchus. The power of modeling—a response to PDDL2.1. *Journal of Artificial Intelligence Research*, 20:125–132, 2003.

[6] D. Bernard, E. Gamble, N. Rouquette, B. Smith, Y. Tung, N. Muscettola, G. Dorias, B. Kanefsky, J. Kurien, W. Millar, P. Nayak, and K. Rajan. Remote agent experiment. ds1 technology validation report. Technical report, NASA Ames and JPL report, 1998.

[7] C. Boutilier, T. L. Dean, and S. Hanks. Planning under uncertainty: Structural assumptions and computational leverage. In Ghallab and Milani, editors, *New Directions in AI planning*, pages 157–171. IOS Press, 1996.

[8] L. Castillo, J. Fdez-Olivares, and A. Gonzlez. Automatic generation of control sequences for manufacturing systems based on partial order planning techniques. *Artificial Intelligence in Engineering*, 4(1):15–20, 2000.

[9] T.-C. Chang. *Expert Process Planning for Manufacturing*. Addison-Wesley, Reading, MA, 1990.

[10] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84, 2003.

[11] DOT. An assessment of the u.s. marine transportation system, a report to congress. Technical report, U.S. Department of Transportation, 1999. 103 pages.

[12] I. Frank and D. A. Basin. Search in games with incomplete information: A case study using bridge card play. *Artificial Intelligence*, 100(1-2):87–123, 1998.

[13] A. Gerevini, A. Saetti, and I. Serina. Planning through stochastic local search and temporal action graphs in lpg. *Journal of Artificial Intelligence Research*, 20:239–290, 2003.

[14] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[15] M. L. Ginsberg. Partition search. In H. Shrobe and T. Senator, editors, *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 228–233, Menlo Park, California, 1996. AAAI Press.

[16] S. K. Gupta, D. A. Bourne, K. Kim, , and S. S. Krishanan. Automated process planning for sheet metal bending operations. *Journal of Manufacturing Systems*, 17(5):338–360, 1998.

[17] S. K. Gupta, W. C. Regli, and D. S. Nau. Manufacturing feature instances: Which ones to recognize? In *ACM Solid Modeling Conference*, 1995.

[18] J. Hoffmann. The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *Journal of Artificial Intelligence Research*, 20:291–341, 2003.

[19] J. Kvarnstrm and M. Magnusson. TALplanner in IPC-2002: Extensions and control rules. *Journal of Artificial Intelligence Research*, 20:343–377, 2003.

[20] P. Laborie and M. Ghallab. Planning with sharable resource constraints. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1643–1649, 1995.

[21] S. Lemai and F. Ingrand. Interleaving temporal planning and execution in robotics domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2004. To appear.

[22] D. Long and M. Fox. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research*, 20:1–59, 2003.

[23] M. Mantyla, D. S. Nau, and J. Shah. Challenges in feature-based manufacturing research. *CACM*, 39(2):77–85, 1996.

[24] D. McDermott. AIPS-98 planning competition results, 1998.

[25] N. Muscettola, P. P. Nayak, B. Pell, and B. C. Williams. Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1-2):5–47, 1998.

[26] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, W. Murdock, D. Wu, and F. Yaman. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, 20:379–404, December 2003.

[27] N. Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann, 1980.

[28] Parametric Technology Corporation. Pro/ENGINEER NC Sheetmetal. ⟨http://www.ptc.com/appserver/it/icm/cda/icm01_list.jsp?group=2%01&num=1&show=y&keyword=342⟩, 2004.

[29] J. Shah, M. Mantyla, and D. S. Nau, editors. *Advances in Feature Based Manufacturing*. Elsevier/North Holland, 1994.

[30] S. J. J. Smith. *Task-Network Planning Using Total-Order Forward Search, and Applications to Bridge and to Microwave Module Manufacture*. PhD thesis, University of Maryland, 1997.

[31] S. J. J. Smith, D. S. Nau, and T. Throop. Computer bridge: A big win for AI planning. *AI Magazine*, 19(2):93–105, 1998.

A Flexible Approach to Quantifying Various Dimensions of Environmental Complexity

Michael L. Anderson
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742, USA.
Tel: (301) 405-1746 Fax: (301) 314-1353
Email: anderson@cs.umd.edu

ABSTRACT— In this paper I propose a flexible method of quantifying various dimensions of the complexity of a test environment, including its information density, variability, volatility, inconsistency, and uncertainty. This allows one to determine the task performance of intelligent agents as a function of such measures, and therefore permits derivative measures of their perturbation tolerance—that is, their ability to cope with a complex and changing environment.

Keywords:
complexity, perturbation tolerance, standards, testbeds

1. INTRODUCTION

There are many important and valid approaches to measuring the performance of intelligent systems: provable optimality; typical (and worst-case) solution times; speed and throughput, to name just a few. However, if one is interested, as my research group is, in complex systems operating in changing environments with time and resource constraints—that is, environments where there may *be* no optimal solution, or no time to calculate one, and where experience may confound expectations in significant ways, and thus where finding *some* way to accomplish something is more important than finding the theoretically best way—then such standard measures of performance may not capture the most interesting and crucial elements of performance.

As is well known, maintaining adequate performance in complex and changing environments has been a perennial stumbling-block for intelligent systems, and an ongoing challenge to AI. A typical AI system designed for a specific task often fails utterly when circumstances take it even slightly outside its task specifications. Thus, the ability to handle unexpected difficulties, even if non-optimally—that is, the ability to muddle through difficult situations without breaking down—seems to us an ability worth specific study, and, if possible, implementation.

We call this general ability to cope with a complex and changing environment "perturbation tolerance". The term is meant as an extension and generalization of John McCarthy's notion of "elaboration tolerance"—a measure of the ease with which a reasoning agent can add and delete axioms from its knowledge base [1]. However, our term is more general than McCarthy's because his is explicitly limited to formal, symbolic systems, and an elaboration is defined as an action taken to change such a system [2]. But since a given intelligent agent may well consist of more than just a formal reasoning system, and flexibly coping with a changing world may therefore involve altering components in addition to, or instead of, its formal reasoner, we define a perturbation as any change, whether in the world or in the system itself, that impacts the performance of the agent. Performance is meant to be construed broadly to encompass any measurable aspect of the agent's operation, although, as will be explained below, we tend to favor measures for such things as average reward and percentage task completion over such things as reasoning speed or throughput. *Perturbation tolerance*, then, is the ability of an agent to quickly recover—that is, to re-establish desired/expected performance levels—after a perturbation.

However, if improving perturbation tolerance is to be among the goals for intelligent agents, it will be necessary to quantify and measure this aspect of performance. And if perturbation tolerance is primarily a matter of maintaining performance in the face of various kinds of complexity and change, then it is just such complexity and change that should be the focus of measurement. Further, it would be best if, instead of each lab and working group devising their own set of standards, there were a common standard, and preferably one that might be applied to say something useful about current testbeds. For instance, is Phoenix [3], [4] more or less complex than Tileworld [5]? And how can one compare the relative complexity of two different Tileworld runs?

To this end, I suggest a way to specify an environment that allows for such factors as its complexity, informa-

tion density, variability, volatility and uncertainty to be measured. From such measures I show how derivative measures of environmentally-relative task difficulty and degree of perturbation can be developed, and suggest some different metrics for measuring task performance.

2. COMPARISON WITH RELATED WORK

First, as mentioned already above, it should be made clear that while the approach defined here *could* be used to build new testbeds, it can *also* be used to characterize the properties of existing testbed environments in a uniform way. Thus, what is offered here is less a blueprint for new standard testbed implementations, and more a suggestion for a standard way of measuring some important properties of the testbed environments within which intelligent agents operate. It is perhaps worth noting that the lack of a standard way to evaluate intelligent agents has prompted DARPA to modify their Cognitive Information Processing Technology research initiative to include Cognitive Systems Evaluation as a focal challenge.[1]

One weakness of some domain specifications, from the standpoint of evaluating perturbation tolerance, is that they focus on controlling the characteristics and interactions of the agents *in* the world, rather than on fine control of the world itself. In MICE, for instance [6], the main goal was "an experimental testbed that does not simulate any specific application domain, but can instead be modified to impose a variety of constraints on how agents act and interact so that we can emulate the different coordination issues that arise in various application domains." This strategy is, of course, perfectly sensible when it is the *coordination strategies* of multi-agent systems that is under investigation, but it provides little foundation for measures of perturbation tolerance *per se*.

Another weakness of some domain specifications is the limited number of environmental features that can be easily isolated and measured. For instance, the Phoenix testbed [3], [4] offers ways of building complex and dynamic environments (in which the main task is fighting forest fires), but does not offer a general method for measuring the complexity and dynamicity of those environments. Even what is perhaps the most popular and adjustable of the standard test domains for simulated autonomous agents, Tileworld [5], suffers somewhat from this defect. The main task in Tileworld is to fill holes with tiles, quickly and efficiently, while avoiding obstacles. Among the strengths of Tileworld is its ability to easily measure the performance trade-off between deliberation and reactivity. Tileworld allows one to set the value of such environmental variables as the frequency with which objects appear and disappear, the number and distribution of objects, and the reward value for filling each hole. However, as important as these environmental variables are, there are also other aspects of an environment with which an intelligent agent must cope, and against which performance should be measured. In addition, it is not clear how to translate the variables governing Tileworld to those governing other environments. Finally, Tileworld tests only planning (and plan implementation) performance. But intelligent agents may also need to be able to perform such tasks as the inference-based categorization or identification of objects; the communication of accurate information about an environment; and the mapping of stable environmental features. The current proposal, in providing a more general approach to measuring environmental complexity, aims to lay a foundation for measuring performance in these tasks as a function of the complexity of the environment, and to make cross-domain and even cross-task comparisons easier.

[1] http://www.darpa.mil/baa/baa02-21mod6.htm

3. COMPLEXITY METRICS

It is proposed that the environment be modeled as an n-dimensional grid[2] with a large number of propositions (including sets of numeric values and node activations, to simulate the operation of perceptual NNs, sonar, etc.) that can characterize each location, or "square", in the grid. Each square may be adjacent to (accessible from) one or more other squares. Each proposition p might or might not hold in each square s. As s comes into the perceptual range of the agent, it "picks up" on the propositions that characterize it (propositions consisting of numeric values "stimulate" the appropriate perceptual systems directly; symbolic propositions are entered directly into the agent's knowledge base (KB), and might be thought of as the sort of structured representations that would typically be delivered to an intelligent system by a complex perceptual system like vision).[3] The combination of a grid of a certain size and shape with its characterizing propositions is called an *overlay* (O).

Any given environment has many different features that determine its complexity, independent of the task to be performed in that environment. Specifying the environment in the terms given above allows one to measure these features as follows.

3.1. Basic Measures

[2] For a discussion of the wide applicability of this model, see the subsection on Generality and Extensibility, below.

[3] It is perhaps worth emphasizing that the only propositions relevant to the specification are those characterizing features of the environment that the agent would be expected to perceive or otherwise pick up. The number of water atoms at a given location would not be a relevant proposition unless the agent in question is capable of seeing and counting water atoms. Note the implication that the more perceptually sophisticated the agent, the richer its domain.

n (overlay size): the number of squares in the overlay. If the number of squares changes during the course of an experiment, this will naturally have to be reflected in the measure; whether it is best to use the average size, the final size, or some other measure may depend on the details of the experiment. Note that to choose the number of squares for an environment is also to choose the spatial granularity of the environment. There can be some hidden difficulties here, for instance in the case where different tasks, e.g. navigating a hallway or picking a lock, require that an agent divide space more or less finely. It may the that in these cases, it will be best to treat moving from one task to another in terms of moving from a low-granularity overlay to a high-granularity one.

ρ_I (information density): the average number of propositions characterizing each square.

V_o (variability): a measure of the degree of difference in the characterizing propositions from square to square. V_o can be calculated as the sum of the propositional difference between each pair of squares in the overlay divided by their geometric (minimum graph) distance:

$$V_o = \sum_{i,j=1}^{n} \frac{D_p(s_i, s_j)}{G(s_i, s_j)} \tag{1}$$

Where $D_p(s_i, s_j)$ is the number of propositions that hold in s_i but not in s_j and vice-versa; $G(s_i, s_j)$ is the distance between the squares and n is the total number of squares in the overlay.

δ_o (volatility): a measure of the amount of change in the overlay as a function of time. δ_o can be measured in a way similar to V_o, except that rather than measure the propositional difference as a function of geographical distance, we measure it as a function of temporal distance.

$$\delta_o = \sum_{i,j=1}^{n,t} \frac{D_p(s_{i,1}, s_{i,j})}{j} \tag{2}$$

Where $D_p(s_{i,1}, s_{i,j})$ is the number of propositions that hold in s_i at time 1, but not in s_i at time j, and vice-versa; t is the total time of the simulation, and n is the number of squares in the overlay.

I (inconsistency): the amount of direct contradiction between the beliefs of an agent (in its KB) and the propositions characterizing the environment. Note this must be a measure of the number of *direct* contradictions between p and $\neg p$, since the inconsistency of any two sets of propositions is in general undecidable [7].[4] I can be measured as the percentage of propositions initially in the overlay that directly contradict elements of the agent's initial KB (e.g., 2%, 5%, 10%, 15%, 25%). In the case where $\delta_o > 0$, a more accurate measure might be the *average* percentage of propositions, over time, that directly contradict elements of the initial KB. Note, however, that this measure should *not* reflect the percentage of direct contradiction between the environment over time *and the KB over time*. I is meant to be a measure of one kind of difficulty an agent might face in its environment, that it needs to overcome (or at least manage) in order to successfully cope with that environment. Thus, only the *initial* KB should be used to determine I, for if, through the efforts of the agent, I approaches zero as the test run proceeds, this is a measure of the success of the agent, and does not represent a reduction of the difficulty of the task the agent faced.

U (uncertainty): a measure of the difficulty of perceiving the contents of the square correctly. Uncertainty can be understood as the ratio of the average number of "false" propositions (p_f) in each square to the average total number of propositions in each square.

$$U = p_f / \rho_I \tag{3}$$

In the case where one is building a testbed, uncertainty requires the designer to seed the squares with false or inapplicable propositions, or perhaps, after assigning propositions to each square, to replace a certain number of them with their negations. In the case of modeling an existing testbed, or where using numeric values rather than propositions, if the percentage of time that the system will make perceptual errors is known, this number can be used here.

D_o (overlay difference): a measure of the propositional difference between two overlays O_1 and O_2. D_o can be measured as the sum of the propositional differences between the corresponding squares of each overlay.

$$D_o = \sum_{i=1}^{n} (s_{O_1,i}, s_{O_2,i}) \tag{4}$$

Two overlays may have precisely the same information density, variability and volatility, and still be charac-

[4] A practical aside: work with Active Logic shows that although an indirect contradiction may lurk undetected in the knowledge base, it may be sufficient for many purposes to deal only with direct contradictions. After all, a real agent has no choice but to reason only with whatever it has been able to come up with *so far*, rather than with implicit but not yet performed inferences. Active Logic systems have been developed that can detect, quarantine, and in some cases automatically resolve contradictions [8]–[13].

terized by different propositions; hence this measure of overlay difference. This is useful for cases where an agent is to be trained in one overlay, and tested in another, and the question is how much the differences in the test and target domains affect performance.

It is not expected that every testbed, nor every test run, will make use of all these measures of environmental complexity. Depending on the capabilities of the testbed, and on what is being tested at the time, only a few of these measures may be appropriate. Note further that, depending on the task, some of these measures can simulate others. For instance, even in a completely stable environment ($\delta_o = 0$), the agent can experience the *equivalent* of volatility if $V_o > 0$, for as it traverses the environment each square will offer different information. This difference may not affect the agent at all if its sole task is to map the environment, but it could make an inference-based task more difficult in the same way that a changing environment would. Likewise for the case where $I > 0$, for as the agent encounters these contradictions, they can offer the *equivalent* of change, since change can be understood in terms of p being true at one time, and not true at another. Naturally, determining what manner of variation affects what tasks, and by how much, is one of the items of empirical interest to AI scientists. Isolating these different kinds of complexity and change can help make these determinations more specific and accurate.

3.2. Derivative Measures

The basic measures discussed above can be combined in various ways to construct any number of derivative measures. One such measure of particular importance is of the overall complexity of the environment.

C (complexity): a measure of the overall complexity of the environment. C can be defined as the product of all the non-zero basic measures:

$$C = n \times \rho_I \times V_o \times \delta_o \times (I + 1) \times 100U \qquad (5)$$

The intuition behind this compound measure of complexity is that there are in fact many different reasons that an environment might be difficult to cope with, all of which, therefore, can be considered to contribute in some way to the overall complexity of the environment itself, or to a measure of the environment's contribution to the difficulty of tasks to be performed there. For instance, a large environment is in some sense more complex than a small one *ceteris paribus*, just because there is more of it to deal with. After all, mapping or locating objects in a large environment is likely to be *harder* than doing it in a small one. Likewise, information density captures the notion that a more *intricate* environment—one that requires a greater number of propositions to describe—will be harder to reason about or deal with than a less intricate one. Sometimes this will mean that an intelligent agent has more to think *about* in trying to act in a more intricate environment, and sometimes this will mean it has more to *ignore*; both can be difficult. The variability and volatility of an environment expresses the intuition that an environment that remains more or less the same from place to place, and from time to time, is simpler than one that does not. Inconsistency expresses the idea that an environment that is very different from one's expectations will be harder to deal with than one that is not, and, similarly, uncertainty captures the fact that if it is harder (for whatever reason) to correctly perceive an environment, then certainly coping with it will also be more difficult. The overlay difference allows one to quantify the notion that moving between different domains can be difficult (and is likely to be more difficult as a function of the difference).

It may well turn out, after further consideration, both that there are *more* factors important to the complexity of an environment, and that each factor contributes to a measurably different *degree* to overall complexity (something that might be expressed by adding various coefficients to equation 5). Likewise, perhaps it will turn out that more accurate expression of overall complexity results from *adding* rather than *multiplying* all or some of the various factors. I would welcome such future developments as improvements of the preliminary suggestions I am offering here. Ultimately, an evaluation of the usefulness of these measures will require, and suggestions for improvement will certainly result from, their attempted application in evaluating the performance of intelligent agents in increasingly complex environments. My hope is only that they are well-specified enough in their current form to lend themselves to such use.

3.3. Generality and Extensibility

I have characterized the test environment in terms of a grid of squares of a certain size and shape. Naturally, such a characterization is most directly applicable to artificial environments in fact composed of such a grid ("grid worlds"). However, it should be noted that whenever it is possible to divide a domain into parts, and characterize (the contents of) those parts in terms of some set of propositions, in the sense defined above, then it should therefore be possible to characterize and measure the complexity of that domain in the terms set forth here. We might call such domains "grid-available".

One obvious case of a grid-available domain is one consisting of a mappable terrain (or space) with discrete, localizable features. There are very many domains of this sort, including those, like the world itself, that are not naturally structured according to a grid, i.e. that are

continuous. It is nevertheless possible, albeit with some abstraction, to usefully divide such a domain into spatial parts, and characterize the features of each part in terms of a set of propositions.

Another class of domains that are grid-available are those that, while not strictly-speaking spatial, nevertheless consist of individualizable information-parts. A database is one such domain, and the World Wide Web is another. In each case, the domain consists of individual parts (records, pages), with specifiable contents, that may be adjacent to (linked to, accessible from) one or more other part(s). Depending on the needs of the experiment, an "overlay" might be defined as an entire database or set of web-pages, or some particular subset, as for instance the recordset returned by a given query.

Finally, well-specified state spaces are also grid-available domains. Each state corresponds to a "square" in the grid, and the agent can take actions that move it between states. The states themselves can be characterized in terms of some set of propositions.

Examples of domains that are not grid-available include truly continuous or holistic domains that cannot be usefully broken into parts and/or have few or no local properties (all properties are properties of the whole). Domains described at the quantum level appear to be of this sort, as global quantum properties are often not determined by local ones, making analysis of the parts far less useful than in classically described domains.

4. SAMPLE PERFORMANCE METRICS

In keeping with the philosophy that *flexibility* and *adaptability*—an ability to get along even in difficult circumstances—are among the paramount virtues of cognitive agents, we suggest that evaluating task performance is more important than evaluating such things as reasoning speed, throughput, or the degree of consistency in a post-test KB. Indeed, for an intelligent agent it may be that maintaining a consistent database is in general less important than being able to deal effectively with contradictions while continuing to operate in a dynamic environment.[5] Consider, for instance, a target location task, where the agent must traverse an environment containing 100 targets (lost hikers, for instance) and find them all as quickly as possible. A simple measure of performance here might be:

$$M = \frac{(TP)}{(tA)} \tag{6}$$

[5]This is because, for any sufficiently complex knowledge base that was not produced by logical rules from a database known to be consistent, and/or to which non-entailed facts are to be added (e.g. from sensory information), it will not be possible to know whether it is consistent, nor to use principled methods to maintain consistency [7]. Thus, contradictions are in this sense practically inevitable.

where T is the number of targets correctly identified,[6] A is the percentage of environmental area covered at the time the measurement is taken (this allows a measure of M to be taken at any time in the run, e.g., when $A = 0.25, A = 0.5, A = 0.75$ etc.), t is time elapsed, and P is the percentage of task completion (percentage of targets, out of all 100, correctly identified). Because a low performance time is generally only desirable when task completion is high, t is divided by P to penalize fast but sloppy performers.

In the case where the identification of the target is inference-based, and therefore liable to error (for instance, the agent has to tell the difference between lost hikers, park rangers, and large animals), tracking not just correct target IDs (True Positives, or TP) but also False Positives (FP), False Negatives (FN), and True Negatives (TN) will allow one to use the following standard performance metrics:

Sensitivity = $\frac{TP}{TP+FN}$

Specificity = $\frac{TN}{TN+FP}$

PPV (Positive Predictive Value) = $\frac{TP}{TP+FP}$

NPV (Negative Predictive Value) = $\frac{TN}{TN+FN}$

Although the bare metric M, and the measures for sensitivity, specificity, PPV and NPV, give one straightforward way to compute the performance of a given agent, and to compare the performance of different systems, when one is dealing with intelligent agents that can *learn*, it is also very important to measure the *change* in performance over time, and as a function of increased environmental complexity. Successive M values can be compared to assess the learning or improvement rate of the system. Likewise, successive values for the environmental complexity measures can be used to assess the agent's improving ability to handle increased environmental difficulty, for instance:

Ct (avg. complexity tolerance) = $\frac{\Delta C}{\Delta M}$

$V_o t$ (avg. variability tolerance) = $\frac{\Delta V_o}{\Delta M}$

$\delta_o t$ (avg. volatility tolerance) = $\frac{\Delta \delta_o}{\Delta M}$

$D_o t$ (avg. domain flexibility) = $\frac{\Delta D_o}{\Delta M}$

Similar metrics can of course be used for measuring changes in sensitivity, specificity, PPV, and NPV as a function of task complexity. These various measures taken together can give a clear picture of the perturbation tolerance of a given cognitive agent.

Finally, because the special abilities possessed by some intelligent agents, such as getting advice, reorganizing one's KB, or changing one's conceptual categories, can be very time-consuming, their worth depends a great deal on the value of accuracy -vs- the need for quickness in a given task. Thus in many cases it is sensible to introduce

[6]The variable T might also be calculated as correct IDs minus incorrect IDs ($TP - FP$, see below).

the domain variable R_V, a subjective measure of the importance of accuracy in the current task-domain. Although the variable R_V does not actually change anything about the domain itself, it can be used to inform the agent about the characteristics of its task. For the autonomous agent with complex cognitive abilities, and the ability to measure and track its *own* performance, R_V can provide a threshold measure as to when (and when not) to stop and ponder.

5. IMPLEMENTATION AND APPLICATION

A general test domain—PWorld—allowing for relatively easy characterization according to the suggested standard has been implemented as a component object model (COM) object on Microsoft Windows. PWorld is an $n \times n$ grid, and all elements of the world, including characterizing propositions, are stored and tracked in a database, with which PWorld communicates using ActiveX Data Objects (ADO). Active elements of the world—e.g. agents, weather, and such things as plants that can wither or grow—are implemented as separate COM objects that can communicate directly with the world, and indirectly with other active elements, by calling PWorld's various methods, such as: **addProposition()**, **sense()**, **move()**, and **eat()**.

PWorld was recently used to measure the perturbation tolerance of an agent using a standard reinforcement learning technique (Q-learning), and to compare it to the perturbation tolerance of an agent using a version of Q-learning that was enhanced with simple metacognitive monitoring and control (MCL) to create a *very simple* cognitive agent. The basic idea behind Q-learning is to try to determine which actions, taken from which states, lead to rewards for the agent (however these are defined), and which actions, from which states, lead to the states from which said rewards are available, and so on. The value of each action that could be taken in each state—its Q-value—is a time-discounted measure of the maximum reward available to the agent by following a path through state space of which the action in question is a part.

The Q-learning algorithm is guaranteed, in a static world, to eventually converge on an optimal policy [14], [15], regardless of the initial state of the Q-learning policy and the reward structure of the world. Moreover, if the world changes slowly, Q-learning is guaranteed to converge on near-optimal policies [16]. This is to say that Q-learners are already somewhat perturbation tolerant. However, we found that the actual performance of a Q-learner in the face of perturbation varies considerably, and, indeed, that post-perturbation performance is negatively correlated to the degree of perturbation ($R = -0.85, p < 0.01$). We further discovered that adding even a very simple metacognitive monitoring and control (MCL) component, that monitored reward expectations and, if expectations were repeatedly violated, instructed the Q-learner to change its policy in one of a number of ways, could greatly improve the perturbation tolerance of a Q-learner. The comparative performance results are summarized in Figure 1. The results show a high degree of correlation between the degree of the perturbation and the ratio of MCL to non-MCL performance ($R = 0.79, p < 0.01$). See [17] for details.

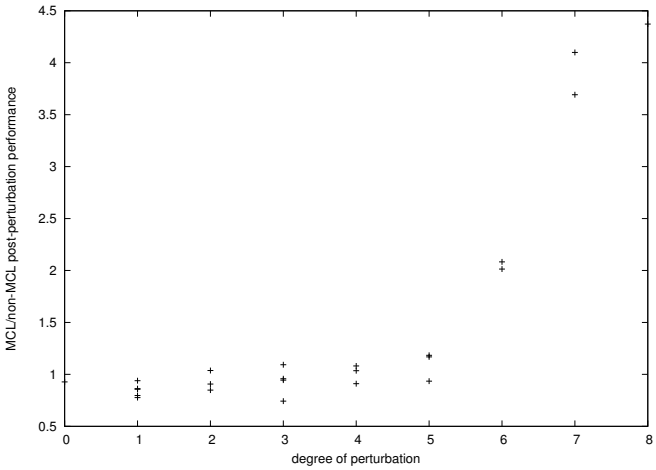


Fig. 1. Ratio of MCL/non-MCL post-perturbation performance, as a function of the degree of perturbation. ($R = 0.79, p < 0.01$)

However, from the standpoint of the current paper, what is important is the evaluation scheme in general, and our estimate of the "degree of perturbation" in particular. For this, the experiment must be understood in some more detail. To establish the above results, we built a standard Q-learner, and, starting with no policy (all Q-values=0), placed the Q-learner in an 8x8 grid-world—the possible states being locations in the grid—with reward r1 in square (1,1) and reward r2 in square (8,8). The initial reward structure [r1,r2] of the world was one of the following: [10,-10]; [25,5]; [35,15]; [19,21]; [15,35]; [5,25]. The Q-learner was allowed to take 10,000 actions in this initial world, which was enough in all cases to establish a very good albeit non-optimal policy. After receiving a reward, the Q-learner was randomly assigned to one of the non-reward-bearing squares in the grid. In turn 10,001, the reward structure was abruptly switched to one of the following: [25,5]; [35,15]; [19,21][7]; [15,35]; [5,25], [-10,10].

Our task-based performance measure for the Q-learner was the ratio of actual average reward per action taken (henceforth, per turn) to the ideal average reward per turn,

[7]Except when the initial structure was [19,21], in which case the post-perturbation structure was [21,19]

i.e., the average reward per turn theoretically available to a Q-learner following an optimal policy in the given environment. To get a handle on the difficulty of each perturbation, we first considered that the learned Q-table can be visualized as a topographic *overlay* on the grid world, where positive rewards are attractors, and negative rewards are repulsors, and the grade of the topography (the differences in the Q-values for each action at each location) corresponds to the degree of attraction to a given reward. Following the policy recommended by the Q-table is equivalent to moving downhill as quickly as possible. For simplicity, we can abstract considerably from this picture, and imagine that each square of the policy-overlay contains a proposition indicating the direction of the slope—toward (1,1), or toward (8,8). For a given perturbation, then, we can get one factor in the difficulty of the change, by counting the number of squares where the propositions characterizing the slope (as determined by an ideal policy) have changed. Thus, for instance, to go from the ideal abstract policy for reward structure [10,-10] (every square says go to (1,1)) to the abstract policy for reward structure [-10,10] (every square says go to (8,8)) involves a large *overlay difference* (D_o) of value 64, but going from [19,21] to [21,19] involves essentially no overlay difference.[8]

Another factor in measuring the degree of perturbation we considered for the current case was any valence change in the rewards. A valence change makes the perturbation greater because it makes it harder for the agent to actually change its abstract policy (one way to think about this might be as the mathematical equivalent of a contradiction). For instance, a negative reward that becomes positive (V^+) is masked from the agent because the policy is strongly biased against visiting that state. Thus, in light of the above considerations, we devised an equation to estimate the degree of perturbation (D_p) in each of the 22 cases:

$$D_p = D_o/16 + 3V^+ + V^- \tag{7}$$

The experiment as described primarily evaluated the perturbation tolerance of the agent in terms of its ability to move effectively between different (abstract) overlays, making the overlay difference the most relevant measure. However, other aspects of the test domain can indeed be measured according to the metrics offered here.

[8]It should be noted that this is an *adaptation* of the meaning of overlay and overlay difference to fit the experimental circumstances, and the nature of the agent being tested. If we understand the task of a Q-learner in terms of uncovering and mapping the reward-based topography of a given region, then this is the relevant difference between two regions that needs measuring when assessing the difficulty of moving from one to the other. Such adaptation of shared definitions and terms to individual circumstances is inevitable, and care must be taken in each case to properly explain individualized uses, and to remain sensitive to the overall goal of allowing cross-experiment comparisons.

n (overlay size) = 64. There are 64 squares in the overlay.

ρ_I (information density)= 3. Three propositions characterize each square: an X value and Y value that correspond to its location, and an R value that corresponds to the reward available there.

V_o (variability)= 0.36. The average minimum graph distance between squares in the grid is 5.5, and the average propositional difference is just above 2 (a square can differ by at most 3 propositions (X, Y and R), however most of the squares differ by 2 (X and Y, X and R, or Y and R), and a few by only 1 (X or Y)).

δ_o (volatility)= 0. The overlay does not change over time.

I (inconsistency)= $0\%/3\%$. Two values are given here, because when the agent begins the experiment, it has no beliefs, and there is therefore no inconsistency. However, when it moves between the two overlays, it has 64 beliefs about the rewards available in each square. Two of these beliefs are in direct conflict with the state of the world ($2/64 = 0.03$). Note the agent also has a number of beliefs about what actions to take in what circumstances to achieve maximum reward; many of these beliefs are false in its new circumstances. However they are not directly about the world, and nothing that the agent can perceive about the world *directly* contradicts any of these beliefs. Therefore, these do not count toward the measure of inconsistency.

U (uncertainty) = 0. The agent had perfect knowledge of its environment.

6. CHALLENGES

As the suggestions I have made are just that—preliminary suggestions meant as the starting point of a potentially long but important investigation, there remain some significant questions and challenges. First, and most obvious: are the elements of the environment identified here in fact the most important? And are the methods suggested for measuring them appropriate? Related to this: how easy will it be in practice to interpret a given test domain according to this proposal? For it is clear that even in the case where a domain is grid-available, and where it is therefore *possible* to apply these metrics, it will not necessarily be easy to do so. Although applying these metrics will be quite easy in domains like the one described above, where the parts and their contents are well defined, and even expressed in terms of the defined partition, it will be much less easy in test environments not designed along this model, for instance video games. Thus, some attention must be paid to developing principled, automated methods for analyzing test domains in accordance with the suggestions outlined here.

7. CONCLUSION

In this paper I have suggested a standard way to characterize the size, information density, variability, volatility, inconsistency and uncertainty of a given test environment, each of which contribute to the complexity of that environment. I have also suggested a way to measure the difference between two different environments of the same size. From these basic measures, I have shown how one can construct more comprehensive measures of the complexity of the environment, and I have given several examples of how the metrics can be used to measure the task performance and perturbation tolerance of cognitive agents. Finally, I showed how some of the metrics were applied to demonstrate that a metacognitive monitoring and control component could enhance the perturbation tolerance of a simple machine-learner. Although significant challenges remain, it is hoped that the paper will prove a useful starting point to the investigation of an important topic.

8. ACKNOWLEDGMENTS

This paper is a revised version of [18]. The research is supported in part by the AFOSR.

9. REFERENCES

[1] J. McCarthy, "Elaboration tolerance," in *Proceedings of the Fourth Symposium on Logical Formalizations of Commonsense Reasoning*, 1998.

[2] E. Amir, "Toward a formalization of elaboration tolerance: Adding and deleting axioms," in *Frontiers of Belief Revision*, M. Williams and H. Rott, Eds. Kluwer, 2000.

[3] M. Greenberg and D. Westbrook, "The phoenix testbed," 1990, technical Report UM-CS-1990-019, Computer and Information Science, University of Massachusetts at Amherst.

[4] P. R. Cohen, M. L. Greenberg, D. M. Hart, and A. E. Howe, "Trial by fire: Understanding the design requirements for agents in compex environments," 1989, technical Report UM-CS-1990-061, Computer and Information Science, University of Massachusetts at Amherst.

[5] M. Pollack and M. Ringuette, "Introducing the tileworld: experimentally evaluating agent architectures," in *Proceedings of the Eighth National Conference on Artificial Intelligence*, T. Dietterich and W. Swartout, Eds. Menlo Park, CA: AAAI Press, 1990, pp. 183–189. [Online]. Available: citeseer.nj.nec.com/pollack90introducing.html

[6] E. Durfee and T. Montgomery, "MICE: A flexible testbed for intelligent coordination experiments," in *Proceedings of the Ninth Workshop on Distributed AI*, Rosario, Washington, 1989, pp. 25–40. [Online]. Available: citeseer.nj.nec.com/durfee89mice.html

[7] D. Perlis, "On the consistency of commonsense reasoning," *Computational Intelligence*, vol. 2, pp. 180–190, 1986. [Online]. Available: http://www.cs.umd.edu/projects/active/doc/papers/86/occr.pdf

[8] K. Purang, "Systems that detect and repair their own mistakes," Ph.D. dissertation, Department of Computer Science, University of Maryland, College Park, Maryland, 2001.

[9] J. Elgot-Drapkin, "Step-logic: Reasoning situated in time," Ph.D. dissertation, Department of Computer Science, University of Maryland, College Park, Maryland, 1988.

[10] J. Elgot-Drapkin, S. Kraus, M. Miller, M. Nirkhe, and D. Perlis, "Active logics: A unified formal approach to episodic reasoning," Univ of Maryland, UMIACS and CSD, Tech. Rep. UMIACS TR # 99-65, CS-TR # 4072, 1993.

[11] J. Elgot-Drapkin and D. Perlis, "Reasoning situated in time I: Basic concepts," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 2, no. 1, pp. 75–98, 1990.

[12] K. Purang, D. Purushothaman, D. Traum, C. Andersen, and D. Perlis, "Practical reasoning and plan execution with active logic," in *IJCAI-99 Workshop on Practical Reasoning and Rationality*, 1999.

[13] M. Bhatia, P. Chi, W. Chong, D. P. Josyula, M. Anderson, Y. Okamoto, D. Perlis, and K. Purang, "Handling uncertainty with active logic," in *Proceedings of the AAAI Fall Symposium on Uncertainty in Computation*, 2001. [Online]. Available: http://www.cs.umd.edu/ mikeoda/papers/aaai01.pdf

[14] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge University, Cambridge, England, 1989.

[15] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[16] I. Szita, B. Takács, and A. Lörincz, "ϵ-MDPs: Learning in varying environments," *Journal of Machine Learning Research*, vol. 3, pp. 145–174, 2002.

[17] M. L. Anderson, T. Oates, W. Chong, and D. Perlis, "Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance," submitted.

[18] M. L. Anderson, "Specification of a test environment and performance measures for perturbation-tolerant cognitive agents," in *Proceedings of the AAAI Workshop on Intelligent Agent Architectures*, 2004. [Online]. Available: http://www.cs.umd.edu/ anderson/papers/aaai_metrics_04.pdf

Framework for the Performance Assessment of Architectural Options on Intelligent Distributed Applications

Günter Haring‡, Carlos Juiz*, Christian Kurz‡, Ramon Puigjaner*, Joachim Zottl‡
‡ Department for Computer Science and Business Informatics, University of Vienna
Lenaugasse 2/8, A-1080 Vienna, Austria
{guenter.haring, christian.kurz, joachim.zottl}@univie.ac.at
*Department of Mathematics and Computer Sciences, University of Balearic Islands
Carretera de Valldemossa, km. 7.5, 07071 Palma de Mallorca, Spain
{cjuiz, putxi}@uib.es

ABSTRACT

This position paper brings together the evaluation of ambient intelligence architectures in context-awareness systems with performance modeling. Thus, firstly appropriate description methods for distributed intelligent applications are summarized. Derived from the system characterization, typical software performance engineering techniques are based on the augmented description of the model regarding performance annotations. However, these annotations are only related with the syntactical view of the architecture. In the next generation of performance assessment tools for intelligent context-awareness systems, the description of the system would be capable of reasoning and acquiring knowledge about performance. Having an appropriate architectural description including performance aspects, any possible design options for intelligent distributed applications can be evaluated according to their performance impact. Therefore, we propose the use of an ontology with performance-related information - not only to evaluate the architecture off-line - but also building a context broker that assesses the performance during execution.

KEYWORDS: *performance evaluation, distributed software performance engineering, context-awareness, ambient intelligence, mobile devices*

1. INTRODUCTION AND MOTIVATION

To be able to create architectures for intelligent distributed systems one has to consider the capabilities and limitations of the devices running the applications. One fundamental aspect is performance issues which have to be included into the decision process when choosing between different architectural options. Performance analysis of architecture options should be integrated in early life cycle stages of a software development process [8].

The term *software architecture* (*SA*) of a program defines the systems structure, which comprises the software components, their external observable behavior and the relationship of these components to each other [1], [3]. A software design method is a systematic approach for creating a system design. During a given design step, the method may provide a set of structuring criteria to help the designer in decomposing the system into its components [4]. However, non functional features of the system, e.g. performance, have not been considered for those software design methodologies. Thus, the performance modeling of systems is based on a certain type of conceptual performance formalism (e.g. queuing networks (QN) and their extension (EQN), stochastic timed Petri nets (SPTN) or stochastic process algebra (SPA)). As the size and complexity dramatically increase, many software (distributed) systems can not provide performance properties as required due to fundamental architecture or design problems. During the last years the UML (Unified Modeling Language) has been widely used to specify, construct and document the functionality of software systems [15]. In order to reduce the gap between functional models and performance evaluation, a software and performance community has emerged to provide (automatically) accessible techniques and tools to include performance annotations for building performance prediction constituting a new topic in Software and Performance Engineering (SPE) [9].

UML diagrams provide key information required for performance analysis so that they describe both behavior and resources. Therefore, sequence, activity, state chart and deployment annotated diagrams may be annotated to express some performance information in a direct or indirect way [19], [20], [21].

In typical software architectures of distributed systems communication between clients and servers has an important role. However, the growing availability of mobile and wireless networks and the expansion of powerful mobile devices define new issues for these software distributed systems. Thus, applications designed for mobile computing are expected to run in a highly heterogeneous and dynamic

environment, due the limited computing, storage and power capabilities of portable devices, the large variance in the communication bandwidth, and maybe the crucial factor, the mobility itself. In that sense, other mobile topics are emerging, e.g. the computing ubiquity, the natural interaction of the systems components and their intelligence. However, less attention has been paid to these last phenomena in the performance evaluation arena because the traditional software architectures for distributed applications are difficult to translate to ad-hoc communication environments [18]. Our position is that performance-related information must be considered not only for performance evaluation of the actors in a changing mobile environment, but also in scenarios where it is possible to reason about the performance activity in an intelligent ambient way and even take actions on it. Thus, the huge amount of knowledge that was researched under the software performance engineering may walk one step beyond to this cutting edge issue.

The remainder of the paper is organized as follows. In Section 2 we overview the different factors to be considered in the performance assessment of ambient intelligent applications (from now we name this approach PA-Ai). Section 3 of the paper summarizes related work, mainly giving an overview of work similar to the scope of this paper. The following section details the structure of the performance evaluation framework. Finally Section 5 summarizes the conclusions of this paper and provides an outlook to future work.

2. FACTORS TO CONSIDER IN PA-Ai

The following factors are the main issues to consider for performance assessment in ambient intelligence applications.

2.1 *Distributed intelligent applications*

A distributed application is an application which is executed based on a distributed system; therefore different parts of the application are processed on different machines. Usually the functionality of the architecture is mapped on the client-server paradigm. However, in mobile applications the client and server roles are not defined so specifically, some times devices are clients and some times they are servers. To add intelligence to such an application usually means that the system can learn from past experiences and make future decisions based onto this knowledge.

One possible scenario for a distributed intelligent application could be a *meeting coordination* system for office or congress use (MC scenario) [5]. In that scenario a congress participant enters the congress area. At that moment his personal digital assistants (PDA) automatically connects to the hotel server. It recognizes the conference participant, accesses his previous behavioral patterns, and immediately sends him information which could be useful for him. This information might be a room map when the conference is entered or the session agenda depending on the room being entered. It may contain a renewed session agenda which might have been altered due to short time changes. Additionally a list of the participants of the conference or a certain session can be offered, or supplementary information like presentation slides can be transferred to the attendees' mobile computer. The mobile device can also allow for communication with other congress members, for example with participants of the same session.

A second scenario might be useful for *office coordination* (OC scenario) [6]. A project manager can locate the members of his team using a "People Locating System (PLS)". This system is able to detect employees inside a companies building. When the project manager is scheduling a meeting the PLS is trying to locate all participants to be invited to be able to deliver them a message about the meeting schedule. Based on the participants behavior when receiving meeting information in the past according to their respective working circumstances, the PLS decides which type of message it delivers. When it finds two people together in a room with several others, it reasons that they are in a meeting and therefore decides to send them only a message notification to their PDA. Other members are located at their working place and thus are considered to be available; therefore they get the full text message onto their computers. Finally, two more members cannot be found on the company's site. The system hence accesses their appointment calendars and finds out that one of them has a meeting with a customer and thus should not be disturbed, and the other one is at his dentist. To both of them the system sends an email detailing the forthcoming meeting.

So the key difference between the traditional client-server architecture and these last scenarios is mainly how the information is represented in this changing environment. Whereas in traditional distributed software systems the representation is meant for computers to process information, i.e. syntactic level, in the ad hoc connected communication systems the representation allows devices to process and reason about information, i.e. semantic level. Therefore, it is necessary to get a semantic description of the components in the architecture [22].

Context-awareness systems not only consider the location but also any information that can be used to characterize the situation of the mobile devices, e.g. the system capabilities, the services offered and sought, the activities among devices and users, and their intentions.

2.2 *Mobile Devices*

Mobile devices, for example PDA's or Pocket PC's, are essential elements in future context-aware systems. Those devices are characterized by limited resources. They have

low processing power, constraints in memory capacity, communication bandwidth, and battery power. Hence, it is important to find a performance optimal architecture for applications using these limited devices.

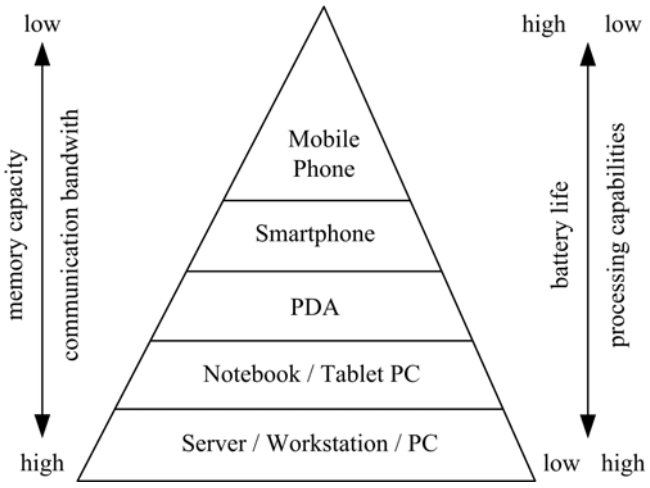


Figure 1: Device Capabilities

Concerning distributed intelligent applications we should at least consider five basic categories of devices which could be involved in performing various tasks for the application (see Figure 1). At the bottom of the pyramid the category consisting of immobile but powerful Servers, Workstations or PCs is located. The Notebook or Tablet PC on top of them is less powerful, but can be moved freely, only being limited by usually low battery endurance of a few hours. On the next higher layer PDAs provide less computing power, a limited user interface but stronger battery life up to usually about ten hours. Smartphones on the layer above have extended battery life, but even less processing capabilities and an even smaller user interface. On top of the pyramid are the mobile phones which can feature battery life of more than a week (not at heavy use), but offer only very limited processing power. Also the potential communication bandwidth and memory capacity is smallest on top of the pyramid and is increasing towards the base of it e.g. for the servers.

2.3 *Evaluation of architecture options*

Currently a number of well established software architectures are known, for example: (i) *Web-Services* are software components which are made useable via application servers. This model is also known as service-oriented architecture (SOA). (ii) In a *Client/Server architecture* resources are concentrated in one or a small number of nodes. So, in this model workload and bandwidth capabilities are unbalanced. (iii) In *Peer-to-Peer-Systems* workload and bandwidth demands will be distributed uniformly among the connected processors. (iv) *Component models* are based on building blocks which describe a well defined functionality. Such components can be accessed through interfaces (e.g. Corba, J2EE or .NET). (v) *Push-Systems* are used for efficient and timely distribution of information to a huge number of users. (vi) In *Event-Based-Systems* users are notified when determined events occur.

These architectures possess different characteristics like structure, degree of hierarchy or degree of coupling. When evaluating architecture options some of them will tend to be more adequate then others, but for one application there might be several suitable architecture options. So, given an application with its requirements and usage patterns a number of open questions arise. Is there only one adequate service architecture? How can several architecture options be assessed and qualified? Which design is the right one according to the given requirements and basic conditions? There may not exist a perfectly fitting architecture or a totally unsuitable one, but architectures which achieve a more or less suitable solution for a given problem and usage. A number of methods and techniques were developed for the evaluation of software architectures, for example: ATAM (Architecture Tradeoff Analysis Method, [10]), SAAM (Software Architecture Analysis Method, [11]), or ARID (Active Reviews for Intermediate Designs, [12]). However, for our purpose the major question is how to express the performance-related information in a context-awareness intelligent application.

3. RELATED WORK

UML diagrams that provide key information required for performance analysis are those that describe behaviour and resources together, therefore augmented sequence, activity, state chart and deployment annotated diagrams may express some performance information. A huge number of approaches have been proposed to derive performance models from software architecture specifications [2]. Basically, the concept can be used in an early stage of the software lifecycle. It uses the SPE architectural decision strategy. From annotated UML diagrams performance models are generated in the corresponding formalism (QN, SPN, SPA, etc.) and then they are offline evaluated through analytical, numerical or discrete-event simulation techniques. Following this procedure, [8] uses the SPE methodology for deriving performance models from software architecture specifications. In [7] a derivation of a QN model from SA is presented. This approach is based on Client/Server software performance evaluation (CLISSPE). In [13] an example to generate stochastic timed Petri net models from UML diagrams is shown. Finally [14] presents an example for the derivation of a performance model from an object-oriented design model. Due to the huge amount and the variety of proposals of 1.X UML performance extensions, new approaches are being developed for

performance modeling built from UML/SPT profile (Schedulability, Performance and Time) annotation [17].

Some performance analysis approaches have been reflected into mobile software architectures from annotated UML diagrams [2]. However, these solutions cover the mobility or location-awareness aspects, referring to the ability of the system to recognise the mobile components and the services (requested/offered) of the distributed system but not about the context or the ambient intelligence. Some performance tools and UML performance annotated design techniques have been connected through XML/XMI files [16].

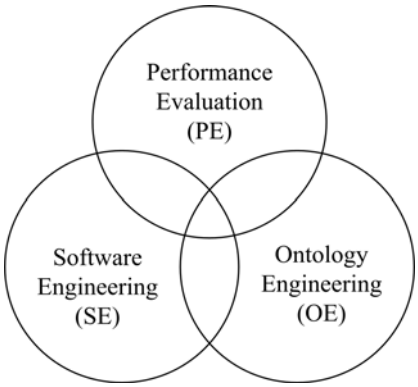


Figure 2: Intersection of PE, SE and OE areas

Several initiatives have been taken to deal with the topic of a joint terminology of context-awareness systems. Sponsored by the W3C, the web ontology language OWL seems to be a de facto standard. The OWL language builds on XML's ability to define customized tagging schemes and the flexible approach to representing data of RDF (Resource Description Framework). OWL is a language for defining and instantiating ontologies [23].

Figure 2 shows some of the research areas involved in the development of a framework to assess the performance of ambient intelligence applications. SE, PE and OE disciplines cannot provide a complete solution by themselves for certain topics, for example, the scope of SPE problems is located at the intersection between SE and PE. In this paper, we focus on the overlapping area between PE and OE and probably should be extended to all three disciplines.

Thus, critical issues in context-awareness research are context modeling, context intelligence (reasoning and knowledge) and context-privacy but other non-functional aspects are not considered, yet, e.g. context-aware performance assessment. In any case, software engineering has moved a bit since there are also early studies to map OWL into UML, but the approach on SPE may be different, as in next section we are going to overview.

4. STRUCTURE OF THE PERFORMANCE EVALUATION FRAMEWORK

There are several issues to be considered when defining a framework for the performance assessment of architectural choices in a context-awareness system (PA-Ai) that are similar to traditional SPE techniques: (i) It must be decided about the way the intelligent system is modeled and therefore, how to add the performance-related information (and which is interesting) into the specification with the minimal interference; (ii) Once the performance aspects of the system are depicted in the model, how to transform the architectural options onto a performance model and finally; (iii) how to evaluate the performance model of every choice. We are going to refer to this as Offline Performance Evaluation to distinguish if from the Online Adaptive Performance Brokerage.

4.1 *Off-line Performance Evaluation*

The framework shall provide an opportunity to compare different alternatives for architectures based on the capabilities of the involved devices and communication infrastructure. Thus an assessment of architecture options with respect to performance for various alternatives is done. This framework gives a strategy for a performance evaluation for architecture options based on relative performance predictions.

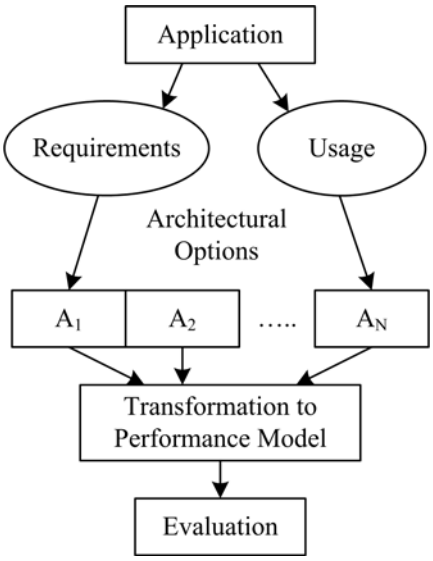


Figure 3: Framework Architecture

The overall architecture of the framework is depicted in Figure 3. The application determines the input parameters which are the requirements and perspective usage of the system. Depending on these parameters, several architectural options A_i may be feasible. In the next stage,

these architectural options (based on an appropriate description) are transformed into a selected performance model which can be evaluated. This strategy does not differ from the traditional performance assessment for distributed applications although it has to consider the semantic representation of the information on the model.

An ontology is an explicit formal description of concepts in the domain composed of classes, properties of each class, and restrictions on properties. Therefore, it expresses the set of terms, entities, objects and classes and the relations between them with formal definitions. The use of ontologies contributes to knowledge sharing and reuse across systems. OWL ontologies are usually placed on web servers as web documents, which can be referenced by other ontologies and downloaded by applications that use these ontologies.

Our position is that performance-related information may be also declared through this new approach, not only for performance evaluation of the actors in a changing mobile environment, but also in scenarios where it is possible to reason about the performance activity in an intelligent ambient way and even take actions based on it.

On the other hand, ontologies can be used to build an information model, as some of the UML diagrams do, which allows the exploration of the information space in terms of the items which are represented, the associations between the items, the properties of the items, and even the links to documentation which describes and defines them (i.e., the external justification for the existence of the item in the model). That is to say that the ontology and taxonomy are not independent of the physical items they represent, but may be developed / explored in tandem. Thus, an ontology may consider performance-related information as description of the architecture of a system. Moreover, OWL should be compatible with other commonly used Web and industry standards. In particular, this includes XML and related standards (such as XML Schema and RDF), and possibly UML. Therefore we may exploit the interchange format between OWL and performance evaluation tools in the same manner as SPE engines. Figure 4 shows part of a simple example of OWL ontology encoded in RDF/XML.

```
<owl:Class rdf:ID="PDA">
  <rdfs:subClassOf rdf:resource="device" />
  ...
</owl:Class>

<owl:Class rdf:ID="performanceDescriptor" />

<owl:ObjectProperty rdf:ID="activity">
  <rdfs:domain rdf:resource="#device" />
  <rdfs:range
rdf:resource="#performanceDescriptor" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="locatedIn">
  ...
  <rdfs:domain
rdf:resource="http://www.w3.org/2002/07/owl#T
hing" />
  <rdfs:range rdf:resource="#building" />
</owl:ObjectProperty>

<owl:Class rdf:ID="demand" />
  <rdfs:subClassOf
rdf:resource="performanceDescriptor" />

<owl:DatatypeProperty rdf:ID="demandValue">
  <rdfs:domain rdf:resource="#demand" />
  <rdfs:range  rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<demand rdf:ID="exp_average">
  <demandValue
rdf:datatype="&xsd;float">4.500</demandvalue>
</demand>
```

Figure 4: Simple OWL performance example

We provide a simple example of a vocabulary for performance-related information for the location information and average demand requirements of PDA devices (some information has been deleted due to space limitation of the text).

4.2 *On-line Performance Assessment*

However, proposing the use of OWL as a language to express similar performance annotated information as other de facto standards may not justify the effort. In this last case, only the syntactical view of OWL should be exploited.

One of the definitive features in ambient intelligence applications is the service discovery, i.e., functions offered by various mobile (e.g. mobile phones, PDAs, notebooks) and non-mobile devices (e.g. servers, printers, panels) that can be described and advertised, so that, they are sought-and-found by others. All of the current service discovery and capability description mechanisms (e.g. JINI, UPnP, JXTA, Bluetooth…) are based on ad-hoc representation schemes and rely heavily on standardization due to devices which were not necessarily designed to work together (such as ones built for different purposes, by different manufacturers, at a different time, etc.) as we experienced in the AKSIS project [5].

Being able to communicate at a high-level of abstraction with other devices, and reason about their services/functionality and performance is necessary for the complete evaluation of different architectural choices.

Thus, an ontology language will be used to describe the characteristics of devices, the means of access to such devices, the policy established by the owner for the use of a device, and other technical constraints and requirements that affect incorporating a device into a ubiquitous computing network. The needs established for DAML-S (DARPA Agent Markup Language) [25] and the RDF-based schemes

for representing information about device characteristics (namely, W3C's Composite Capability/Preference Profile (CC/PP) and WAP Forum's User Agent Profile (UAProf)) directly relate to this use case and the resource infrastructure which will support mobile applications and dynamically configure/negotiate ad-hoc networks. Thus, the performance information about resources, activities, actions, etc. in the context may be included as subproperties and datatypes in an extended vocabulary for OWL. This performance-related information and several simple operational rules and heuristic knowledge may be used for reasoning during execution about the performance of devices and services. Therefore, scenarios as OC or MC may be implemented through a team of context brokers. The context brokers would be running on stationary servers. A service discovery infrastructure will meet devices and servers, and the ontology will acquire information and reason about users, location, privacy and also performance. For example, in the OC scenario the ontology must include identifiable places in order to infer about location context. Reasoning about the spatial situation can predict performance improvements for example by mirroring services or automatically by disabling inactive device connections. To support reasoning with the device hardware/software descriptions, the ontology not only has to include profiles that would be extensions of [24] but also about PDAs and mobile phones to implement the MC scenario. Inferring about the device profiles may play an important role for capacity planning during context execution. The DAML ontology is a temporal ontology for expressing time-related properties. An extended OWL would have to consider this crucial information for performance prediction since it could be used to know the throughput of servers, the latency of a connection, the utilization of a device, etc. Moreover, location and temporal reasoning may be correlated for performance assessment purposes learning about inconsistencies among offered/required services in the scenarios.

5. CONCLUSION AND FUTURE WORK

This position paper tries to address the use of ontology as the solution to evaluate the performance of intelligent context-aware systems. Our preliminary study shows that OWL is not only a requirement for knowledge sharing in pervasive ambience, but also for acquiring performance-related information and the subsequent reasoning. However, the first step is to show that the syntactic use of ontologies for performance evaluation may incorporate the same information as annotated modeling languages in the SPE area. Thus, the off-line performance evaluation of architectural choices would be computed from the object properties and datatype definitions with performance constraints. The interconnection between the annotations and the performance tools for analytical solving or discrete-event simulation would use the XMI/XML interchange formats. Although this work is only overviewed in this paper, it could represent a primary step for evaluating the performance of context-awareness systems.

A more ambitious project would be the utilization of context brokers in order to assess performance during context execution. The advantage of the OWL description of the ambient may use the semantics to infer performance knowledge. Even the off-line performance evaluation relies on the annotated constraint values; it seems to be possible to get information on-line about the relationships in the context and to reason about them. Thus, a team of context brokers would implement the architecture in various aspects of pervasive computing, e.g. location, timing, device profiling, etc. and performance.

6. REFERENCES

[1] Bass L., Clements P., and Kazman R., *Software Architecture in Practice*, Addison-Wesley, 1998

[2] Cortellessa V., and Mirandola R., "PRIMA-UML: A Performance Validation Incremental Methodology on early UML Diagrams", *Science of Computer Programming*, vol. 44, pp. 101-129, 2002

[3] Dustdar S., Gall H., and Hauswirth M., *Software-Architekturen für Verteilte Systeme*, Springer-Verlag, 2003

[4] Gomaa, H., *Software Design Methods for Concurrent and Real-time Systems*, The SEI Series in Software Engineering, N. Habermann (ed.), Addison-Wesley, Reading, Massachusetts, 1993

[5] Hummel K.A., "Meeting Coordination", http://www.ani.univie.ac.at/~karin/ambience/scenarios/meeting.pdf

[6] Hummel K.A., "Office Communication", http://www.ani.univie.ac.at/~karin/ambience/scenarios/communication.pdf

[7] Menascè D.A., and Gomaa H., "On a Language Based Method for Software Performane Engineering of Client/Server Systems", in Proceedings of the 1st International Workshop on Software and Performance, pp. 63-69, 1998

[8] Smith C.U., and Williams L.G., "Performance Evaluation of a Distributed Software Architecture", in Proceedings of the 1st International Workshop on Software and Performance, pp. 164-177, 1998

[9] Smith C.U., and Williams L.G., *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*, Addison-Wesley, 2002

[10] Kazman R., Klein M. Barbacci M., and Lipson H. Longstaff T. Carriere S.J., "The Architecture Tradeoff Analysis Method", in Proceedings of the 4st IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 68-78, 1998

[11] De Simone M., and Kazman R., "Software Architecture Analysis: An Experience Report", in Proceedings of the

1995 conference of the Centre for Advanced Studies on Collaborative research, 1995
[12] Clements P., "Active Reviews for Intermediate Designs", Technical Note CMU/SEI-2000-TN-009, Software Engineering Institute, Carnegie Mellon University
[13] King P., and Pooly R., "Derivation of Petri Net Performance Models from UML Specification of Communication Software", in Proceedings of the 1997 Computer Measurement Group Conference, 1997
[14] Smith C.U, and Williams L.G., "Performance Engineering Evaluation of Object Oriented Systems with SPE•ED", in *Computer Performance Evaluation: Modelling Techniques and Tools*, Springer-Verlag, 1997
[15] Object Management Group (OMG): Unified Modeling Language Specification, version 1.3, http://www.omg.org/uml
[16] Object Management Group (OMG): XML Metadata Interchange (XMI) Specification, version 1.2, http://www.omg.org/cgi-bin/
[17] Object Management Group (OMG): UML Profile for Schedulability, Performance and Time Specification, March 2002
[18] Beer, W., Christian, V., Ferscha, A. and Mehrmann L., "Modeling Context-Aware Behavior by Interpreted ECA rules". In Proceedings of Euro-Par 2003, H. Kosch, L. Böszörményi and H. Hellwagner (Eds.), LNCS 2790, pp. 1064-1073, 2003
[19] Proceedings of the Second Workshop on Software and Performance (WOSP 2000), Ottawa, Canada, September 2000, ACM Press
[20] Proceedings of the Third Workshop on Software and Performance (WOSP 2002), Rome, Italy, July 2002, ACM Press
[21] Proceedings of the Fourth Workshop on Software and Performance (WOSP 2004), San Francisco, USA, January 2004, ACM Press
[22] Berners-Lee, T., Hendler, J. and Lassila, O. "The Semantic Web", Scientific American, May 2001
[23] Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel- Schneider, P.F. and Stein, L.A. "OWL Web Ontology Language reference", 2002, http://www.w3c.org/TR/owl-ref/
[24] Foundation for Intelligent Physical Agent. FIPA Device Ontology Specification, pc00091a edition, 2001
[25] Hobbs, J.R. "A DAML Ontology of Time", http://www.cs.rochester.edu/daml

The Semantics of Ontology Alignment

Todd C. Hughes and Benjamin C. Ashpole
Lockheed Martin Advanced Technology Laboratories
Cherry Hill, NJ
{thughes, bashpole}@atl.lmco.com

ABSTRACT

Ontology alignment is a foundational problem area for semantic interoperability. We discuss the complexity faced by automated alignment solutions and describe an ontology-based approach for describing and evaluating alignments.

KEYWORDS: ontology, ontology alignment, ontology mapping, Semantic Web

1 THE CHALLENGE OF ONTOLOGY ALIGNMENT

The vision of semantic interoperability, the fluid sharing of services and digitalized knowledge, is often thought to hinge on a common, formal language that machines can somehow understand. However, protocols and data formats such as XML tags and schemas have proven to be inadequate solutions primarily because the burden of meaning is still on humans, who still must learn implicit semantics of foreign systems in order to make them work with their native systems. Semantic Web languages like RDF and OWL begin to ameliorate the problem by adding explicit semantic relationships and logical constraints between elements (i.e., classes, properties, and restrictions) in the form of ontologies, an extension of schemas. However, programs that read OWL documents that conform to a particular ontology cannot understand other OWL documents conform to a different ontology *unless* there is an explicit mapping between the ontologies. Creating this mapping is the alignment problem, and solving it is the first step to semantic interoperability.

Alignment between ontologies is a critical challenge for semantic interoperability. There are $(n*m)$ possible *individual*, undirected alignments for ontology graphs of size n and m. Optimal graph matching algorithms run in exponential time due to the NP-complete nature of the search space. For large ontologies with tens of thousands of elements, purely manual alignment methods are clearly impractical [1], and semi-automated approaches are not suitable for real-time applications.

Semantic interoperability requires fully automated ontology alignment approximation techniques. This cannot be accomplished solely by lexical comparison between element names in different ontologies, since names (like tags) can be abbreviations, acronyms, phrases, in different languages, misspelled, or used in unexpected, jargon-specific ways. In addition, the size, structure, and scope of ontologies must be considered. There is no guarantee that two ontologies in the same domain will have terms that all *precisely* and *completely* overlap: in one ontology, an element name might be equivalent to several—or none—in another. Clearly, alignment techniques must be sensitive to a number of ontology features to find corresponding elements. [4]

A number of prototype ontology alignment applications have been developed to meet this challenge. However, it is difficult to assess the efficacy of these tools because their developers each use their own alignment formats, test data sets, and evaluation metrics. Do et al [2] have made a notable effort to compare alignment tools using standard metrics, but at this point in alignment research there is still no formal, broadly used language to describe the output of an aligner and to judge the value of one aligner relative to another.

2 ALIGNMENT SEMANTICS

We have developed, appropriately enough, a set of ontologies intended to capture the semantics for relevant metrics for automated ontology operations, including ontology alignment. These ontologies are part of an ongoing effort to focus the ontology alignment community on canonical set of challenge problems, research objectives, and evaluation criteria. Here we describe some of the classes and properties of our ontologies, which are available on our website [1].

2.1 Alignment and Equivalence

Alignment is distinct from equivalence for at least two fundamental reasons. First, an ontology alignment provides only a relation between ontology elements: any particular element alignment will depend on the alignments between other elements. An ontology alignment is the most stable set of element alignments, at least in the opinion of the aligner. This leads to a second difference, namely, that element alignments can (and often do) have degrees of confidence associated with them. That is, the aligner cannot say with certainty any particular alignment is true, only that it is the most probable alignment given other alignments.

The differences suggest that current Semantic Web terms for expressing equivalence, such as `owl:sameClassAs`, `owl:samePropertyAs`, and `owl:sameAs` are not adequate for expressing alignments.

These properties are intended to capture logical, not relative, equivalence. At this time, there are no broadly accepted semantics for describing the uncertainty of equivalence statements made using these properties, nor is it clear there should be.

2.2 Alignment File

These considerations have led to a different approach to describing alignments formally. We describe an ontology alignment in a Semantic Web document called an `AlignmentFile`. An Alignment File declares instances of the class `Alignment`, where each instance states that an element from one ontology (`elementA`) corresponds to an element from the other ontology (`elementB`) with some probability (`alignmentConfidence`). An example alignment is shown in Figure 1.

```
:Alignment1 a ao:Alignment;
      ao:elementA
      <someOntologyA#ClassA>;
      ao:elementB
      <someOntologyB#ClassB>;
      ao:alignmentConfidence "0.5".
```

Figure 1. An example instance of an Alignment, shown in the N3 language with simplified URI's. [1] The classes and properties are all defined in the Alignment Ontology (referred via the ao: prefix).

The Alignment File format easily allows for 1-n and n-1 element alignments. Should they become useful in the future, it also allows for more unusual alignments, such alignments between a class and a property or a (group of) instances and a class.

2.3 Alignment Evaluation File

Alignment Files deliver the output of ontology alignment algorithms. To assess the performance of that algorithm, one may compare the output to a document that contains the correct (or best) element alignments for the ontologies in question. We refer to this document using the property `trueAlignment`, and we create this document by hand. When an automated grader compares an alignment file to a true alignment file, it delivers another document called an `AlignmentEvaluationFile`.

There are two broad categories of metrics to consider when evaluating an alignment: experiment metrics and performance metrics. The first category concerns the behavior of the aligner in the experiment, independent of the true ontology alignment. Experiment metrics include but are not limited to:

- `meanGlobalCardinality`: For 1:n alignments between elements between Ontology A and Ontology B, this property expresses the average value for n. (Based on Do et al [2] local/global cardinality metric.)
- `sdGlobalCardinality`: For 1:n alignments between elements between Ontology A and Ontology B, this property expresses the standard deviation value for n. (Also based on Do et al [2] local/global cardinality metric.)
- `unalignedElements`: The number of elements in Ontology A for which no corresponding element in Ontology B has been found.
- `alignedProportion`: The proportion of elements from Ontology B that were aligned to elements from Ontology A.
- `uniqueElements`: The proportion of resources not shared (i.e, having different URIs) between Ontology A and Ontology B.
- `alignmentChallenge`: The proportion of unique elements between Ontology A and Ontology B to the total number of elements in Ontology A and Ontology B.

The second category of metrics concerns the correctness of the element alignments contained in the alignment file. A number of these metrics are derivative of well-known metrics from the information retrieval domain.

- `truePositives`: The number of correct alignments an alignment file contains.
- `falsePositives`: The number of incorrect a alignments an alignment file contains.
- `falseNegatives`: The number of correct alignments missed in an alignment file.
- `precision`: The proportion of correct alignments among those found, (truePositives / (truePositives + falsePositives).
- `recall`: The proportion of correct alignments found (truePositives / (truePositives + falseNegatives)).
- `fMeasure`: The harmonic mean of precision and recall (2*(precision*recall)/(precision + recall)).
- `alignmentPerformance`: Indicates performance given the proportion of overlapping resources between Ontology A and Ontology B (alignmentChallenge * fMeasure).

All of the above performance metrics with the exception of last one are borrowed from Do et al [4]. A partial example of an Alignment Evaluation File is shown in Figure 2. The metrics provide a fairly comprehensive account of the performance of an alignment algorithm, while the ontological framework allows the addition of new metrics as needed.

```
<AlignmentEvaluation123.n3>
      a ae:AlignmentEvaluationFile;
      oe:evaluates
<AlignmentFile123.n3>;
      ae:trueAlignment
<TrueAlignmentAB.n3>;
      oe:grader <Grader1.n3>;
      ae:meanGlobalCardinality "0.5";
      ae:sdGlobalCardinality "0.5";
      ae:unalignedElements "0.5";
      ae:alignmentProportion "0.5";
      ae:uniqueElements "0.5";
      ae:alignmentChallenge "0.8";
      ae:truePositives "0.8";
      ae:falseNegatives "0.0";
      ae:precision "0.8";
      ae:recall "0.8";
      ae:fMeasure "0.8".
      ae:alignmentPerformance "0.64".
```

Figure 2. Partial Example Alignment Evaluation File with Simplified URIs. Prefixes refer to the Alignment Evaluation Ontology (ae:) and an "upper" Ontology Operation Evaluation Ontology (oe:).

3 CONCLUSION

The purpose of creating these ontologies is not only to facilitate our own experimentation with alignment algorithms, but also to facilitate greater collaboration among members of the ontology alignment research community. With a common representational scheme for stating and evaluating alignments, it becomes significantly easier to compare alignment algorithms. Of course, in addition to this framework canonical data sets are also needed to ensure fair and accurate comparisons.

To that end, we have made these ontologies freely available on our website, which also includes sample data sets and an Experiment Set Platform for administering ontology alignment experiments. We have collaborated with NIST to establish an ontology alignment competition based on the model of the Text Retrieval Conference (TREC), called the Information Interpretation and Integration Conference (I^3CON). This event will be the first systematic comparison of ontology alignment algorithms.

Finally, it bears mentioning that ontology alignment is not valuable for its own sake, but is worthwhile only in the service of some other function that requires it. We envision considerable value in automated ontology alignment capabilities for agents that semantically interoperate with heterogeneous (particularly legacy) data systems. As such, the ontologies for ontology alignment should grow to encompass semantic interoperation use cases. These new concepts should allow us to articulate in a formal way the impact of ontology alignment on agent mission success.

4 REFERENCES

[1] Berners-Lee, Timothy. "Getting into the Semantic Web and RDF using N3." http://www.w3.org/2000/10/swap/Primer.html

[2] Do, Hong-Hai, Sergey Melnik, and Erhard Rahm. "Comparison of Schema Matching Evaluations." *Proceedings of the 2nd Int. Workshop on Web Databases*. German Informatics Society, 2002. Available at: http://citeseer.ist.psu.edu/do02comparison.html

[3] McGuinness, Deborah L., Richard Fikes, James Rice, and Steve Wilder. "An Environment for Merging and Testing Large Ontologies." *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*. Breckenridge, Colorado, USA. April 12-15, 2000. Available at: http://www.ksl.stanford.edu/software/chimaera/

[4] Rahm, Erhard, and Phillip A. Bernstein. "On Matching Schemas Automatically." Technical Report for Univ. Leipzig, Dept. of Computer Science, Database Group. 2001. Available at: http://dol.uni-leipzig.de/pub/2001-5

[5] The Ontology Alignment Source, http://www.atl.lmco.com/projects/ontology/

[6] Information Interpretation and Integration Conference, http://www.atl.lmco.com/projects/ontology/i3con.html

Introducing I³CON

The Information Interpretation and Integration Conference

LOCKHEED MARTIN

Todd Hughes, Ph.D.
Senior Member, Engineering Staff
Advanced Technology Laboratories

I³CON: Motivation

- Semantic integration will be one of the first major accomplishments for ontology-based applications
  - Heterogeneous information system and resource Interoperability is a major concern for military, government, industry
  - Many view this as the a fundamental technical challenge of the Semantic Web
- To answer this challenge, there have been new developments in automated ontology and schema:
  - Markup
  - Alignment
  - Merging
  - Translation
  - Learning
- Much of this research has been funded by DARPA programs, but today the largest sponsors are EU programs

I³CON: Observation

- **Semantic integration research community resembles the text retrieval community of 15 years ago**
  - Critical mass of globally distributed research programs
  - Large variety of technical approaches
  - Generally, but not universally, accepted metrics
  - No meaningful basis of evaluating one technical approach over another
- **The success of text retrieval technology was due in large measure to the Text Retrieval Conference (TREC)**
  - Promoted well-defined concepts for measuring success
  - Clarified metrics
  - Established realistic benchmarks
  - Created canonical challenge problems

The NIST TREC model has a proven record of success!

NIST TREC Model

1. Define the metrics
2. Develop experiment format for easy participation by researchers
3. Create development data sets and test data sets; publish the former
4. Distribute test data sets to experiment participants
5. Collect automatically generated results data
6. Collate and compare results data
7. Hold assessment workshop and end of cycle

I³CON: Timeline

- **March 2004:** Met with NIST, "pilot" conference as PerMIS special session proposed
- **March-June 2004:**
  - Formed Organizational Committee
  - Recruited participants
  - Created ontology alignment format
  - Developed test ontology pairs
  - **May 25:** Gave presentation at DAML PI Meeting
- **June 15 2004:** Released test ontology pairs
- **July 16, 2004:** Collected alignment results data
- **July 16-August 20, 2004:** Compiled and analyzed results data
- **August 25, 2004:** I³CON special session at PerMIS

http://www.atl.lmco.com/projects/ontology/i3con.html

I³CON and the TREC Model

1. Define the metrics
2. Develop experiment format for easy participation by researchers
3. Create development data sets and test data sets; publish the former
4. Distribute test data sets to experiment participants
5. Collect automatically generated results data
6. Collate and compare results data
7. Hold assessment workshop and end of cycle

→Precision, Recall, fMeasure

→Ontology Alignment Ontology; Experiment Set Platform

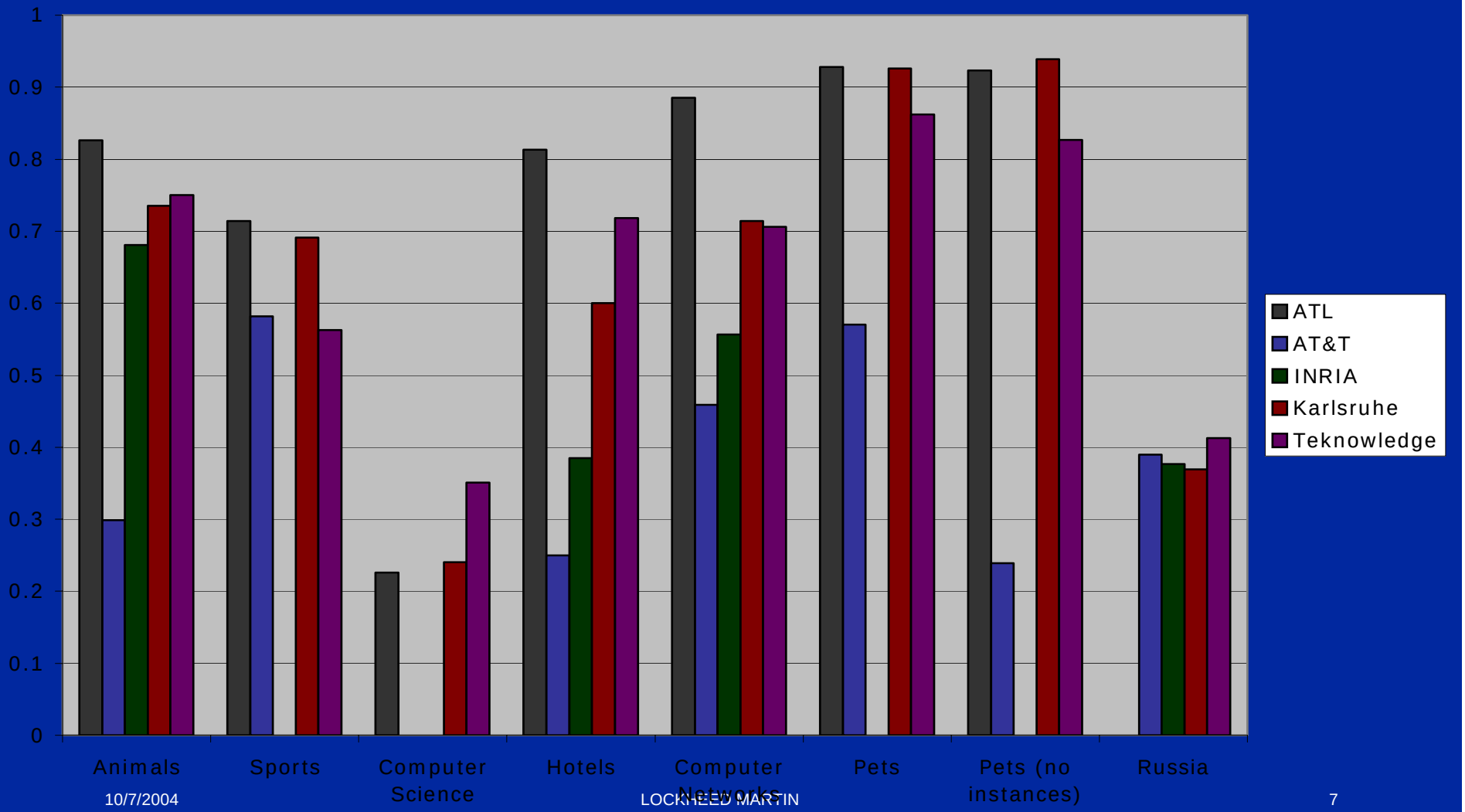
→2 development ontology pairs; 8 test ontology pairs

→5 participants

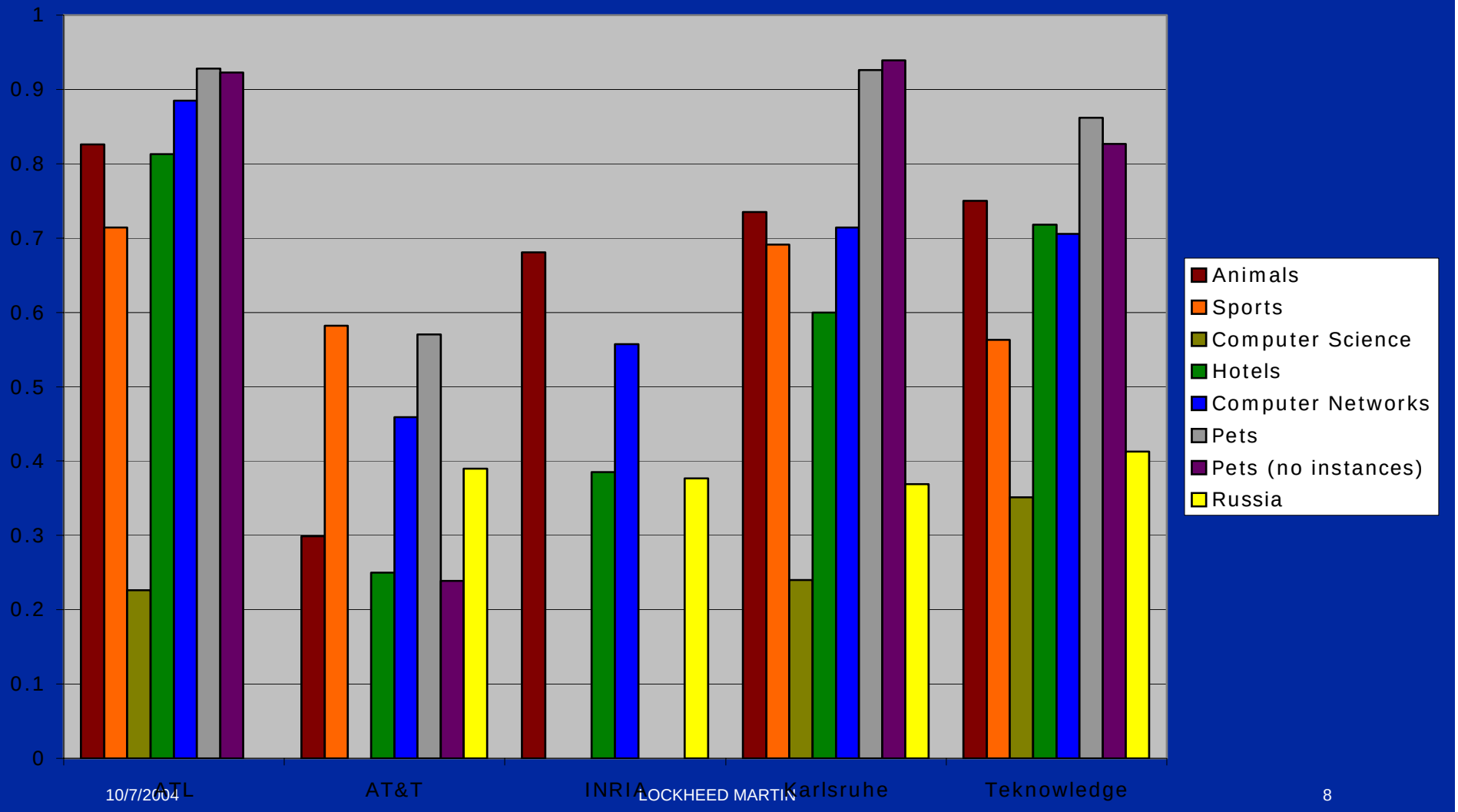
→Most participants submitted alignment data for all ontology pairs

→Where we are today

I³CON: Experiment Results Overview



I³CON: Experiment Results Overview



I^{3}CON Experiment: Lessons Learned

- No single technical approach performed best on *all* test ontology pairs
- No single ontology pair was best for *all* technical approaches
- All approaches performed >0.5 fMeasure on at least one ontology pair
- All approaches performed <0.5 fMeasure on at least one ontology pair

There is much more to be learned from the I3CON experiment data.

I³CON: Special Thanks

- **Organizational Support**
  - Larry Reeker (NIST)
  - Elena Messina (NIST)
- **Technology and Data**
  - Ben Ashpole (ATL)
  - Liz Palmer (ATL)
  - Emil Macarie (ATL)
  - Yun Peng (UMBC)
  - Rong Pan (UMBC)
- **Experiment Participants**
  - Jerome Pierson (INRIA)
  - John Li (Teknowledge)
  - Lewis Hart (AT&T)
  - Marc Ehrig (University of Karlsruhe)
- **Guest Speakers**
  - Bill Andersen (Ontology Works)
  - Mike Pool (Information Extraction and Transport)
  - Yun Peng (University of Maryland Baltimore County)
  - Mike Gruningner (University of Maryland)

EON 2004

- **Evaluation of Ontology-based Tools 3rd International Workshop**
  - **http://km.aifb.uni-karlsruhe.de/ws/eon2004/**
- **Located at the 3rd International Semantic Web Conference (ISWC 2004)**
  - **November 8, 2004**
  - **Hiroshima Prince Hotel, Hiroshima, Japan**
- **EON Ontology Alignment Experiment**
  - **Provides participants with a complete test base of ontology pairs**
  - **Test is based on one particular ontology dedicated to a very narrow domain and a number of alternative ontologies of the same domain**

Semantic Interoperability: Case Study in Ontology-Based Solutions

Mike Pool (mpool@iet.com)

Information Extraction and Transport, Inc.

August 25, 2004

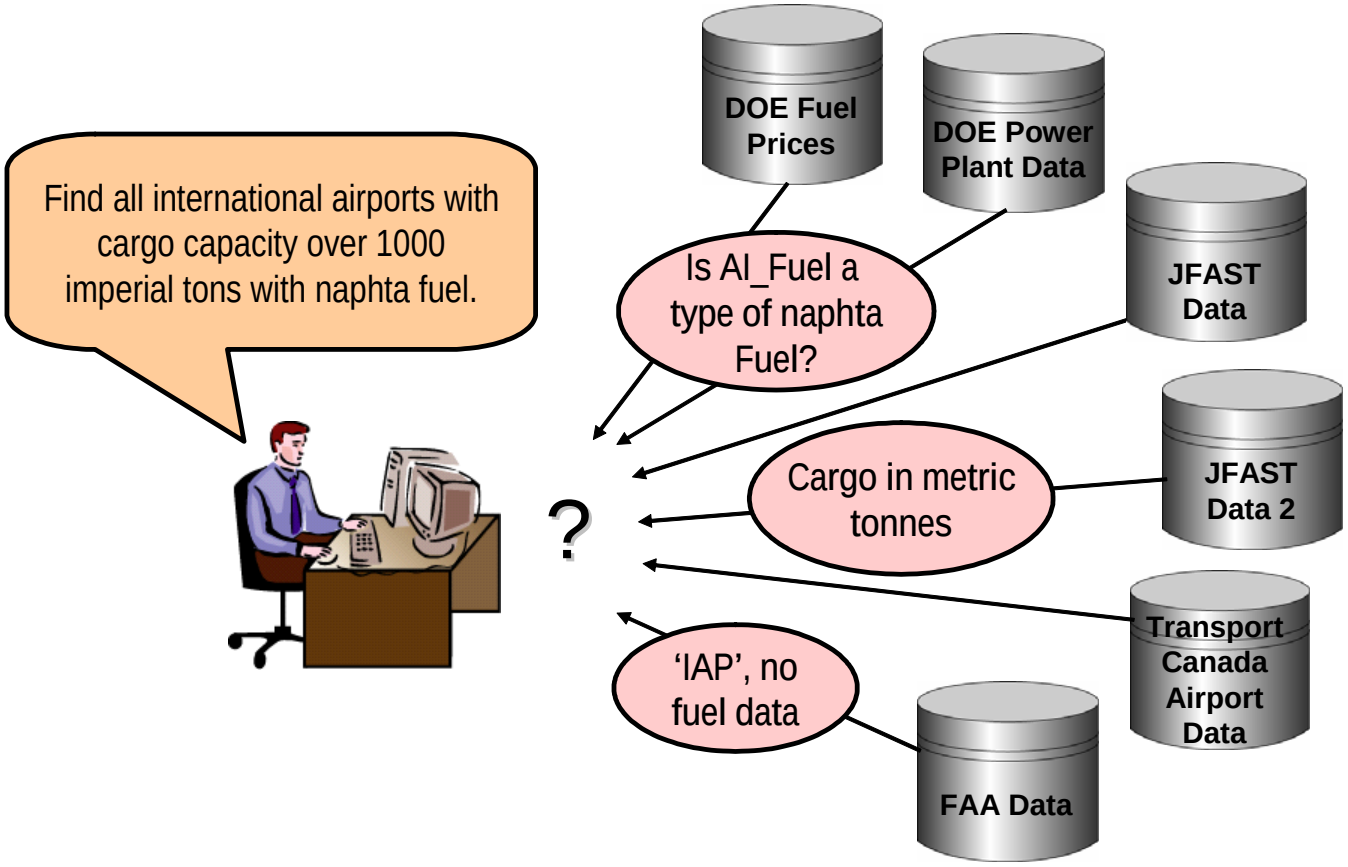
Semantic Interoperability

- "Semantic interoperability is defined as the enablement of software systems ... to interoperate at a level in which the exchange of information is at the enterprise level. This means each system (or object of a system) can map from its own conceptual model to the conceptual model of other systems, thereby ensuring that the meaning of their information is transmitted, accepted, understood, and used across the enterprise." –Obrst *et al*

- How and to what extent do ontologies facilitate *semantic interoperability*?

AIXE: IET's Semantic Integration Tool

- Information Extraction & Transport, Inc. (IET) is developing the Application Information Exchange Environment (AIXE), as a Phase II SBIR for the Navy, to:
  - allow users to quickly map new, dynamic and legacy data sources to the system.
  - integrate diverse data at query time to generate a single integrated data/knowledge base for answering queries.


Find all international airports with cargo capacity over 1000 imperial tons with naphta fuel.
?
DOE Fuel Prices
DOE Power Plant Data
JFAST Data
Is Al_Fuel a type of naphta Fuel?
JFAST Data 2
Cargo in metric tonnes
Transport Canada Airport Data
'IAP', no fuel data
FAA Data

AIXE General Approach

- Use an ontology/logic-based foundational data scheme that implements OWL markup plus other tools (translation scripts, Bayesian reasoning) for interoperability
- Simple ontology and logic-aided schema extension tool that logic-naïve users can implement

AIXE General Approach

- We extend the central ontology as necessary for each new data source (database tables, spreadsheets, structured web pages, etc.) and then define a translation scheme to wrap (or rewrap) the data sources with Class and property wrappers from the central ontology.
- For each data source, we define a mapping to our ontology on a field by field basis.

1	GEO	NAME	TYPE	CC	LAT	LON	RLAT	RLON	CINC	ICAO	COUNTRY	USEF
2	AFSD	Aalmeria	PRT	SP	365000N	0022800W	0.642863	4.31E-02	4		Spain	
3	AFSF	Aalmelo	STG	NL	522000N	0063800E	0.913389	-0.11577	4		Netherland	
4	AFSJ	Aalmena	CTY	20	395332N	0994224W	0.696251	1.74021	7		Kansas	
5	AFSL	Almeria	CTY	31	414933N	0993118W	0.729998	1.736981	7		Nebraska	
6	AFSM	Abdullex	MAP	ID	264240N	1205623W	0.832716	2.07589	7	WISI	Indonesia	
7	AFSN	Abbeville	MAP	FR	410601N	13758W	0.734788	1.983266	7	LFOW	France	
8	AFSP	Alimo	APT	23	364141N	0881642W	0.640444	1.540748	7	ETMA	Kentucky	
9	AFSQ	Alimond	CTY	9	353905N	0915850W	0.622234	1.605364	7		Arkansas	
10	AFSR	Alimond	CTY	16	421920N	0774410W	0.738662	1.356795	7		New York	
11	AFSS	Alimelo	CTY	NL	522157N	0064060E	0.913956	-0.1165	4		Netherland	
12	AFSV	Alimont	CTY	7	383953N	1065044W	0.674827	1.864807	7		Colorado	
13	AFSW	Alimont	CTY	22	415514N	0860242W	0.719105	1.239409	7		Michigan	
14	AFSX	Alimont	CTY	32	464331N	1013008W	0.81551	1.771548	7		North Dak	
15	AFSY	Almira	CTY	9	342422N	0912434W	0.6005	1.595396	7		Arkansas	
16	AFT1	Almiranto	PRT	PM	091700N	0822300W	0.162025	1.43786	6		Panama	

```
<rdf:Description rdf:about="#AFSP">
 <rdfs:label>Alimo</rdfs:label>
 <rdf:type rdf:resource="&aixeFds;#APT"/>
 <aixeFds:locationOfObject rdf:resource="&aixeFds;#ST23"/>
 <aixeFds:latitudeNumD rdf:datatype="&aixeFds;#LatLongDAFormat">41060N</aixeFds:latitudeNumD>
 <aixeFds:longitudeNumD rdf:datatype="&aixeFds;#LatLongDAFormat">0881642W</aixeFds:longitudeNumD>
 <aixeFds:latitudeRad rdf:datatype="&aixeFds;#LatLongRadians">0.640444</aixeFds:latitudeRad>
 <aixeFds:longitudeRad rdf:datatype="&aixeFds;#LatLongRadians">1.540748</aixeFds:longitudeRad>
 <aixeFds:icaoCode>ETMA</aixeFds:icaoCode>
</rdf:Description>
```

The mapping allows us to convert the data into AIXE format when we need it.

Reasoning Applications

- Identifying infrastructure objects in a given area
- Identifying potential dependencies
- Analyze “what if” scenarios.
- Collecting all information relevant to a particular object, location, etc.

- This presentation: Consider challenges that arise in integrating disparate data
  - How does the ontology and supporting inference tools ease integration of disparate data and what are the limitations?
  - Consider in terms of example questions that we might pose to the system

- Interoperability Issues
  - Identity and Glossary Control
  - Power of Transitivity Reasoning
  - The Space Carving Problem
  - Up and Down the Subclass Hierarchy (Granularity, Part 1)
  - Faceting
  - Combining Hierarchies
  - Format and Unit Translation
  - Granularity, Part 2
  - Credibility

Identity and Glossary Control

- Example Query: Find all civilian airports selling fuel of type F12

Suppose that other data sources use different labeling convention for fuel types, i.e., they refer to F-12 fuel with a different name. This points to an obvious ontology application, call it glossary control, the management of different labels for single objects and managing the polysemy of labeling terms.

CODE	FLIP	NATO	AKA	EEFC	DEFINITION
A	115	F-22		BA	115/145 octane gasoline, leaded, MIL-L-5572F (PURPLE)
B	100	None			100/130 octane gasoline, leaded, MIL-L-5572F (GREEN)
C	None	None		B91	91/96 octane gasoline, leaded, No MIL Spec.
D	80	F-12	887		80/87 octane gasoline, leaded, MIL-L-5572F (RED)
F	None	None	80NL		80 octane gasoline, unleaded, No MIL Spec.
G	None	None	AvGas		Aviation Gasoline (AVGAS), octane unknown.
H	None	None			108/135 octane gasoline, leaded, No MIL Spec.
K	None	None	73NL		73 octane gasoline, unleaded, No MIL Spec.
L	100LL F-18	B95,B100			100/130 MIL Spec, low lead, aviation gasoline (BLUE)

- Approaches to glossary control: (i) Reify a new object for each term used, and use identity reasoning **or** (ii) attach different labels to single objects?

```
:Flip_80                                    (i)
  a   owl:Class;
  rdfs:subClassof  LowOctaneGasoline.

:Nato_F12
  a   owl:Class;
  owl:equivalentClass    Flip_80.

:AKA_887
  a   owl:Class;
  owl:equivalentClass    Flip_80.
```

```
:Flip_80                                    (ii)
  a   owl:Class;
  rdfs:subClassof     LowOctaneGasoline.
  natoLabel:  “Nato_F12”;
  akaLabel:   “AKA_887”;
  flipLabel:  “Flip_80”.

:natoLabel
   a      owl:AnnotationProperty;
   rdfs:subPropertyOf     rdfs:label.
```

- Approach (i) to glossary control:
  - Use annotation properties:
    - Simply map each term to the object via ‘label’ or create subproperties of ‘label’ that allow us to quickly distinguish different labeling sources.
      - e.g., (subProperty natolabel label).
      - This keeps our ontology lean and mean, distinguishing annotation issues from reasoning and representation issues.

- Challenges:
  - This is a straightforward way to realize the interoperability but it becomes more difficult to use the data implementing that label or query using the terms.
    - Consider, if our data source indicates that

      (fuelTypeAvailable Airport639 AKA_887)

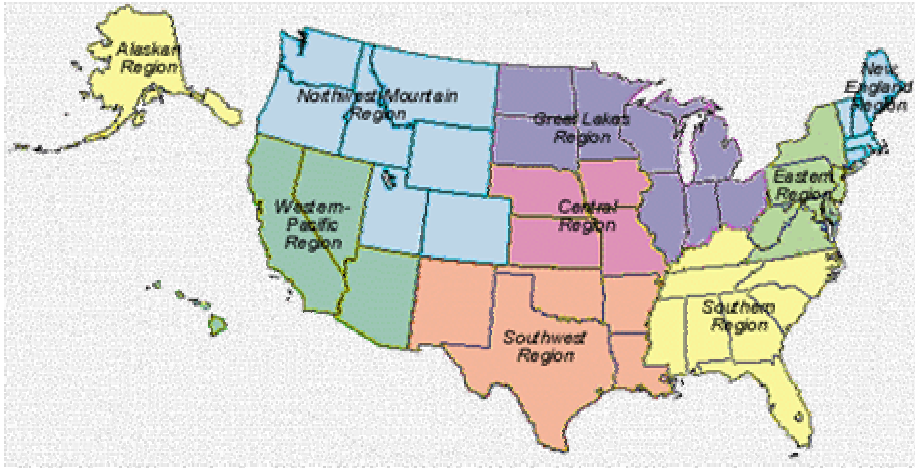
    if “AKA_887” is just a label in our ontology, we need to replace it with a direct reference to the object that it denotes, i.e., Flip_80. Similarly, “AKA_887” can’t be directly used in queries if it’s only a label, not a direct denotation of a reified object.

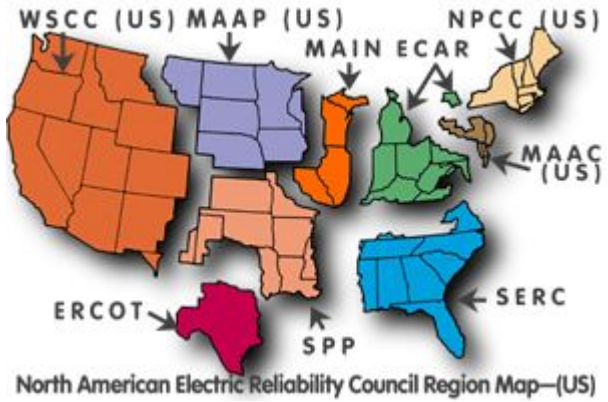
- Approach (ii) to glossary control:
  - Reify an object for each new name and then declare them as identical.
    - This simplifies data transformation and querying.
  - Challenges:
    - This may complicate inferencing depending on means of supporting identity reasoning, by dramatically increasing the size of the knowledge base or failing to support all the identity reasoning.
    - We conflate annotation issues with representation issues in our ontology.

Power of Transitivity and “Space Carving”

- Example Query: Find any objects in Western Pacific FAA region dependent on objects in NERC Region, SPP.

- Two Challenges:
  - Dependency linkages
  - Integrate the asset location and dependency information with information about two distinct federal region breakdowns, i.e., FAA regions and NERC regions. There are many ways to subdivide the physical regions into subregions and our system must reason across each.





Power of Transitive Reasoning

	A	B	C	D	E	F	G
1	DEPENDENCYID	ASSETID	SUPPORTSDEPENDENCYID	STARTTIM	ENDTIME	CRITICAL	RISK
2	1077	5161	1072				
3	1078	5723	1077				
4	1079	5612	978				
5	1080	5340	1079				
6	1081	5320	1080				
7	1082	5300	1079				
8	1084	5056	1083				
9	1085	5660	983				
10	1072	5910	1085				
11	1087	5911	1086				
12	1088	5891	1085				
13	1089	5561	1085				
14	1090	5560					
15	983	5066					
16	1092	5910					
17	1093	5191	1092				
18	1094	5189	1091				

→ *(dependentOn 5723 5066)*

Suppose this table specifies dependencies between assets.
We can extend the reasoning by enforcing the transitivity of dependence.
This query is more difficult in straight SQL, easy with transitive reasoning.

Space Carving

- The integration challenge arises from the need to integrate asset information with different geographical information.
  - (location ASSET_5066 City345)
  - (subRegionOf City345 New Mexico)
  - (subRegionOf New_Mexico SPP) → (location Asset_5066 SPP)

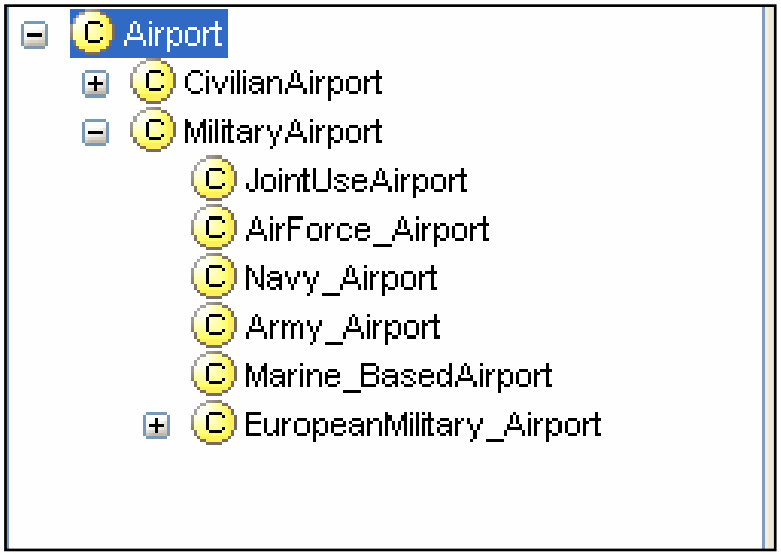
  - (location ASSET_5723 City234)
  - (subRegionOf City234 California)
  - (subRegionOf California Western_Pac_Reg) → (location Asset_5723 WPR)
- The integration of different “space carvings” requires:
  - That the ontology contain the high level parts in terms of which we can define the distinct space carvings.
  - The ability to represent and reason about the transitive parthood relations, i.e., that B’s parts are A’s parts if B is part of A.

Up and Down the Subclass Hierarchy

- Example Query: Find all military airports in the northwest

- Challenge: The challenge here lies in the fact that some data sources distinguish between air force airports, naval airports and other DOD-controlled airports. Similarly, some distinguish between joint-use airports (military and civilian) and military airports. Others simply distinguish between military and civilian airports. (Also, system needs to integrate geographical information and recognize all parts of the northwest.)

- This is addressed rather straightforwardly, i.e., by utilizing subtyping.



Interested users can query at the desired level of specificity. However, a more general query will also capture instances of more specific subclasses. The utilization of hierarchies overcomes *some* of the challenges associated with representations at different granularity levels.

Note that the class hierarchy also allows users to quickly extend the ontology and map to existing schemes. And, users can do extensive querying with a lot of ignorance of the original data schemes.

Faceting

- Example Query: Find [city, airport, fuel type] most similar to [city, airport, fuel type] X.

- Challenge:
  - Different data schemes carve up concepts with respect to different properties. Airport subtyping might be done with respect to location, size, functionality, etc. Similarly, fuel typing might be done in terms of basic chemical makeup (e.g., kerosene vs. gasoline) and/or kinds and levels of additives, (octane, lead, deicer).
  - Answering the above question, and integrating new data into the ontology depends on the ability to quickly determine the different ways in which the reasoning space is carved up.

- Integrating these different representational schemes requires hierarchical reasoning but also some kind of “faceting” or partitioning of the reasoning space. Ideally, our integration ontology allows us to partition or carve up the workspace in different ways. One solution, second order classes:
  - AirportsByFunction = {CivilAirport, MilitaryAirport, JointUseAirport, …}
  - AirportsBySize = {MetropolitanAirport, MidSizeAirport, SmallRegionalAirport, …}

- We must recognize what the different representations have in common, e.g., all are subclasses of airport, but also allows us to focus on different ways to subdivide the reasoning space.
- This approach requires both multiple inheritance and second order classes (beyond DL reasoners).
- This facilitates data retrieval and the mapping of new concepts into the domain, i.e., it becomes easier to find the different ways in which the domain is partitioned/faceted.

Combining Hierarchies

- Example Query: Find all training facilities in VA controlled by the DoD.

- Challenge: Relevant information is stored in up to four different data sources, i.e., geographical information about VA, subclass hierarchies about military infrastructure, parthood information about military infrastructure, and military organization charts.
- Here we're doing more than simple "isa" reasoning, we're trying to reason about the extent to which properties of the whole apply to the part, and vice versa.
  - Consider the DoD, many properties of its parts don't apply to the whole, but some do. We need to write more subtle rules to reason about this.
    - "All things controlled by suborganizations of an organization are controlled by the organization"
- This starts to push us further beyond simple DL-based ontologies, this is most easily accomplished with horn rules or other representation and reasoning tools beyond DLs.

Ontology Limitations

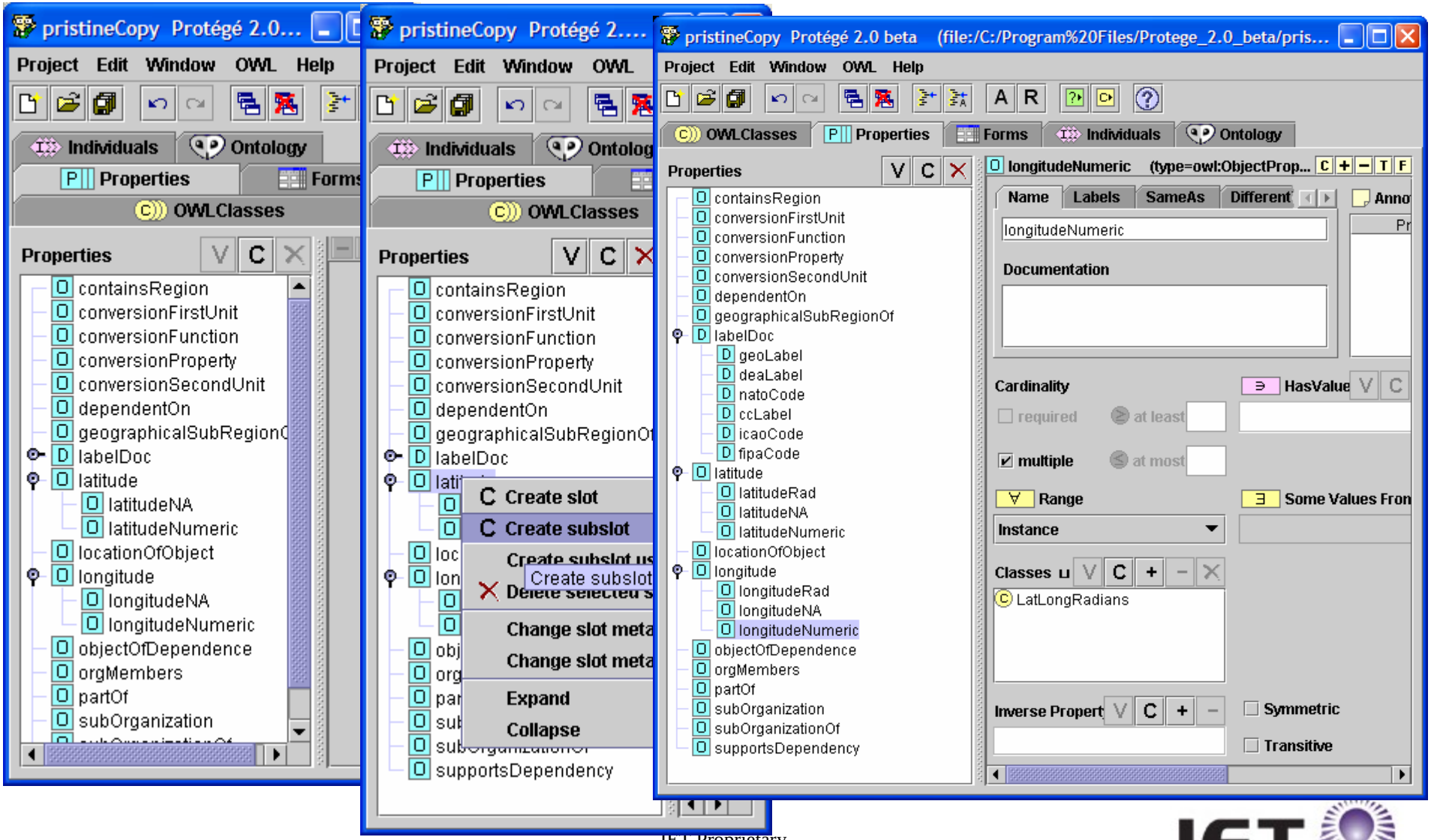
- Obviously, the more reasoning we can do the easier it is to query and integrate disparate data sources, but what kinds of things can't we do with ontologies alone?
  - Different formats
  - Some granularity challenges
  - Credibility reasoning

Format and Unit Translation

- Example Query: Find all military assets between 70 and 55 W and 30 and 40 N capable of carrying over 100 metric tonnes/day.

- Challenge: One of our data sources represent location information in terms of radians, and most of them represent cargo capacity in terms of imperial tons.
- How can an ontology help here?
  - We use the ontology to track datatypes and create datatype property hierarchies for purposes of guiding calls to translation tools.

Subproperty hierarchies are used to guide translation



Format and Unit Translation

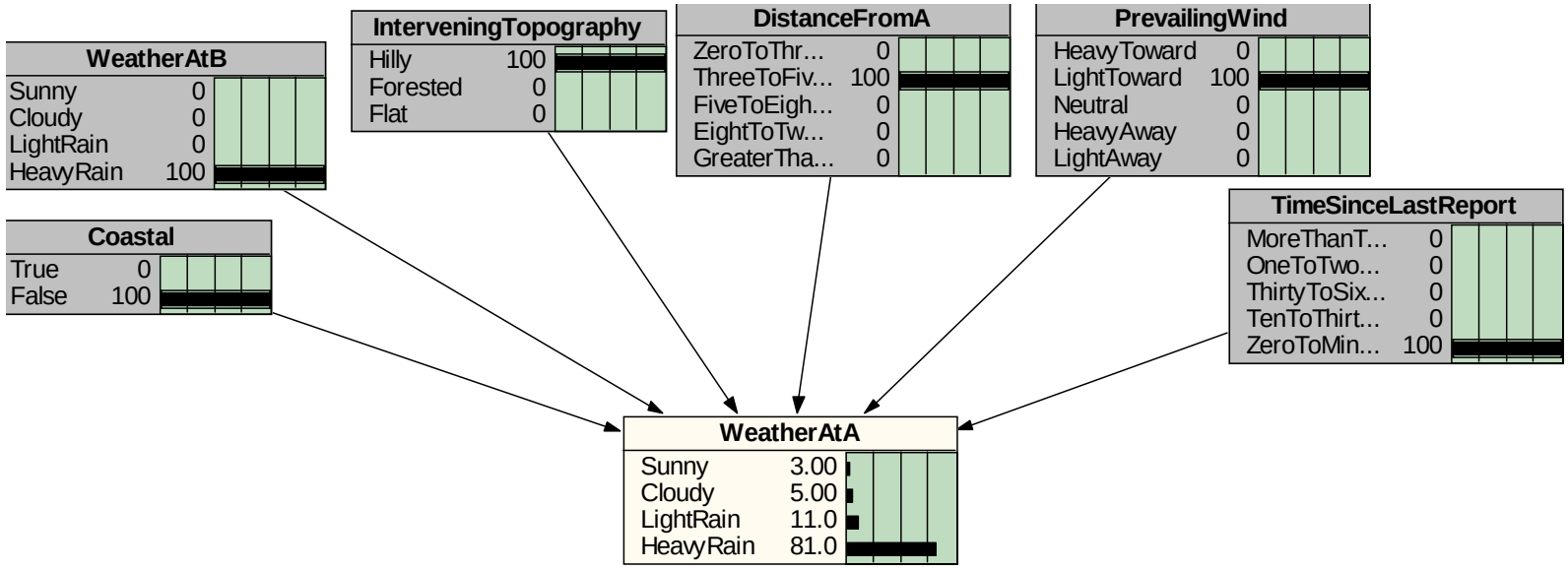
- Parse query and remove all translatable properties
  - Subquery to determine relevant “sibling” properties.
  - (<aixeFds:latitudeDegree>,<rdf:subPropertyOf> ?X)(?PROP rdfs:subPropertyOf ?PROP)
  - And look for property pairs for which a translation function is defined
- SELECT ?apt ?lat ?long WHERE
  - (?apt, <rdf:type>,<aixeFds:Military-Airport>)(<aixeFds:latitude> ?apt ?lat)(aixeFds:longitude,?apt,?long)
  - Note that this will return latitudeNumeric, latitudeRad and latitudeNA (these are the subproperties) and then we invoke appropriate translation tools. Ontology helps to render the search reasonable.

Granularity, Part 2

- Example Query: Describe terrain at region3352
- Example Query: Is it raining at location T?

- Challenge:
  - We know the terrain in three subregions of X, how do we integrate that into a terrain assessment for X?
  - We know weather in three different locations surrounding T, how do we approximate weather at T?

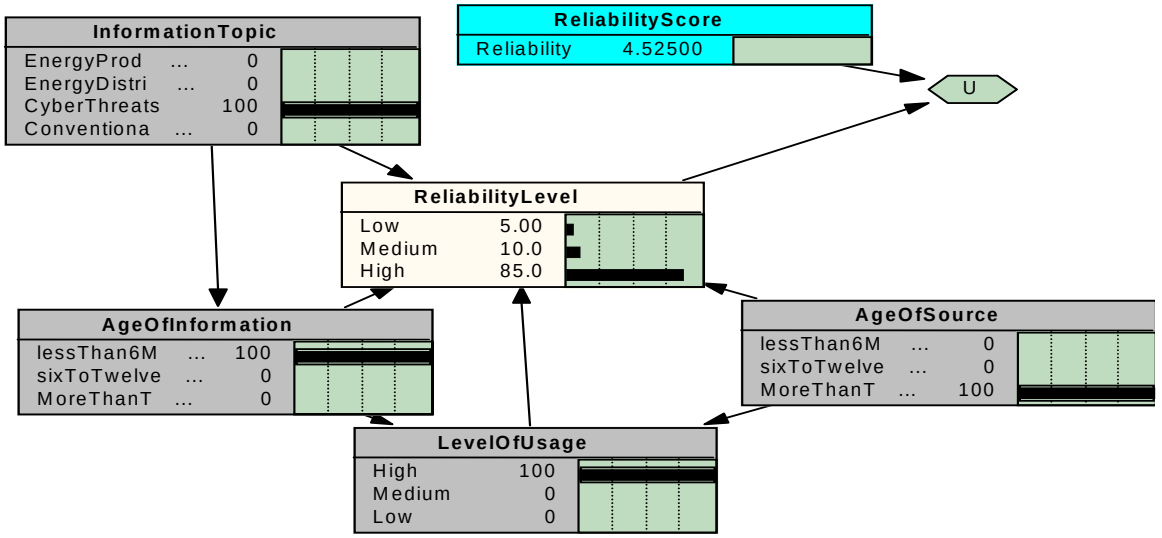
- Here we may have to resort to other reasoning means to reason from one granularity level to another or to reapply known information to the question at hand:



Credibility

- Consider other challenges:
  - How do we resolve contradictory or differing reports from amongst the different data sources?

- Challenge:
  - Use metadata to evaluate new data sources



Ontology Translation Protocol (Ontrapro)

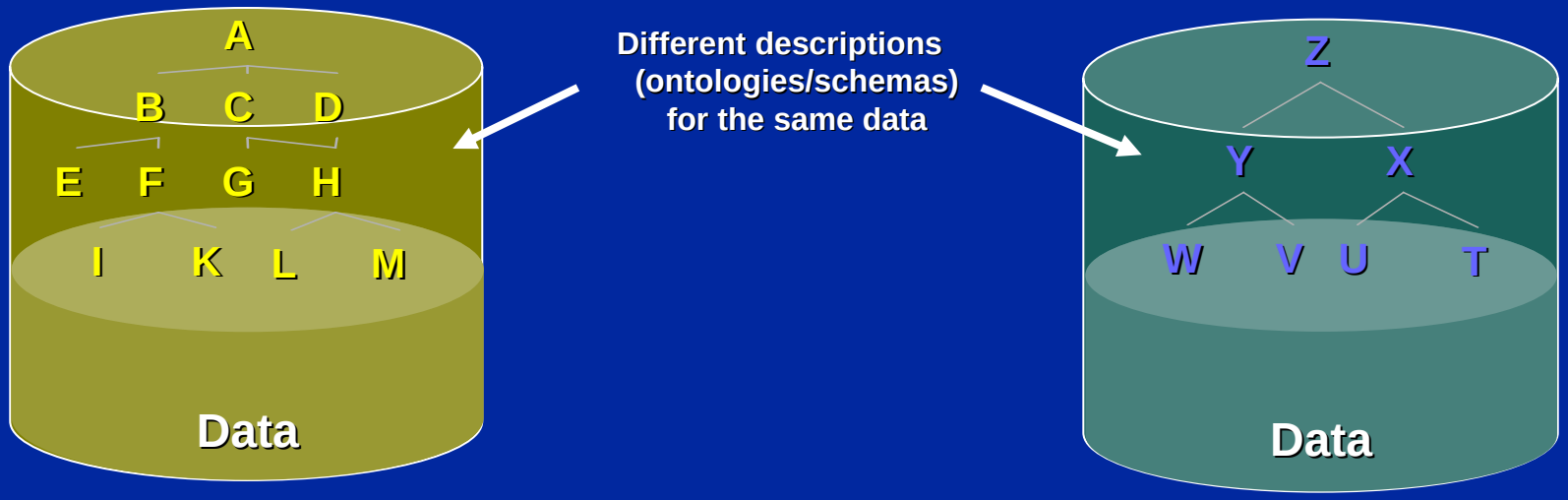
Automated Ontology Alignment

LOCKHEED MARTIN

Ben Ashpole
Member, Engineering Staff
Advanced Technology Laboratories

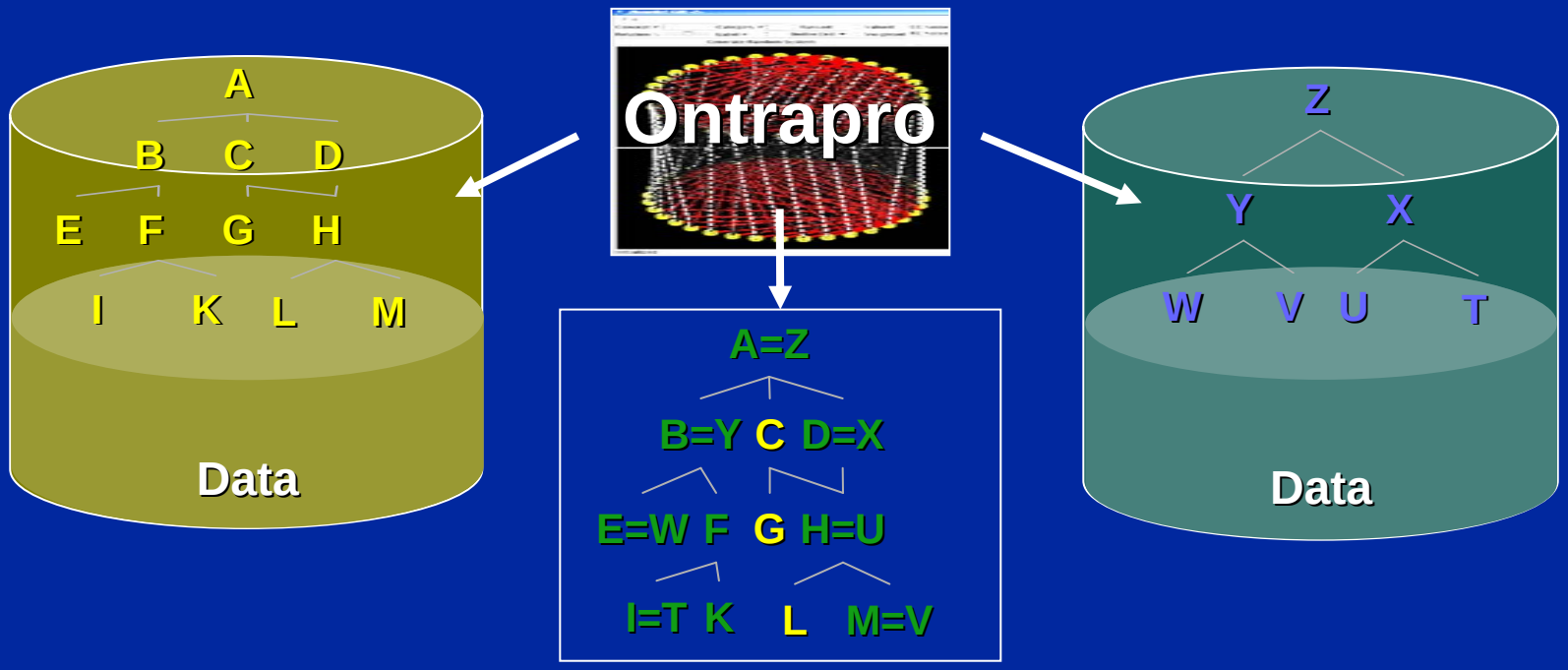
Ontrapro

- Semantic integration is needed when different information systems have different formal descriptions of the same type of data
- E.g., Inter-agency intelligence data sharing, joint sensor surveillance, multinational command and control
- Semantic Integration is also a key challenge of the Semantic Web

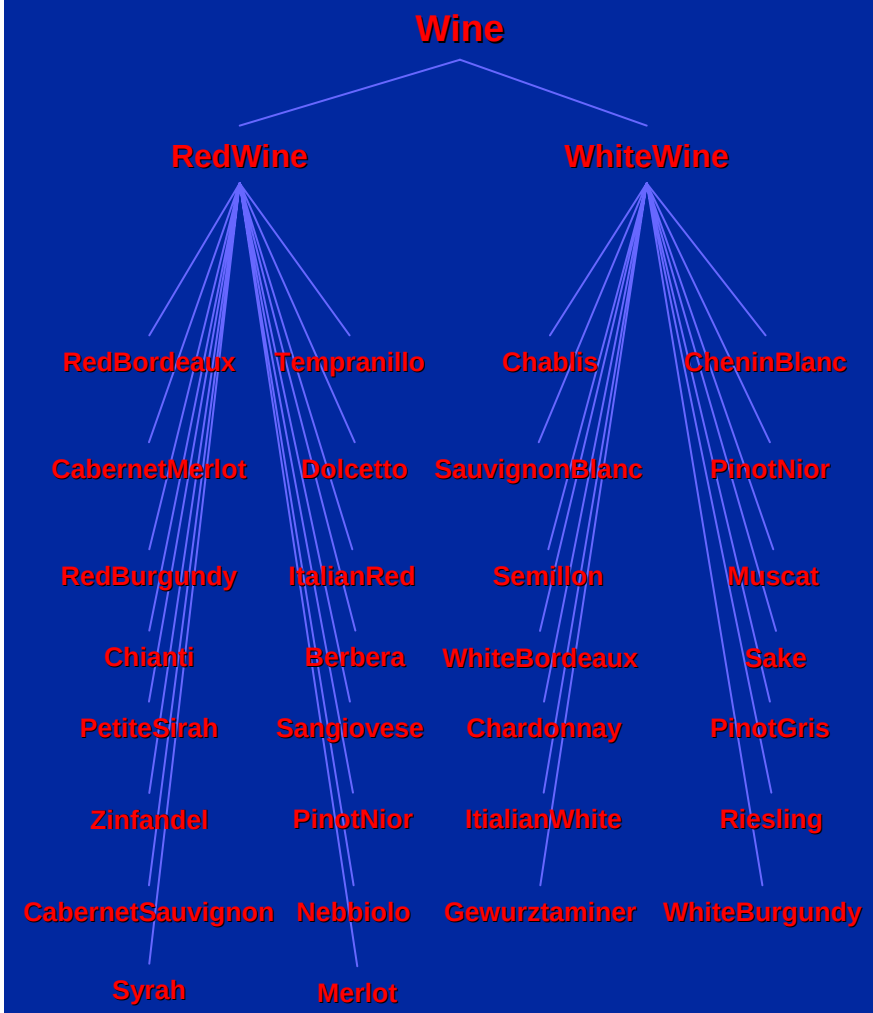


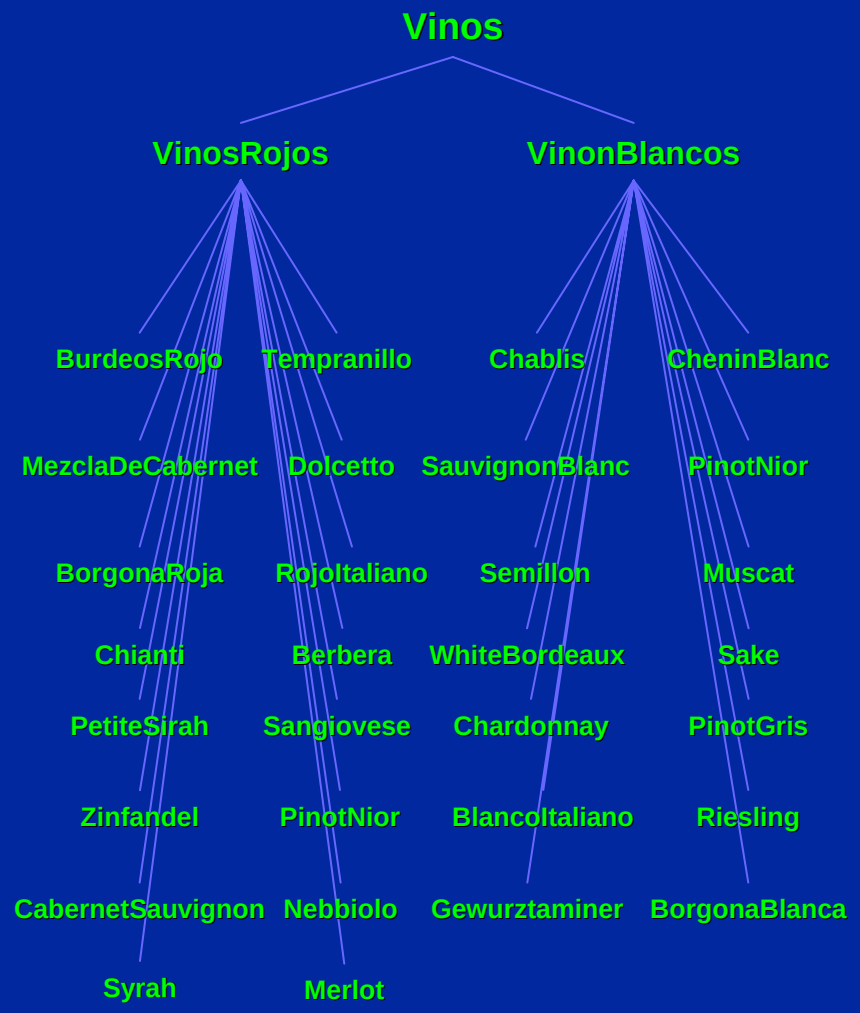
Ontrapro

- **Our approach is to develop and/or integrate a suite of best-of-breed aligner algorithms**
  - Ontrapro seeks to automate the process of aligning the data descriptions of *fielded* information resources
  - Ontrapro discovers semantic correspondences between the elements of ontologies and schemas

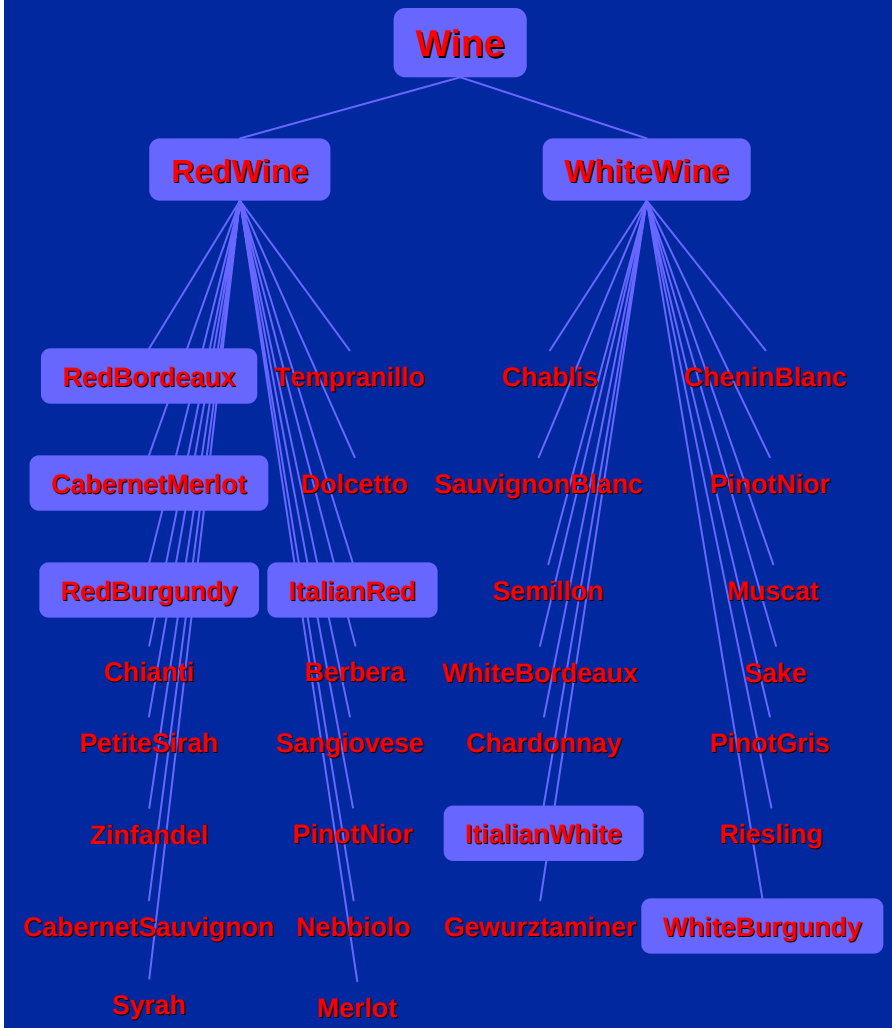


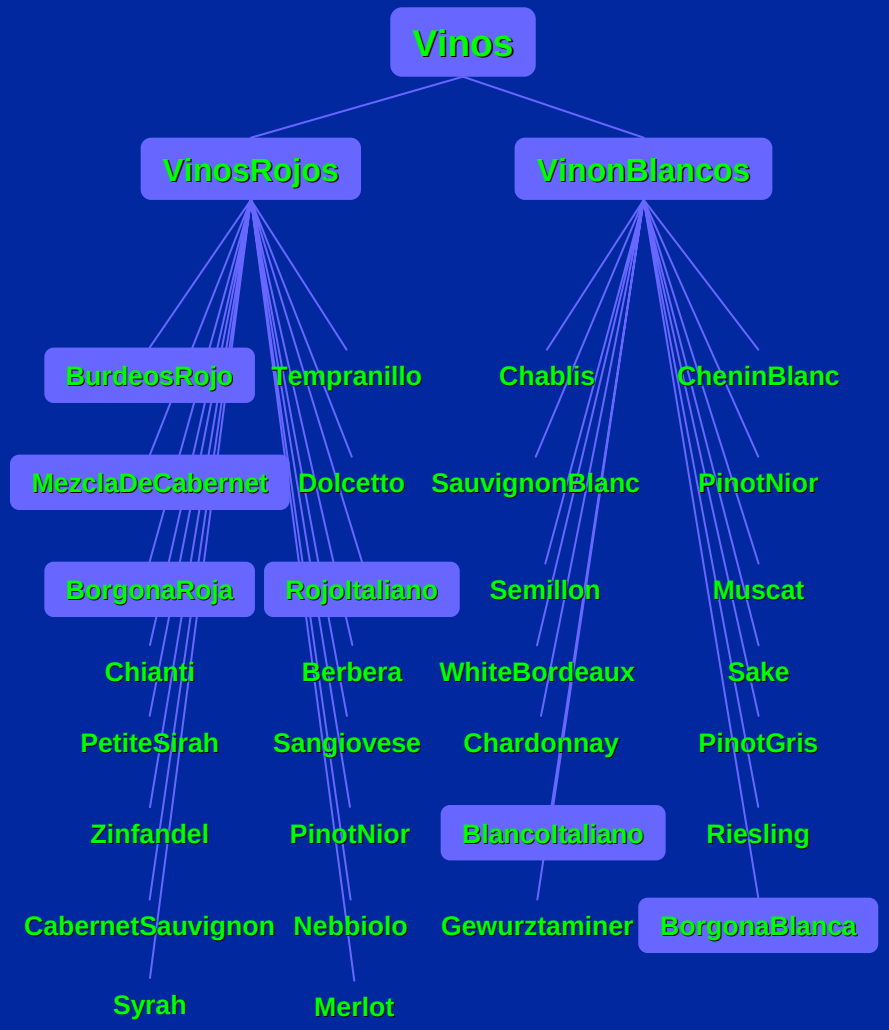
Example: Wine Ontologies



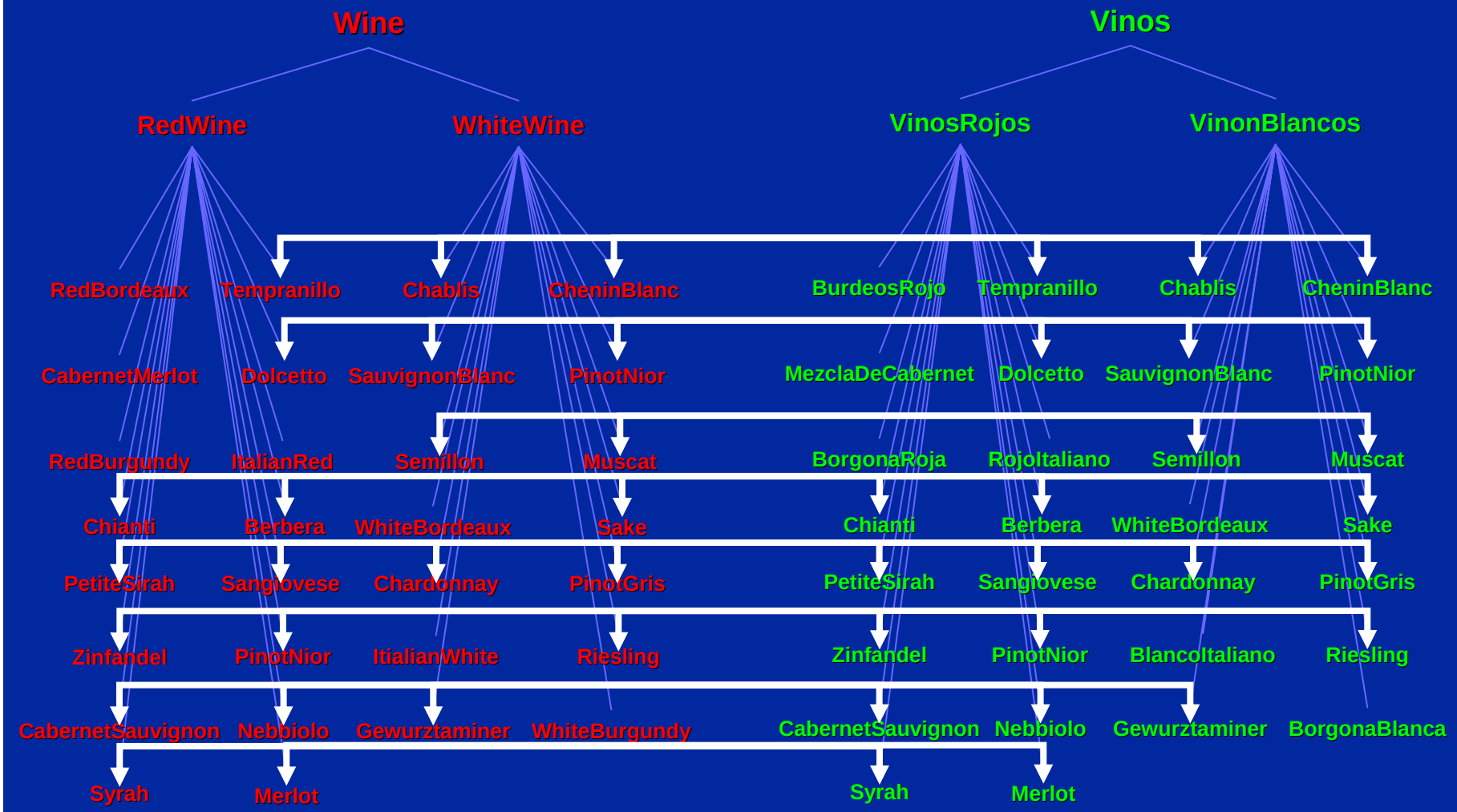


Example: Wine Ontologies
Term Dissimilarities

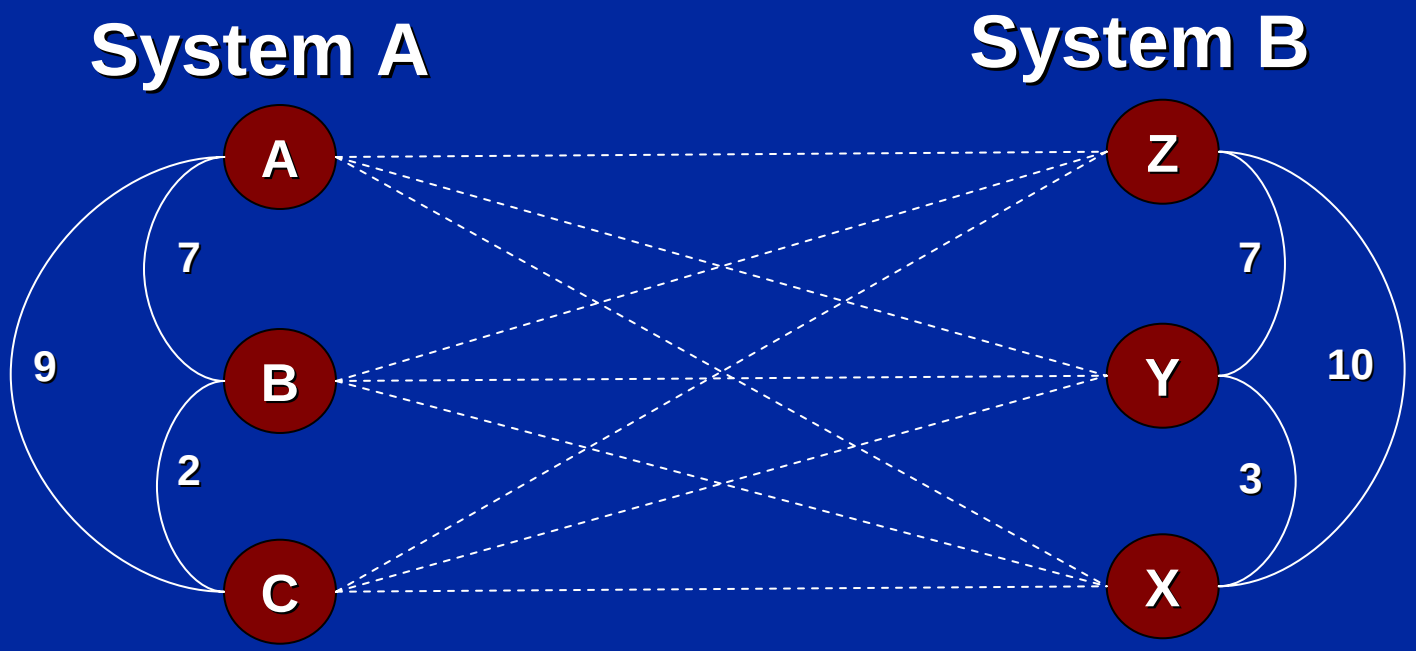




Example: Wine Ontologies
Edit Distance Mapping

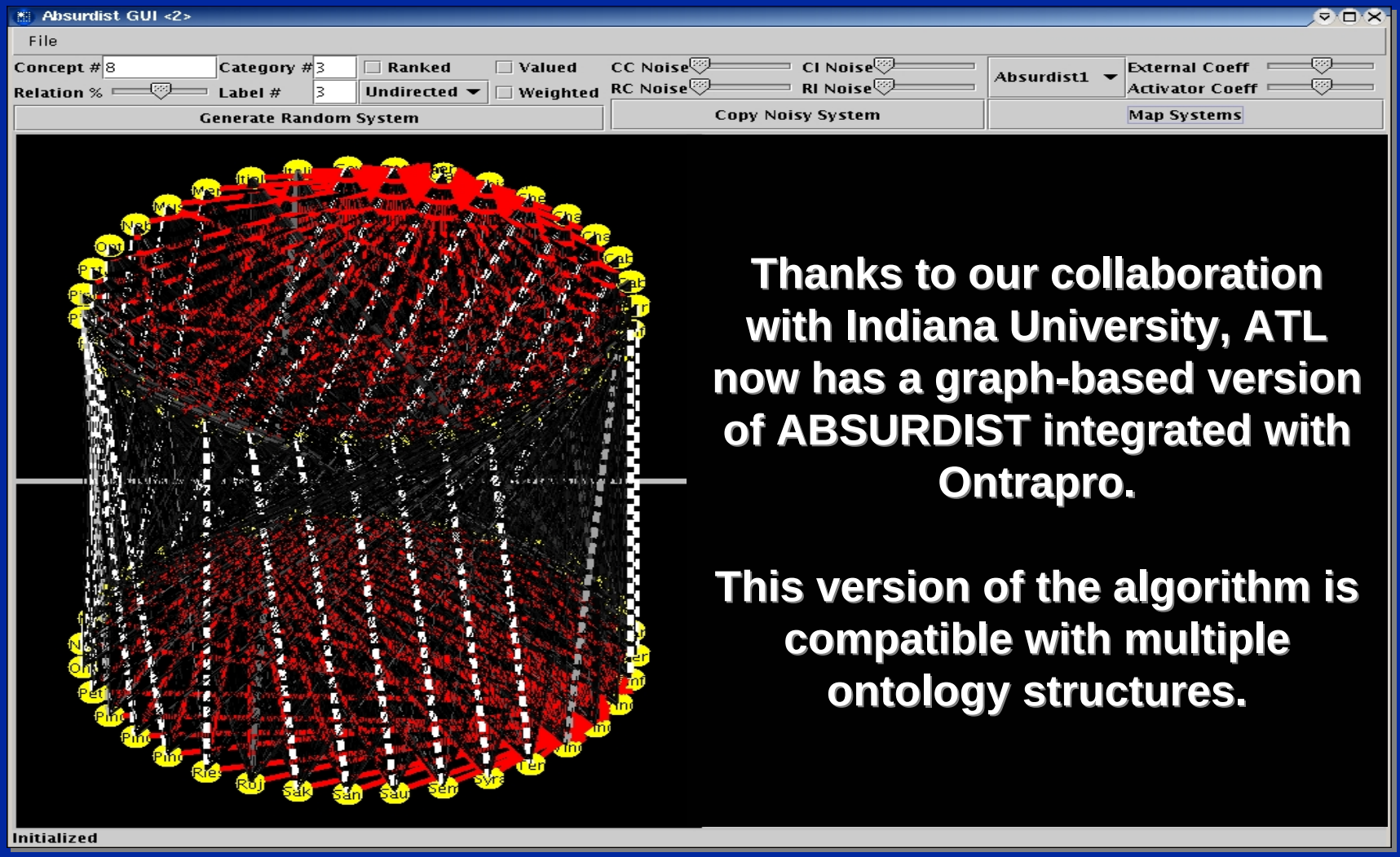


Example: Wine Ontologies
Structure Mapping



Original ABSURDIST (Aligning Between Systems Using Relations Derived Inside Systems for Translation) algorithm generates a list of term correspondences by iterating through similarity distances between systems.

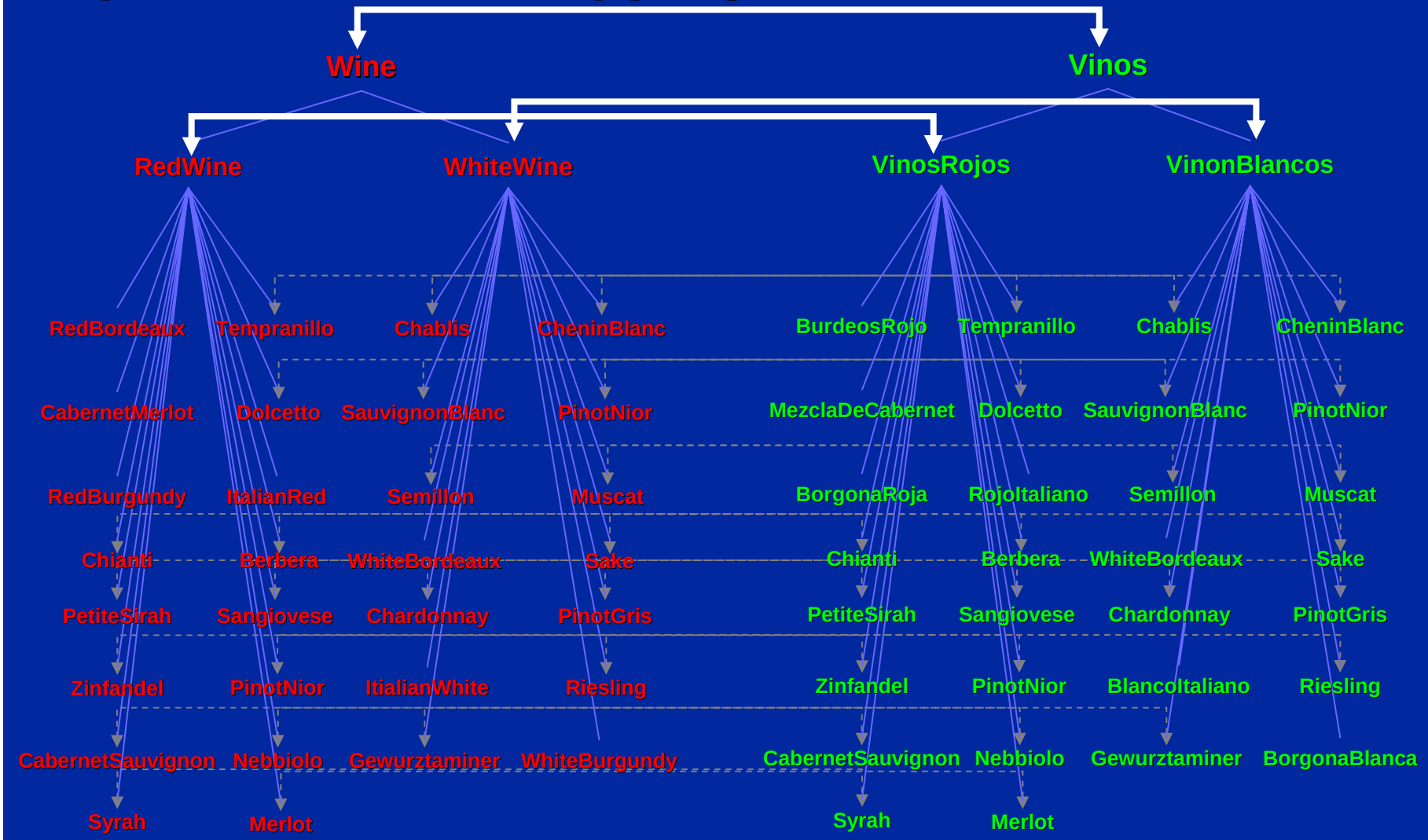
Example: Wine Ontologies
Structure Mapping



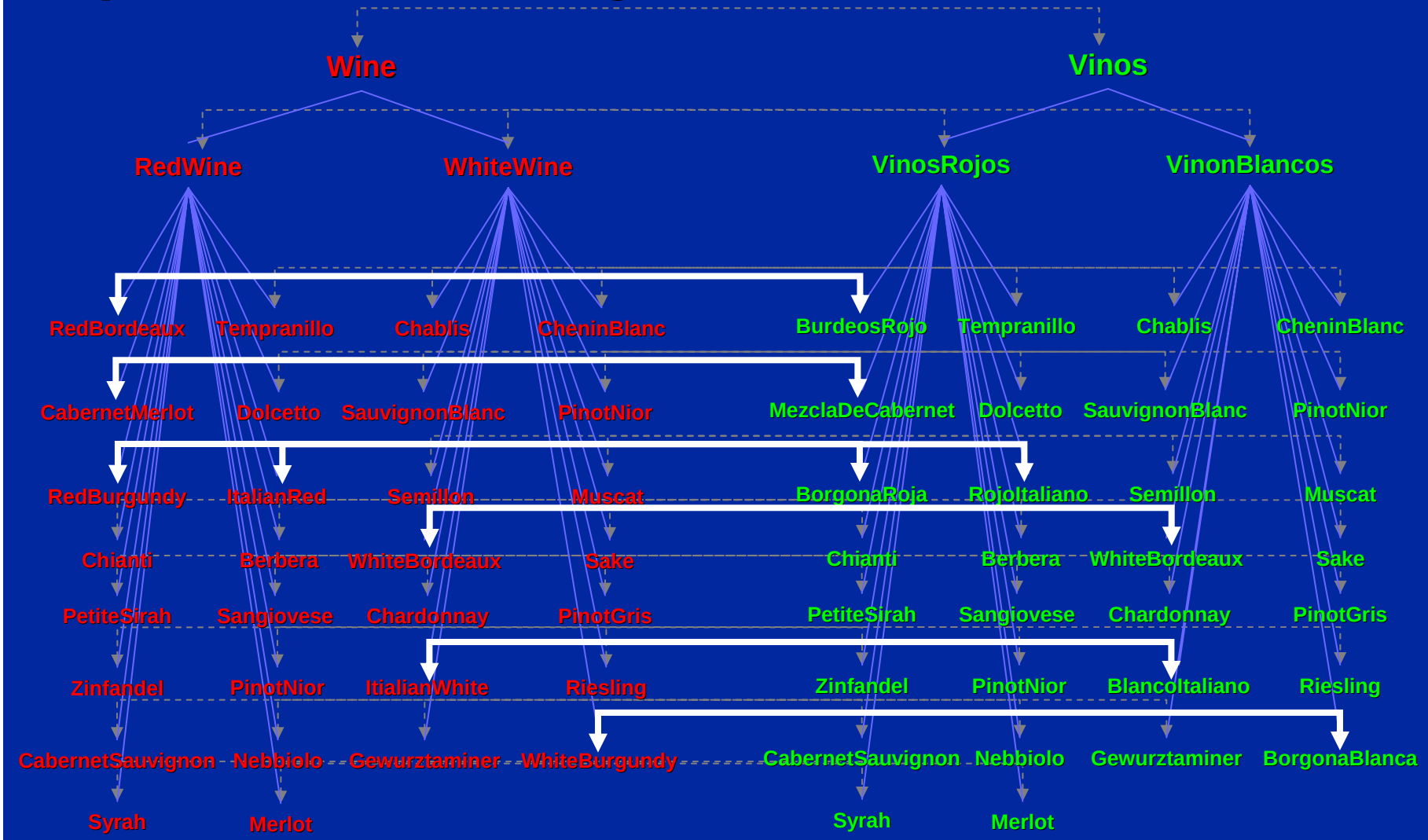
Thanks to our collaboration with Indiana University, ATL now has a graph-based version of ABSURDIST integrated with Ontrapro.

This version of the algorithm is compatible with multiple ontology structures.

Example: Wine Ontologies
…plus structure mapping

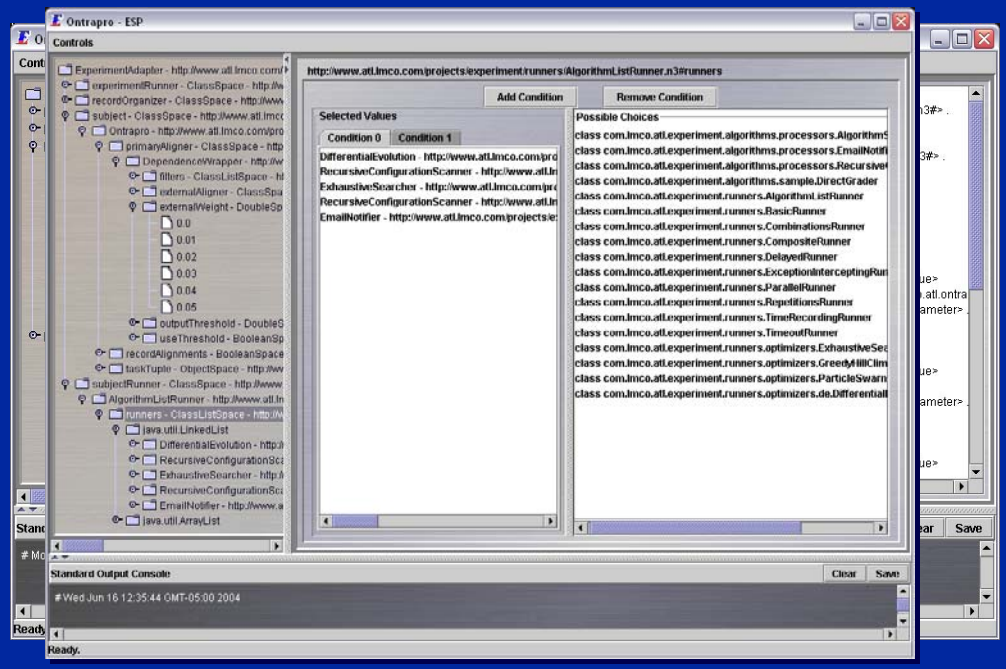


Example: Wine Ontologies
…plus stable marriage



Experiment Set Platform

- **ATL has developed a platform for**
  - — **Semi-automating experiment setup**
  - — **Automating experiment execution**
  - — **Automating data collection**
  - — ***Used to grade alignments***
- **Employs a core set of ontologies**
  - — **General Experiment**
  - — **Ontology Alignment**
  - — **Alignment Evaluation**
  - — **Ontology Operation**
  - — **Operation Evaluation**



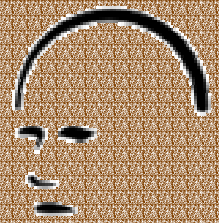
http://cvs.sourceforge.net/viewcvs.py/ontology/

Ontrapro Software and Tools

- **Developed for Ontrapro**
  - —ABSURDIST
  - —Various aligners and filters
  - —Experiment tools
    - E.g., AlignmentGrader, CombinationsRunner
  - —Query tools
  - —Integration framework
- **Adopted for Ontrapro**
  - —Jena
  - —Similarity Flooding
  - —TreeJuxtaposer
  - —KavaChart
  - —IsaViz

Ontology Alignment Source

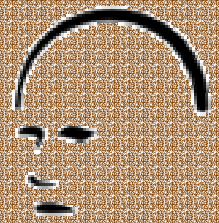
http://www.atl.external.lmco.com/projects/ontology



Application of Formal Ontology to Database Schema Alignment - an Outline

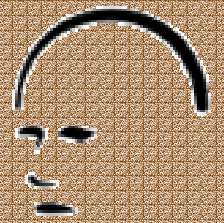
Bill Andersen
Ontology Works, Inc.

Schema alignment?



- UML-based database schemas are impoverished
- OWL-DL-based ontologies suffer the same problem
- What is needed to effect alignment of databases built on these formalisms?
- Depends on what we mean by alignment in the context of federated query

Oracle Sales DB



Employee

Id	Name	Dept

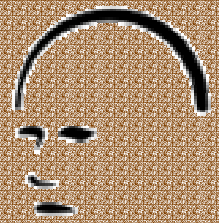
Does ‘Employee’ correspond to a type?

Are the employee ID numbers given by the company that owns the database or are they surrogate keys?

Employees of what company? It’s not in the schema.

And the IDs are useless for combining information about employees across databases from different companies

Oracle Sales DB



Product

Id	Name

Order

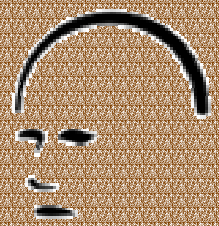
Id	Prod	Qty

Are products individual objects like employees?

Are product names comparable across DBs?

What are orders?

Oracle Sales DB



Department

Id	Name	Mgr

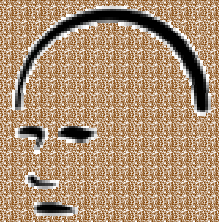
Does ‘Department’ correspond to a type?

Is an ID of a ‘Department’ like an ID of an employee?

Departments of what company? It’s not in the schema.

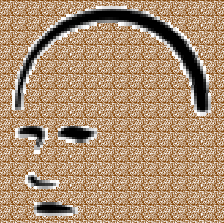
Does combining information about Departments across companies even make sense?

Heuristics aren’t enough



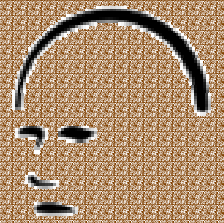
- Lexical matching of schema element names doesn’t answer these questions
- Structural heuristics that consider statistically significant clusters of matches don’t provide the answers either
- What could?

Formal Ontology



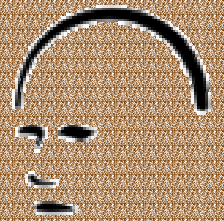
- Concerned with the most general structures of (domain-independent) reality
  - Identity
  - Mereology
  - Dependence
  - Modality and change
- Here we can find some tools that provide the additional information needed to answer our questions

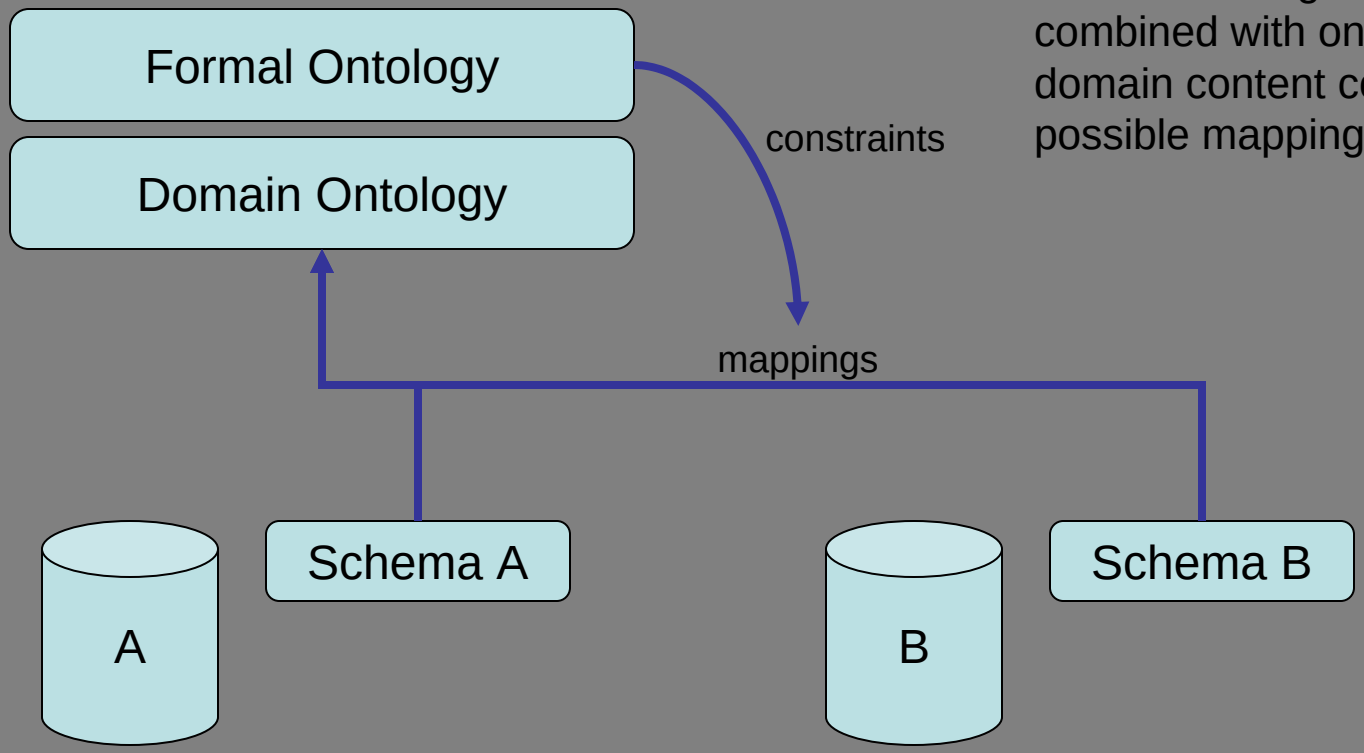
This stuff can't be relevant, can it?



- Dependence
  - Departments and employees are dependent entities … dependent upon organizations
- Mereology
  - Departments are ultimately part of organizations that are not dependent
- Identity
  - Cross-DB individuation of employees and can't come from employee IDs
- Modality
  - Employeehood is a contingent matter

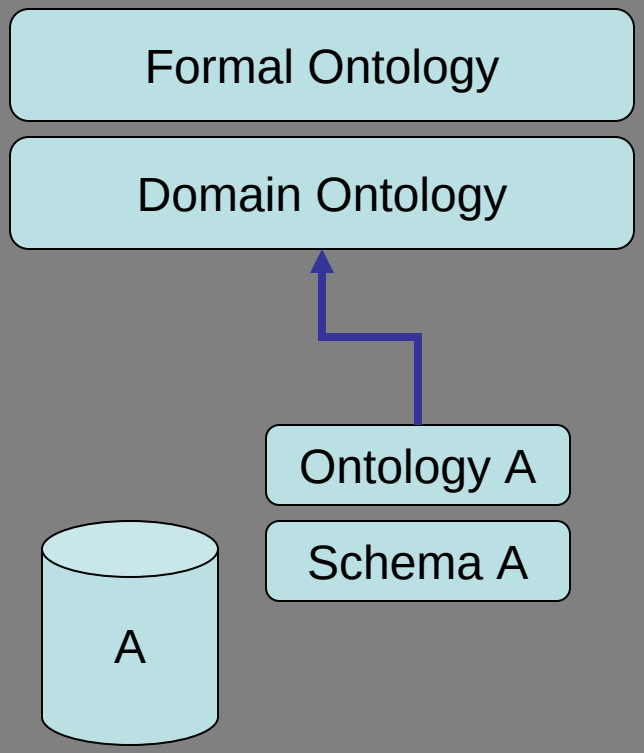
Ontology-based federation





Formal-ontological principles combined with ontologized domain content constrain possible mappings

A sketch of a process



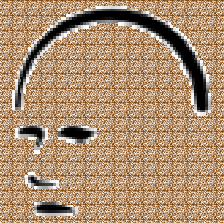
Map Schema A elements to Domain Ontology

Construct elaboration Ontology A based on dependence relations

Construct views that disambiguate information in Database A

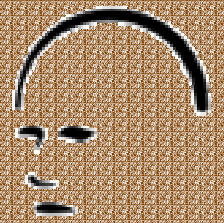
Identity criteria point to where inter-DB concordances are required

Requirements

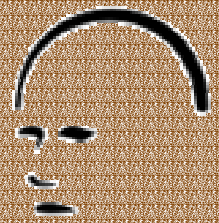


- Ontology language must support limited second-order logic (e.g., SCL)
  - Full power of logic not needed once mappings established (or else we're in trouble)
  - OWL-Full may be sufficient
- Lambda abstraction may be necessary
- Suitable formal ontological content is required (e.g., BFO, DOLCE)

Summary



- Formal-ontological notions provide needed information to control search for mappings
  - Constraints augment heuristic techniques
- Formal-ontological notions fill in missing information
  - Improving accuracy of cross-database query
- Formal-ontological basis provides neutral model for integration of arbitrarily many databases
  - Avoids ontological short-cuts



Questions?

Uncertainty in Ontology Mapping:

A Bayesian Perspective

Yun Peng, Zhongli Ding, Rong Pan

Department of Computer Science and Electrical engineering
University of Maryland Baltimore County
ypeng@umbc.edu

UMBC
an Honors University in Maryland

Outline

- Motivations
  - Uncertainty in ontology representation, reasoning and mapping
  - Why Bayesian networks (BN)
- Overview of the approach
- Translating OWL ontology to BN
  - Representing probabilistic information in ontology
  - Structural translation
  - Constructing conditional probability tables (CPT)
- Ontology mapping
  - Formalizing the notion of “mapping”
  - Mapping reduction
  - Mapping as evidential reasoning
- Conclusions

Motivations

- Uncertainty in ontology engineering
  - In representing/modeling the domain
    - Besides ***A*** *subclasOf* ***B***, also ***A*** is a *small* subset of ***B***
    - Besides ***A*** *hasProperty* ***P***, also *most* objects with ***P*** are in ***A***
    - ***A*** and ***B*** overlap, but none is a subclass of the other
  - In reasoning
    - How close a description ***D*** is to its most specific subsumer and most general subsumee?
    - Noisy data: leads to over generalization in subsumptions
    - Uncertain input: the object is *very likely* an instance of class ***A***

Motivations

- In mapping concepts from one ontology to another
  - Similarity between concepts in two ontologies often cannot be adequately represented by logical relations
    - Overlap rather than inclusion
  - Mappings are hardly 1-to-1
    - If ***A*** in onto1 is similar to ***B*** in onto2, ***A*** would also be similar to the sub and super classes of ***B*** (with different degree of similarity)
- Uncertainty becomes more prevalent in web environment
  - One ontology may import other ontologies
  - Competing ontologies for the same or overlapped domain

Bayesian Networks

- Why Bayesian networks (BN)
  - Existing approaches
    - Logic based approaches are inadequate
    - Others often based on heuristic rules
    - Uncertainty is resolved during mapping, and not considered in subsequent reasoning
      - Loss of information
  - BN is a graphic model of dependencies among variables:
    - Structural similarity with OWL graph
    - BN semantics is compatible with that of OWL
    - Rich set of efficient algorithms for reasoning and learning

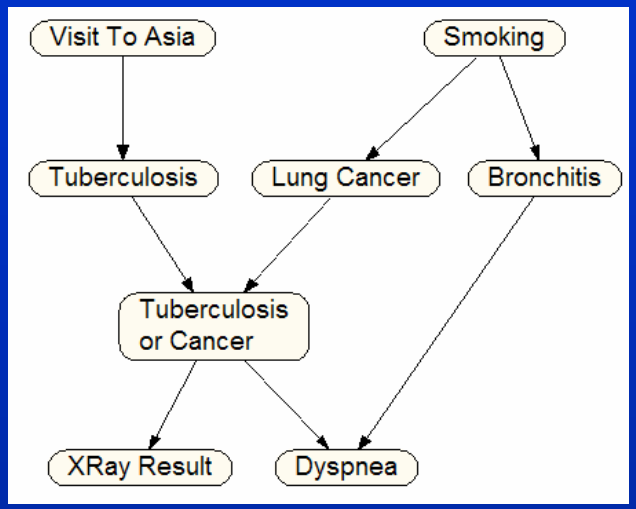
Bayesian Networks

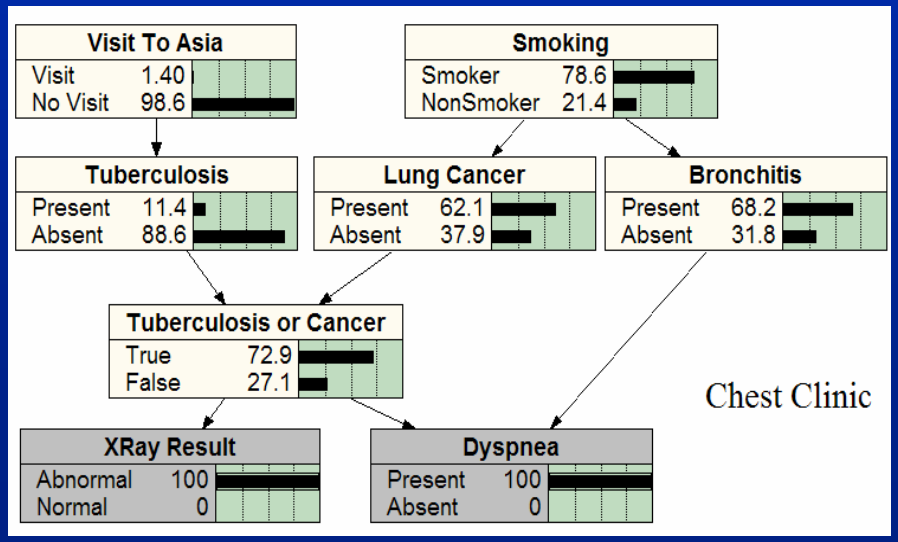
- Directed acyclic graph (DAG)
  - Nodes: (discrete) random variables
  - Arcs: causal/influential relations
  - A variable is independent of all other non-descendent variables, given its parents
- Conditional prob. tables (CPT)
  - To each node: $P(x_i \mid \pi_i)$ where π_i is the parent set of x_i
- Chain rule:
  - $P(x_1,\ldots x_n) = \Pi_i P(x_i \mid \pi_i)$
  - Joint probability as product of CPT

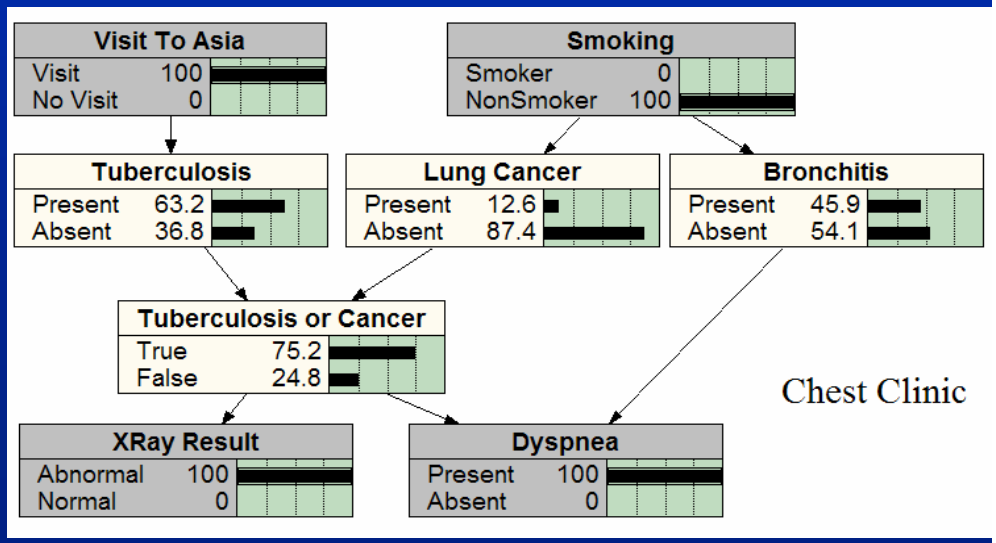
Bayesian Networks

Visit	No_Visit
1.000	99.000

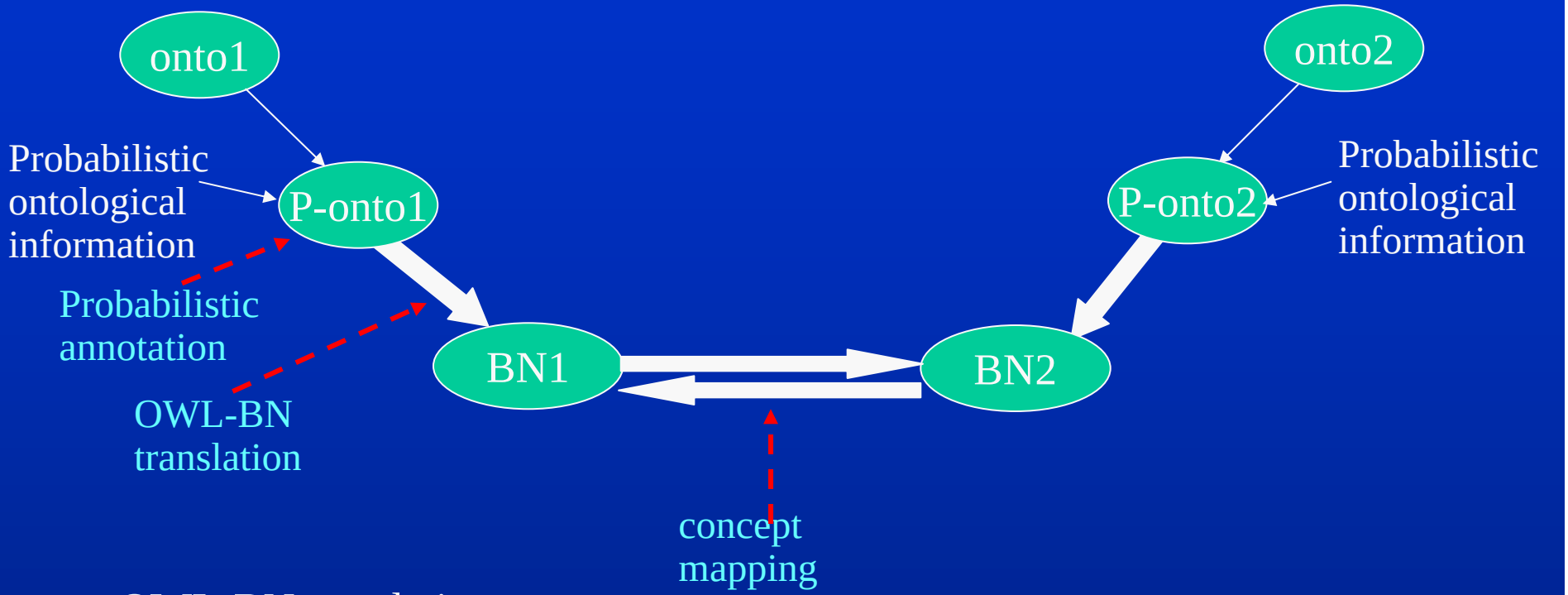
Tuberculosis	Cancer	Tb...
Present	Present	True
Present	Absent	True
Absent	Present	True
Absent	Absent	False







Overview of The Approach



- OWL-BN translation
  - By a set of translation rules and procedures
  - Maintain OWL semantics
  - Ontology reasoning by probabilistic inference in BN

- Ontology mapping
  - A parsimonious set of links
  - Capture similarity between concepts by joint distribution
  - Mapping as evidential reasoning

OWL-BN Translation

- Encoding probabilities in OWL ontologies
  - Not supported by current OWL
  - Define new classes for prior and conditional probabilities
- Structural translation: a set of rules
  - Class hierarchy: set theoretic approach
  - Logical relations (equivalence, disjoint, union, intersection...)
  - Properties
- Constructing CPT for each node:
  - Iterative Proportional Fitting Procedure (IPFP)
- Translated BN will preserve
  - Semantics of the original ontology
  - Encoded probability distributions among relevant variables

Encoding Probabilities

- Allow user to specify prior and conditional Probabilities.
  - Two new OWL classes: “PriorProbObj” and “CondProbObj”
  - A probability is defined as an instance of one of these classes.
- P(A): e.g., P(Animal) = 0.5

```
<prob:PriorProbObj rdf:ID="P(Animal)">
  <prob:hasVariable><rdf:value>&ont;Animal</rdf:value></prob:hasVariable>
  <prob:hasProbValue>0.5</prob:hasProbValue>
</prob:PriorProbObj>
```

- P(A|B): e.g., P(Male|Animal) = 0.48

```
<prob:CondProbObjT rdf:ID="P(Male|Animal)">
  <prob:hasCondition><rdf:value>&ont;Animal</rdf:value></prob:hasCondition>
  <prob:hasVariable><rdf:value>&ont;Male</rdf:value></prob:hasVariable>
  <prob:hasProbValue>0.5</prob:hasProbValue>
</prob:CondProbObjT>
```

Structural Translation

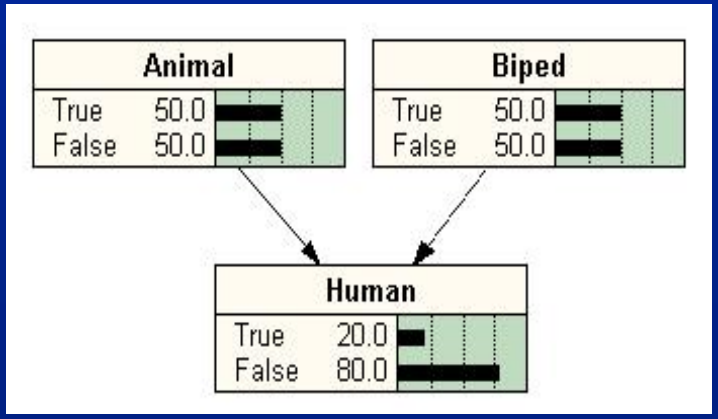
- Set theoretic approach
  - Each OWL class is considered a **set** of objects/instances
  - Each class is defined as a node in BN
  - An arc in BN goes from a superset to a subset
  - Consistent with OWL semantics

```
<owl:Class rdf:ID="Human">
  <rdfs:subclassOf rdf:resource="#Animal">
  <rdfs:subclassOf rdf:resource="#Biped">
 </owl:Class>
```

RDF Triples:

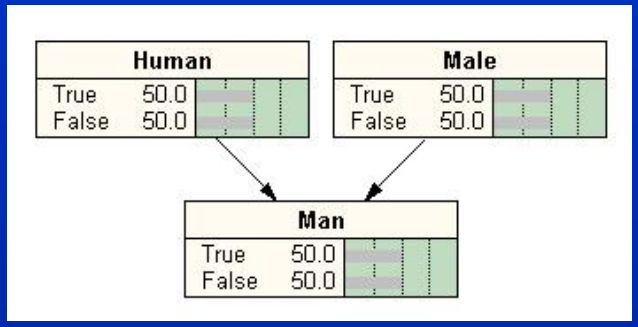
(Human rdf:type owl:Class)
(Human rdfs:subClassOf Animal)
(Human rdfs:subClassOf Biped)

Translated to BN



Structural Translation

- Logical relations
  - Some can be encoded by CPT (e.g.. Man = Human ∩ Male)



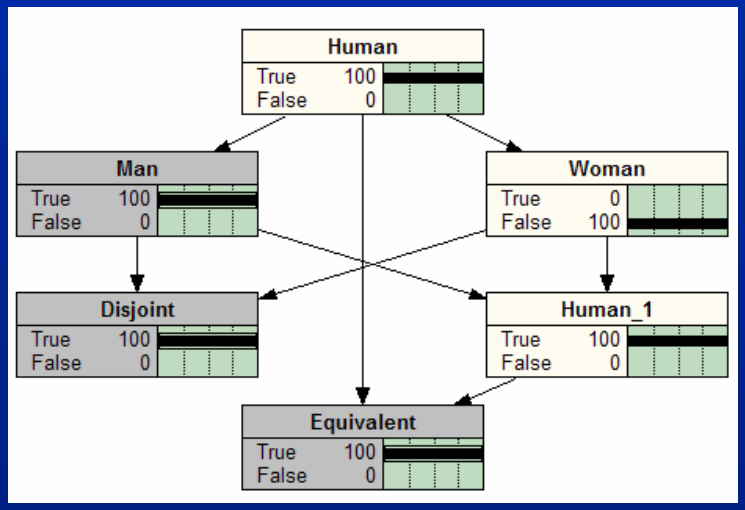
Human	Male	True	False
True	True	100.00	0.000
True	False	0.000	100.00
False	True	0.000	100.00
False	False	0.000	100.00

  - Others can be realized by adding control nodes

    Man ⊂ Human
    Woman ⊂ Human
    Human = Man ∪ Woman
    Man ∩ Woman = ∅

auxiliary node: Human_1

Control nodes: Disjoint, Equivalent

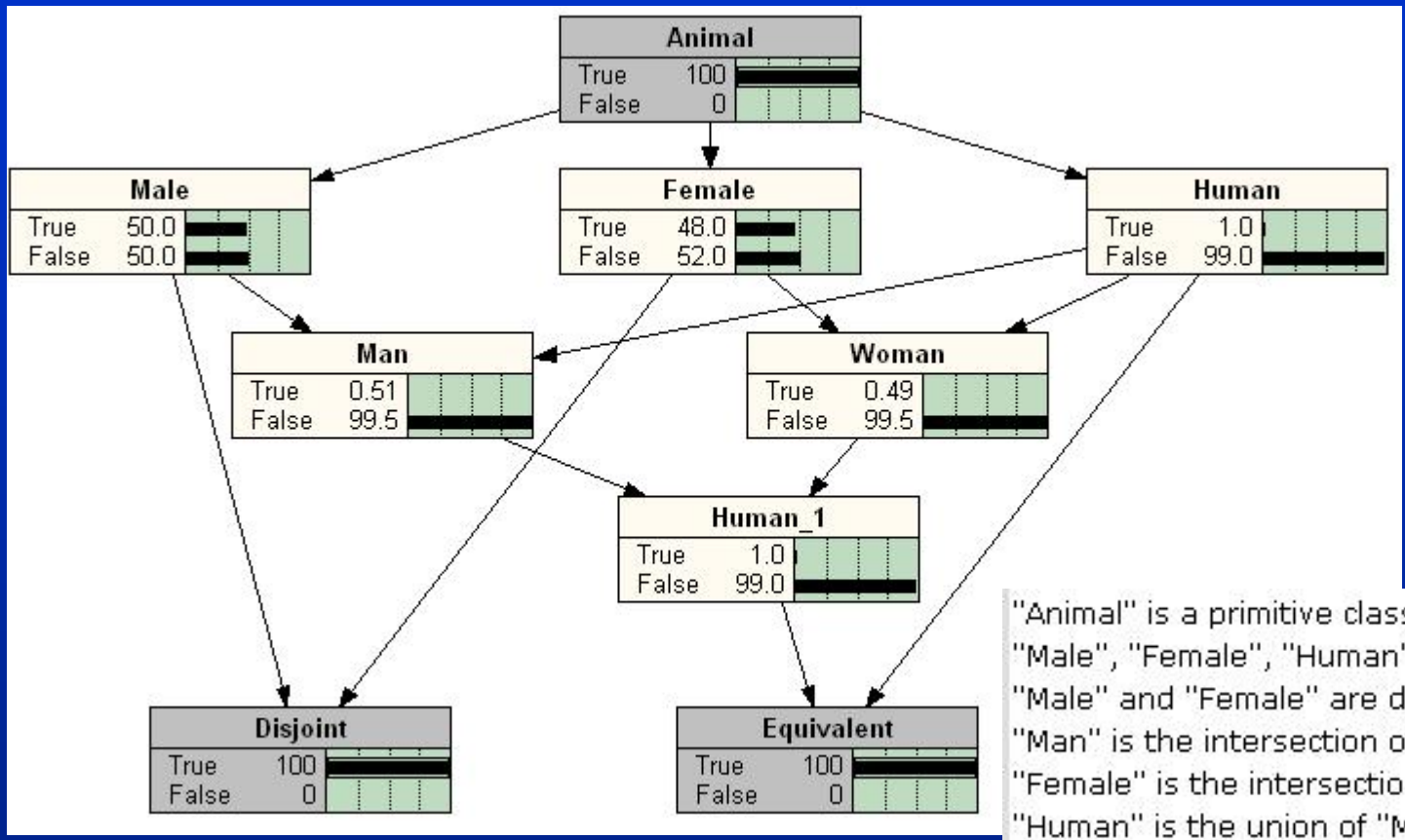


Constructing CPT

- Imported Probability information is not in the form of CPT
- Assign initial CPT to the translated structure by some default rules
- Iteratively modify CPT to fit imported probabilities while setting control nodes to *true*.
  - IPFP (Iterative Proportional Fitting Procedure)

    To find $Q(x)$ that fit $Q(E_1)$, … $Q(E_k)$ to the given $P(x)$
    - $Q_0(x) = P(x)$; then repeat $Q_i(x) = Q_{i-1}(x)\ Q(E_j)/\ Q_{i-1}(E_j)$ until converging
    - $Q_\infty(x)$ is an I-projection of $P(x)$ on $Q(E_1)$, … $Q(E_k)$ (minimizing Kullback-Leibler distance to P)
  - Modified IPFP for BN

Example



"Animal" is a primitive class
"Male", "Female", "Human" are subclasses of "Animal"
"Male" and "Female" are disjoint with each other
"Man" is the intersection of "Male" and "Human"
"Female" is the intersection of "Female" and "Human"
"Human" is the union of "Man" and "Woman"

Probability information:

P(Animal) = 0.5
P(Male|Animal) = 0.5
P(Female|Animal) = 0.48
P(Human|Animal) = 0.01

Ontology Mapping

- Formalize the notion of *mapping*
- Mapping involving multiple concepts
- Reasoning under ontology mapping
- Assumption: ontologies have been translated to BN

Formalize The Notion of Mapping

- Simplest case: Map concept E^1 in Onto^1 to E^2 in Onto^2
  - How similar between E^1 and E^2
  - How to impose belief (distribution) of E^1 to Onto^2
- Cannot do it by simple Bayesian conditioning

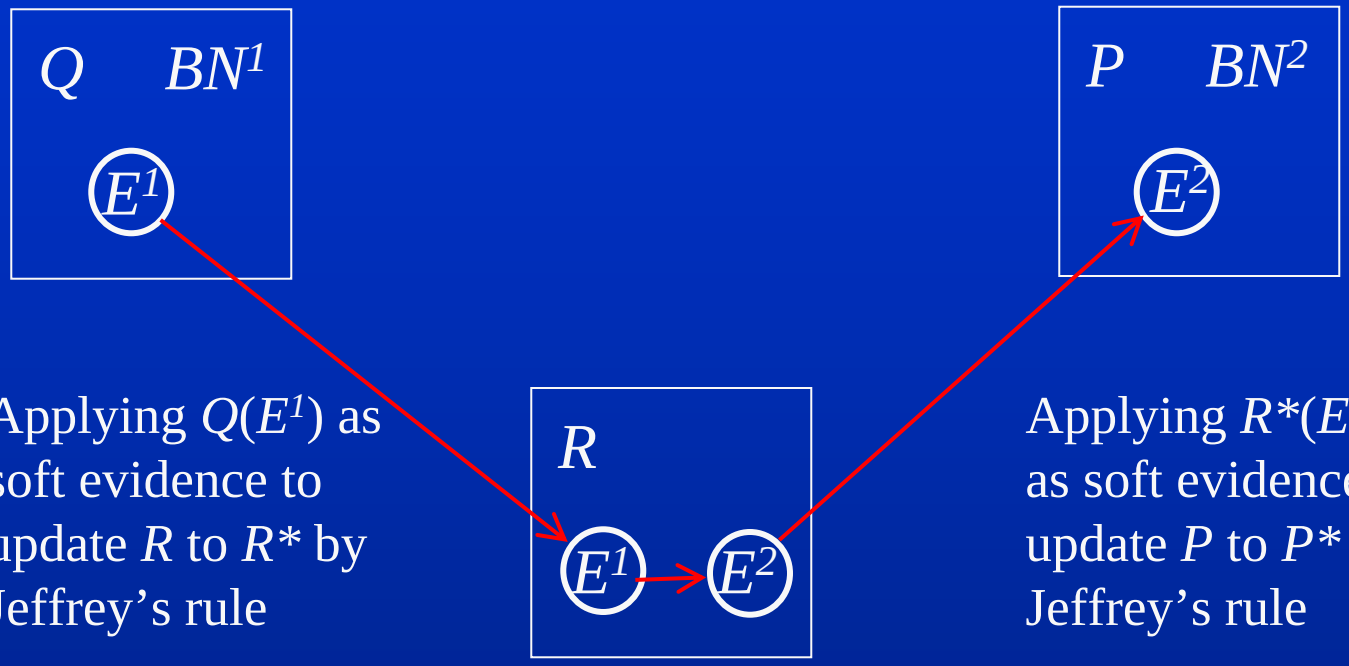
  $P(\mathrm{x}|\, E^1) = \sum_{E^2} P(\mathrm{x}|\, E^2) P(E^2 \mid E^1)$ similarity(E^1, E^2)
  - Onto^1 and Onto^2 have different probability space (Q and P)
    - $Q(E^1) \neq P(E^1)$
    - New distribution, given E^1 in Onto^1: $P^*(x) \neq \sum P(x|E^1)P(E^1)$
  - similarity(E^1, E^2) also needs to be formalized

Formalize The Notion of Mapping

- Jeffrey's rule
  - Conditioning cross prob. spaces
  - $P^*(x) = \Sigma\, P(x|E^1)Q(E^1)$
  - P^* is an I-projection of $P(x)$ on $Q(E^1)$ (minimizing Kullback-Leibler distance to P)
  - Update P to P^* by applying $Q(E^1)$ as soft evidence in BN
- similarity(E^1, E^2)
  - Represented as joint prob. $R(E^1, E^2)$ in another space R
  - Can be obtained by learning or from user
- Define

  $\text{map}(E^1, E^2) = <E^1, E^2, BN^1, BN^2, R(E^1, E^2)>$

Reasoning With map(E^1, E^2)



Applying $Q(E^1)$ as soft evidence to update R to R^* by Jeffrey's rule

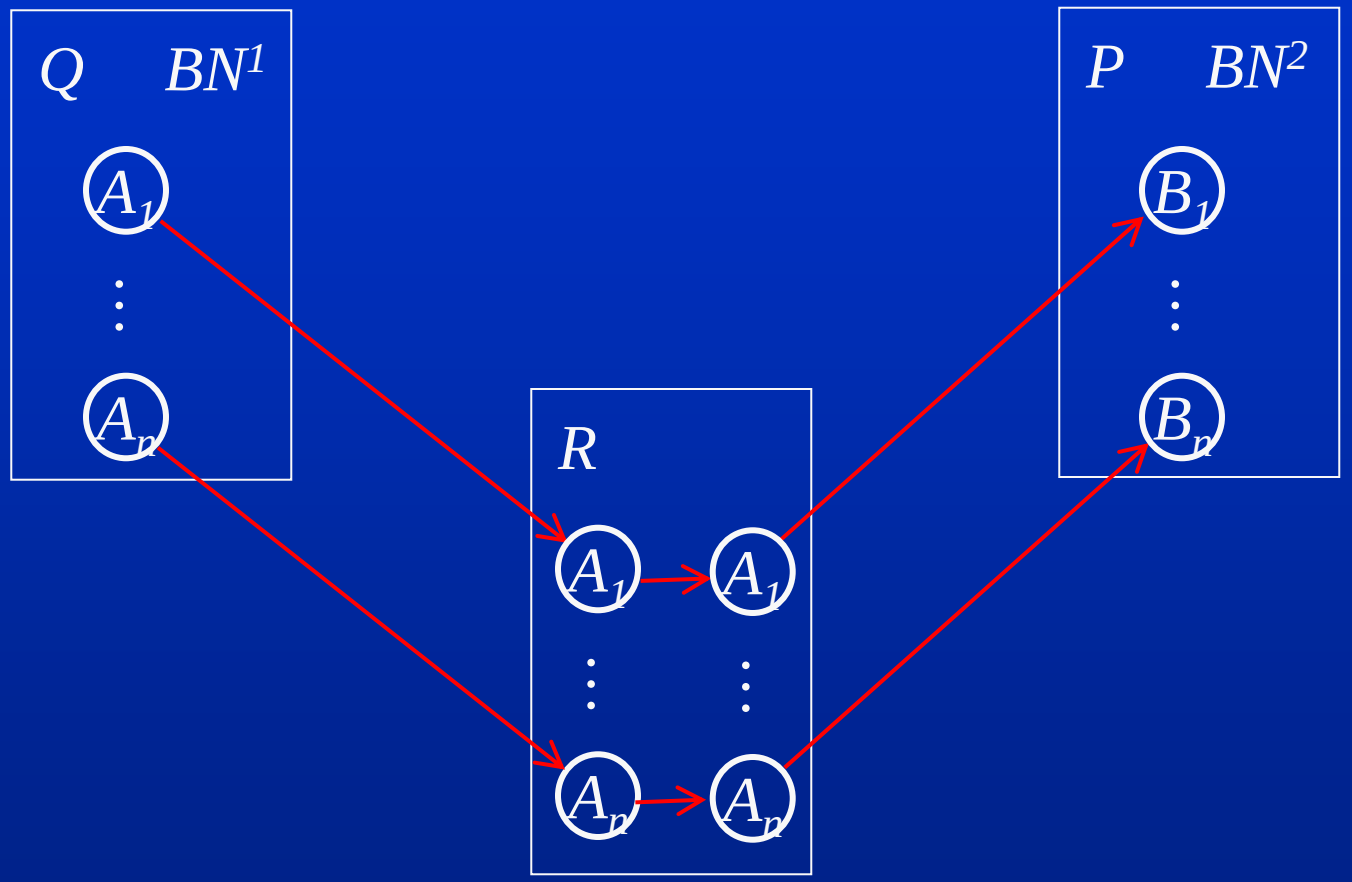
Applying $R^*(E^2)$ as soft evidence to update P to P^* by Jeffrey's rule

Using similarity(E^1, E^2):

$R^*(E^2)$

$= R^*(E^1, E^2)/R^*(E^1)$

Reasoning With Multiple map(E^1, E^2)



Multiple pair-wise mappings: map(A_k, B_k):
Realizing Jeffrey's rule by IPFP

Mapping Reduction

- Multiple mappings
  - One node in BN1 can map to all nodes in BN2
  - Most mappings with little similarity
  - Which of them can be removed without affecting the overall
- Similarity measure:
  - *Jaccard-coefficient*: $\text{sim}(E^1, E^2) = P(E^1 \cap E^2)/R(E^1 \cup E^2)$
  - A generalization of subsumption
  - Remove those mappings with very small sim value
- Question: can we further remove other mappings
  - Utilizing knowledge in BN

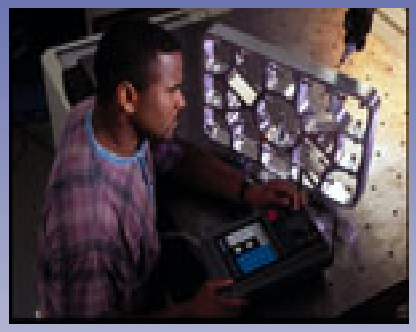
Conclusions

- Summary
  - A principled approach to uncertainty in ontology representation, reasoning and mapping
- Current focuses:
  - OWL-BN translation: properties
  - Ontology mapping: mapping reduction
- Prototyping and experiments
- Issues
  - Complexity
  - How to get these probabilities

Using Model-Theoretic Invariants for Semantic Integration

Michael Gruninger
NIST /
Institute for Systems Research
University of Maryland

Interoperability

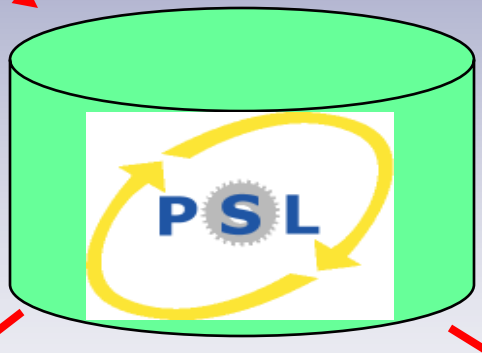


Process Modeler

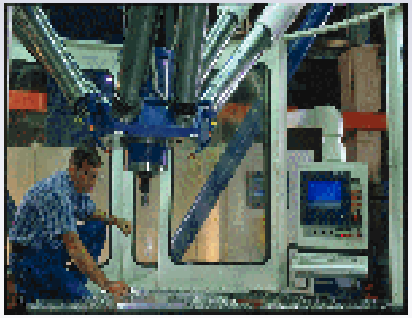
(ProCAP / KBSI)

Process Planner

(MetCAPP/Agiltech)



Simulator (Quest / Dessault)



Scheduler

(ILOG Scheduler)

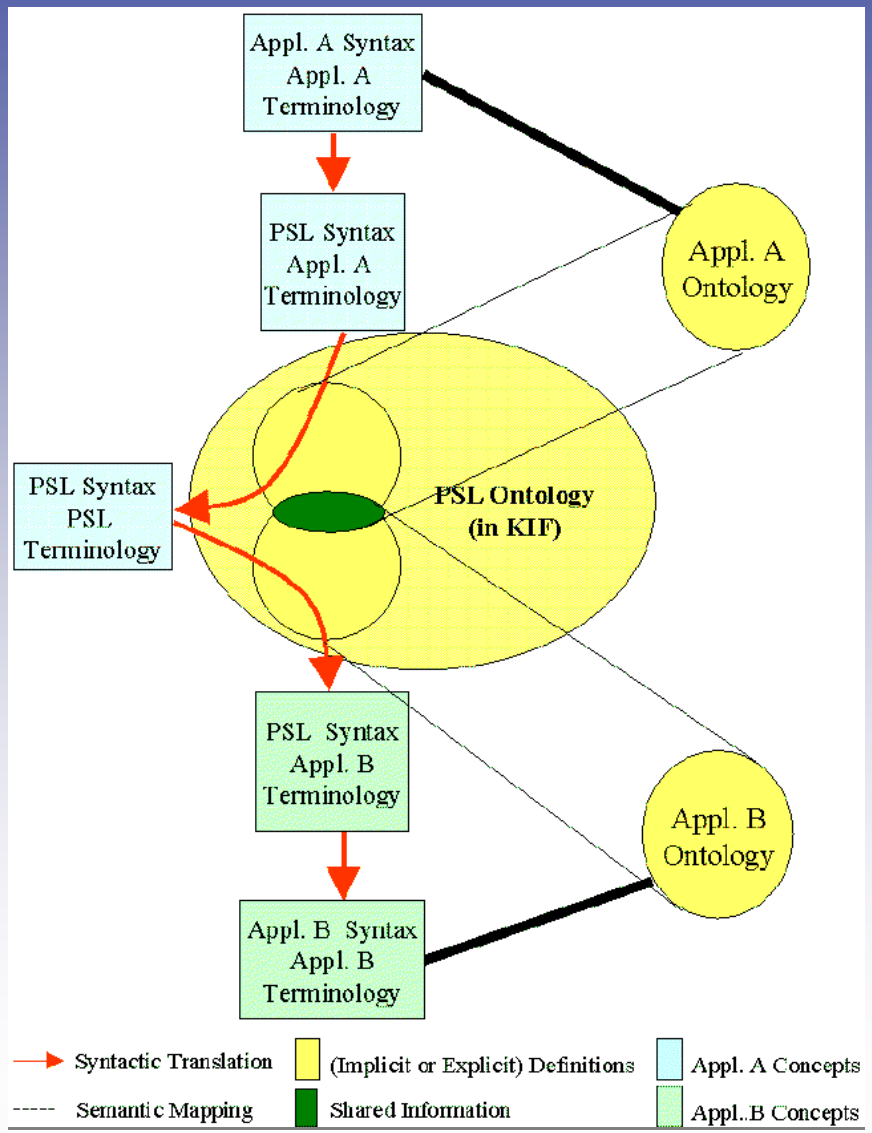
Semantic Translation

Translation definitions specify the mappings between PSL and application ontologies.

Example: The ***AtomicProcess*** in OWL-S maps to the ***activity*** concept in PSL only if the activity is atomic and its preconditions and effects depend only on the state prior to the occurrences of the activity.

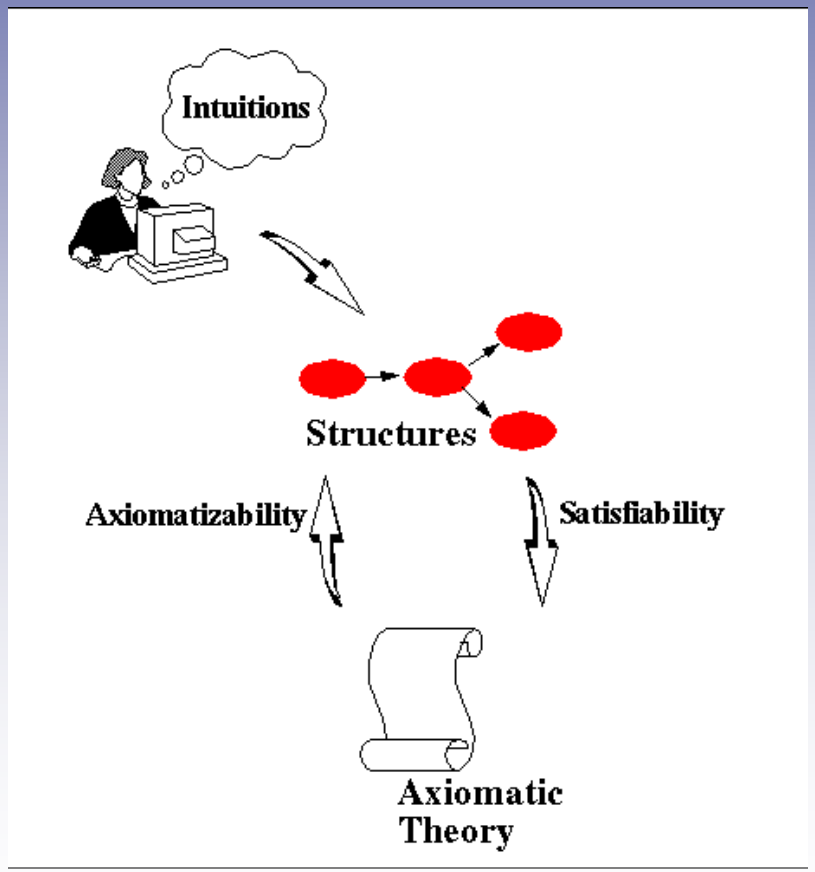
```
(forall (?a)
        (iff     (AtomicProcess ?a)
                 (and    (atomic ?a)
                         (markov_precond ?a)
                         (markov_effects ?a))))
```

Semantic Interchange Protocols



- What are sufficient conditions for an ontology to support this approach to semantic integration?

Verified Ontologies

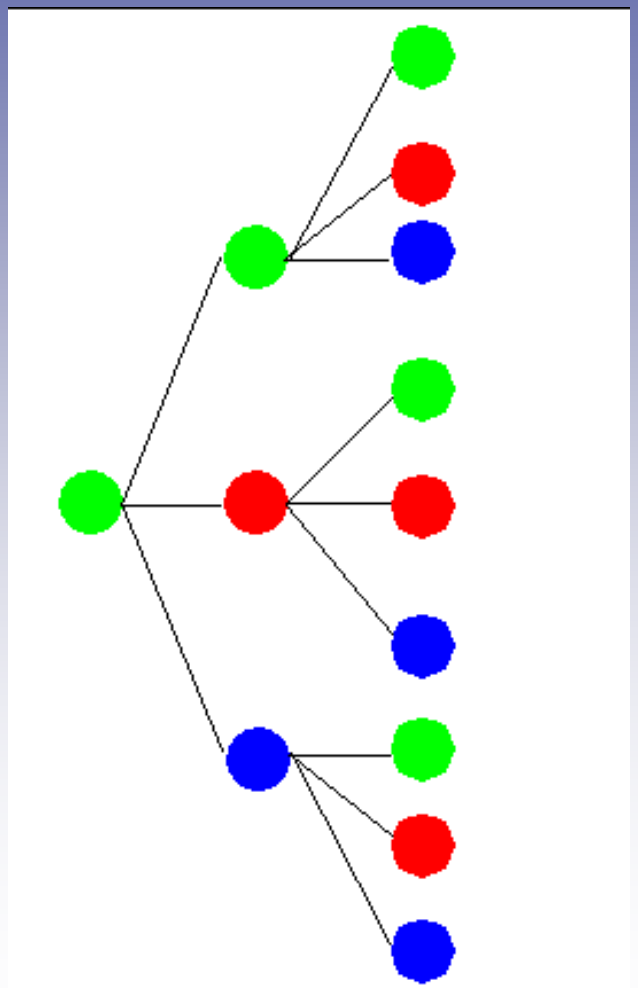


- The meaning of terms in the ontology is characterized by models for first-order logic.
- The ontology provides a first-order axiomatization of the class of models

Definitional Extensions

- Preserving semantics is equivalent to preserving models of the axioms.
  - *preserving models = isomorphism*
- We classify models by using *invariants* (properties of models that are preserved by isomorphism).
  - automorphism groups, endomorphism semigroups
- Classes of activities and objects are specified using these invariants.

Models in PSL



- Occurrence trees
- Fluents (state)
- Activity trees

Twenty Questions

How can we generate translation definitions?

- Each invariant from the classification of models corresponds to a different question.
- Any particular activity or object will have a unique value for the invariant.
- Each possible answer to a question corresponds to a different value for the invariant.

Limitations

- Not all theories have complete sets of invariants
- Invariants may not be first-order definable
- How do we determine the correctness of the translation definitions?

Interoperability Hypothesis

- We are considering interoperability among complete first-order inference engines that exchange first-order sentences.

- *Why first-order logic?*
  - Soundness and completeness guarantees that a sentence is provable from a theory if and only if it is satisfied in all models of the theory.

Ontological Stance

$\Sigma \cup \text{Ontology} \models \Phi$

Summary

- The meaning of terms in the ontology is characterized by models for first-order logic.
- The PSL Ontology has a first-order axiomatization of the class of models.
- Identify invariants of the models
- By axiomatizing these invariants, translation definitions can be shown to preserve semantics between software applications.

Further Questions?

Michael Gruninger
gruning@nist.gov
(301) 975-6536

http://www.nist.gov/psl

LOM: A Lexicon-based Ontology Mapping Tool

John Li
Teknowledge Corporation
1800 Embarcadero Road
Palo Alto, CA 94303

ABSTRACT

Ontology mapping is important to knowledge sharing and semantic integration but hard to completely automate. LOM is a semi-automatic lexicon-based ontology-mapping tool that supports a human mapping engineer with a first-cut comparison of ontological terms between the ontologies to be mapped, based on their lexical similarity. This paper will explain the algorithms used, the tests performed, and the applications developed using the results of this approach. It will also discuss the limitations of this approach as well as the future research and development issues in this field.

KEYWORDS: *ontology, mapping, lexicon, semantic web, semantic integration, alignment, interoperability*

1. INTRODUCTION

Ontology mapping is an important step to achieving knowledge sharing and semantic integration in an environment in which knowledge and information have been represented with different underlying ontologies. As more applications exploit semantic interoperability by employing an increasing number of ontologies developed by diverse communities, the demand for rapid ontology mapping is arising. Many efforts have been spent on machine-assisted ontology mapping [1]. However, this task is by nature very difficult to automate because heterogeneous ontologies may reflect fundamentally or subtly different perceptions of the domain by the creators of these ontologies. The evidence for the difficulty in producing a fully automated method for ontology mapping can be traced back to an early survey on automated database schemata alignment and to a recent one on the state of the art in ontology mapping [1, 2].

We view ontology mapping as a learning process, by human or machine, to find a morphism between the concepts of the given ontologies. Given two ontologies, A and B, a mapping from A to B is a set of pairs *(a, b)* where *a* is a concept expressed in A and *b* is its translation in B. Note that *a* and *b* can be represented in terms or expressions. Obviously the mapping is partial and not necessarily one-to-one depending on the ontologies under consideration. A good mapping tool should find the maximal number of potential mapping pairs. Naturally, if there is no overlapping of concepts in the two ontologies, there is no mapping that can be found between them.

As ontologies are logical theories that contain vocabularies and axioms for concepts, the first step in ontology mapping is to find the morphism between their vocabularies. LOM was just designed for that purpose. It is a prototype lexicon-based ontology-mapping tool developed at Teknowledge, under the Agent Semantic Communication Services (ASCS) project [3] for DARPA Agent Markup Language (DAML) Program [4]. LOM supports a human mapping engineer with a first-cut comparison of ontological terms between the ontologies to be mapped, based on their lexical similarity. We call LOM a semi-automatic method because it requires human validation at the end of the process. The output of LOM, which is a list of matched pairs of terms with scores ranking their similarity, will be reviewed by the human for the final decision. The finally approved matched vocabulary will serve as the basis for the axiom translation.

The development of LOM was based on the following two observations: (1) Human intervention in ontology mapping cannot be totally avoided but human labor can be reduced by mechanic comparisons done by intelligent software, and (2) The lexicon-based mapping is feasible because most ontologies bear lexical similarity in their vocabularies describing the same concepts when the natural languages underlying the vocabularies are the same (such as English). This linguistic connection exists naturally since most ontologies are developed by humans and are required to be understood by both humans and agents. That provides a good opportunity for our software to explore the common language base of the heterogeneous ontologies and to use syntax and semantics to identify the similarity between the terms. Like most mapping tools, LOM does not guarantee accuracy nor correctness in its suggested mappings. It saves human labor by changing their job from tedious and time-consuming search and matching tasks to much easier ones of approval and validation.

This paper is a work-in-progress report since LOM is still under development. In the next section we will present the algorithms used in LOM (Section 2). Section 3 describes the results of some tests as well as some semantic web applications using the mappings developed

by LOM, followed by a discussion on the future development of LOM and possible improvements. Section 4 briefly reviews some related work. Section 5 contains a summary.

2. ALGORITHMS

LOM uses four methods to match the vocabularies from any two ontologies. They are (1) whole term matching; (2) word constituent matching; (3) synset matching; and (4) type matching. We will explain each method in detail below. As the first step, vocabularies should be separated into lists of classes, predicates and instances, and then compared class vs. class, predicate vs. predicate, etc. However, sometimes it is desirable to compare whole vocabularies without such classification since some authors may represent similar concepts with different types of terms.

LOM takes two lists of terms from ontologies A and B and produces a list of matched pairs. Each pair contains two terms: one from the source, A, and the other from the target, B. Each term can be multi-word, such as "BiologicalParent" or "office-phone-number", etc. The matched pairs are then found through the following procedures:

(1) Whole term matching: This is the first as well as the simplest procedure to be executed. The terms in both ontologies are converted to lowercase and then compared for an exact name string match. The matched pairs are given a score of 1. Otherwise, the score is zero.

(2) Word constituent matching: This is the second procedure to be executed. Each term is broken into words wherever there is a capital letter, a hyphen or an underscore. Stop words such as "a", "the", "of", "in", etc. are dropped from multi-word terms. Remaining words for each term are morphologically processed and compared in exact string match to words of each term from the target ontology. Every matched pair has a score from 0 to 1, inclusive, representing the ratio of the number of the words matched with regard to the total number of word constituents. Then, for each term, among all its matched pairs, only the *best-fit* pairs (the highest scorers) are recorded and presented to the user. Using this procedure, unobvious matching term pairs such as "written-by" and "wrote", "meeting-place" and "place-of-meeting" can be found.

(3) Synset matching: This is the third procedure to be executed. It explores the semantic meanings of the word constituents by using the WordNet [5] synsets to help identify synonyms in matching. A synset is a WordNet term for a sense or a meaning by a group of synonyms. This procedure is similar to the method in (2) in decomposing multi-word terms into their word constituents except that it does not perform direct matching between the words. For each word in each term in each ontology, if it is in WordNet, then it must belong to one of the synsets and have at least one WordNet synset index number. The procedure associates the WordNet synset index numbers of the constituent words with the term. The two terms which have the largest number of common synsets are recorded and presented to the user. Their score is calculated and recorded in the same method as that in (2). Using this procedure, the terms "auto-care" and "car-maintenance", for example, can be matched.

(4) Type matching: This is the last procedure to be called by LOM, and it explores the ontological category of each word constituent for matching. It uses the mappings from WordNet synsets to the formal ontologies SUMO (the Suggested Upper Merged Ontology) [6, 7] and MILO (the Mid-level Ontology) [8]. SUMO and MILO together contain about six thousand ontological terms at the upper and middle level. The most popular WordNet synsets have been mapped into this set of terms [9]. LOM takes the source terms that are unmatched in the above-mentioned three procedures, collects the set of SUMO/MILO terms that their synsets map to, and then compares the SUMO term sets to their counterpart for each term in the target ontology. If there is a match, the matched terms are recorded and given a score based on the method of calculation in (2) and (3). The matched terms with the highest score for each term are recorded. Using this procedure, terms that cannot be matched by previous methods, either string comparison or sense comparison, will be matched if they represent classes or properties of the same type. For example, the terms "tank" and "armed-personal-carrier" can be matched since they are both military vehicles.

There are several caveats about the methods we mentioned above. First, the morphological processes used in procedure (2) are standard for the English language and we will not describe them here. However, if other languages are used, the morphological processes need to be replaced with rules for the other languages. Second, to do an ontology mapping from A to B, each term in source A is tested against *every* term in target B. Thus the algorithm runs in $O(nm)$ time where n and m are the length of the two input term lists respectively. During the execution, the list in B does not decrease although that in A may, as the matched ones in the source may leave the game. Third, one may think the most efficient way to execute these four procedures is to follow the sequence and let each procedure process the leftover of the previous procedure. To determine what constitute the leftover, the user needs to determine the thresholds for all methods except (1), which has only two scores: 0 and 1. If the score of a matching pair is below that threshold, the source term in the pair will be left to the next procedure to continue the process. Finally, after all methods are applied, the leftover in the source list are

unmatched. Another way of executing these methods is to filter out the matched pairs after the first procedure is executed but leave those from the second or the third procedures in the game and let them do alternative matching. To help it, LOM identifies in its output the method it uses to reach the matching together with the score of the matching. One advantage of the second way of execution is that there is no need for the artificial thresholds. Either way, each procedure does not need to repeat the process done by the previous procedure, such as breaking-down the multi-word terms, morphologically processing words, and finding synsets, etc. The second way of execution creates more opportunities for the mapping but requires more time when the ontologies are big. Fourth, one may easily find that the precision of the matching differs from procedure to procedure. Obviously the mapping through type matching can be very inaccurate since there are a limited number of ontological categories at the upper and middle level. This method is used as the last resort.

Here we have presented an algorithm for LOM and explained some of its features. The whole software is implemented in Prolog. In the next section we will report some of the tests LOM underwent and some applications it had contributed to. We will talk more about the issues and possible improvements to LOM after that.

3. TESTS AND APPLICATIONS

3.1 *Tests*

LOM has been tested extensively in-house to evaluate its functionality and performance. As this paper is written, it is participating in a competition at I^3CON (Information Interpretation and Integration Conference) [10]. In its early development stage we had run an experiment with the test data created by the SENSUS development team at the Information Science Institute (ISI) of the University of Southern California [11]. The data consists of 102 pairs of matched terms between SENSUS and CYC. LOM took the terms from both the SENSUS ontology and the CYC ontology and generated a set of mappings that were compared to the manual mappings that ISI and Cycorp created by hand. Then, metrics like precision and recall as used in the information retrieval were computed. According to our calculation, precision was 54/76 (71%) and recall was 54/94 (57%) for this experiment. Note that in this experiment we were using an early version of SUMO and an incomplete mapping from the WordNet synsets to the SUMO, so the procedures (3) and (4) did not help much in the mapping. Following that we did many test runs with the ontological terms developed by the DAML ontology community. The metrics generally improved but still varied depending on the contents and the representations of the ontologies to be mapped. On the performance measure, the time to run inputs of about 100 terms per ontology is in seconds on a 500MHz laptop. The same machine with an increased RAM size (512MB) and an increased stack size can run inputs of over one thousand terms per ontology. The ability to perform a first-cut mapping on big ontologies has been the target of our performance improvement efforts because that ability is exactly the goal of the LOM development.

3.2 *Applications*

LOM is an important component in the ASCS [4] tool set. ASCS was intended to provide semantic search and translation functions to semantic web applications. Teknowledge's DAML/OWL [12] Semantic Search Service crawls web pages, gathers semantically marked contents into a repository, and provides a search engine that allows people to query the repository and get data as the answers to their queries. Its most recent version even allows people to publish their own data into the repository via URL registration, and to register their queries and get automatic notification when the conditions for the queries are met. Obviously, with such extensive and diverse authorship, the number of ontologies underlying the data is increasing steadily. Envisioning the massive growth of diverse ontologies, the OWL designers created a set of OWL terms such as "equivalentClass", "equivalentProperty" and "sameAs" to help the authors of the ontologies to align their creation with others. Our semantic search engine not only can use these relations to seek equivalent data but also can reason with other ontological relating predicates such as "subClassOf", "subPropertyOf" and "inverseOf" to perform semantic search.

Despite all these relating predicates and the superb search capability of our search engine, the semantic search remains a problem if the authors did not actually produce the equivalence instances using these predicates. Without these instances, the data would still be isolated islands. To a search engine developer, that means a query based on one ontology will not be able to match data across the ontology boundaries although they are semantically answerable. It can be an even severer problem for the semantic-search query language designers if they have to choose a certain ontology as the base for the query language because whatever ontology the query language is based on, the answers will stop within that ontology, if these instances do not exist. We took this opportunity to test the usability of LOM. With the help of LOM, we quickly located the matching pairs from a group of ontologies and generated a big set of equivalence instances over certain domains, such as bibliography and terrorisms. In the bibliography domain,

for example, we mapped six ontologies to SUMO and generated about 300 instances of equivalent classes and properties in a very short time period. As expected, these instances greatly expanded the search range of our search engine and enabled it to answer queries with data marked in different ontologies from diverse sources. In addition, they enabled us to reduce our query interface to a much simpler one. The users do not need to remember nor specify the multiple terms and the multiple ontologies they had to use when they formed the query because there is only one ontology underlying the query language. With our recently developed Restricted English Query Interface [13], the user only needs to enter a conjunct English query (using *What, Who, When, Where* and other regular English words) and the interface will translate it into a logic form based on the SUMO and execute it. Since our repository is populated with the equivalence instances relating terms from other ontologies to those in the SUMO, our search engine will be able to gather data from multiple diverse sources using these relations.

3.3 *Discussion*

Although the experiments and applications showed that LOM made contribution to the ontology-mapping tasks, we realize that there are many places where LOM can be improved. To strengthen the word constituent matching method, LOM needs to recognize proper names, shorthand and abbreviations correctly. For example, it may need to use some fuzzy syntactic analysis method to learn that "SemWeb" is the shorthand for "SemanticWeb". As more low-level domain ontological terms are being developed and deployed, the mappings from WordNet synsets to SUMO will be updated to achieve higher accuracy. Both the synset matching method and the type matching method can benefit from the enhanced accuracy and find closer sense or type matching between the terms. As a lexicon-based ontology mapping method, LOM has its limitation in handling ontologies built with abstract symbols or codes, such as those used in chemistry, mathematics, or medicine. We plan to implement a structural mapping method that may alleviate the weaknesses of the lexicon-based approach by recognizing structural similarity between the ontologies.

4. RELATED WORK

Information integration has been a research topic for the database and KR communities for many years. With the emergence of the Internet and the advent of DAML/OWL language, semantic interoperability issues and solutions are gaining a greater audience. Among the vast number of publications related to the ontology alignment, we recently found the proposed conceptual alignment process in [14] had suggested the usage of syntactic and lexical analyses for similarity measuring, similar to what we have developed for LOM, although the development of LOM started in 2001, one year earlier than the proposal was published. Besides the difference between a proposal and an implementation, LOM has an additional method - type matching. Nevertheless, this paper provides some ideas about the integration of different methods that, as well as those in [15], might help us to explore the future development of LOM. Among the similarity learning algorithms, we found the similarity flooding algorithm [16] might be useful to the future development of LOM. Multi-strategy learning for ontology mapping was explored in [17].

5. SUMMARY

We have developed a lexicon-based ontology-mapping tool as one of many approaches in the ontology mapping research and development arena. This approach explores the lexical similarities between ontological vocabularies by using its four matching procedures: whole term matching, word constituent matching, synset matching and type matching. We have used it in some experiments and some semantic web applications in which it showed its strengths and weaknesses. As we view ontology mapping as a machine learning process, we will use this tool as the starting point to pursue multi-strategy learning of similarities between the ontologies that will take advantage of the strengths of various approaches. We expect that there will be some research and development issues ahead of us before all the desirable features can be integrated into this tool. There are vast and important applications (such as semantic integration, semantic web services) for ontology mapping in the real world. We are looking forward to continuing our research in this field and the practical deployment of our mapping tool to serve real-world users.

6. ACKNOWLEDGEMENT

The algorithms used in LOM were designed with help from my former colleagues, Ian Niles and Adam Pease. I also wanted to thank Professor Eduard Hovy for his generosity in sharing his SENSUS mapping data with us to support our early experiment. The development of LOM was partially supported by the DARPA Agent Markup Language (DAML) program.

7. REFERENCES

[1] Kalfoglou, Y., and Schorlemmer, M., "Ontology mapping: the state of the art," *The Knowledge Engineering Review*, Vol. 18:1, pp. 1-31, 2003.

[2] Sheth, A. and Larson, J., "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM computing Surveys* 22(3) pp. 183-230 1990.

[3] The DARPA Agent Markup Language (DAML) program, Internet Location: http://www.daml.org

[4] Agent Semantic Communication Services (ASCS), Internet Location: http://reliant.teknowledge.com/DAML/

[5] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. "Introduction to WordNet: An On-line Lexical Database," *International Journal of Lexicography* 3(4) pp. 235-244 1990.

[6] Niles, I., and Pease, A., "Toward a Standard Upper Ontology," *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems* (FOIS-2001), Chris Welty and Barry Smith, (Eds.). 2001.

[7] Pease, A., Niles, I., Li, J., "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications", in Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web. 2002.

[8] Niles, I., and Terry, A., "The MILO: A general-purpose, mid-level ontology," *Proceedings of the International Conference on Information and Knowledge Engineering* (IKE'04), Las Vegas, Nevada. 15-19. 2004.

[9] Niles, I. and Pease, A., "Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology," In Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE'03), Las Vegas, Nevada, June 23-26, 2003.

[10] Information Interpretation and Integration Conf. (I^3CON), August 24-26, 2004. Gaithersburg, MD., Internet Location: http://www.atl.external.lmco.com/projects/ontology/i3con.html

[11] Hovy, E.H., "Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses," In Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC). Granada, Spain. 1998.

[12] OWL Web Ontology Language, Internet Location: http://www.w3.org/TR/owl-features/

[13] The Restricted English Query Interface, Internet Location: http://ibis.teknowledge.com:8080/DAML/query.jsp

[14] Silva, N. and Rocha, J., "Merging Ontologies using a Bottom-up Lexical and Structural Approach," *Proceedings of The Seventh International Society for Knowledge Organization Conference* (7th ISKO), Granada, Spain, July 2002.

[15] Ehrig, M. and Sure, Y., "Ontology mapping – an integrated approach," Internet Location: http://www.aifb.uni-karlsruhe.de/WBS/meh/mapping/combinationTR.pdf

[16] Melnik, S., Garcia-Molina,H., Rahm, E., "Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching," *Proceedings of The 18th International Conference on Data Engineering* (ICDE'02), San Jose, California, February 26 - March 01, 2002.

[17] Doan, A., Madhavan, J., Domingos, P., Halevy, A., "Ontology Matching: A Machine Learning Approach," *Handbook on Ontologies in Information Systems*, S. Staab and R. Studer (eds.), Springer-Velag, 2004. pp. 397-416.

The Challenge

Of Knowledge Soup

John F. Sowa

26 August 2004

PerMIS 2004 Workshop at NIST
Gaithersburg, Maryland

Outline of This Talk

1. **Thesis:**

   Support interoperability among heterogeneous systems by defining all concepts precisely and unambiguously.

2. **Antithesis:**

   **"There are more things in heaven and earth, Horatio,
   Than are dreamt of in your philosophy."**

   William Shakespeare

3. **Synthesis:**

   Develop more flexible methods of knowledge acquisition by simulating the human cognitive cycle.

Aristotle's Syllogisms

System of logic based on four sentence patterns:

1. ***Universal affirmative*. Every employee is human.**
2. ***Particular affirmative*. Some employees are customers.**
3. ***Universal negative*. No employee is a competitor.**

4. *Particular negative*. Some customers are not employees.

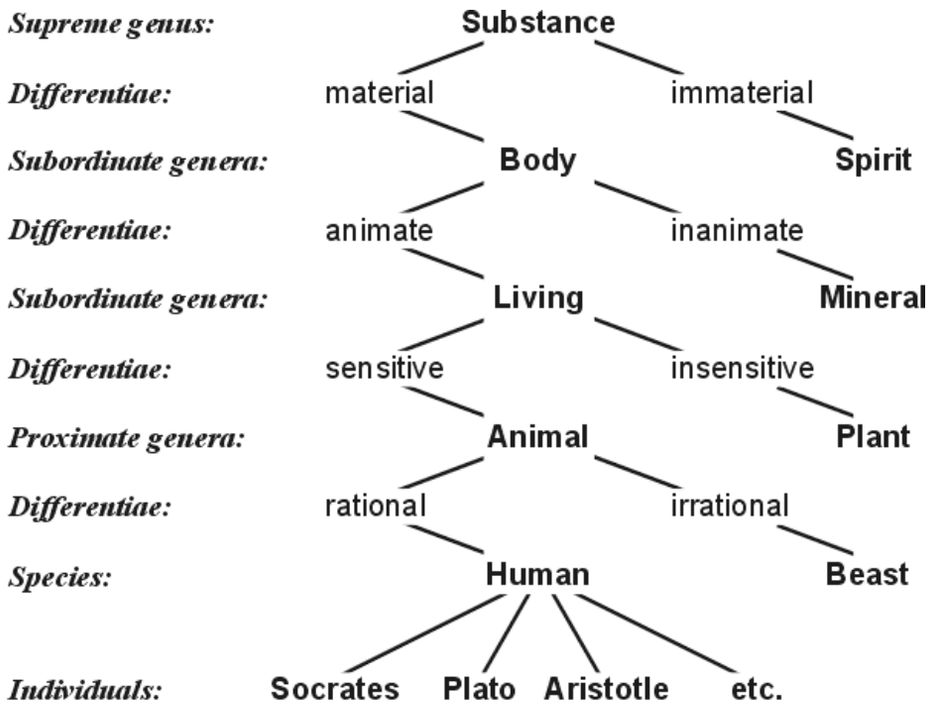
Affirmative patterns for stating inheritance.

Negative patterns for stating constraints.

Description logics are based on Aristotle's syllogisms.

Tree of Porphyry

Shows inheritance of *differentiae* from genus to species:



This diagram was translated from a version by Peter of Spain (1239).

Gottfried Wilhelm Leibniz

Encoded Aristotle's categories as integers:

- Prime numbers to encode primitive concepts,
- Products of primes for compound concepts.
- Concept X is a subtype of Y iff Y divides X.
- The result is a *lattice* with multiple inheritance.

But he never realized his grand hope:

> ***The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate, without further ado, in order to see who is right.***

Immanuel Kant

Proposed twelve categories as a replacement for Aristotle's:

Quantity	Quality	Relation	Modality
Unity	Reality	Inherence	Possibility
Plurality	Negation	Causality	Existence
Totality	Limitation	Community	Necessity

But he never realized his grand hope:

> ***If one has the original and primitive concepts, it is easy to add the derivative and subsidiary, and thus give a complete picture of the family tree of the pure understanding. Since at present, I am concerned not with the completeness of the system, but only with the principles to be followed, I leave this supplementary work for another occasion.***

Académie Française

Primary mission:

- **Défense de la langue française.**

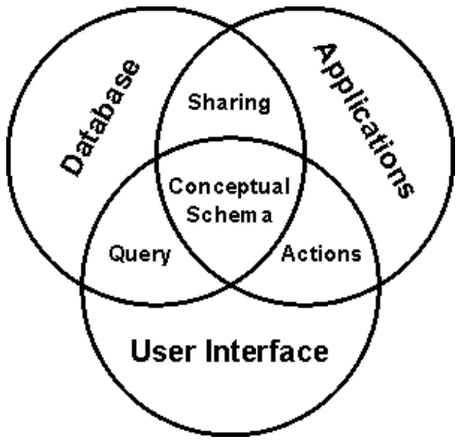
Strategy:

- **Create a dictionary that freezes the meaning of every French word.**

Result:

- **Uncontrollable growth of slang terms that never appear in the dictionary.**
- **Wholesale borrowing of new words from English.**

Conceptual Schema



ANSI SPARC, 1978.

ISO Standards Project, R.I.P. 1999.

Born again as the Semantic Web.

World's Largest Ontology Project

Cyc project started in 1984 by Doug Lenat.

- **Name comes from the stressed syllable of *encyclopedia*.**
- **Goal: implement the commonsense knowledge of an average human being.**
- **After $70 million and 700 person-years of work,**

  600,000 categories
  defined by 2,000,000 axioms
  organized in 6,000 microtheories.

Project Halo

Project for evaluating methods of knowledge representation.

Goal: Build an intelligent tutor.

Test case: Encode knowledge from a chemistry textbook in order to answer questions on a freshman chemistry exam.

Participants: Cycorp, OntoPrise, SRI International.

Results:

- **Average score: about 40% to 47% correct.**
- **Cost to encode knowledge: average about $10,000 per page from the textbook.**
- **Despite its large knowledge base, Cyc had the lowest score.**

Utterance by a 3-year-old Child

When I was a little girl, I could go "geek, geek" like that;
but now I can go "This is a chair."

Enormous logical complexity in one short passage:

- **Subordinate and coordinate clauses**
- **Tenses: Earlier time contrasted with "now"**
- **Modal auxiliaries: *can* and *could***
- **Quotations: "geek, geek" and "This is a chair"**
- **Metalanguage about her own linguistic abilities**
- **Contrast shown by *but***
- **Parallel stylistic structure**

Observations

The child has much less technical knowledge than Cyc.

But her learning ability is far more flexible and far more efficient:

Only three person-years of effort.

No need for knowledge encoding at $10,000 per page.

Can our computer systems ever be as flexible?

Limitations of Current Approaches

The technologies of the Semantic Web (XML, RDF, and OWL) are useful for many applications, but there is nothing new:

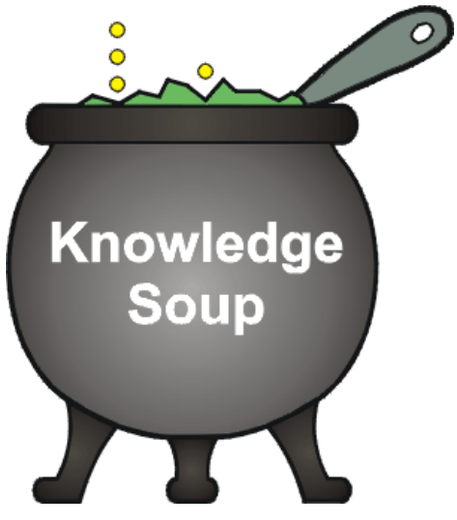
- **They're bracketed on the low end by Aristotle's syllogisms and on the high end by Cyc.**

The cost of $10,000 to encode one page from a textbook is a major barrier to widespread use.

In recent years, the Cyc knowledge base has expanded from 100,000 axioms to 2,000,000 axioms — but the cost of adding new knowledge has not gone down.

There's no evidence that an expansion from two million to two billion would make much, if any difference.

The Challenge



The fluid, loosely organized, dynamically changing contents of the human mind.

Examples of Knowledge Soup

- **Overgeneralizations: Birds fly.**

  But what about penguins? A day-old chick? A bird with a broken wing? A stuffed bird? A sleeping bird? A bird in a cage?

- **Abnormal conditions: If you have a car, you can drive from New York to Boston.**

But what if the battery is dead? Your license has expired? There is a major snowstorm?

- Incomplete definitions: An oil well is a hole drilled in the ground that produces oil.

  But what about a dry hole? A hole that has been capped? A hole that used to produce oil? Are three holes linked to a single pipe one oil well or three?

- Conflicting defaults: Quakers are pacifists, and Republicans are not.

  But what about Richard Nixon, who was both a Quaker and a Republican? Was he or was he not a pacifist?

- Unanticipated applications: The parts of the human body are described in anatomy books.

  But is hair a part of the body? Hair implants? A wig? A wig made from a person's own hair? A hair in a braid that has broken off from its root? Fingernails? Plastic fingernail extender? A skin graft? Artificial skin used for emergency patches? A band-aid? A bone implant? An artificial implant in a bone? A heart transplant? An artificial heart? An artificial leg? Teeth? Fillings in the teeth? A porcelain crown? False teeth? Braces? A corneal transplant? Contact lenses? Eyeglasses? A tattoo? Make-up? Clothes?

Devil in the Details

Most banks offer similar services with similar terminology:

- Checking, savings, loans, mortgages...

Banks interoperate on electronic funds transfer.

But when two banks merge, they never merge their databases.

Two common strategies:

- Keep running both databases indefinitely, or
- Close some or all accounts of one bank, and open new accounts in the database of the other bank.

There are too many incompletely documented details.

Limits of Definability

- Immanuel Kant:

Since the synthesis of empirical concepts is not arbitrary but based on experience, and as such can never be complete (for in experience ever new characteristics of the concept can be discovered), empirical concepts cannot be defined.

Thus only arbitrarily made concepts can be defined synthetically. Such definitions... could also be called *declarations*, since in them one declares one's thoughts or renders account of what one understands by a word. This is the case with *mathematicians*.

- **Wittgenstein's *family resemblance*:**

  Empirical concepts cannot be defined by a fixed set of necessary and sufficient conditions. Instead, they can only be taught by giving a series of examples and saying "These things and everything that resembles them are instances of the concept."

- **Waismann's *open texture*:**

  For any proposed definition of empirical concepts, new instances will arise that "obviously" belong to the category but are excluded by the definition.

Limits of Logic

Alfred North Whitehead, *Modes of Thought*:

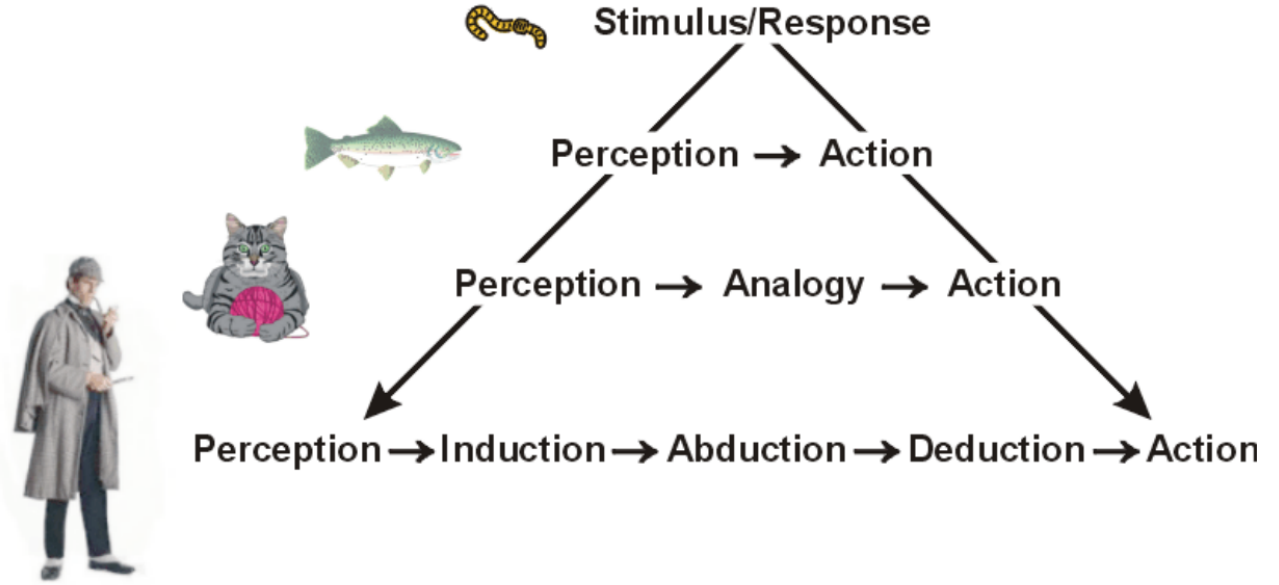
- **"Both in science and in logic, you have only to develop your argument sufficiently, and sooner or later you are bound to arrive at a contradiction, either internally within the argument, or externally in its reference to fact."**

- **"The topic of every science is an abstraction from the full concrete happenings of nature. But every abstraction neglects the influx of the factors omitted into the factors retained."**

- **"The premises are conceived in the simplicity of their individual isolation. But there can be no logical test for the possibility that deductive procedure, leading to the elaboration of compositions, may introduce into relevance considerations from which the primitive notions of the topic have been abstracted."**

Summary: "We must be systematic, but we should keep our systems open."

Evolution of Cognition



Every organism retains the capabilities of all earlier forms.

Peirce's Classification of Reasoning

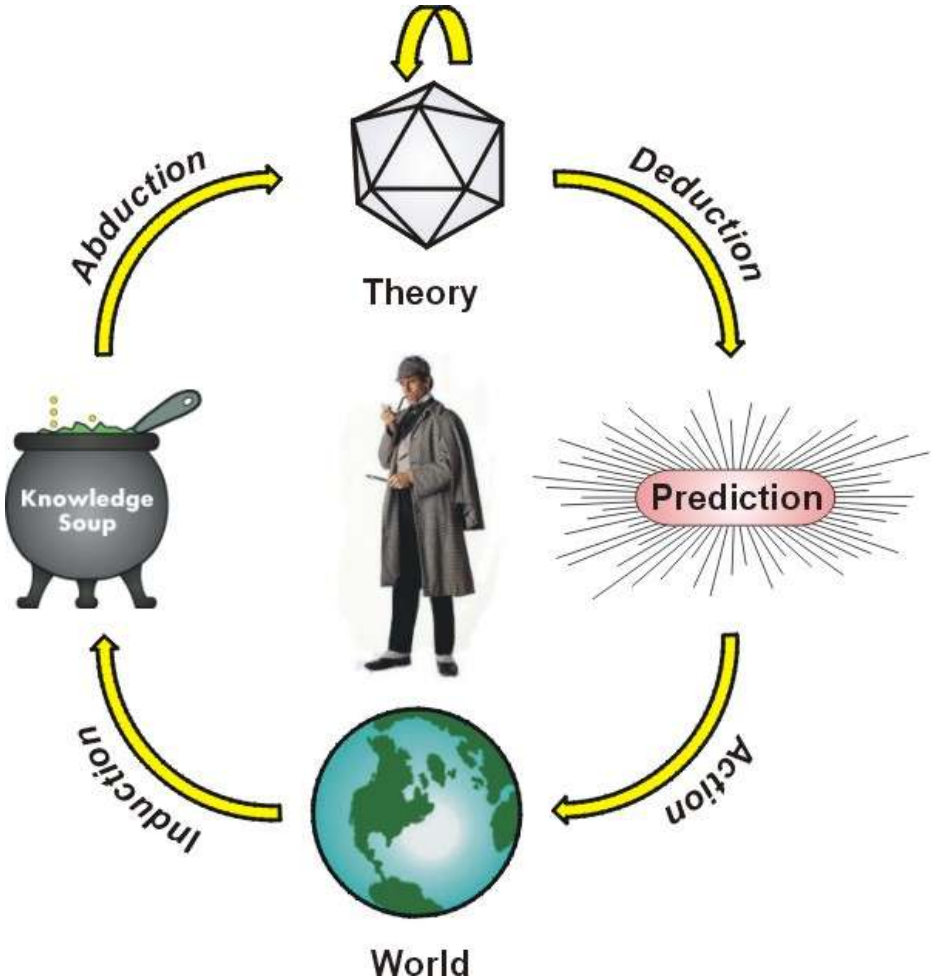
Three methods of logic plus analogy:

1. **Deduction: Deriving implications from premises.**
2. **Induction: Deriving general principles from examples.**
3. **Abduction: Forming a hypothesis that must be tested by induction and deduction.**
4. **Analogy: "Besides these three types of reasoning there is a fourth, analogy, which combines the characters of the three, yet cannot be adequately represented as composite."**

Analogy is more primitive, but more flexible than logic.

The methods of logic are disciplined ways of using analogy.

Peirce's Cycle of Cognition



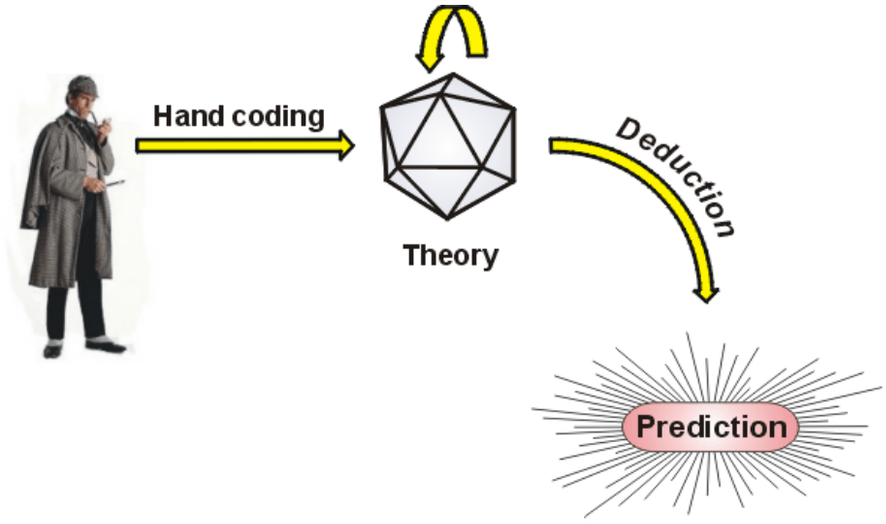
A Continuum of Reasoning Processes

Peirce's cycle characterizes reasoning processes at every level of difficulty and for time periods of any length:

- **Real-time operations, as described by Boyd's OODA loop (Observe, Orient, Decide, Act), may happen in seconds or milliseconds.**
- **Problem-solving cycles may take minutes to days.**
- **Scientific research may take months to decades.**

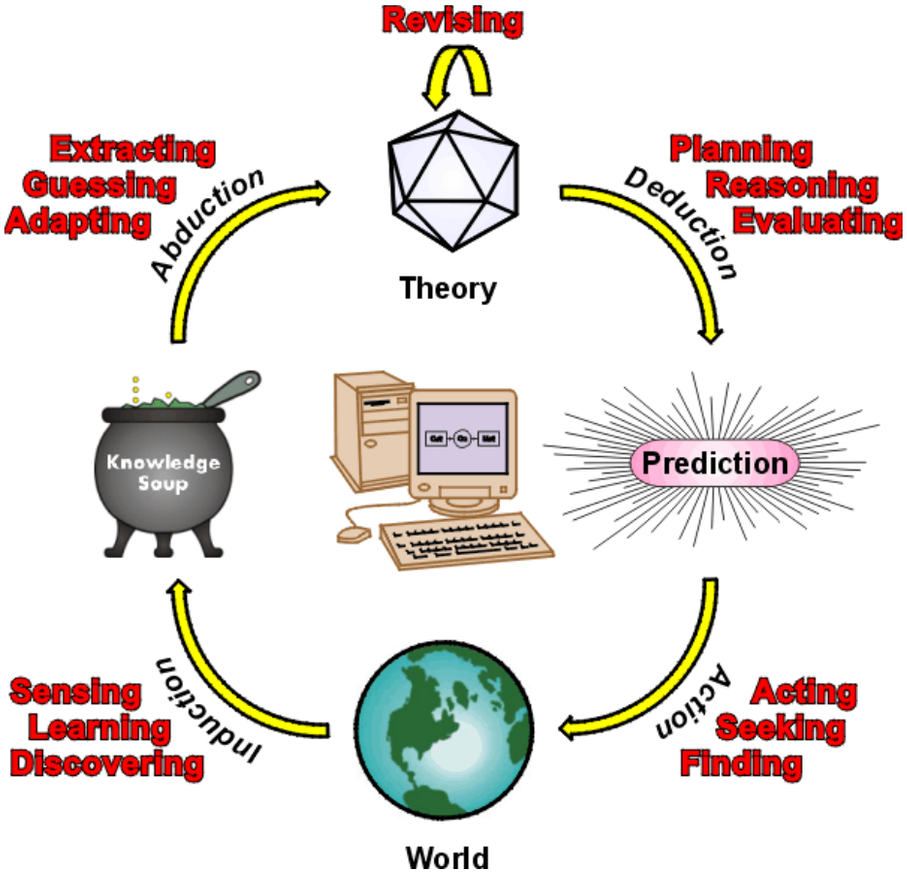
The central feature of Peirce's pragmatism is the grounding of the reasoning process in perception at one end and action at the other.

Cyc's Piece of the Pie

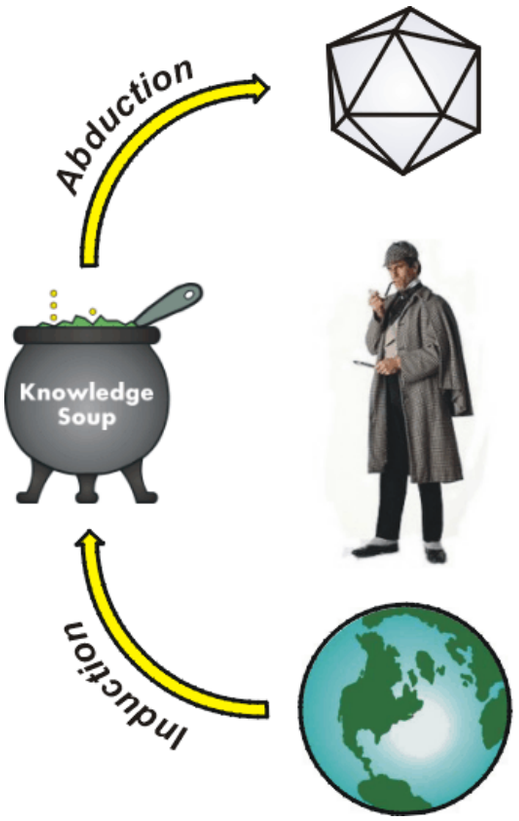


- **Cyc does not automate Sherlock Holmes.**
- **It requires people like him to write axioms.**
- **At a cost of $10,000 to encode one page from a textbook.**

Deduction is only 25% of the Cycle



The Challenge of Knowledge Soup

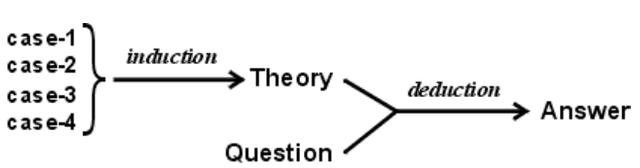


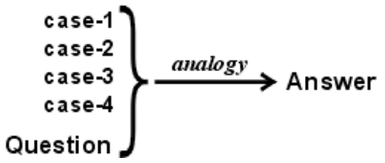
- Computer systems are better at deduction than most people.
- But the greatest challenges and opportunities are on the other side.
- How is new knowledge added to the soup?
- How is structured knowledge derived from the unstructured soup?
- How is relevant knowledge found and used when needed?
- And how can those processes be automated?

Ibn Taymiyya Contra Aristotle

- Fourteenth-century Islamic legal scholar.
- Admitted that deduction is necessary for pure mathematics.
- But for reasoning about the world, deduction is limited to the accuracy of the induction.
- Given the same data, analogy can replace induction + deduction.

Ibn Taymiyya's Argument





- **A theory can be very useful when available, as in mathematics, science, and engineering.**
- **But analogy can be used when no theory exists, as in law, medicine, business, and everyday life.**

Structure Mapping

Mapping one conceptual structure to another can have four logical effects:

1. **Equivalence: $CS_1 \equiv CS_2$**
2. **Generalization: CS_1 implies CS_2**
3. **Specialization: CS_2 implies CS_1**
4. **Similarity: Neither one implies the other.**

Analogy uses all four.

Logic uses only the first three.

The same mechanisms, both computational and neurophysiological, underlie both.

VivoMind Analogy Engine

Structure-mapping methods used in analogy:

1. ***Matching labels:***
   - **Compare type labels on conceptual graphs.**
2. ***Matching subgraphs:***
   - **Compare subgraphs independent of labels.**
3. ***Matching transformations:***
   - **Transform subgraphs.**

Methods #1 and #2 take (N log N) time.

Method #3 takes polynomial time (analogies of analogies).

Using VAE for Knowledge Fusion

Challenge:

- **A large corporation needed to analyze and re-engineer mainframe software and documentation dating back to 1962.**
- **Extract and combine knowledge from highly structured software and unstructured English.**
- **Cross-index all the sources.**
- **Detect inconsistencies.**
- **A major consulting firm estimated that it would take 40 people two years to do the analysis.**

Understanding unrestricted natural language is still an unsolved problem.

But if the structured data is processed first, the results can be used to interpret the unstructured English.

VAE was used to find and compare structures from different languages when translated to conceptual graphs.

Legacy Re-engineering Task

Compare three different languages:

- **1.5 million lines of COBOL.**
- **Several hundred JCL scripts.**
- **100 megabytes of English documentation — text files, e-mails, Lotus Notes, HTML, and transcriptions of oral communications.**

Intellitex parser used a different grammar for each language:

- **First translate the structure declarations from COBOL and JCL to conceptual graphs.**
- **VAE was used to find and interpret relevant passages from the English documentation.**
- **Internally all information was represented in CGs.**
- **Output was translated to UML diagrams and English text.**

Results

Job finished in 8 weeks by two programmers, Arun Majumdar and André LeClerc.

- **Four weeks for customization:**
  - **Design and logistics.**
  - **Additional programming for I/O formats.**
- **Three weeks to run Intellitex + VAE + extensions:**
  - **24 hours a day on a 750 MHz Pentium III.**
  - **VAE handled matches with strong evidence.**
  - **Matches with weak evidence were confirmed or corrected by Majumdar and LeClerc.**
- **One week to produce a CD-ROM with integrated views of the results:**

  Glossary, data dictionary, UML diagrams.

Conclusions

Cyc has demonstrated that deduction alone, even with large knowledge bases, does little to improve the flexibility and efficiency of knowledge acquisition.

The OWL ontologies are much smaller than Cyc, and even if they were as large as Cyc, the OWL reasoners are not as powerful as Cyc.

New breakthroughs will have to come from improvements in reasoning methods other than deduction: induction, abduction, and analogy.

VAE demonstrates that a high-speed analogy engine has great potential.

But much more R & D needs to be done.

Related Readings

For further analysis of the knowledge soup, see Chapter 6 of

Sowa, John F. (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co., Pacific Grove, CA.

A discussion of knowledge soup and its relationship to formal theories:

Crystallizing Theories out of Knowledge Soup
http://www.jfsowa.com/pubs/crystal.htm

A description of the VivoMind Analogy Engine and the Intellitex parser, co-authored with Arun Majumdar:

Analogical Reasoning
http://www.jfsowa.com/pubs/analog.htm

The relationship of ontology to logic, metadata, metalanguages, and semiotics:

Ontology, Metadata, and Semiotics
http://www.jfsowa.com/ontology/ontometa.htm

Philosophical issues about the effect of knowledge soup on the development and application of ontologies:

Signs, Processes, and Language Games: Foundations for Ontology
http://www.jfsowa.com/pubs/signproc.htm

Model-theoretic foundation of logics with multiple metalevels and nested contexts:

Laws, facts, and contexts: Foundations for multimodal reasoning
http://www.jfsowa.com/pubs/laws.htm

Graphic and language interfaces to intelligent systems:

Graphics and Languages for the Flexible Modular Framework
http://www.jfsowa.com/pubs/gal4fmf.htm

A description of how the VivoMind Analogy Engine was used to support legacy re-engineering:

LeClerc, André, & Arun Majumdar (2002) "Legacy revaluation and the making of LegacyWorks," *Distributed Enterprise Architecture* 5:9, Cutter Consortium, Arlington, MA.

Evaluating Intelligence in Unmanned Ground Vehicle Teams

Sesh Commuri[1], Yushan Li[1], Dean Hougen[2], Rafael Fierro[3]

[1]School of Electrical and Computer Engineering
University of Oklahoma, Norman, OK
[2]School of Computer Science
University of Oklahoma, Norman, OK
[3]School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK

ABSTRACT

Evaluation of intelligence in Teams of Unmanned Ground Vehicles (UGVs) requires the development of consistent metrics and benchmarks. This is a complicated process as the implementation of the UGVs is problem and domain specific. Different performance requirements give rise to different set of metrics making the comparison of performance between two implementations difficult. In this paper, we focus on three aspects of intelligence, namely reconfiguration, adaptation and learning, and communications in UGV teams and investigate the development of metrics for measuring their performance. We also investigate the available benchmarks for intelligent systems and verify their suitability for measuring the performance of UGV teams. A hierarchical architecture called Adaptation and Learning at All levels (AL^2) for the UGV teams is presented. This architecture is designed to allow for a modular and hierarchical approach to implement deliberative and reactive behaviors in teams of autonomous vehicles. In this implementation, system intelligence is incorporated at all levels of the hierarchy. The performance of the proposed architecture is evaluated using the metrics identified.

Keywords: *Performance Metrics, Intelligent Systems, UGVs, Adaptation and Learning.*

1. INTRODUCTION

The high cost associated with the acquisition and deployment of mobile robots motivates the development of low cost multi-robot teams that can function cooperatively to achieve specified goals. As the performance requirements get more stringent and the application realm becomes more diverse, embedding intelligence in system becomes a critical part of the realization of the Unmanned Ground Vehicles (UGV) Teams.

The notion of "intelligence" and the requirements for such intelligent systems has been discussed in detail in [1]. For a control system, at a very minimum level, system intelligence implies the ability to sense the environment, to make the control decisions based on the task requirements and to take the necessary corrective actions. At a higher level, system intelligence may include the ability to recognize objects and events, to represent the knowledge in a world model and to plan for the future. Intelligence at its highest level provides the ability to perceive and understand, to predict outcomes based on actions, to choose wisely between actions, and to maximize the chances of success under variety of circumstances. In general, intelligence embodies the ability to learn from experience and adapt successfully to the environment. For successful implementation, however, "intelligence" has to be formalized and the required metrics for its measurement developed.

The challenges and issues in defining performance metrics for intelligent systems are discussed in [2, 3]. The analysis of the system architecture and configuration is proposed to develop a measure of "Machine Intelligent Quotient (MIQ)" in [4]. In [5], requirement specifications and system verification are used to develop a formal method to specify the performance metrics. Techniques to assign metrics to intelligent systems are also explored in [6, 7].

Designing intelligent systems is a complex task requiring the integration of a diverse set of hardware and software components. Intelligence can be formally defined as "the ability of a system to behave appropriately in an uncertain environment" where "appropriate behavior maximizes the likelihood of the system's success in achieving its goals [1]". Such an intelligent system should be able to respond to sensory feedback at every level such that goals are achieved despite perturbations and unexpected feedback. Since intelligence responds to sensory feedback at all levels, overall effectiveness requires such 'system intelligence' to be distributed in nature. Therefore, any measure of the "intelligence" must account for the "intelligence" at each level of the system. Typical components that are to be looked at are:

a. Sensors and actuators
b. Knowledge representation and world model
c. Planning and control
d. Learning and adaptation

In addition to the above, metrics are required to measure the level of system autonomy. Some of the measures

proposed in literature measure the sensitivity of the implementation to environment and the learning algorithms that are implemented [8]. For a system comprising a single robot, some possible metrics [9] are (a) the ability to choose strategies or algorithms; (b) the ability to generate reactive and deliberative behaviors; (c) effectiveness in accomplishing goals and objectives; (d) efficiency of operations.

While the above metrics are adequate to describe high level system performance, they do not give insight into the functioning of UGV teams. In the case of UGV teams the mission complexity and uncertainty in the environment impose more stringent requirements on the "system intelligence". In this case, the design must address additional issues such as:

1) Dynamic reconfiguration of the UGV teams to meet mission requirements. This situation is typically encountered when a new team has to be formed, or when a team has to be augmented with additional resources. Dynamic modification of teams also occurs during formation control of UGVs.
2) Coordination and Cooperation between team members, and
3) Distributed real-time communications between team members and other teams.

These requirements have strong impact on the implementation of every level in the hierarchy of the system. Thus, in addition to the metrics for the overall system performance, the performance analysis requires the definition of metrics for each level of the hierarchy in the implementation.

In this paper, a hierarchical architecture called Adaptation and Learning at All Levels (AL^2), that allows system intelligence to be incorporated at all levels of the hierarchy is proposed. This architecture is modular, scalable and flexible. Specific requirements on the UGV teams are listed and their impact on the entities in each layer is discussed. The development of metrics is then discussed based on this implementation framework. The rest of this paper is organized as follows: Section 2 presents the AL^2 architecture for teams of UGVs. The metrics needed to evaluate the performance of the team are discussed in Section 3. The implementation of the proposed architecture for the development of intelligent teams of unmanned ground vehicles is then presented in Section 4.

2. AL^2 ARCHITECTURE

In this section, architecture is proposed that enables the design of complex hierarchical systems using simple components whose performance can be rigorously analyzed. This architecture, called Adaptation and Learning at all Levels (AL^2), allows for intelligence to be implemented at all levels of the hierarchy and for adaptation and learning occurring at different granularities throughout the system.

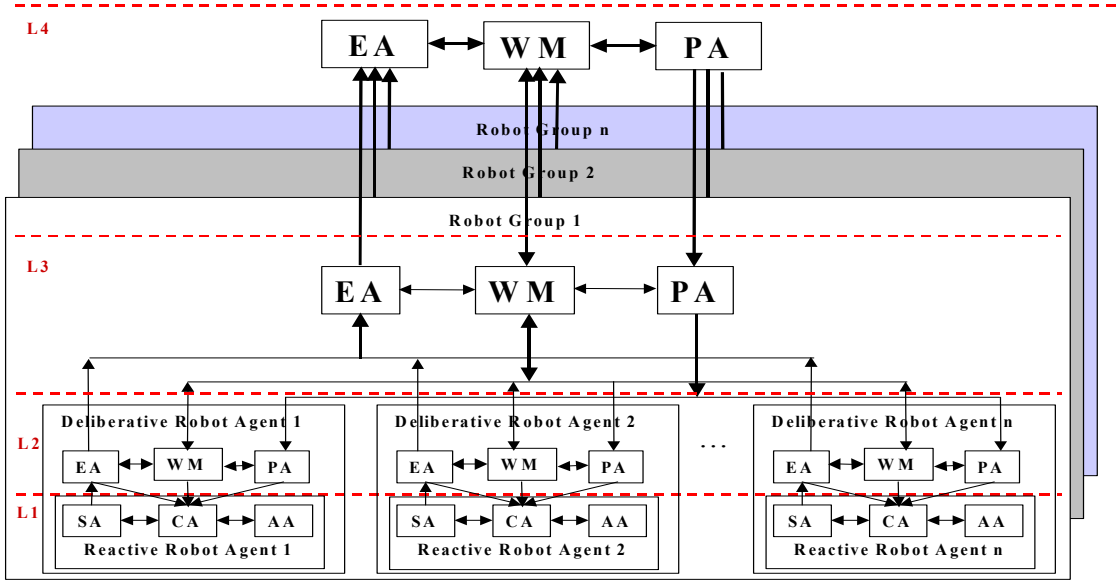


Fig. 1 Architecture for Adaptation and Learning at All Levels (AL^2)

The proposed architecture (AL^2) envisions one or more robotic agents working as a group. At the lowest level (L1), each robot agent has a control agent (CA), an actuator agent (AA), and a sensor agent (SA). The control agent is responsible for attaining the commanded system performance at the lowest level. It can command the sensor agent to override its output values, recalibrate its signal, as well as perform rudimentary signal processing like filtering. The AA, CA, and SA have the lowest level of autonomy. This level (L1) is characterized by stringent real-time requirements and deterministic behavior. At a very fundamental level, this design is adequate for a robotic agent to function and perform repetitive tasks in a structured environment. Note that, because of our distributed communication infrastructure, the sensor, actuator, and control sources (and the corresponding sub-agents) for a single robot agent need not be present on the same physical platform. For example, a platform lacking a camera and image processing capabilities could still perform leader-following if the leader platform had a rear-facing or omnidirectional camera (or other sensor) that could be used to sense the relative position of the follower platform.

In order to meet the requirements of fault tolerance, uncertainty in the system model and the environment, we propose a distributed architecture wherein the higher layer (L2) incorporates elements that instill higher-level intelligence in the robot. In this layer, the sensory signals from Layer L1 are processed by the Estimator Agent (EA). The output of the Estimator is then used to modify/update the local representation of the World Model (WM) and as input to the Control Agent. The distributed intelligence paradigm that is proposed means that EA can now include algorithms for fault detection, dynamic sensor reconfiguration, and sensor fusion at the level of a deliberative robot agent. The WM entity in the robot agent maintains information about the environment that is necessary for the successful tasking of the robot. Typically, this would include local map information, friend/foe classification, targets and obstacles etc. The Planning Agent (PA) utilizes the information from the local model of the world (WM) and the high-level task requirements to generate a plan that is communicated to the control agent in layer L1. The PA implements algorithms for path planning, obstacle avoidance, optimization, etc., for an individual robot. Level L2 is characterized by increased autonomy and less stringent real-time requirements.

A team of robots consists of a number of individual robot agents possibly with differing sensor/actuator suites and capabilities. The coordination between these agents is managed by the PA entity at the level of the robot group (L3). Information sharing between L2 entities is controlled by the entities in L3. This increases the security of the implementation because the L2 entities can function independently of each other, while still functioning in a coordinated manner. The primary function of the entities in Layer 3 is to coordinate the working of the robot agents in the group. L3 handles all reassignments of tasks between different robot agents in L2. Introduction of new robot agents or sensor suites, etc., are the exclusive domain of L3. The outputs of all the EAs in layer L2 provide the input to the EA module in L3. Team-level sensor fusion amongst the different robotic agents is accomplished by the EA at L3. This EA module is used to update the world model (WM) in Layer 3. This WM also manages the information sharing among the robot agents in L2. The planning agent (PA) in this layer does the task decomposition from the mission requirements and updates the individual PAs in L2. It is to be noted that the architecture specified is independent of hardware and software implementations and individual elements in L2.

Layer 4 (L4) manages the coordination between groups of robot agents. The highest level of intelligence and autonomy and the lowest level of real-time criticality characterize L4. Dynamic reassignment of the responsibilities of each group is handled by L4.

The proposed architecture will enable the development of groups of unmanned ground vehicles that can be dynamically configured and retasked. The architecture is flexible and is not dependent on the type of controllers or algorithms implemented in any given layer.

3. DEVELOPMENT OF METRICS FOR SYSTEM INTELLIGENCE

In this section, the metrics required to evaluate the performance of the UGV teams with respect to reconfiguration, cooperation and the real-time communication between team entities are addressed.

3.1 Dynamic Reconfiguration of Hardware and Software

To meet the operational requirements for different tasks, the system should have the intelligence to implement different software and hardware configurations dynamically at the every level of the hierarchy. Plug-and-play sensors and actuators require relevant signal processing elements and software drivers to be loaded and the data made available seamlessly to the application. Fault handling on the other hand might necessitate the routing of signals dynamically through a different part of the system in order to bypass a faulted element. The capability of the system to handle these requirements can be evaluated by the following set of metrics.

a. Is the system reconfigurable?

The rigidity of the implementation can be assessed by checking the amount of reconfigurable entities in the implementation. A typical implementation may consist of fixed hardware and software components and some

modules whose functionality can be modified at run-time either by the user or by other processes. The amount of reconfigurable resources as a ratio of the overall system resources is a measure of the reconfigurability of the system.

b. Is the system reconfiguration static or dynamic?

Static reconfiguration requires the system to be taken off line and reconfigured before it is deployed. On the other hand, dynamic reconfiguration can take place while the system is under operation. Dynamic reconfiguration is essential when it is not feasible to take the system off-line to implement changes.

c. Can the system be fully / partially reconfigured?

Full and partial reconfigurations are important aspects of the design of intelligent systems. Systems typically need to execute special tests at startup to verify proper system functioning. Once the startup tests are complete, the system can transition to the "run-time" mode. While it is easy to load test software to run system tests at startup, the tests that can be run are constrained by the hardware. By incorporating the ability to change the configuration of the hardware, for example by using FPGA devices, the same hardware can be used for system tests at startup and then "fully" reconfigured for run-time operations. The ability to "fully" reconfigure the hardware is also essential to the retasking of the individual robot. When the robot is retasked, sensor and actuator configurations can be selected that adapt the robot for the specified task. Since the embedded hardware can be optimized for the specific task, the overall performance can be improved without an increase in system cost.

Often, it is required to re-route the signals to accommodate for faults or add additional circuitry to handle signals from new sensors that come on-line. In such circumstances, unused portions of the FPGA can be configured to handle this requirement while the rest of the device is unaffected. Such reconfiguration, called Partial Reconfiguration, is essential to support retasking of individual robots, plug-and-play transducers, and for fault accommodation.

3.2 Coordination and Cooperation Between Team Members

Traditional control theory enables the design of controllers in a single mode of operation in which the task and the controlled plant are fixed. Contrary to this, in the case of UGV teams, team members usually interact with each other and with uncertain or unstructured environments. The team is to reach a goal destination, negotiate around obstacles and satisfy constraints on the formation. Thus, any measure of performance of the UGV teams has to address the ability of the teams to coordinate and cooperate with another in order to successfully accomplish the overall mission objective. The effectiveness of the implementation of UGV teams for coordination and cooperation among team entities can be measured by the following set of metrics.

a. Is the UGV team capable of executing simple formations?

The ability of the UGV team to execute simple formations like leader-follower, straight line, and convoy are a good indication of the coordinated activity among the members of the UGV team.

b. Is it possible for individual members in the team to seamlessly share sensory information, world models, and data?

The performance of the UGV team can be significantly improved if the sensors could be calibrated using reference data gathered by an external entity. The ability to use sensory information from other team members also extends the capability of a team in the face of sensory failures. Successful implementation of intelligent UGV teams requires the ability for each member of the team to benefit from the knowledge gained by other team members. Thus by sharing the world models and the knowledge, UGVs can demonstrate behaviors that are not programmed. Similarly, sharing of performance data between members is critical to the efficient operation of the UGV team.

c. Can an UGV team be dynamically modified by the addition or removal of a team member?

Operational damage to an UGV or changing mission requirements often requires augmentation of an UGV team with additional resources. The ability to recognize the availability of additional resources and retask each of the UGVs in the team is a measure of the dynamic adaptability of the UGV team. This characteristic is also important in cases where an UGV team moving in a formation has to navigate around a dynamic obstacle. In this case, the team has to split into two sub-teams, maintain sub-formations while avoiding the obstacle and then rejoin in the original formation.

3.3 Distributed Real-Time Communications

Real-time communications are essential to support other functions within the UGV teams. The communication scheme has to be flexible and allow for communication of differing transmission rates, media, and security. The ability of the system to dynamically select channels of communication to improve performance and reduce power consumption is critical to the performance of the team. The following characteristics can be used as a measure of the performance of the communications scheme.

a. Is there a mechanism for the intra-layer and inter-layer communications?

Teams of UGVs have to communicate and coordinate at several levels [11]. Therefore in each layer, modular implementation of the entities with appropriate communication interfaces is crucial for the successful coordination between UGVs. The communication mechanism will have to provide visibility into each entity at every level of the implementation. For example, the ability to share sensory data, world models, or plans between different UGVs in a team is essential to the implementation of intelligent UGVs. Intra-layer and inter-layer communication is essential for the intelligence of an UGV while communication between different UGVs and teams of UGVs is crucial for the implementation of intelligent teams of UGVs.

b. Is the communication scheme used flexible?

The higher layers in the implementation of an UGV are typically characterized by abstract entities where intelligent decisions are made. The lower layers on the other hand, are characterized by real-time control modules where traditional closed loop control decisions are taken. The varied nature of communication at each level in the implementation of an UGV implies that the communication scheme employed must be flexible enough to enable the dissemination of high-level abstract information as well as the low-level real-time data.

b. What are the communication mechanisms and the guaranteed performance?

Successful operation of the UGV teams requires a number of communication techniques. Commonly implemented ones are the ability to provide query-and-response mechanism, broadcast and periodic transmission of data between different entities in the implementation. Key properties such as bandwidth, transmission rates, protocol overhead, transmission error rates, error correcting methods etc. are to be analyzed to ensure that the communications do not become a bottleneck in the performance of the overall system.

4. CASE STUDY

The proposed framework is tested by implementing the L1 layer of the proposed architecture on the Xilinx Virtex-II Pro platform [13]. This platform was selected based on its capability to implement reconfigurable architectures, and the excellent development tools and product support. The Virtex-II Pro XC2VP4 has a PowerPC core, 6768 logic cells, 504 KBits BRAM, 4 3.125 Gbps RocketIO transceivers, and 3.01 Mbits configuration space.

The Xilinx Virtex-II Pro device is a user programmable gate array with embedded PowerPC processor and embedded high-speed serial transceivers. The Xilinx Virtex architecture is coarse grained and consists of a number of basic cells called configurable logic blocks (CLBs). These logic blocks are arranged in rows and columns, with each CLB consisting of four logic cells arranged in two slices. Each CLB also contains logic that implements a four-input look up tables (LUTs). Each slice contains two function generators, two storage elements, arithmetic logic gates, large multiplexers, wide function capability, fast-carry look ahead chains, and horizontal cascade chains. The function generators are configurable as four input look up tables (LUTs), sixteen bit shift registers, or as sixteen bit selective RAM memory. Each CLB also has fast interconnect and connects to a generalized routing matrix (GRM) to access general routing resources. The Virtex-II Pro has SelectIO-Ultra blocks (IOBs) that provide the interface between the package pins and the internal configurable logic. Active Interconnect Technology connects all these components together. The overall interconnection is hierarchical and is designed to support high speed designs.

The programmable elements in the Virtex-II Pro, including the routing resources, are controlled by values stored in the static memory cells. The device is configured by loading the bitstream into the internal configuration memory. These values can be reloaded to change the functions of the programmable elements. The Xilinx Virtex family of FPGAs supports both partial as well as dynamic reconfiguration. Partial reconfiguration can be achieved in one of the two ways, namely Module-based partial reconfiguration and difference-based reconfiguration. In the module-based reconfiguration, the entire module can be reconfigured. The height of the reconfigurable module is the height of the device and the module can cover one or more columns. In difference-based reconfiguration, the reconfiguration is done by making a small change in the design, and then generating a bit-stream based only on the differences in the two designs. Switching the configuration from one implementation to another is easy and very quick.

The system is designed with PPC405 processor core, SDRAM controller connected to Processor Local Bus (PLB) and general purpose Inputs-Output (GPIO) devices like Leds, Push buttons, UART and dip switches are connected to its On-chip Peripheral Bus (OPB). These are the components available on

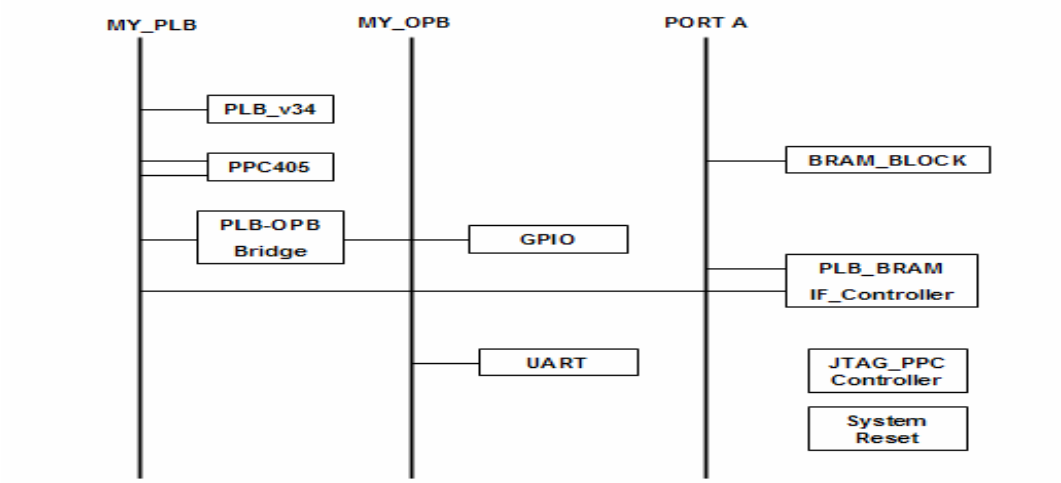


Fig. 2. Implementation of PWM motor control with dynamic reconfiguration for fault accommodation.

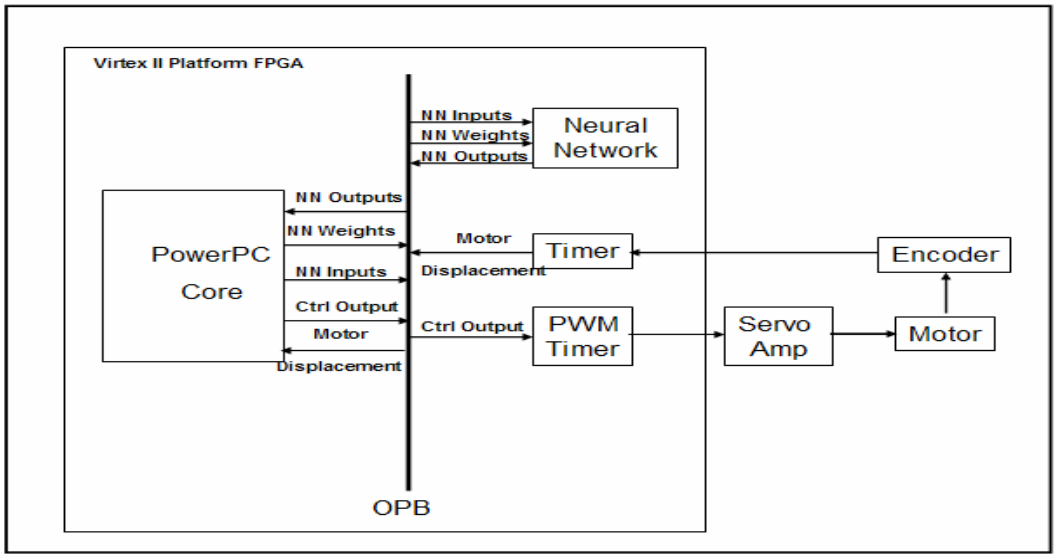


Fig. 3. Neural Network based compensation of actuator nonlinearities.

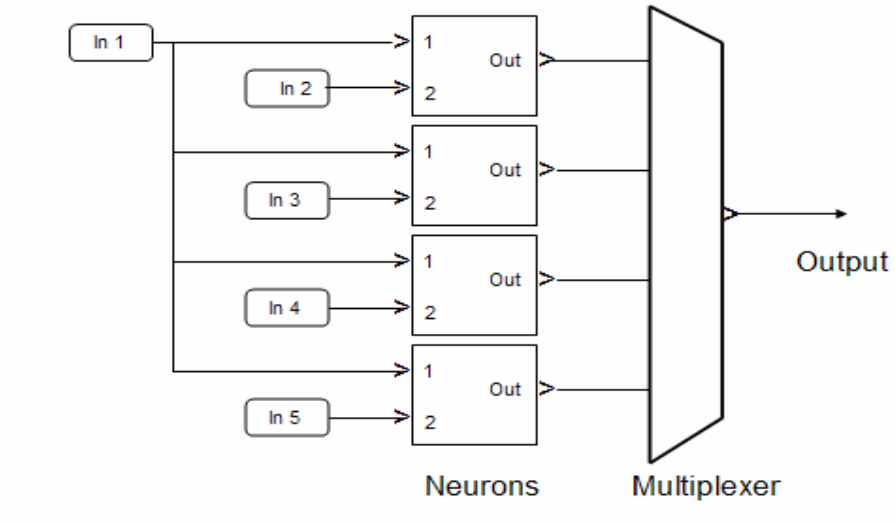


Fig. 4. Implementation of a one-layer Neural Network in Simulink using Xilinx Toolset.

the board, so the first step is to verify the proper functionality of all these components. To do this, the processor boots up with a configuration file to test all the components. On successful completion of built in self test, the processor fetches the second configuration file to configure itself and the board, switching into operational mode. If subsequent reconfiguration of the system requires additional components, then the configuration files can be loaded dynamically into the FPGA. For example, if serial communication capability is required, then a preconfigured UART module can be loaded into the FPGA and the relevant software drivers activated. This can be achieved during the operation of the system with configuration times of the order of micro-seconds.

In the second design example, a PWM generator is implemented in the hardware to control the drive motors of the robot (Fig. 2). Timer 1 (pwmTimer) is configured to generate the PWM signal while Timer 2 (opbTimer) is configured to sense the feedback signal. If a fault is detected in the "sense" circuit during operation, then a different sensing circuit is dynamically created and the signals routed through it. The control cycle in this example was executed in real time with a sampling rate of 20 milli-secs. The time for reconfiguration was of the order of a few micro-seconds showing that dynamic fault accommodation was achieved in real time.

In the third design example, a Neural Network (NN) is implemented in the FPGA to compensate for actuator deadband (Fig.3). The controller continuously monitors the output and dynamically instantiates the neural network in the FPGA when the performance degrades significantly due to load dependent deadband in the actuator dynamics. The Neural Network is modeled and designed in Simulink using the Xilinx toolset provided by Mathworks Inc. (Fig. 4). Once the design has been successfully validated, a configuration bit stream can be generated that allows for dynamic creation of the NN module in the hardware.

5. CONCLUSIONS

In this paper, architecture was presented that facilitates the implementation of teams of intelligent UGVs. Three aspects of intelligence, namely reconfiguration, adaptation and learning, and communications in UGV teams were investigated and metrics for measuring their performance were proposed. The performance of the proposed architecture was evaluated using the metrics identified.

REFERENCES

[1]. J. S. Albus. Outline for Theory of Intelligence. IEEE Trans. System, man and Cybernetics, Vol. 21, No.3, pp473-509, 1991.

[2] J.M. Evans, E.R. Messina, Performance Metrics for Intelligent Systems. . Proc. of Performance Metrics for Intelligent Systems Workshop. 2000 PerMIS Workshop, 2000.

[3] L.A. Zadeh. The Search for Metrics of Intelligence- A Critical View. Proc. of Performance Metrics for Intelligent Systems Workshop. 2000 PerMIS Workshop, 2000.

[4] E.C. Chalfant, S. Lee. Measuring the Intelligence of Robotic Systems: An Engineering Perspective. Proceedings of International Symposium on Intelligent Systems. Gaithersburg, MD. October, 1999.

[5]. Y.Zhang, A.K. Mackworth. Formal Speification of Performance Metrics for Intelligent Systems. Proc. of Performance Metrics for Intelligent Systems Workshop. 2000 PerMIS Workshop, 2000

[6]. C. Schlenoff, L. Welsch, R. Madhavan, N. Zimmerman, Towards Measuring the Performance of Architectural Components of Autonomous Vehicular Systems. 2002 PerMIS Workshop August 13-15, 2002.

[7]. J. S. Albus. Features of Intelligence Required by Unmanned Ground Vehicles. Proc. of Performance Metrics for Intelligent Systems Workshop. 2000 PerMIS Workshop, 2000.

[8] A.Yavnai. Metrics for System Autonomy, Part I: Metrics Definition. Proc. of Performance Metrics for Intelligent Systems Workshop. 2000 PerMIS Workshop, 2000

[9] R. Finkelstein. A Method for Evaluating the 'IQ' of Intelligent Systems. Proc. of Performance Metrics for Intelligent Systems Workshop. 2000 PerMIS Workshop, 2000.

[10] Sesh Commuri, Yushan Li, Dean Hougen, Rafael Fierro. Designing for System Intelligence. Proceedings of the 5th IFAC Symposium on the Intelligent Autonomous Vehicles, Lisbon, Portugal, July, 2004.

[11] R.Muthuraman, A. Fajebe, S.Commuri. Intelligence in Embedded Controls-A Case Study. Proceedings of the IEEE Region 5 Technical Conference, April 2004.

[12] S. Donti, and R. L. Haggard, "A survey of dynamically reconfigurable FPGA devices, " Proc. IEEE., vol. 8, pp. 422-426, 2003.

[13] Xilinx, Inc., Virtex-II Pro: Platform FPGA Handbook, UG012, v 2.0., 2002.

Metrics for Intelligent Autonomy

S. O'Day
Eagan, McAllister Associates
47332 Eagan, McAllister Lane
P.O. Box 986
Lexington Park, MD 20653

M. Steinberg, C. Yglesias
Office of Naval Research

K. Bonnevier, T. Moulds
NAVAIR
Patuxent River Naval Air Station, MD

R. Sincavage, M. Ricard, M. Cleary
Draper Laboratory
555 Technology Square
Cambridge, MA 02139

M. Curry, T. Sliski, J. Tierno
Alphatech, Incorporated
6 New England Executive Park
Burlington, MA 01803

J. Clark
Lockheed Martin
Fort Worth, TX

F. Snyder
Northrop Grumman Electronic Systems
Pittsburgh PA

ABSTRACT

Intelligent Autonomy (IA) is a multi-year program within the Office of Naval Research (ONR) Autonomous Operations (AO) Future Naval Capabilities (FNC) program. The primary goal of the effort is to develop and demonstrate technologies for highly automated and fully autonomous mission planning and dynamic re-tasking of multiple classes of Naval unmanned systems and minimization of human intervention in unmanned vehicle operations. This technology is being applied to both individual and teams of unmanned air, surface, ground, and undersea vehicles for a variety of mission areas including reconnaissance/search, persistent surveillance, tracking, and some limited application to strike. Autonomy technologies will be matured through a series of phased demonstrations to allow low risk transition to current and future Navy and Marine Corps systems. Demonstrations will be done using both real vehicles and simulation. Some of the major simulation demonstrations will be done within the context of a simulated warfare environment at the Naval Air Systems Command based around the Air Combat Environment Test & Evaluation Facility (ACETEF) and the Unmanned System Research and Development Lab (USRDL). The demonstrations at NAVAIR will utilize much of the architecture and many of the assets from the NCW4.0X Virtual Laboratory (V-LAB) project. Metrics for testing of IA software in this environment are currently being developed. This paper will discuss some candidate performance metrics that are currently being considered for evaluation of the Intelligent Autonomy technologies.

KEYWORDS: *unmanned aerial vehicle, unmanned undersea vehicle, mission planning, autonomous control, JIMM*

1. INTRODUCTION

The Intelligent Autonomy Effort under the Autonomous Operations Future Naval Capability is developing a range of technologies that will be challenging to effectively evaluate using traditional metrics. The goals of the Intelligent Autonomy (IA) effort IA are to

•Provide autonomy software for highly automated and fully autonomous dynamic retasking of multiple classes of Naval unmanned systems to perform littoral reconnaissance, search, persistent surveillance, and to a limited extent tracking and strike.
•Minimize human intervention via autonomous and highly automated mission planning/replanning functions and operator aids such as alert management and plan understanding for individual vehicles and teams of vehicles.
•Enable limited automated surveillance and reconnaissance data processing for surface and shoreline object detection and classification to provide autonomous replanning based on sensed information, bandwidth reduction, and operator workload reduction.

The technologies demonstrated under the IA product lines will be applicable to multiple types of Naval unmanned vehicles including unmanned air, undersea, ground, and surface vehicles with a focus on air and undersea vehicles and control stations. This effort will leverage numerous DOD programs in autonomy to support specific Navy and Marine Corps unique and essential needs. Intelligent Autonomy technologies will be demonstrated through a series of phased demonstrations to allow low risk transition to current and future Navy and Marine Corps systems. The primary areas being developed and demonstrated under the IA program are:

UxV High-Level Planning/Replanning
Lead Performers: Alphatech, Draper Laboratory
Allocate mission tasks to available platform/payload types out of a team of 5-10 heterogeneous vehicles and determine an optimal sequence of mission tasks with associated time windows/constraints based on high-level tasking (platform availability, team mission tasks, priorities, and constraints).

UAV Dynamic Replanning
Lead Performer: Lockheed Martin, Ft. Worth
Produce UAV mission plan that optimizes survivability & employment of on-board capabilities while meeting an ordered set of mission objectives and constraints.

UUV On-Board Dynamic Mission Replanning
Lead Performer: Draper Laboratory
Generate minimum-cost, energy efficient, safe routes to achieve combination of mission tasks within constraints.

Alert Management & Replan Assessment
Lead Performer: Lockheed Martin, Ft. Worth
Replan Assessment to analyze mission plan changes, monitor/assess contingencies, assess contingencies, and trigger a replan or alert if necessary. Alert Management to determine the level and type of alerts received, store alerts received and forward them to be displayed, and support the operator in recovering the context of tasks interrupted.

Mixed-Initiative Interface Manager
Lead Performers: Lockheed Martin ATL, Charles River Analytics, and Aptima
Display relevant mission information, provide plan-understanding capabilities, and enable the operator task the vehicles and set the level of autonomy.

Distributed Cooperative Control
Lead Performer: Alphatech
Enable autonomous mission replanning among teams of vehicles with limited communications

Maritime Image Understanding
Lead Performer: Northrop Grumman Electronic Systems
Develop video processing technology in the river and harbor domain. Detect and classify mission relevant objects to support autonomous navigation and surveillance.

2. PROJECT DESCRIPTION AND SCHEDULE

Initially, the developers will demonstrate the functionality and capability of their products using medium to high-fidelity simulation models at their facilities. Later, software algorithms are planned to be integrated with Naval control stations or vehicles to enable testing and maturation of the software products. The algorithms will be demonstrated in both simulation and hardware/in-water demos.

NAVAIR will provide a test bed for development using the Air Combat Environment Test & Evaluation Facility (ACETEF) and the Unmanned System Research and Development Lab (USRDL). A specific Joint Integrated Mission Model ACETEF (JIMMACE) port scenario database will serve as the warfare environment for mission planning system development and the Tactical Control Station (TCS) will serve as a baseline for operator station development.

3. WARFARE ENVIRONMENT

JIMMACE may cede control of specific player tactics and system functions to external IA hardware or software assets that interface to JIMMACE shared memory. Object instances

are defined in the JIMMACE scenario database (SDB). The extent of asset control is defined via statements in the control database (CDB). The types of objects (players, platforms and systems) in the environment, the tactics that these objects execute as well as the command, communications and control architecture are being defined in the JIMMACE type database.

The JIMMACE model will provide the warfare environment for integrated IA demonstrations. IA assets access and control parts of the JIMMACE warfare environment via shared memory network interfaces. DIS, HLA and UDP datagram socket connection protocols are used to transmit the data between IA assets and shared memory. The JIMMACE model creates and fills information into shared memory based on databases written in its native language (which consists of English language phrases which are combined in a straightforward syntax).

4. OPERATOR ENVIRONMENT

The IA program is focused on operator functions related to mission management only. It is assumed that either the vehicles are highly autonomous or that there are additional operators or operator functions concerned with vehicle management issues, such as shipboard recovery and air traffic management issues for UAV's. The system operators will interface with the IA hardware and software assets through the TCS interface and the new IA operator interface modules. Serving as a network interface between the operator interface (TCS and the new IA OI's) and the warfare environment will be the Unmanned Simulation System (USS) stimulator. The USS will provide a way for the evaluators to insert real-time scenario modifications such as vehicle or sensor cautions, warning, and emergencies. This will be used as part of the evaluation metrics in testing the interaction between the IA technologies and the operator(s) in relation to SA and workload.

The baseline operator environment will be a 1-3 person team that may consist a mission commander, vehicle operator, and/or a sensor operator. The number of operators depends on the type and number of unmanned vehicles and the complexity of the mission tasks. A secondary baseline derived from the JUCAS concepts will be a five-person team that will dynamically split vehicle control aspects with sensor control and C4I aspects.

The USS will interface with JIMMACE via a HLA interface. The USS is a CORBA based architecture and supports custom interfaces via UDP and TCP.

5. CANDIDATE METRICS

This section will describe some of the major candidate metrics that are being considered for use on the IA effort. There are a variety of useful measurements that can characterize the engineering quality of unmanned vehicle simulation, intelligent autonomy, and operator control station software. These can be roughly categorized as:

- response of system components to a range of initial input parameters
- human factors of operator mission management and situational awareness
- response of system components to changes in the simulated warfare environment during execution

Metrics relating the unmanned vehicle simulation system performance to mission goals include:

- optimization of the number and mixture of unmanned vehicles to maximize the number of successful missions
- impact of mission re-planning time on mission success
- optimum distance between assets and targets for maximal mission success
- impact of reactive/creative maneuvers on mission success
- loss of assets in mission completion/objectives completed
- operator/vehicle ratio

The response of the system to different initial conditions can be measured in terms of time and impact on mission success. Top-level parameters, which can be varied, include:

- geographic size of gaming area and placement of assets
- weather, terrain and other environmental factors
- number of missions
- number of each type of unmanned vehicle
- mission re-planning cycle time
- extent of UV intelligent tactics
- types of missions within a particular scenario

Each proposed system has a number of engineering metrics which also relate to mission flexibility and success.

5.1 *Mission Software Component Metrics*

The IA project accepts certain scenario dependent mission coverage metrics to evaluate the IA mission-planning components. The time it takes to turn around a mission plan or re-plan is an obvious metric. This can be dependent on the number of constraints, number of total missions underway and on a variety of warfare environment parameters. The goal is then to maximize the number of simultaneous missions that can be executed within the context of the following:

- Number of simultaneous mission tasks that the system can handle
- Number of mission task types that the system can handle
- Planning exception rate (dropped tasks over total tasks)
- Fraction of mission constraints not met (if feasible)

In the area of dynamic performance metrics, it is hoped that the time elapsed between task appearance and completion is

minimized within the context of the following scoring ratios $S(t)$ where *t* is the time from the beginning of the scenario:

- Optimization of mission-specific cost-measures
- Discounted optimization of mission-specific cost-measures to emphasize timeliness

Stability/sensitivity metrics are used to avoid frequent changes in plans that may have a detrimental effect on mission performance and operator situation awareness. One such measure is that of "thrashing" in tasking where one takes the time elapsed between execution of one task and the last change of the preceding task. This can also be examined as a function of communication bandwidths, error rates, and scenario variation.

5.2 *Operator Station Metrics*

Since the IA program is focused on operator functions related to mission management only, there are some differences from the types of metrics traditionally used for operators directly controlling the vehicle. Of particular concern is the neglect-tolerance of the system. This concerns both how well the autonomous system behaves when there is limited human intervention and how well the human operator is able to maintain situation awareness when not constantly in the loop or when managing multiple vehicles and mission tasks. The impact of human performance on the overall system can appear at several levels:

- system interoperability level (with external assets)
- software system level (e.g. efficiency and accuracy of UV mission prosecution by this system)
- operator station component level

System level measures can be used to identify the decision-making roles in which the human is most influential and effective relative to the capabilities of the automation. Measures should be made under varying levels and types of human intervention for factors such as

- Speed and accuracy for decisions and actions
- Time to respond to critical events
- Duration of mission activities
- Ratio for completion of "Mission-Critical Objectives" vs. "Secondary Objectives"

Task loading metrics are critical for estimating the required number of operators required for a mission. This can be drawn out from the speed and accuracy of task completion for different levels of task demands associated with the mission (e.g., the number and rate of required tasks for successful mission completion; complexity of the mission, etc.). Objective measures can also be used to identify the points at which the operator begins either shedding tasks or failing to achieve accurate task completion.

Another important type of metric is subjective workload measures (i.e. NASA-TLX). These enable operators to rate their experience of mission difficultly/cognitive demands for both the overall workload of the mission and the workload associated with select critical incidents and mission phases. These measures are helpful for identifying the appropriate distribution of task load and organizational structure for a team of operators and the areas where additional automation may be desired.

Examples of relevant metrics are:

- the quality and extent of operator station training that is needed for operators to be effective in using the system
- speed of task completion vs. mission completion requirement speed
- accuracy of task completion
- identification of points at which critical tasks are dropped
- mission workload (overall and for critical tasks)
- reduction of required operators without impact to mission effectiveness

For the operator to achieve effective mission management of the system, it is important to maintain situation awareness to the progress of the mission. There are a number of subjective measures that can be used where operators rate their understanding of the situation. There are also objective measures such as "blanking the screen" and asking the operator to answer questions about key features of the situation & make predictions about expected mission progress. A final area is SA for critical incidents. These measures can be used to identify the effectiveness of user interface displays in allowing the operator to monitor key events relevant to mission tasks.

Some subjective human performance measures include:

- operator understanding of the mission complexity
- SA during the mission
- operator correctly using automated capabilities
- operator trust of automated capabilities
- efficiency and accuracy of decision-making
- operator effectiveness in mission prosecution

These measures can also be used to judge the effectiveness of the system concept of employment.

Factors, which will affect the outcome of such measurements, are:

- the background of the operator
- the number of operators
- the quality and extent of operator training
- quality of the operator interface software
- the commonality to other operator interfaces of relevant systems
- complexity and tempo of the mission
- the level of autonomy of the vehicle

- the complexity of the scenario

5.3 *Maritime Detection and Classification Metrics*

Images collected and processed by unmanned vehicles are useful to mission planning if they provide detailed topological and/or object location and identification information.

These capabilities may be described by the following measures:

- accurate spatial digitization of objects and environment by the image processing algorithms
- robustness of image processing algorithms in varied environments
- ability of algorithms working with COTS products to do real-time image processing and object recognition
- ability to derive understanding/real-time map of harbor and river environments from vehicle ISR image processing
- efficiency with which real-time image processing results can feed mission re-planning

5.4 *JIMMACE System Modeling*

The JIMMACE model in conjunction with a shared memory interface can be used to model an idealized mission planner and fully automated mission control operator station. JIMMACE tactics can be employed to simulate a systems with varying degrees of autonomy. JIMMACE simulated players can assume the role, mission and function of operators and mission planners. Metrics can be extracted from the model output for comparison with the mission planning components and manned operator stations to identify inefficiencies.

6. SUMMARY

There is still a great deal of uncertainty about how best to use metrics to evaluate future autonomous system. This paper discussed a range of approaches to metrics that are currently being examined for use in planned demonstrations with a wide variety of autonomy components. Experimentation with different metrics over the course of these demonstrations will be help better define under what circumstances a particular metric is appropriate and useful.

7. REFERENCES

[1] J. McMaster, S. O'Day, J. Sarratt, "Millennium Challenge 2002 Fleet Battle Experiment Juliet Analytical Excursions with JIMMACE", 9th ITEA Modeling and Simulation Workshop, Las Cruces, NM, December 8-11, 2003.

Measuring Autonomous UAV Surveillance Performance

Michael Freed, Robert Harris, Michael Shafto
{mfreed, rharris, mshafto}@arc.nasa.gov
NASA Ames Research Center

ABSTRACT: *We describe an approach to evaluating algorithmic and human performance in directing UAV-based surveillance. Its key elements are a decision-theoretic framework for measuring the utility of a surveillance schedule and an evaluation testbed consisting of 243 scenarios covering a well-defined space of possible missions. We apply this approach to two example UAV-based surveillance methods, an algorithms called 2-opt and a human-directed approach, then compare them to identify general strengths and weaknesses of each method.*

UAV-based Surveillance

Aerial reconnaissance, surveillance, and other observation tasks have been primary aircraft applications since the early days of powered flight. They remain key activities in domains from military and security operations to land management and scientific research. However, airborne observation is typically a deadly dull process that strains the vigilance and morale of human pilots and makes poor use of their costly, hard-won skills. Thus, following the rule of "dull, dirty or dangerous," it is considered an excellent application for autonomous vehicles. Unmanned aerial vehicles (UAVs) have been employed in this capacity for decades, though almost exclusively for reconnaissance (DoD 2002). Technological improvements combined with increasing investment and interest in UAVs promise to increase their capabilities and availability, thus enabling more diverse and demanding missions. Of particular interest to several operational communities are missions using UAVs to maintain "situation awareness" by continuous or periodic surveillance.

Autonomous surveillance of spatially separated sites raises issues beyond those related to reconnaissance at a single site. In particular, since a given UAV can only be at one place at a time, it must be treated as a limited resource that needs to be allocated as effectively as possible. Effectiveness, in this case, means providing the best possible information to the user at the best possible time – i.e. maximizing the value of returned information. For any surveillance agent, airborne or otherwise, this entails a variety of interlinked choices about which sites to visit over the course of a mission, how often to visit each site, what paths to take, how long to spend observing, and what kind of measurements to take (cf. Sacks (2003) for a related discussion on police patrol, Carbonell (1969) regarding human visual scanning of instruments and Koopman (1956) regarding submarine-based search).

Factors specific to aerial vehicles affect what kind of algorithms can most effectively make these decisions. For instance, Massios et al. (2001) have studied the problem of optimizing surveillance for autonomous ground vehicles (UGVs) operating inside buildings. In this case, the problem of deciding where to go next is highly constrained by the structure of the building while the problem of how to get to a location not immediately adjacent requires path-planning. With UAVs, the situation is reversed. Sites of interest may all be accessible by a direct path, reducing the need for path-planning but leaving the problem of where to go next physically unconstrained. A second factor, wind, usually has little effect on UGVs, but has a large effect on UAVs, increasing or reducing required traverse time between almost any two sites. Algorithms for UAV-based surveillance should thus treat expected wind conditions (including variability) as a central parameter and should adapt dynamically to changes in wind speed or direction.

Differences in vehicle mobility and vantage together create a third significant difference between UGV- and UAV-based surveillance. Because of its altitude, a UAV will frequently be able to observe a site from a distance without obstruction and thus may not have to travel the full distance to that site. And, due to the low friction on an air vehicle in aerodynamic flight, a UAV making fast-time observations may be able to retain most of its speed when transitioning between approach to one site and approach to the next. A surveillance algorithm that takes advantage of these aviation-specific factors should perform significantly better than one that does not.

Our work on UAV-based surveillance represents one part of a larger project to develop a practical and flexible UAV observation and data-delivery platform. The Autonomous Rotorcraft Project (Whalley et al. 2003) is an Army/NASA collaborative effort combining advanced work on avionics, telemetry, sensing, and flight control software in addition to software for high-level autonomous control. The base platform selected for the project, a Yamaha RMAX helicopter, has been enhanced in a variety of ways that increase its potential effectiveness as a surveillance vehicle. Flight control software allowing it to fly aerodynamically extends the vehicle's speed and improves its fuel-efficiency, thus extending both operating range and base flight duration (60 minutes hovering with full payload). The vehicle includes a range of sensors and the capacity to integrate and control additional sensors as demanded by particular missions. Its high-level autonomy component, Apex (Freed 1998), incorporates reactive planning and scheduling capabilities needed for mission-level task execution, navigation, response to health/safety contingencies and

interaction with human users. To enable the system to become highly effective for surveillance, scheduling capabilities must be extended based on algorithms of demonstrated effectiveness in diverse mission scenarios relevant to the Army and to NASA.

The diversity of possible surveillance missions poses particular challenges. First, an algorithm that performs well in certain kinds of missions may perform poorly in others. For instance, an algorithm that does well optimizing observations for a small number of closely spaced sites may not scale well to missions involving a large number of sites spread out over a wide area. Similarly, an algorithm that assumes that information obtained at different sites becomes obsolete at equal rates or that the value of making an observation at one site necessarily equals that at another will not perform well when such assumptions do not hold. It is not yet well-understood which attributes are most significant in distinguishing one mission from another. While the number of sites to be observed is clearly an important factor, the importance of other factors, e.g. the centrality of the takeoff/land location with respect to the set of target sites, is less clear. Finally, for a single system to provide autonomous surveillance capability for a broad range of missions requires an underlying theory of surveillance. If users need to communicate mission goals in terms of that theory, its generality is likely to pose difficulties for most users (Freed et al. 2004). For instance, a theoretical foundation based on mathematics unfamiliar to most users (as will be described below) may require them to specify the mission in terms of seemingly exotic mathematical parameters.

To meet these challenges requires: (1) developing methods for measuring the effectiveness of a given algorithm and for comparing the performance of an algorithm to that of human operators (i.e. to current practice); (2) creating planning and scheduling algorithms that perform surveillance effectively in significant parts of the space of possible missions; and (3) addressing issues of usability in the specification of missions by non-expert users. In this paper, we describe our work in the first of these areas to create a framework for evaluating algorithm performance and human performance at surveillance tasks. We then illustrate the application of the framework using two example surveillance techniques – a modified 2-opt algorithm and human-directed surveillance.

Measuring Surveillance Performance

The first issue in devising an evaluation framework is to define what it means to do a good job at surveillance. Intuitively, the purpose of surveillance is to return information on a set of targets to some user or set of users. Performance at the surveillance task will depend on the information's quantity, accuracy, importance and timeliness. As will be discussed, there are many variations on the general problem. To accommodate the diversity of surveillance missions, we start with a very general, decision-theoretic formulation of the overall goal: to maximize the utility of returned information over a defined interval.

Like Massios et al. (2001), we characterize information value in the negative – i.e. in terms of the cost of not having observed a target for a given interval rather than the benefit of having observed the target at a given time. Consider the example of maintaining surveillance over a set of buildings, any of which might catch fire at any time. Observing the building allows us to call the fire department if necessary, and thus limit the amount of damage. The longer we go without observing, the more likely it is that a fire will have occurred (though the probability may still be very small) and the more damage any such fire is likely to have inflicted. Thus, the expected cost of not observing the building (and thus remaining ignorant of its state) for a given interval depends on the fire's probability and expected cost of occurrence. Specifically, the expected cost of ignorance (ECI) for having not observed a target τ during the interval t_1 to t_2 is:

$$\mathrm{ECI}_{\tau}(t_1, t_2) = \int_{t=t_1}^{t_2} p(t) \cdot \mathrm{cost}(t_2 - t)\,dt$$

where *p(t)* is probability density function for the occurrence of some cost-imposing event E (e.g. a fire breaking out) and *cost(d)* is a function describing the expected cost imposed by E as a function of the time from occurrence to intervention. In other words, the cost of ignorance is the sum, for all points in the interval, of the probability of the event occurring at that point[1] times the expected cost if it occurs at that point. If more than one kind of event can occur at a target, and the event-types are uncorrelated, the expected cost of ignorance is simply the sum of the ECI values for each.

Over the course of a surveillance mission, an interval running from t_{start} to t_{end}, expected cost accumulates at each target[2]. If the target is never observed during that period, the total mission ECI for that target is determined by the above equation for ECI_{τ} with $t_1=t_{start}$ and $t_2=t_{end}$. Otherwise, observations divide the target's mission timeline into a sequence of intervals I_{τ} where the target's total mission ECI equals the sum of ECIs for each interval.

$$\mathrm{ECI}_{\tau\text{-mission}} = \mathrm{ECI}_{\tau}(t_{start}, t_{end}) = \sum_{i}^{I_{\tau}} ECI(i_{start}, i_{end})$$

[1] Here we assume expected detectability latency $l_0 = 0$ and refer to the time of occurrence of an event rather the time it becomes detectable. Values of $l_0 > 0$ can be accommodated by integrating from $\max(0, t_1 - l_0)$ rather than from t_1.

[2] Time t_{start} represents a reference start time at which costs begin accruing.

The effect of observation occurring at t_2 is to reduce the maximum expected cost of an event occurring at $t < t_2$ from $cost(t_{end} - t)$ to $cost\ (t_2 - t)$. This reduces the total mission ECI and also constrains its maximum. For example, *cost(t)* may asymptote at \$5M, corresponding in our example to the building burning to the ground. If, e.g., the building is observed every 30 minutes and *cost(30 minutes)* is \$1M, the ECI over the course of the mission for that target cannot exceed \$1M.

With this way of determining the mission ECI for a target, the total mission ECI can be defined simply as the sum of mission ECIs for all surveillance targets. The performance of a surveillance algorithm in a given mission is thus measured by its success in minimizing this total expected cost. We define ECI_{max} as the total mission ECI if no targets are observed during the course of a given mission and $ECI_{<method>}$ as the total mission ECI resulting from an observation schedule generated by a particular method. Thus:

$$value_{<method>} = ECI_{max} - ECI_{<method>}$$

Modeling a Mission

The choice of what probability function and what cost function to use to model ignorance cost at a given target depends on the kind of cost-imposing event(s) that may occur there. Some events are once-only, meaning that we assume they can occur at most once during the course of a mission (e.g. theft of an item). Others can re-occur serially (e.g. a security gate left open which can be closed and then left open again) or in parallel (e.g. an individual entering an area illegally). Event probability may vary with some regular event (e.g. rush hour, night time), contingent upon some other event (e.g. rain) or may remain constant. For the work described here, we have assumed that all events are once-only and that occurrence probability is constant assuming no prior occurrence. Thus, the exponential function $1 - e^{-at}$ describes the probability that event E has occurred by time t (assuming the start of the mission $t_{start} = 0$); its derivative yields the probability density function $p(t) = ae^{-at}$.

The cost function combines a number of factors. Most important is how the physical process initiated by an event unfolds and how cost accrues as a result. For instance, a building fire may start out slowly, at some point begin increasing rapidly in intensity, then eventually taper off as flammable material runs out and the cost of the fire approaches the total value of the building. This suggests an s-shaped cost function such as a sigmoid. Other factors include the initial cost c_0 of the event (e.g. from an explosion that causes a fire), the maximum cost m that may accrue from an event (e.g. the cost of the building plus fire cleanup costs), the expected intervention latency l_1 (e.g. how much time it takes firefighters to get to the site and put out the fire) and expected reporting latency l_2 (e.g. how long it takes to get in range to alert firefighters). The work described here assumes that all events are modeled using a sigmoid normalized to intercept the y-axis (cost) at c_0 and to asymptote at m.

$$\text{cost}(d) = c_0 + \left(\frac{2}{1 + e^{-k(d + l_1 + l_2)}} - 1 \right)(m - c_0)$$

Multiplying the probability and cost functions with initial-cost and latency factors factored out for simplicity ($c_0 = l_1 = l_2 = 0$), we get the ECI equation below for evaluating the expected cost of not observing a specified target (associated with parameters a, k and m) during the interval t1 to t2 (each a displacement from the mission start time t0=0).[3]

$$\text{ECI}_{\tau}(t1, t2, a, k, m) = \int_{t=t1}^{t2} ae^{-at} m \left(\frac{2}{1 + e^{-k(t2 - t)}} - 1 \right) dt$$

From this framework, a clear process emerges for how a user can specify mission parameters, apply a surveillance decision method and then evaluate the output of that method with respect to the mission. The first step is to specify the mission. This involves defining a start/end location, mission duration, surveillance vehicle (with range, kinematics, sensors and other characteristics) and set of target locations. Each target is associated with one or more events, and each event with parameterized probability density and cost functions. Given our previously described assumptions about these functions, users would specify three parameters for each event: a, k and m. The value m is simply the maximum (asymptotic) cost of the event. To determine the probability rate parameter a, a user should specify some reference probability interval for the event. For instance, the user may specify that the probability of the event is 0.2 during a 60 minute interval. Solving for a yields the value .00372. To determine the cost rate parameter k, the user should specify a reference cost interval such as \$1M during the first 30 minutes following occurrence. Solving for k yields the value .0135.

Second, after specifying all elements of the mission, this information is made available to the algorithm or person responsible for generating a surveillance schedule. The algorithm or person's output may take the form of a repeatable sequence that must be translated into a schedule. For example, the sequence ABCAB denotes that targets A, B and C will be visited repeatedly and in order, skipping C on alternate circuits and breaking off just in time to return to the start location before the mission end time. A schedule

[3] Though, we implemented a closed-form solution for the integral, we found that simple numeric approximation methods provide equivalent speed and accuracy.

specifying at what times each target is observed over the course of the mission can be generated by simulation based on vehicle characteristics, weather and map information. The resulting schedule is then used to compute $value_{<method>}$ as described in the previous section, providing a measurement of the expected benefit of performing surveillance using a given method.

Comparative Evaluation Testbed

In the previous section, we addressed the question of how to measure the performance of a surveillance method in a given mission. The next step is to make it possible to compare different methods so as to learn their relative strengths and weaknesses. Such comparisons serve two important practical purposes. First, the process of developing and refining surveillance algorithms depends on knowing what weaknesses should be addressed and on being able to measure the effect of intended improvements. Second, this kind of analysis might allow a system to automatically select the best method for a newly defined mission by matching to the most appropriate method.

Comparative analysis requires testing surveillance methods against a set of significantly different mission types. This raises the question of what features are likely to differentially affect the performance of different methods. A set of such features would provide a basis for classifying missions into different types and thus for creating a stable testbed mission set. Unfortunately, it is not altogether clear which are important. It is not clear, for example, what features should be considered at all, what tradeoffs exist in the design of algorithms that are likely to impact sensitivity to a given feature and what features tend to vary significantly in missions arising in real operations.

We have created an initial testbed mission set consisting of 243 missions based on 5 feature types (dimensions), each with 3 values. Feature types include: **N**, the number of targets to be observed, with possible values 4, 8 and 16; **spatial scale**, representing the size of the map in which the mission takes place, with possible values .002, .02 and 0.2 of the range of the vehicle; **spatial distribution**, the degree to which targets are clustered, with possible values of uniform, globular and 2-cluster; **maxcost distribution**, representing the variability across targets of the parameter *m*, with possible values of fixed, uniform, and 2-cluster; and **cost rate distribution**, the variability across targets of the cost rate parameter *k*,[4] with possible values fixed, uniform, and 2-cluster. All missions use the mission modeling framework described above and all have the following features in common: mission duration is fixed at 60 minutes (the worst-case flight duration of our RMAX helicopter); start/end point is located at the centroid of mission targets; the probability of occurrence of all events is fixed at .2 per hour; and initial cost (c_0) = detection latency (l_0) = response latency (l_1) = reporting latency (l_2) = 0.

Because we expect to enlarge and refine the testbed repeatedly as our understanding of user needs and algorithm design tradeoffs grows, we have created software that lets us easily create and modify testbeds, and run evaluation experiments with both algorithms and human subjects. The software includes a model of the flight characteristics of the RMAX, allowing us to accurately compute travel time between targets. This is likely to be especially important for evaluating the impact of spatial scale, particularly where targets are relatively near one another, since turn rate in aerodynamic flight, acceleration to cruise speed and other UAV characteristics are likely to have large and varying effects on travel time.

Case Study: 2-opt vs. Human Performance

To illustrate the described evaluation framework, we describe its application to two surveillance methods. The first method is based on a 2-opt exchange algorithm (Reinelt 1994) for the Traveling Salesman Problem (TSP). The algorithm has been modified in a number of ways in order to (a) generate a repeating cycle of visits that start and end on a given location but do not visit it in the interim and (b) make use of a flight dynamics model requiring that travel time between locations is not a constant, but instead varies with initial speed, initial turn angle and end turn angle. It is important to note that while these modifications make the 2-opt algorithm applicable to our surveillance problem, any TSP-based approach is likely to perform well only in those parts of the space of possible missions where TSP-like assumptions hold – e.g. where there are fixed value distributions for cost-rate and maxcost. In general, we expect different algorithmic approaches to perform well in different parts of the mission space.

Second, we evaluated human-decision making in directing surveillance, the method corresponding to current practice. A human subject selected surveillance paths for each of the 243 mission scenarios in our testbed. Each mission was represented graphically as a map showing all relevant dimensions. Targets were represented as icons colored to indicate cost rate (urgency) and with shape varied to represent maximum cost (importance). The start/end (home) point was displayed as a distinctive icon and spatial scale as a circle centered on the home point whose radius represented .002 of the vehicle's specified flight range. The subject used a mouse to select and modify a route and were allowed as much time as they wished on each mission. In contrast to the 2-opt method which always attempted to visit all targets, humans were allowed to exclude targets from the surveillance route if they wished.

[4] Specific values of m are {10,20,30,40} with 30 used when maxcost distribution = fixed. Specific values of k are based on {20,40,60,80} minutes to reach .9 maxcost with 60 minutes used when cost rate distribution = fixed.

Pct. Adv.			N	Spatial								
			4			8			16			
Scale	Rate	Cost	2-Cluster	Globular	Uniform	2-Cluster	Globular	Uniform	2-Cluster	Globular	Uniform	
0.002	Fixed	Fixed	0	0	0	0	**14**	0	6	**31**	-3	
		Clustered	0	0	0	0	**16**	5	7	**29**	7	
		Uniform	0	0	-1	**42**	**24**	5	2	10	**10**	
	Clustered	Fixed	0	0	0	0	0	**11**	0	0	**11**	
		Clustered	0	0	0	0	0	0	0	0	1	
		Uniform	0	0	0	0	0	0	0	0	1	
	Uniform	Fixed	**143**	**47**	0	**130**	**19**	**47**	**10**	**93**	**47**	
		Clustered	**22**	**28**	**23**	0	**12**	6	3	8	0	
		Uniform	**61**	**20**	-2	0	**18**	7	2	5	9	
0.02	Fixed	Fixed	0	0	0	1	1	0	0	1	0	
		Clustered	0	0	0	1	1	0	0	1	0	
		Uniform	0	0	0	1	1	0	1	2	0	
	Clustered	Fixed	0	0	0	1	0	0	1	6	0	
		Clustered	0	0	0	1	0	0	0	9	1	
		Uniform	0	0	0	1	**16**	0	0	6	0	
	Uniform	Fixed	0	0	0	1	0	0	2	2	0	
		Clustered	0	0	0	1	0	0	2	2	1	
		Uniform	0	0	0	1	0	0	0	2	1	
0.2	Fixed	Fixed	1	0	0	0	-1	0	-2	**-22**	4	
		Clustered	1	0	0	-1	5	-10	-6	**-15**	3	
		Uniform	0	0	-1	-2	4	-8	-5	**-19**	4	
	Clustered	Fixed	0	0	0	8	0	**14**	9	-1	10	
		Clustered	0	0	0	9	-1	**-11**	**11**	**-14**	5	
		Uniform	0	0	-1	**16**	8	3	5	**-10**	4	
	Uniform	Fixed	3	0	**23**	**14**	-7	**22**	2	-7	**10**	
		Clustered	**16**	0	**23**	9	-3	**-12**	9	**-16**	8	
		Uniform	**23**	0	**31**	**15**	4	2	-3	**-26**	5	

Table 1. Percentage difference in performance between 2-opt and human-directed surveillance

Our initial expectations were that performance would vary significantly between the methods based on certain strengths and weaknesses. In particular, the 2-opt method, with a computer's advantages in speed and precision, would presumably do well on small scale maps where aerodynamic factors would favor complex paths that minimize turn angle rather than (only) distance between targets. It would likely perform poorly on maps with varying max-cost and rate parameters since it could not reason about that information. Humans, with natural visual-spatial capabilities that exceed any computer-based technique, might perform well when targets are spatially grouped. And, allowed to exclude targets from the surveillance schedule, people would likely perform well on maps where non-fixed distributions of max-cost and rate make some targets worth skipping and on large scale maps where the importance of being selective is especially great.

Table 1 shows data for all 243 missions. Data entries represent the percentage difference in performance between the two methods, with positive values indicating 2-opt advantage and negative values human advantage. Values outside the range -10% to 10% are in boldface to indicate where the greatest differences in performance lie.

Overall performance was comparable, with 2-opt doing 4.9% better on average. In the human best case, the subject outperformed TSP by 26%, whereas the best 2-opt case had a 146% advantage. The latter was almost certainly due to human error, as the mission in which it occurred was similar to others where the subject performed well. This may indicate a phenomenon favoring algorithmic methods in general: human tendency to err when making surveillance decisions.

Across the five independent variables, scale and cost-distribution stood out as especially significant in differentiating human from 2-opt performance (standard deviations of 6.3 and 5.6 respectively). In all 9 cases where humans outperformed 2-opt by at least 10%, scale was large (0.2). In 24 of 36 (66%) cases where 2-opt was better by at least 10%, scale was small (.002). N was least significant (s.d. = 1.3), though 7 of the 9 cases with human advantage >= 10% were with N=16. The ability to exclude least-important targets is most likely to prove valuable in large scale maps with large numbers of targets. That human performance was best in those cases suggests that this ability was the principal human advantage.

Contrary to expectations, 2-opt performed relatively well with non-fixed cost and rate distributions. It performed particularly well when the rate distribution was uniform, performing at least 10% better in 24 cases. Human advantage >= 10% occurred with uniform rate in 3 cases. Confirming prior expectations, human performance was better in cases with spatial structure (globular and 2-cluster),

especially when N was high – 7 cases with 10% advantage vs. 1 for 2-opt. This suggests a constraint on the conditions in which people will be able to judge which targets to exclude.

This comparative evaluation was intended only to test and illustrate our technique. In particular, the data on human-directed surveillance came from a pilot study involving only a single subject. Given the limits of our current data, we limit our interpretation of the results to the identification of general patterns that deserve further study. We are currently running human subjects in a new version of the experiment that incorporates training in effective surveillance decision-making and provides computational decision aids through the UAV control interface. The new data should provide a more reliable measure of human performance in a genuine operational context, and thus a better baseline against which algorithmic performance can be measured.

Next Steps

As described in the first section, a practical and effective UAV-based surveillance capability requires efforts in three areas. The first is to develop means to evaluate and compare different surveillance methods. There are numerous ways to improve the presented approach. The mathematical framework should be extended to include more event types (e.g. sequentially reoccurring), more event features (e.g. detection latencies) and more diverse probability and cost functions. The mission testbed should be refined and extended to include additional features and a greater range of values for each feature type (e.g. N=100). And the whole framework should be extended to accommodate multiple surveillance agents including not only multiple UAVs, but also heterogeneous human and robotic observers.

The second area of work is to develop new and better surveillance algorithms, iteratively refining them based on comparative analyses of their strengths and weaknesses. A particularly important class of algorithms are those that make and/or modify surveillance decisions at execution-time in response to changing conditions (e.g. wind shifts, changes in user information needs). Though our framework has been described as a way to evaluate surveillance schedules prior to execution and without regard to such changes, it applies equally to post-hoc evaluation of schedules generated reactively (at execution-time) in response to unfolding events. As significant changes in physical conditions and user needs are likely to occur frequently in realistic missions, we anticipate that this framework will ultimately be more useful for evaluating reactive surveillance methods than for methods that schedule exclusively in advance. In particular, we anticipate applying it to assess ongoing scheduler enhancements to the Autonomous Rotorcraft Project helicopter's mission-level autonomy component (Apex).

Finally, these approaches must be made "usable" in real operational contexts where limits on time, knowledge and user expertise are likely to constrain interactions with the surveillance agent. On issue of particular concern is to enable users without a background in decision-theory or mathematics to specify mission parameters. Though users may be experts in the operational domain, eliciting the required utility and probability knowledge from them is notoriously difficult, though useful techniques exist (French 1986) and continue to emerge (Wang and Boutilier, 2003).

Acknowledgements

This work was supported by the NASA CICT / Intelligent Systems program. We thank Peter Cheeseman, Jeremy Frank, Roger Remington, Byron Roe, David Smith and Matthew Whalley for many useful discussions.

References

Wang, T. and Boutilier, C. (2003) Incremental utility elicitation with the minimax regret decision criterion. In *Proceedings of the 2003 International Joint Conference on Artificial Intelligence.* Acapulco, Mexico.

Carbonell, J.R. (1969) A queueing model for many-instrument visual sampling. IEEE Transactions on Human Factors in Electronics, 1966, HFE-7, 157-164.

DoD (2002) *Unmanned aerial vehicles roadmap 2002-2027*. Report to the Office of the Secretary of Defense.

Freed, M., Harris, R. and Shafto, M. (2004) Human-interaction challenges in UAV-based autonomous surveillance. In *Proceedings of the 2004 Spring Symposium on Interactions Between Humans and Autonomous Systems Over Extended Operations*, AAAI Press.

Freed, M. (1998) Managing multiple tasks in complex, dynamic environments. In *Proceedings of the 1998 National Conference on Artificial Intelligence*. Madison, Wisconsin.

French, S. (1986) *Decision Theory*. Halsted Press, New York.

Koopman, B.O. (1956) The theory of search; Pt I: Kinematic Bases. *Operations Research,* 1956, 4, 324-346.

Massios, N., Dorst, L. and Voorbraak, F. A Strategy for Robot Surveillance Using the Hierarchical Structure of the Environment. In Proceedings of the IJCAI 2001 Workshop on Reasoning with Uncertainty in Robotics, 43-50, Seattle, USA.

Reinelt, G. (1994) *The Travelling Salesman. Computational Solutions for TSP Applications*, volume 840 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg New York, 1994.

Sacks, S. (2003) Evaluation of Police Patrol Patterns. *Working papers 2003-17*, University of Connecticut, Department of Economics.

Whalley, M., Freed, M., Takahashi, M., Christian, D., Patterson-Hine, A., Schulein, G. and Harris R. (2003) The NASA/Army Autonomous Rotorcraft Project. In *Proceedings of the American Helicopter Society 59th Annual Forum*. Phoenix, Arizona.

INTELLIGENT AUTONOMY AND PERFORMANCE MEASURES FOR COORDINATED UNMANNED VEHICLES

A. Scott Lewis[+] and Lora G. Weiss[++]
[+] Research Associate, asl103@psu.edu
[++]Senior Research Associate, lgw@enterprise.arl.psu.edu
Applied Research Laboratory/The Pennsylvania State University (ARL/PSU)
State College, PA

ABSTRACT

This paper describes an autonomous Intelligent Controller (IC) architecture directly applicable to the design of unmanned autonomous vehicles and performance measures associated with intelligent autonomy. The vehicles may operate independently or cooperate to carry out complex missions involving disparate sensors or payload packages. An approach to measure the performance achieved with collaborative control is presented and simulation scenarios are provided to demonstrate how the metrics are applied.

KEYWORDS: *Emergent capability, collaborative control, intelligent control, unmanned vehicle, performance metrics*

1. INTRODUCTION

This paper describes an autonomous Intelligent Controller (IC) architecture directly applicable to the design of unmanned autonomous vehicles and collaboration and coordination between them. Two fundamental issues associated with multiple unmanned vehicle control are: how is collaboration enabled within the architecture and how is performance measured. This paper presents a behavior-based control approach for intelligent autonomy for a group of coordinated vehicles and it describes a metric for assessing collaborative performance.

There are many variants of behavior-based architectures. One of the earliest was the subsumption architecture of Rodney Brooks[1]. The basic concept is that the control system is constructed around a collection of largely independent operational capabilities referred to as behaviors or behavior-generating elements.[2,3] Prototype designs of such systems have shown that the overall capability of a system can exceed that of more conventional architectures, and sometimes to a surprising degree.

For multi-vehicle collaborative control, Chandler and Pachter[4] summarize research issues involved in autonomous control of tactical UAVs. They conclude that decision making through planning and management are the essence of the autonomous control problem, and they determine that hierarchical decomposition is a promising approach.

Stipanovic et al.[5] use decentralized overlapping control for a formation of UAVs. The dynamic model of the formation with an overlapping information structure constraint is treated as an interconnecting system with overlapping subsystems. Their approach, though, does not enable dynamic reconfiguration or the ability to reconfigure the mission plan.

Boskovic et al.[6] present a multi-layer control architecture for UAVs with four layers: (1) Fault-tolerant redundancy management, (2) Trajectory generation, (3) Path planning, and (4) Decision making. They propose a model switching method to address different failure scenarios.

Measuring performance of groups of intelligent systems is a topic of recent interest. Jacoff et al[7] present performance metrics for urban search and rescue robots with emphasis on pertinent robot capabilities and different robotic implementations. Yang et al[8] use performance metrics in the development of a collision avoidance and warning system. Zadeh[9] has introduced the concept of machine IQ (MIQ) to measure the intelligence of smart machines. However, as a metric of intelligence, the MIQ is product specific and does not involve the same dimensions as the human IQ. It is relative and the MIQ of a camera made in 1990 would be a measure of its intelligence relative to cameras of the same era and would be much lower than the MIQ of cameras made today.[10]

Several workshops on performance metrics for intelligent systems have been organized by the National Institute of Standards and Technology (NIST) and their results encapsulated in proceedings. Evans and Messina[11] discuss challenges and issues in defining performance metrics for intelligent systems. They cite government agencies basing major programs on intelligent capabilities and emphasize that there is no consensus on how to define or measure an intelligent system. However, they summarize the traits that an intelligent controller might have including: adaptability, capability of learning, doing the right thing or acting appropriately, non-linearity, autonomous symbol interpretation, goal-oriented, and knowledge-based.

An engineering perspective is given by Lee et al[12] in which they present several questions that should be asked prior to the definition of the metric of system intelligence. Among those are (1) should the intelligence measure be goal-dependent or goal-independent (2) should the intelligence measure be time-varying or time-invariant and (3) should the intelligence measure be resource-dependent or resource-independent? DeLeo[13] proposes measuring classifier

intelligence by computing the area under the receiver operating characteristic (ROC) curve and using the concept of the separation index he introduces. Feddema et al. discuss their view of emergent behavior with regard to finite state machines[14].

Albus[15] claims a barrier to the development of intelligent systems is the lack of metrics and quantifiable measures of performance and that there cannot be a science of intelligent systems without standard units of measurement. While the determination of performance metrics and measures with regard to physical entities is precisely defined and accepted, performance metrics and standards for intelligent systems are loosely defined and no acceptable standards exist. This paper presents an intelligent control architecture for collaborative control and an approach for measuring performance.

The Intelligent Controller (IC) described in this paper was initially based on the subsumption approach, but actual system needs presented more challenging requirements. This resulted in a newer, more novel approach to intelligent control architectures[16, 17, 18] .

Extending the IC architecture for collaborative behaviors also resulted in a unique approach for coordinated control. This resulted in the derivation of a measure of the performance gained by operating as a coordinated group of autonomous vehicles vs. a group of autonomous vehicles operating on their own accord. This metric is discussed in this paper and calculated for two scenarios.

2. INTELLIGENT CONTROLLER (IC) ARCHITECTURE

To appreciate the approach to measuring performance, one must understand the underlying architecture. This section provides the architecture used for collaborative control. The Intelligent Controller (IC) architecture developed at Penn State University's Applied Research Lab (ARL/PSU) is composed of two main modules: Perception and Response. The Perception module is where sensor data is analyzed, information is integrated, and interpretation of the events is generated. The Response module is where the situation is assessed, plans are generated, and re-planning or plan execution occurs. Figure 1 illustrates the IC modules for a single controller. The responses from the Response module are in the form of commands and communications to vehicle subsystems or to external systems. These systems and subsystems may be other ICs, conventional control systems (effectors), or human collaborators.

2.1 Perception Module

The role of the Perception module is to create an internal representation of the external world relevant to the IC, using sensor data streams as inputs. A key capability of the Perception module is to make correct inferences and recognize the existence of properties in the representational objects (e.g., obstacles) from incomplete and potentially erroneous input data. The Perception module contains data fusion algorithms and Continuous Inference Networks (CINETs).[19] CINETs are used to infer properties or events, such as "target" or "friend", by appropriately combining multiple pieces of information in an automatic recognition process.

2.2 Response Module

The role of the Response module is to plan and execute *in real time* a course of action to carry out a specific mission, given the situational awareness derived by the Perception module. The Response module is decomposed into three levels: A Mission Manager, Behaviors, and Actions. The Mission Manager retains the big picture and specifies a mission plan, which is a list of relevant Behaviors to be executed. Each Behavior has its own plan to execute, which is a list of Actions to be conducted. Control is cycled and interrupted appropriately using an Execution Engine within the Response module. The Execution Engine is an application-independent component of the Response module that calls the functions in the Response module in an appropriate order. Figure 2 illustrates components of the Response module for a UAV with a selected set of Behaviors (blue) and Actions (pink).

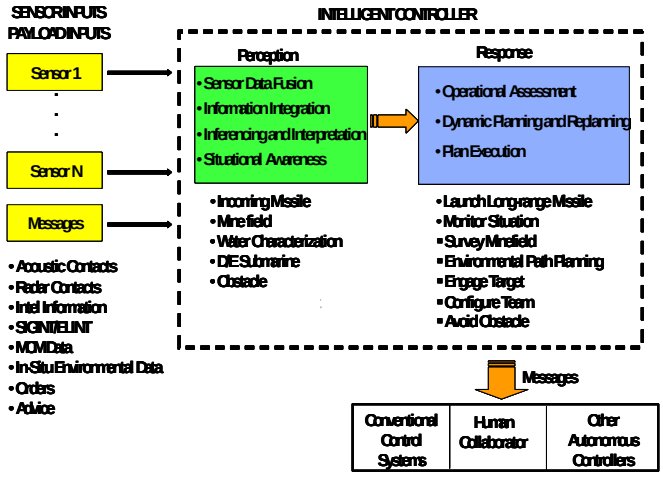


Figure 1: High Level Decomposition for a Single IC.

2.3 Multiple ICs

Multiple ICs can be integrated in a hierarchy for coordinated control. Each IC retains the same architecture yet has a different local mission to execute (such as UAVs searching different local areas). This replication of architecture allows management of complexity and considerably simplifies the problem of designing multiple, interacting, intelligent controllers for complex systems.

Just as a collection of Behaviors within an IC needs a Mission Manager for coordination and arbitration, a collection of ICs within a system also requires a supervisor. This role is

assumed by a Supervisory IC. Multiple ICs may also exist within a single vehicle in this hierarchical architecture, as determined by design decisions. Figure 3 depicts how multiple ICs communicate.

For multiple vehicles, a significant portion of a mission may be achieved by having the capability of the individual autonomous units carry out their own tasks while operating in a group. Given that the individual vehicles are intelligent and capable of inferring the behavior of other cooperating vehicles through sensing and observation, only a limited number of direct communications may be necessary to achieve significant performance enhancements.

The overall capability of the aggregate system is then an emergent property stemming from the collaboration among individual vehicles coupled with their own abilities to carry out autonomous operations over a range of variations in the missions. For a properly designed collaborative unmanned vehicle system, this emergent capability is greater than that of a system composed of vehicles acting alone.

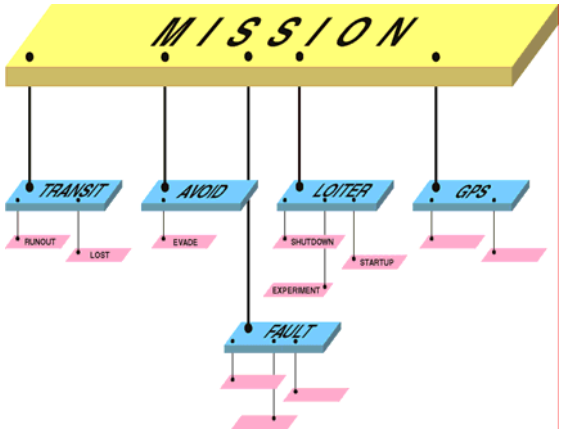


Figure 2. Sample Elements of Response Module.

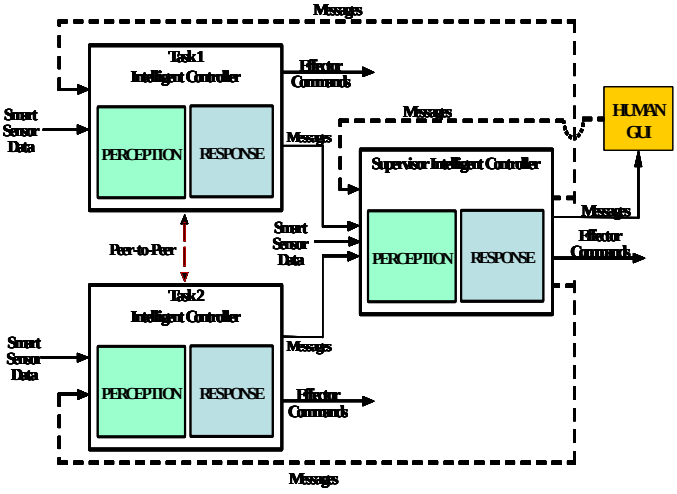


Figure 3. Architecture for Multiple IC Communications.

3. SYSTEMS OF ICs

The capability of a single IC is the aggregate of the capabilities of its modules. In particular, its operational capability is determined by the aggregate of its Behaviors. To expand the scope of its operational capability, a Behavior is simply added. To remove a capability, a Behavior is removed, which will not affect operation of the remaining Behaviors.

This approach can be extended to a group of ICs, where the architectural relationship between the Mission Manager and its set of Behaviors is replicated to define the relationship between a Supervisor IC and the set of vehicle ICs under its supervision. Figure 3 illustrates this extended Architecture. The hierarchy can be expanded arbitrarily in horizontal and vertical directions.

3.1 Communications

In contrast to the internal communication paths of Response, communication between one IC on a vehicle and another IC on a separate vehicle may not be possible at a given time due to the characteristics of the external medium. Consequently, the architecture supports both peer-to-peer communications and peer-to-supervisor communications.

Permitted message types for peer-to-peer (and peer-to-supervisor) communications are Sensor-data, External-Advisories, and Queries. Sensor data passed among ICs serves to extend the senses of the receiving IC and consequently potentially improves its performance. Bandwidth limitations imposed by the external medium may require that the communicated data be in highly processed form for sufficient compactness. External-Advisories may be used by an IC to inform partners within communication range of its operational status or its interpretation of its local environment. Queries may be sent by an IC to its partners asking for information it needs that they may be able to provide.

There is no requirement that all ICs understand exactly the same language; messages containing Words that are not understood by the Perception processing of the receiving IC may simply be ignored or routed through a central unit that may serve as a translator.

3.2 Collaborative Operations

The operating characteristics of a group of autonomous, coordinated controllers constitutes an autonomous intelligent control system, and their design based on this architecture can be summarized as follows. The autonomous system is composed of one or more ICs, where one of the ICs may possibly be a supervisor ICs. Each IC's objective is to carry out a local mission but in coordination with the global mission and defined by the set of Orders for the collective system. These Orders may be altered during mission execution by receipt of new Orders from a human or a Supervisor IC. Within the constraints of its current Orders and its design, each IC is operating autonomously. There does not exist an "optimal control law" for the system of ICs, and the designers do not attempt to derive one.

Rather, the objective in the design of each IC is for it to be able to operate "optimally" as an individual, given its current set of Orders and its perceived world as created by its Perception processing. This perceived world may include a Representational Class, say "Partner," where an instance of "Partner" represents the status of a partner or peer, including its operational plans and objectives to the extent known. This knowledge will generally be incomplete and changeable.

An autonomous intelligent control system such as this would appear to constitute a close parallel to biological systems such as beehives, ant colonies, and football teams. The system is composed of a collection of individuals with certain specialized characteristics and with some ability to communicate with each other. Operating together, the resulting system can have emergent properties, strengths, and survivability that go beyond the sum of that of the individual units.

The IC architecture also supports collaborative control of heterogeneous vehicles with varying architectures and levels of autonomy. This is accomplished by developing standard interfaces for communications and by providing the level of capability of each vehicle to the other vehicles through a database, where the database is accessed to determine the capabilities of the other vehicles before issuing

4. MEASURING EMERGENT COLLABORATIVE CAPABILITIES

4.1 The Collaborative Gain Metric

Collaborative gains result from the communication between platforms, where exchange and interpretation of information is crucial, and from coordinated mission control. Collaboration involves interpretive and behavioral adaptation among the platforms as a result of integrating communicated data into the internal representations of the central system.

Multi-platform collaboration should increase the likelihood of achieving mission objectives of the system over and above that achievable by a base capability. However, a collaborative gain is not automatically positive, since it is possible to design a system in which the communication and collaboration results in conflicts, causing deterioration in the system's ability to perform its mission.

One approach to evaluating the collaborative contribution of a system involves defining an index to measure the collaborative gain (CG) of this system. One such metric is:

$$CG = \sum_{i=1}^{M} \alpha_i \beta_i T_i \,, \tag{1}$$

where α_i is a weight reflecting the importance of the task to the overall mission, β_i denotes the success (0 or 1) of the task, and T_i is the i^{th} of *M* mission tasks. The goal is to maximize the CG.

For example, consider two UAVs, A and B, that are similarly capable of identifying a target. Assume platforms A's target has a higher priority and that A has been damaged and incapable of fulfilling its top priority. If B is about to identify its target but has only enough fuel to identify platform A's target, its priority will be reconfigured so that its original target will be ignored and its new target will be that originally assigned to A. Based on the extent of damage to A, platform A might be reassigned to another lower priority task. This type of cooperation leads to controlled emergent behavior, but mathematically quantifying it is an open task. By changing B's target, then β_i for A is 1 instead of 0 resulting in a higher CG.

In cases where it is possible to measure partial success of an unmanned vehicle accomplishing its mission, the value of β_i becomes a variable, i.e.,

$$0 \le \beta_i \le 1 \tag{2}$$

This allows partial success of an individual UAV or group of UAVs to be captured within the context of this same measure.

4.2 Multiple, Coordinated UAV Control and Evaluation

Controllers based on the IC architecture have been designed for multiple UAVs capable of executing an individual mission and collaborating with other UAVs to execute a larger, overall mission. Performance of one such design is described below.

For this collection of UAVs, the functional capabilities of the prototype group include Navigation, Avoidance, Search, Investigate, Attack, Assist, Communicate, and Supervise. Each of these operations is implemented as an independent Behavior that operates autonomously within its scope, where each conducts real-time planning and analysis of the situation relative to mission execution, and each responds appropriately to the results of that analysis.

Each behavior has one or more Actions that are responsible for carrying out sub-operations and reacting directly to objects of interest as represented in the Perception module. These reactions consist of commands to vehicle subsystems such as an autopilot or sensor control system.

The initial UAV controllers developed are actual controllers operating closed-loop in a virtual environment consisting of a simulation of the external world and vehicle subsystems. These simulated subsystems include an assumed set of sensor systems, effector systems (e.g., an autopilot or other conventional control systems), vehicle dynamics, and interfaces. This closed-loop operation, in which actual controllers are stimulated with synthetic data, allows for issues to be addressed that are indicative of those expected to be encountered by a collaborating group of UAVs.

Multi-UAV control was exercised and evaluated by using a simulation of the external environment and vehicle subsystems, including assumed sensing systems for navigation, avoidance, and target detection. The simulation

stimulated the actual controllers. The operational functionality of the prototype collaborative UAV control capabilities was exercised in various scenarios during the design process. Two of these scenarios are described below.

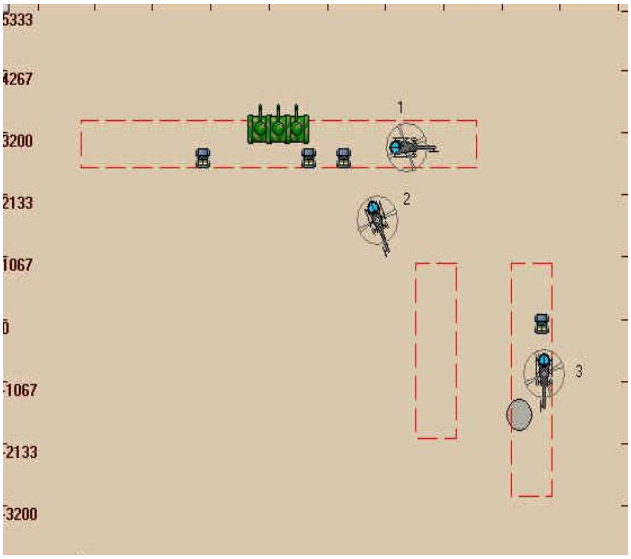


Figure 4: Screen Capture of Three UAVs Collaborating to Search and Attack Large and Small Targets.

Scenario 1: A screen capture of the first scenario is presented in Figure 4. In this scenario, three UAVs are provided search areas (red boxes) with the objective being to attack large and small targets. The UAVs are authorized to attack large targets (tanks) without independent target confirmation. However, they are required to obtain independent target confirmation prior to attacking small targets (jeeps). Included in this scenario are way points (x) and an obstacle (large circle), so that the Navigation and Avoidance behaviors are activated. Also in this scenario, towards the end of the overall mission, UAV #1 is shot down before completing its mission. The other UAVs re-plan their missions to ensure the mission of UAV #1 is completed even after it has been removed from the engagement. Such a scenario allows evaluation of the collaborative capability of the multiple UAVs to execute the overall mission of clearing out targets in the specified area.

The performance associated with this scenario can be captured in the Collaborative Gain metric given by equation (1). If the gain is measured by individual UAV success, and where α_i, β_i, and T_i are associated with UAV #*i*, *i*=1,2,3, then the performance of the individuals is given by summing the success of each individual unit accomplishing its mission. In this example, for the individual performances,

$$\alpha_i = 1, i = 1,2,3$$

$$\beta_1 = 0, \beta_2 = 1, \beta_3 = 1$$

so that without collaboration, the performance of the units is given by:

$$CG = 2.$$

If partial success is allowed to be measured (e.g., using equation (2)), then

$$\beta_1 \sim .42,$$

where β_1 includes the success of eliminating the targets (highest priority that had a 0.833 success rate) followed by successfully returning to a rendevous point for updates (lower priority with 0 success). Thus, the performance of the individual UAVs is given by

$$CG \sim 2.42.$$

When the UAVs are allowed to cooperate to execute the mission, UAV #2 completes the mission of UAV #1, so that

$$\beta_1 = 1$$

and

$$CG = 3.$$

Thus, enabling collaboration between the units provides a level of payoff in mission success.

Scenario 2: As another example of the coordinated control, one of the evaluation scenarios evolved as follows: Five UAVs were collaborating. UAV #1 was designated as the supervisor and delegated missions (search areas and rendezvous points) to the five individual UAVs. See Figure 5. During the course of operations, UAVs #2, #4, and #5 crash, and UAVs #1 and #3 return to a rendezvous point. The supervisor UAV requests the status of individual UAVs and reassigns the search areas of the downed UAVs to itself (UAV #1) and to UAV #3. See Figure 6. To further demonstrate the collaboration, the scenario then has UAV #3 go down. When the supervisor returns to the rendezvous point and identifies that UAV #3 is unresponsive, it re-plans and searches the area that UAV #3 was responsible for, and it completes the overall mission.

The performance associated with this scenario can also be captured in the Collaborative Gain metric given by equation (1). If the gain is measured by individual UAV success, and where α_i, β_i, and T_i are associated with UAV #*i*, *i*=1,..,5, then the performance of the individuals is given by summing the success of each individual unit accomplishing its mission. In this example, for the individual performances,

$$\alpha_i = 1, i = 1,\ldots,5$$

$$\beta_1 = \beta_3 = 1$$

$$\beta_i = 0, i=2,4,5$$

so that without collaboration, the performance of the units is given by:

$$CG = 2.$$

If partial success is incorporated into this metric, the performance will be slightly higher, but not as high as for cooperating units. When the units are allowed to cooperate to execute the mission, UAV #1 completes the mission of the other UAVs, so that

$$\beta_i = 1,\ i=1,\ldots,5$$

and

$$CG = 5.$$

Again, enabling collaboration between the units provides a level of payoff in mission success.

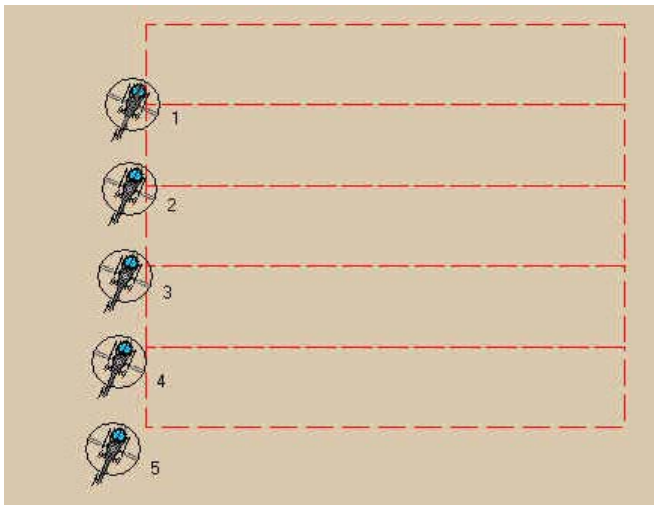


Figure 5: Screen Capture of Five Collaborating UAVs and Their Assigned Search Areas.

5. CONCLUSIONS

ARL/PSU's experience in unmanned vehicles lead to the development of a unique, robust, universal architecture for the design of intelligent autonomous vehicles. This IC architecture provides a reliable approach to the design of a single unmanned vehicle or of a system of autonomous intelligent units that collaborate with each other and with humans to carry out complex missions.

When this high level of intelligent autonomy is integrated into a system of collaborating, unmanned vehicles, an even larger gain results. This gain is measured with the quantity called Collaborative Gain. Examples indicate that the performance of collaborating units can exceed that of individual units operating on their own accord, and the improvements can be significant.

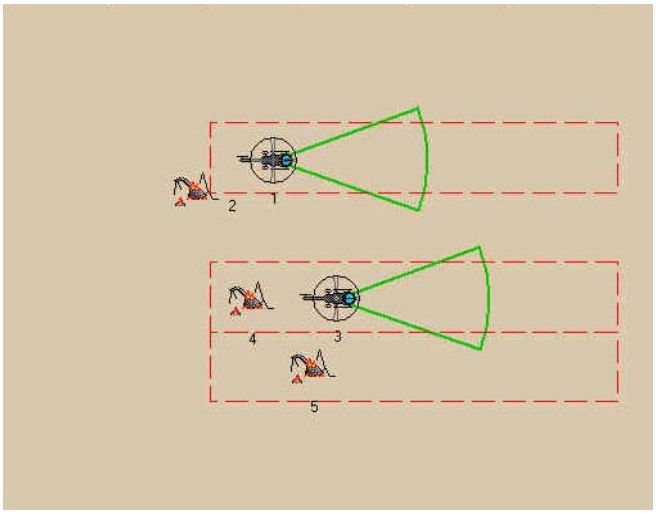


Figure 6: Screen Capture of Two Collaborating UAVs Completing the Missions of Three Downed UAVs.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge several researchers at Penn State's Applied Research Laboratory, including M. Roeckel, J. Stover, S. Neal, J. Miller, and C. Walker.

7. REFERENCES

1. Brooks, R.A., "Intelligence Without Representation, "Artificial Intelligence," 47, pp. 139-159, 1991.
2. Albus, J.S. Albus, "Outline for a Theory of Intelligence," IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No. 3, pp. 473-509, 1991.
3. Kumar R. and J.A. Stover, "A Behavior-Based Intelligent Control Architecture with Application to Coordination of Multiple Underwater Vehicles," IEEE Transactions on Systems, Man, and Cybernetics, 30, pp. 767-784, 2000.
4. Chandler, P.R. and M. Pachter, Proceedings of the American Control Conference, 1998, Vol.: 1, 24-26 June 1998, pp. 394-398.
5. Stipanovic, D. M., Inalhan, G., Teo, R. and C.J. Tomlin, Proc. of the 41st IEEE Conference on Decision and Control 2002, Vol.: 3, 10-13 Dec. 2002, pp. 2829-2835.
6. Boskovic, J.D., Prasanth, R. and R.K. Mehra, Proceedings of the American Control Conference, 2002, Vol.: 3, 8-10 May 2002, pp. 1825-1830.
7. Jacoff, A., Messina, E., Weiss, B. A., Tadokoro, S. and Y. Nakagowa, Proceedings of the 2003 IEEE International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, October 2003, pp. 3396-3403.
8. Yang, L., Yang, J.H., Feron, E., and V. Kulkarni, Proceedings of the Intelligent Vehicles Symposium, June 9-11, 2003, pp. 316-321.

9. Zadeh, L. A., "Fuzzy Logic, Neural Networks, and Soft Computing," Communications of the ACM, 37 (3), pp. 77-84, 1994.
10. Zadeh, L.A., "The Search for Metrics of Intelligence-A Critical View," Proceedings of the 2000 PerMIS Workshop, pp. 105-111, Aug. 14-16, 2000, Gaithersburg, MD.
11. Evans, J. M. and E.R. Messina, "Performance Metrics for Intelligent Systems," Proceedings of the 2000 PerMIS Workshop, pp. 101-104, Aug. 14-16, 2000, Gaithersburg, MD.
12. Lee, S., Bang, W. C. and Z.Z. Bien, "Measure of System Intelligence: An Engineering. Perspective," Proceedings of the 2000 PerMIS Wksp., pp. 112-117, Aug. 14-16, 2000, Maryland.
13. DeLeo, J., "Measuring Classifier Intelligence, Measuring the Performance and Intelligence of Systems," Proceedings of the 2002 PerMIS Workshop, pp. 26-31, Aug. 13-15, 2002.
14. Feddema, J. T., Robinett, R. D., and B.J. Driessen, "Designing Stable Finite State Machine Behaviors Using Phase Plane Analysis and Variable Structure Control," Journal of Intelligent and Robotic Systems, No. 36, pp. 349-370, 2003
15. Albus, J.S., "Metrics & Performance Measures for Intelligent Unmanned Ground Vehicles," Proceedings of the 2002 PerMIS Workshop., pp. 61-68, Aug., 2002.
16. Stover, J. A. and R.E. Gibson, "Modeling Confusion for Autonomous Systems," SPIE, Science Artificial Neural Networks, 1710, pp. 547-555, 1992.
17. Stover, J. A. and R.E. Gibson,"Controller for Autonomous Device", US Patent #5,642,467, Issued June 1997.
18. Stover, J. A. and R.E. Gibson, "PIC 95 Project Report," Tech. Mem. #00-128, Applied Research Lab., Penn. State Univ., State College, PA, Feb. 2002
19. Stover, J.A., Hall, D.L., and R.E. Gibson, "A Fuzzy-Logic Architecture for Autonomous Multisensor Data Fusion," IEEE Transactions on Industrial Electronics, 43, pp. 403-410, 1996.

Traversable Terrain Modeling and Performance Measurement of Mobile Robots

A. Shirkhodaie, R. Amrani, N. Chawla, and Vicks, T.

Intelligent Tactical Mobility Research Laboratory
Department of Mechanical and Manufacturing Engineering
Tennessee State University
3500 John Merritt Blvd., Nashville, TN 37209

ABSTRACT

In this paper, we have described a technique for terrain traversability assessment modeling of mobile robots operating in natural terrain and presented a fast near-optimum algorithm for autonomous navigational path planning of mobile robots in rough terrain environments. The proposed method is based on visual sensing of terrain salient features and analysis of geo-location coordinates of the salient features. Using an algorithmic image processing technique, both free and obstacles spaces are differentiated and multiple candidate terrain paths are generated for optimization of trajectory terrain path of the robot. The algorithm uses a fuzzy logic terrain classifier to categories different salient features of the terrain. A virtual simulation is developed for terrain perception modeling and verification of generated trajectory path plans of the robot. The developed path-planning algorithm is computationally efficient, and suitable for implementation onboard autonomous robotic systems. Several different terrain conditions have been tested to validate the proposed approach.

KEYWORDS: *Traversable Terrain Modeling, Visual Path Planning, Autonomous Robots, Measure of Performance*

1. INTRODUCTION

Autonomous navigation of outdoor terrain in an active pursuit of department of defense in battlefield reconnaissance and surveillance military operations and NASA in exploration of remote planetary surfaces by robotic rovers. There are a number of challenging issues with outdoor terrain navigation. A robot must have the ability to operation autonomously and intelligently on unstructured terrain with minimal interaction with remote human operators. Robot navigation system must provide sufficient onboard intelligence for long-range traverse in sanding, muddy, rocky, and poorly structured natural terrain, without jeopardizing the robot health and mission failure.

This paper focuses, in particular, on the terrain visual sensing and terrain salient features recognition and characterization. We have described a method for terrain salient features detection using imaging technique and proposed a method for terrain rocks formation modeling. Furthermore, we have presented a method for near optimum visual path planning of mobile robots in natural terrain similar to the planet Mars surface. Section 2 addresses some of the challenges with terrain navigation. Section 3 presents a method for visual terrain sensing and salient feature extraction and recognition. Section 4 discusses a near optimum method for navigation path planning of the robot. Section 5 presents our experimental setup. Finally, section 6 presents the conclusion of this research effort.

2. TERRAIN NAVIGATION

Robot navigation in natural terrain is challenging due to uncertainty in recognition of salient features of the environment. Natural terrain is deceiving, superficial, and complex. Without a good perception model of the environment, a robot cannot reliably navigate a terrain and traverse to its goal successfully. Terrain traversability and path planning of all-terrain robots have been addressed by a number of researchers. Howard et. al., [1] presented a technique for terrain traversability assessment learning for outdoor mobile robot navigating. Using human-embedded logic in real-time, they demonstrated a technique for development of terrain perception based on features extracted from imagery data. In their method, they introduced a fuzzy logic framework and vision algorithms for analysis of terrain. Golda et. al., [2] presented a probabilistic modeling technique suitable for analysis of high-speed rough-terrain mobile robots. They have experimentally shown that their model can accurately predict robot performance in simple, well-known terrain, however, in unstructured environment, their stochastic method performance was degraded. A combination of terrain complexity and unaccountable uncertainty measures were found as leading causes in degradation of their predictive terrain assessment model.

For long-range terrain navigation, accurate maps of the terrain are critical for robot navigation system. Without such maps, a robot may spend much time and energy venturing along what turns out to be a dead end. Olson et. al. [4] developed a method based on visual terrain mapping of Mars rovers. Using a visual stereo imaging fusion technique, they have demonstrated a reliable method for high fidelity terrain mapping and robot world perception modeling. By compiling terrain map images using a system that unified multi-resolution models, they were able to integrate Mars descent and orbital images to obtain 3D terrain maps used by rovers for navigational purposes.

Autonomous robot navigation in natural terrain can be divided into three closely related tasks: (1) *Route Planning*, (2) *Navigation Planning*, and (3) *Mobility Planning*. As illustrated in Figure 1, in *Route Planning,* objective is that of assessment of the terrain in bulk from a large distance away and deciding on feasible terrain courses that maximize traversability potential of the robot as well as its safety and health. Another objective is to identify pertinent intermediate landmarks. The landmarks are used as waypoints that robot can easily identify along its path and localize itself with respect to them if necessary. A typical route path plan may consist of many waypoints laid out on either an aerial map or a landscape image captured from a ground level, or a series of global position coordinates defined manually. Typical range of applicability of route planning ranges from 10 to 1000's feet or higher.

Navigation planning, on the other hand, can be considered as localization of obstacles, treacherous, hills, and slopes, positive and negative objects. Typical range of applicability of navigational planning varies from 2 to 12 feet for a slow moving robot and to a higher range for a fast speed robot. In navigation planning a set of short traverse path segments from current robot location to next intermediate landmark or the goal are decided. Navigational planning and following are coupled and achieved by means of a map or some model of environment. Localization and navigational error recovery is critical at this phase, in order to keep the robot as close as possible to its designated path or model frame of reference between its intermediate landmarks. Due to loss of environmental details in the field of view, navigational certainty in correct recognition of landmarks tends to diminish with the distance of landmarks from the robot. To keep the navigational certainty under control, typically the range of effectiveness of localization is within a few feet radius from the robot. Depending on the navigational speed of the robot, terrain intricacy, and sensory reliability, this range can be adjusted appropriately.

The purpose of *Mobility planning* is that of describing a set of low-level drive actions that moves the robot through its obstacle terrain while negotiating and/or avoiding obstacles along its path. An important consideration in this phase is the dynamic interaction of the robot with its environment, in particular, detection of robot wheel traction losses during steering while avoiding wheel traps, wheel supports, and tip over states [6]. Another consideration is optimization of robot's safety, energy, and reliability [5]. Typical range of applicability of mobility planning is from a few inches to a few feet depending on physical ability (i.e., wheel diameter) of the robot, and complexity of the terrain. In [6], dynamic modeling of robot and environment is shown reliable for simple, well-characterized terrain. However, on rough terrain, unknown terrain, model fidelity degrades due to imprecise knowledge of terrain parameters.

For terrain navigation, it is very difficult if not possible to obtain a precise mathematical model of the robot's interaction with its environment. Even if the dynamics of the robot itself can be described analytically, the environment and its interaction with the robot through sensors and actuators are difficult to capture in a mathematical sense. The lack of precise and complete knowledge about the environment limits the applicability of conventional control system to the domain of autonomous robotics. If one draws an analogy to navigational skills of animals and humans in nature terrain, they both perform route, navigation, and mobility planning simultaneously and intelligently without getting trapped in the middle of terrain obstacles. Vision plays the most critical role, though physical mobility ability plays an equal important role in achieving navigational mobility objective. This observation motivated what it follows here. In this paper, we have presented an approach that solely relies on visual terrain characterization and intelligent path planning based on localization of salient features of the environment.

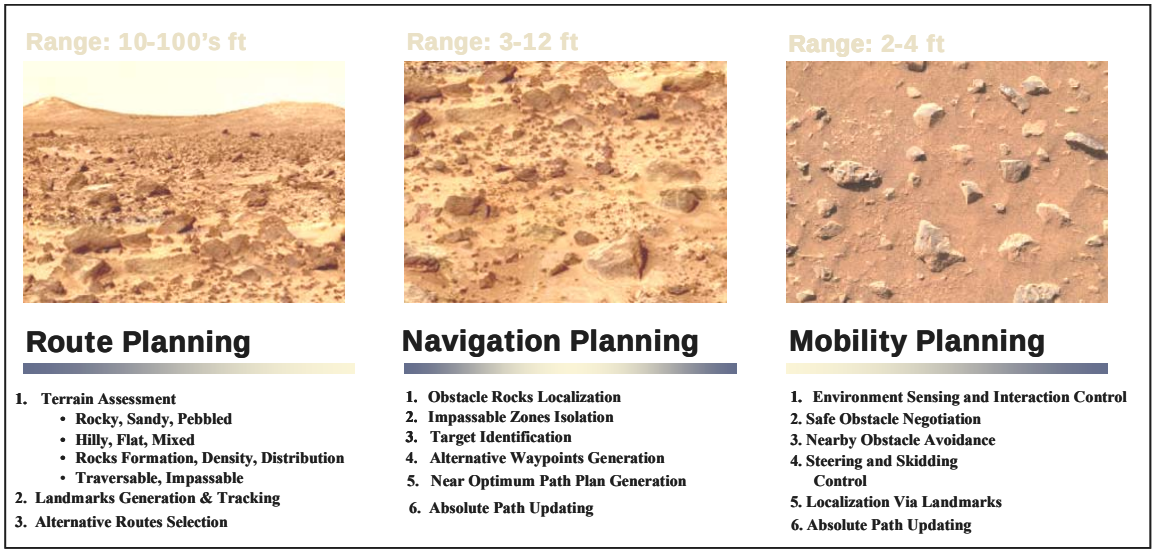


Figure 1. Three Stages of Navigational Path Planning in Natural Terrain

3. TERRAIN PATH PLANNING

The proposed visual navigational path-planning scheme is comprised of three phases. In the first phase, an image of natural terrain is obtained for detection of salient features of the environment. Using imaging techniques, the salient features are isolated and characterized. The environment features we are interested about include: obstacle free areas, rocks formations, sand piles, hills, and slopes. By salient features analysis, we can determine measures characterizing the environment surroundings and classify them appropriately. Furthermore, additional information such as rocks density and geolocation distribution, instable sand piles, and high risks ground conditions (i.e., hills, and slopes) for mobility can be discovered and avoided during navigational path planning of the mobile robot. In the second phase, our algorithm generates a number of waypoints that mark out a near optimum traversable trajectory path for the robot to follow toward its specified target location. In the last phase, a virtual model of the terrain is constructed based on geo-location coordinates and physical attributes of the salient features of the environment. The terrain model serves the robot as a reference model that embeds the robot's state of perception of its surroundings. The terrain model is applicable for simulation verification of robot's generated path plans; in particular, it is appropriate to give the operator tele-presence sense of robot's situation in the remote terrain. Figure 2 presents an example of a rough terrain that our algorithm has analyzed along with the salient features that it has characterized. It also shows a generated world perception model of the rough terrain. The followings present details of this visual terrain traversability assessment technique.

Initially, we divide an image of terrain captured by the robot into a matrix of smaller sub-windows and apply an adaptive binarization technique on each sub-window. This adaptive binarization method locally selects an optimum threshold for binarizing of each sub-window. The resultant binary image is a nonlinear map of geo-location of obstacles in the image frame and thresholds small rocks that are of insignificant size from the background. We explained the non-linearity issue of obstacles geo-location map in next paragraph. The resultant image can now be thought of a binary image where center of each black pixel represents the location of the obstacle centers in the original image. Next, a weight factor is assigned to each black pixels of the resultant binary image. By convoluting the compressed binary image, each black blob cell corresponds to an obstacle center is assigned a weight proportional on number of black blob cells surrounding it. This operation results obstacles appearing farther away from central obstacle receive less or equal weight depending on number of obstacle cell surrounding it. The result of this convolution operation is a hilly field map of the terrain environment. Note that obstacles relative gap from one another in the image frame is not linear. This is due to the fact that obstacles farther away from camera appear compressed in the depth of view. The depth perception diminishes rather

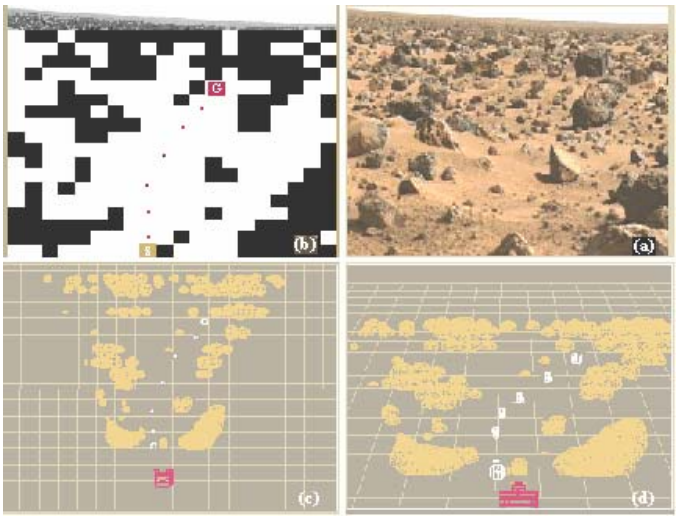


Figure 2. (a) A Rough Terrain, (2) Obstacles Geo-Location Map
(c) Generated Free Path, (d) Close-up View of Generated Free Path

with the square distance of obstacle from the camera. This fact is obvious by observation of Figure 2a. Much wider area of terrain is apparent in the upper part of image than in the lower part of image. Prior to analysis of the image field map, therefore, we map image coordinates of each sub-window to the 3D world coordinate using a camera calibration method described in [7].

Next, we apply a generic algorithm to obtain a free path for the robot among the obstacles. In the generic algorithm, size of the robot is assumed the same size as the width of diagonal distance of terrain area covered by the sub-window in the bottom row of the obstacle geo-location map. Note that physical area represented by image sub-window at upper rows is greater than that of image sub-window at the first row. The generic algorithm is a fast divide-and-conquer approach that searches the obstacle geo-location map row by row. It starts at a row above the first row and at the column where the robot is located. In that row, it searches left and right free spaces to the robot and measures a risk factor for each free space that is proportional to summation of obstacle centers around it. The algorithm determines the most suitable waypoint that minimizes the traveling distance of the robot progressively from start to goal. In an essence, it finds the safest waypoint in the current row that is closest to a straight path connecting the position of the robot to its desirable target location. This optimization method further attempts to intelligently choose a waypoint that is away from any major obstacle along its path. The multi-valued optimization process is carried out layer-by-layer and at each layer through this process a new waypoint is designated for the robot to follow.

Navigation planning is crucial for short-range robot steering. Due to close proximity of robot camera to ground very useful details about salient features of the terrain can be detected at

this stage and characterized with higher degree of confidence. As a robot traverses natural terrain, images may be periodically acquired for traversability assessment. In each period, condition of the terrain can be classified based on the relevant features extracted from the images. One good method of embedding human knowledge is to apply fuzzy logic classification system. Human knowledge can be integrated in terms of knowledge base (i.e., fuzzy rule sets and membership functions). One method also for terrain condition assessment is to perform object surface texture analysis. Objects in different natural terrain have certain distinctive texture properties. The common surface texture attributes include: contrast, variance, energy, entropy, and homogeneity. Image texture is basically due to arrangement of pixel intensities variations that form certain repeated pattern(s). Image repeated patterns are caused by physical surface properties of terrain objects, such as rocks roughness, sand piles waviness, and so on. They could be result of light reflectance from surface of an object. One reliable way to classify textures is to apply quantitative statistics.

An image is a matrix of pixel intensities, $I_{i,j}$. We can define co-occurrence of image matrix as $P_{i,j}$ such as every entry in co-occurrence matrix, $Pd_{i,j}$, is difference in intensity between a pair of image pixels(i and j), that are distance d pixels apart in original image in a given direction. With this notation, the *Energy* associated with an image that is a measure of textural uniformity of an image is defined as:

$$Energy = \sum_i \sum_j P_d^2(i, j) \quad (1)$$

Image Energy reaches its highest value when its image pixel intensity level distribution has either a constant or a periodic form. Furthermore, *Image Entropy* is a measure of disorder of an image and achieves its largest strength when all elements in the P are equal. Entropy is inversely proportional to Energy and is defined as:

$$Entropy = -\sum_i \sum_j (i-j)^2 \log P_d(i, j) \quad (2)$$

Image contrast, on the other hand, is a difference of the P and it measures the amount of local variation of an image. The image contrast is measured by:

$$Contrast = \sum_i \sum_j (i-j)^2 P_d(i, j) \quad (3)$$

Image homogeneity is inverse different moment measures of image and achieves its largest value when image pixel intensity repetitions are concentrated near the main occurrence matrix diagonal. The image homogeneity is defined as:
In order to minimize the computation requirement, we choose

$$Homogeneity = \sum_i \sum_j \frac{P_d(i, j)}{|i-j|^2}, \qquad (i \neq j) \quad (4)$$

the contrast, variance, and energy texture attributes as basis for terrain surface texture analysis. They provide reliable statistical assessment of a terrain object surface texture, in particular, when small image window are analyzed to assess the terrain condition.

In our approach, we divide a full size terrain image into hundreds of finite (small) sub-windows. For each finite image sub-window, we perform surface texture analysis and apply fuzzy logic rules for classification of the terrain. This process is analogous to stress state analysis of a loaded mechanical part when subjected to Finite Element Analysis (FEA). In a FEA process, stress state of each finite element is individually computed under physical constraints and restriction and then stress states of computed finite elements are aggregated to assess the complete stress state of the object under applied loads. We follow a similar approach. We initially conduct terrain traversability assessment on individual finite sub-windows of image independent of context of the whole image, and then aggregate the results to achieve terrain traversability assessment measures and classifying salient features of the nature terrain vigorously.

This method has two benefits. First, it allows much simpler, yet more inclusive fuzzy rule system to be developed for terrain traversability assessment purposes. Secondly, the method offers an opportunity for performing parallel image processing since each sub-window image can be independently analyzed. This feature can expedite terrain traversability assessment considerably if the robot has onboard parallel computation capability. For development of fuzzy rules and membership functions, we chose a set of natural terrain image samples randomly selected from among many salient features of different terrains. We asked some terrain experts to classify terrain conditions at locations where the finite image sub-windows were taken. In parallel, for each finite image sub-window we computed their corresponding image surface texture properties. We compiled over 200 different natural terrain data patterns. By considering range of texture attributes variation of these sample data, we developed a set of suitable fuzzy logic membership functions and fuzzy rule systems that closely mimic the human expert's judgment of the terrain traversability.

4. TERRAIN TRAVERSABILIY ASSESSMENT ALGORITHM

Table 1 summarizes the algorithm we discussed above. The algorithm performs identification, localization, and recognition of rock formations and generates a collision free traverse path based on the optimization technique discussed earlier in section 3. The algorithm initially enhances the image by removing noises and applies a Canny edge detector to extract out rocks edges. Next, an adaptive binarization convolution method is applied to binarize the image. The result is the blobs of rock formations with enhanced edges.

The algorithm sorts the rock blobs according to their size and ignore the smaller rocks that can be negotiated with the robot during mobility planning phase. The remaining rock blobs are treated as obstacles that the robot needs to avoid. The algorithm computes center location and principle moments of the rock blobs.

The algorithm uses a Fuzzy Logic Terrain Classifier (FLTC) to compute a certainty confidence factor for each detected rock blob. The objective of FLTC is to verify that indeed the detected blob has surface texture similarity close to that of a rock. Using texture analysis method, we compute image intensity contrast, energy, and variance, as well as rock blob area. The first three attributes yield the most consistent indication of the rock texture formation. The last attribute helps with the physical size classification of the rock. The image texture attributes are used as input to the FLTC. Figure 3 presents nine cases by which we geometrically characterize the shape formation of the rocks. Figure 4 presents the fuzzy membership functions corresponding to these image texture attributes. The fuzzy inference engine of FLTC classifies the rocks based on a certainty confident level. Figure 5 illustrates the fuzzy output membership functions considered for this purpose for classification of rock formations. After verification of the rocks, we apply a geometrical analysis scheme to characterize geometrical formation of each detected rock.

Table 1. Algorithm for Terrain Traversability Path Planning and Assessment

1. *Apply a low-pass filter to reduce noise.*
2. *Apply an Adaptive Binarization to binarize the image.*
3. *Apply a blob detector to isolate rocks bigger than a size threshold level.*
4. *Compute texture attributes of next largest rock blob.*
5. *Apply fuzzy logic rock classifier to determine rock detection certainty.*
6. *If rock detection certainty is above a threshold value, compute center location and principle moments of the rock blob; otherwise continue with step 4.*
7. *Characterize and model the rock geometry and log in its coordinates, orientation, and geometrical configuration and dimension; if more rocks are remaining for analysis, continue with step 4.*
8. *Map location of each rock to the world*
9. *Apply the divide-and-conquer collision free traverse pathfinder as described in section 3 to generate near-optimum traversable trajectory paths for the robot.*
10. *Select a feasible trajectory path with the highest measure of traversability.*

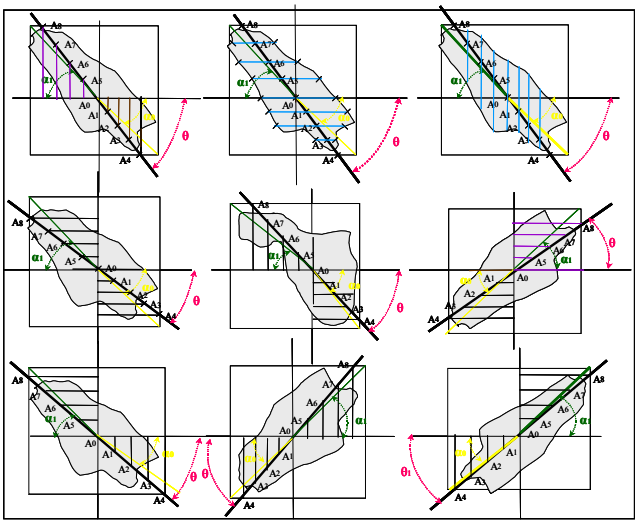


Figure 3. Nine rock formation models and their dimensional measurement probing methods.

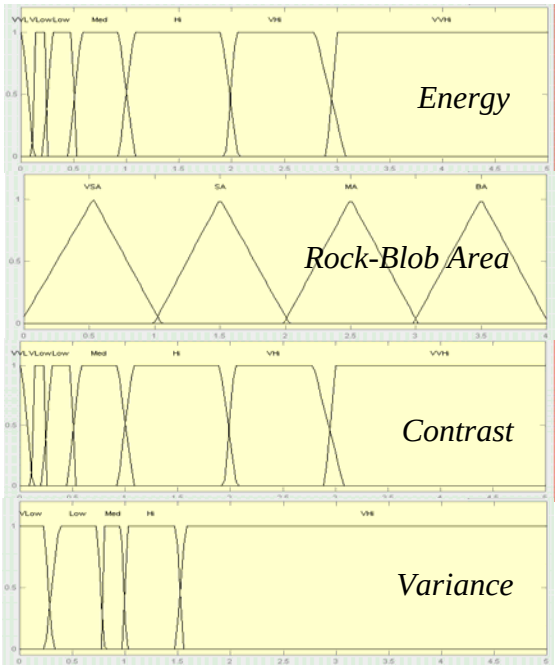


Figure 4. Input Membership Functions of Fuzzy Logic Rock Classifier

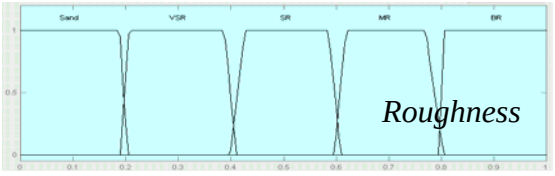


Figure 5. Output Membership Function of Fuzzy Logic Rock Classifier

Of a particular importance is boundary formation of each rock around its first moment axis and mass distribution of the rock around its first moment. Two factors that control the shape appearance of the rock. We model the shape of rocks in our virtual simulation environment using either hemi-spherical or ellipsoidal geometrical representations. We use hemi-spherical rock representation to model that class of rocks appeared to have square dimensional proportionally. For the rocks appeared to have rectangular configuration, however, we use one of the nine models depicted in Figure 3. We construct an ellipsoidal object using a lofting technique applied in CAD software for construction of solid objects. A loft is result of surface formation achieved by connecting contour of several parallel construction polygons together. To model each rock as a loft, we use the first principle moment axis of the rock and create eight spaced, parallel, semi-circles construction polygons with varying diameters proportional to circumferential diameter of the rock in normal direction to its first principle moment axis. Depending on the orientation of the rock's first principle moment axis, the algorithm selects an appropriate rock model representation. The circumferential diameter of the rock is measured in incremental distance along longitudinal direction of the rock's first principle moment axis. Line segments A0 through A8 shown for each rock model in Figure 3 illustrates position and direction of measurement of the rock's circumferential diameters. Measurement is accomplished by image probing along the line segments A_i's in the binary image. See Figure 6b. In the final step of this algorithm the path planning technique described in section 3 is applied to generate a near optimum free path for the robot. Figure 6 presents terrain modeling and geo-location distribution of rocky surface of a Martian Terrain.

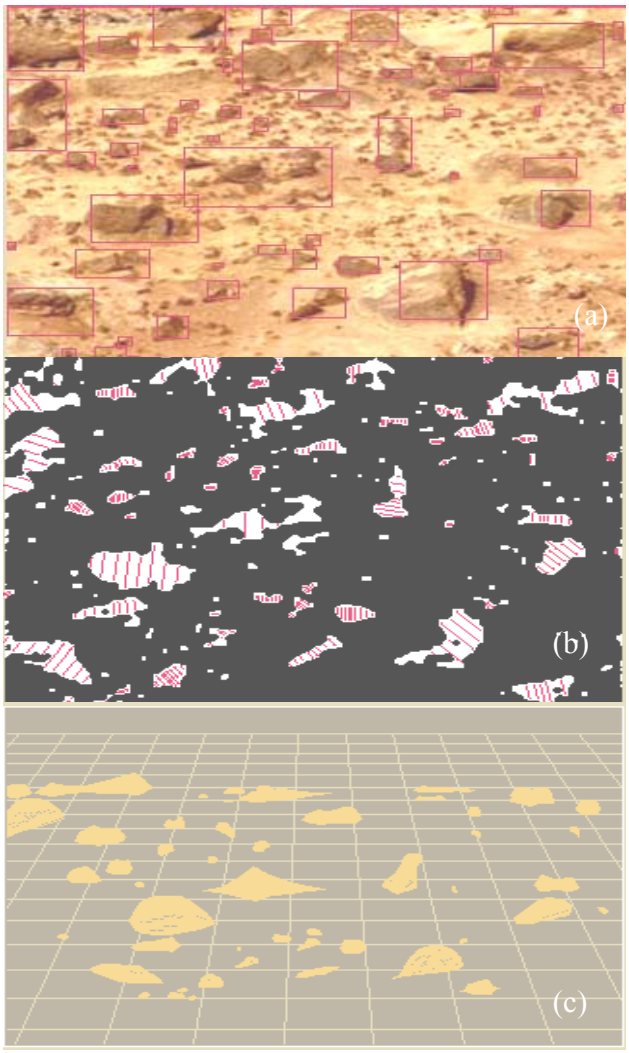


Figure 6. (a) An Image of Mars Terrain, (b) Localized Rock Blobs (c) Modeled Terrain

5. PERFORMANCE MEASUREMENT

The proposed terrain traversability assessment method was tested for assessment of several natural terrains. We chose several B/W and Color images taken by Pioneer and Opportunity Mars rovers from Mars terrains. The images were in size 320x240 pixels. We chose square sub-window frames of size 20x20 pixels for terrain assessment sampling. Figure 7 and 8 shows the results analysis of four different types of Marian terrains. The six terrain classifications are performed for each terrain. The classes of sub-terrain conditions that were classified were: (S) sandy, (V) very small rock, (m) rough, (M) very rough, (B) big rocks, and (U) uncertain. The algorithm classifies those areas of terrain that are unknown to its internal fuzzy logic model as "uncertain". Another situation that algorithm classified terrain as uncertain is at the border near to horizon line. Due to loss in depth of view, objects appearing near to horizon line cannot be uniquely differentiated and classified properly. As shown in Figure 7b, the terrain assessment map shows uncertainty about the terrain context at two rows near to horizon line. Depending on pitch angle of the camera, the perception of terrain weakens in the image frame as we move from bottom of the image toward to top. This problem is more severe during route planning phases where the camera is looking directly straight toward the horizon with pitch angle near to zero. However, this problem tends to be troublesome for navigational planning phases (since the camera is pitched more toward the ground and terrain content appears manifestly in the image frame). Notice this observation in Figure 7c. The Terrain Traversability Assessment Measure (TTAM) that we used to evaluate the performance of the fuzzy logic terrain classifier was:

$$TTAM = \frac{\sum_j \sum_i w_{ij} . M_{ij}}{\sum_j \sum_i w_{ij}} \text{ x } 100\% \qquad (5)$$

Where, M_{ij} is either 1 when terrain is classified correctly, otherwise 0. w_{ij} is a weight factor that reflects value criticality of terrain classification. It may be assumed flat unity for all different classes of terrain condition, or assigned a range of values based on a degree of belief one may have on reliability of classification of the system. We chose the latter alternative. We selected the range of values of w_{ij} according to the following scheme:

Sandy Terrain:	1.0	Very Rough Terrain:	1.3
Very Small Rocks:	1.0	Rocky Terrain:	1.5
Rough Terrain:	1.2		

5. EXPERIMENTAL SYSTEM SETUP

To experimentally verifying the proposed rock detection techniques and navigational path planning technique, we constructed a terrain mock up similar to Planet Mars surface. Figure 9 presents the physical Mars mock up test bed. The test bed provides a sand pile in size of 16 x 12 feet and rocks of various size and shapes. Our laboratory has several all-terrain robotic platforms. Our robots are equipped with laser scanner, sonar, and stereovision cameras. Many smaller rocks cannot be detected using either laser scanner nor the sonar sensor suit located around the robot since the elevation of the sensors is too high related to the height of the smaller rocks we have. This necessitated usage of camera to visually inspect the rough terrain during navigation.

As a part of this original investigation, we did not employ stereovision capability of our robot, though it is possible to fuse visual and stereovision results to achieve more robust analysis of the terrain [4]. Moving on soft sand pile creates not a major problem for our all-terrain robots. Our robots are relatively heavy and compacts the soil under theirs wheels that results good traction for the robot. They also benefit from all-terrain tire rims that provide a sturdy traction with the ground. However, traction losses become on steep slopes. Due to the later problem, dead-reckoning cannot be respected and visual localization relative to landmarks is a better method for localizing the robot. We are currently implementing the algorithms on our robot supervisory controller called CORMI. CORMI stands for Cooperative Robotic Man-Machine Interface that is developed using FMCell robotic modeling and control software. FMCell provides an intuitive man-machine interface tools for control, sensor and image processing of cooperative robotic systems. Our reference [11] provides a detail description of our Cooperative Robotic Man-Machine Interface (CORMI).

6.

Figure 8. (a) A Mars Terrain, (b) Fuzzy Logic Terrain Assessment Results. The Designation Are: (S) Sandy, (V) Very Small Rocks, (m) Rough Terrain, (M) Very Rough Terrain, (B) Big Rock, (U) Uncertain, (c) Another Mars Terrain, (d) Fuzzy Logic Terrain Assessment Results. Figures (b) and (d) are colored coded.

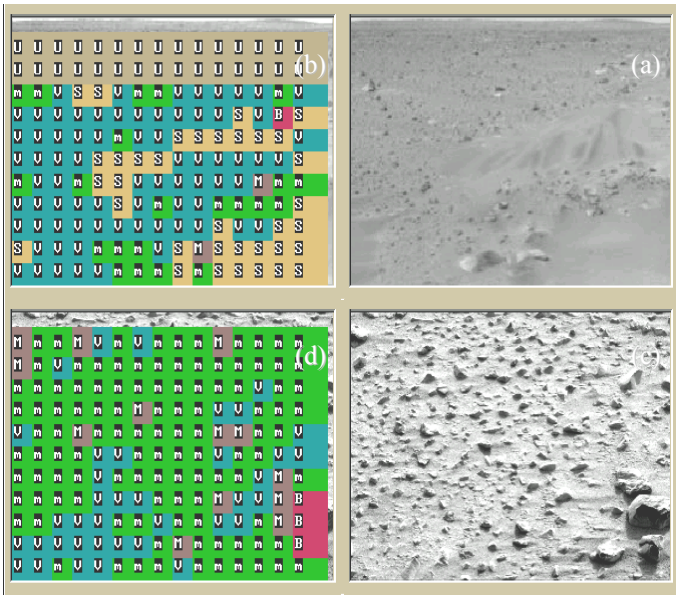
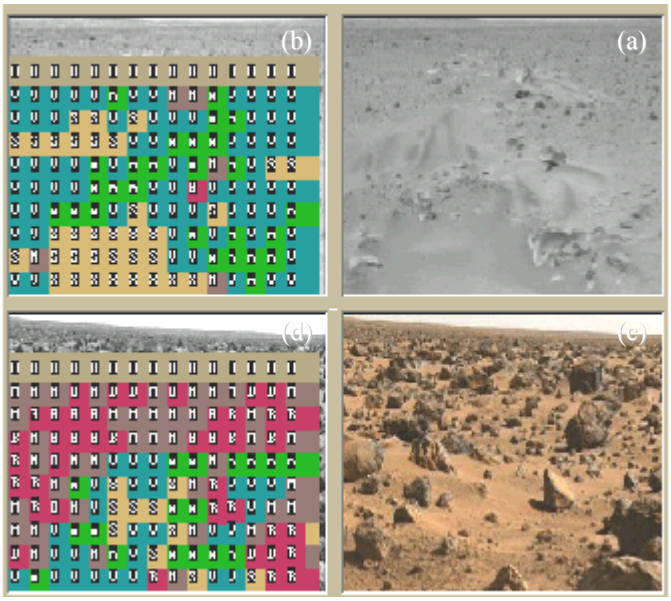


Figure 7. (a) A Mars Terrain, (b) Fuzzy Logic Terrain Assessment Results. The Designation Are: (S) Sandy, (V) Very Small Rocks, (m) Rough Terrain, (M) Very Rough Terrain, (B) Big Rock, (U) Uncertain, (c) Another Mars Terrain, (d) Fuzzy Logic Terrain Assessment Results. Figure (b) and (d) are colored coded.



CONCLUSION

In this paper, we have discussed some of the navigational challenges associated with path planning of mobile robots in natural terrain navigation traversability assessment and presented a method for route planning and navigation planning in rough terrain. The proposed method applies imagery techniques for detection and localization of terrain salient features. For route planning, we have proposed a divide-and-conquer near-optimum path planner method that is both fast and robust in generating collision free trajectory path plans. We have also presented a method for terrain traversability assessment and characterization based on a fuzzy logic approach. Both approaches are fast and can be readily implement onboard mobile robot to assist the robot with terrain path planning and traversability assessment online.

7. ACKNOWLEDGEMENT

This research was conducted under NASA Faculty Award for Research grant under contract number 1248358. The support from NASA is acknowledged. Furthermore, we acknowledge Dr. Edward Tunstel, Senior member of Engineering staff at JPL for technical monitoring of this research project and providing invaluable recommendations for enhancement of this research effort.

Figure 9. Mars Mockup Arrange at Intelligent Tactical Mobility Research Laboratory at TSU.

REFERENCES

[1] Howard A., Tunstel E., Edwards D., and Carlson A., "Enhancing Fuzzy Navigation Systems for Mimicking Human Visual Perception of Natural Terrain Traversability," in the Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf., pp. 7-12, Vancouver, B.C., Canada, July 2001.

[2] Kubota T., Nakatani T., and Yoshimitsu T., "Path Planning for Planetary Rover Based on Traversability Probability," In 7th Int. Conf. on Advanced Robotics, Sant Feliu (Spain), Sept. 1995.

[3] Cherif M., and Laugier C., "Dealing With Vehicle/Terrain Interaction When Planning the Motions of a Rover," in the IEEE International Conference on Intelligent Robots and Systems, Munich (Germany), Sept. 1994.

[4] Olson, C. F., Matthies, L. H., Wright J. R., and Li R., and Di K., "Visual Terrain Mapping for Mars Exploration," IEEE AC paper #1176, 2003.

[5] Golda D., Iagnemma K., and Dubowsky S., "Probabilistic Modeling and Analysis of High-Speed Rough-Terrain Mobile Robots," in the Proc. of the 2004 IEEE Int. Conf. on Robotic and Automation, New Orleans, LA, April 2004.

[6] Iagnemma, K., Golda D., Spenko, M., and Dubowsky S., "Experimental Study of High-speed Rough-terrain Mobile Robot Models for Reactive Behaviors," Proc. of the Int. Symp., on Experimental Robotics, Italy, 2002.

[7] Sirisha R., and Shirkhodaie A., "Visual-Servoing and Navigation of Mobile Robots for Targeting Applications With Laser Sensory Data Fusion," IEEE SSST 33rd Southeastern Symposium on System Theory, March 18-20 2001.

[8] Shirkhodaie A., "Second Annual Research Progress Report," submitted to JPL, Tennessee State University, Department of Mechanical and Manufacturing Engineering, July 2004.

[9] Shirkhodaie, A., "Visual Telerobotic Task Planning of Cooperative Mobile Robots," 1st NASA/JPL Project Progress Technical Report, July 2003, Tennessee State University, Department of Mechanical and Manufacturing Engineering.

[10] Shirkhodaie, A., "Visual Telerobotic Task Planning of Cooperative Mobile Robots," 1st NASA/JPL Project Progress Technical Report, July 2003, Tennessee State University, Department of Mechanical and Manufacturing Engineering.

[11] Devgan, T. P., "Man-Machine Interface for Teleoperation And World Perception Modeling of Cooperative Mobile Robots," M.S. Thesis, Department of Mechanical and Manufacturing Engineering, Tennessee State University, August 2003.

Introduction to Issues in Developing Adaptive Intelligence

Gary Berg-Cross

What are the issues to developing and measuring the intelligence of adaptive systems, how are they defined and what is the background to the issues? Given these questions it seemed to make a sense to present a top-down introduction to this in order to facilitate discussion such as we might have in a panel.

A major context for a track on developing adaptive intelligence includes the foundational assumptions laid out by Albus (1999) in "The Engineering of Mind" which lays out a reference model architecture that serves as a principle vein of work on the design of intelligent systems. Jim Albus (1999) is based on functional, control and goal-oriented pragmatic principles using 2 assumptions. His first assumption is that the function of the mind/brain is to generate and control intelligent behavior. A second assumption is that intelligent behavior is understood as "appropriate action" in an uncertain environment where appropriate action is understood as action which increases the probability of success in achieving high priority goals. While Albus' approach includes pragmatic concepts it grows out of control theory and what Haugeland calls GOFAI, Good Old Fashion AI" . To this meta model Albus adds a specific architecture for intelligent behavior as the result of goals and plans interacting at many hierarchical levels with knowledge represented in a multi-resolutional world model (discussed at greater length in Albus and Meystel, 2002). A general theoretical reference model for mind is axiomatically defined by Albus using 5 key concepts and their assumptions enables the approach. Briefly, these are function components, structural storage, interconnected computational modules in a control system architecture, hierarchical layering of components and attentional selection. These are depicted in Figure 1 and detailed as follows:

1. The functional elements of an intelligent system are: behavior generation, sensory perception (filter, detect, recognize, interpret), world modeling (to store knowledge, predict and simulate future), and value judgment (to compute cost, benefit, and uncertainty attributes). These elements are detailed by sub-activities. As an example behavior generation is based on planning and control of actions designed to achieve behavioral goals. The planning a process itself is complex and:

    1. assigns responsibility to agents/ computational elements for jobs, and allocates resources to agents to perform assigned jobs,
    2. hypothesizes strings of actions (plans) for agents from a "vocabulary" of possible actions to accomplish jobs,
    3. simulates and predicts the results of executing these hypothesized plans,
    4. evaluates the predicted results of the hypothesized plans,
    5. selects the hypothesized plan with the most favorable results for execution.

In general Albus' functional elements, while computational in principle, evoke many appealing cognitive concepts such as meaningful transformations, risk, benefit, importance, advantageous goals etc. Biological correlates to functions are noted (but since this is a moving target, may need to be revised over time).

2. These functional elements of an intelligent system are supported by a knowledge database that stores (both long-term and short-term) a priori and dynamic information about the world in the form of state variables to record state-of-the-world, symbolic entities (as in GOFAI), symbolic events, rules and equations, structural and dynamic models, task knowledge, signals, images, and maps. It is worth noting in passing that Joslyn (2000) understands Albus's overall approach to be one of semiotic control system using models, such as stored in a

knowledge base providing the special property of intelligence and combined with control systems to purposely organize functions such as listed above.

3. In the reference architecture the functional elements and knowledge database described above can be implemented by a set of computational modules that are interconnected to make up nodes in a control system architecture. Exemplars of such nodes are found in the Real-time Control System (RCS) built at NIST. Each node is part of a control system that performs the 4 basic processes -sensing, maintains a world model, computing values, and generating behavior and their supporting sub-processes (planning, task decomposition etc.). To Albus a node corresponds to a functional set of brain neurons closing the loop between afferent and efferent neural pathways. Interestingly conceptually this architecture does not directly address components that do not lop between afferent and efferent parts and thus makes them a secondary sub-component.

4. The complexity inherent in intelligent systems can be managed through hierarchical layering, which is a common method for organizing complex systems that has been used in many different types of organizations throughout history for effectiveness and efficiency of command and control. A key to hierarchical control is that higher level nodes have broader scope and longer time horizons, with less concern for detail, while lower level nodes have narrower scope and shorter time horizons, with more focus on detail. Elements may be deliberative or reflexive throughout the hierarchy

5. The complexity of the real world environment can be managed through a strategic process of focusing attention. Since the world is not uniformly important this serves the needs of an agent with limited computational resources.
Attention focuses sensors on task-relevant objects.

Taken as a whole Albus believes that work based on this reference model helps advance the scientific inquiry into of the nature of mind, and also will very likely also lead to practical improvements in intelligent machine systems technology for many fields. A number of practical achievements can be pointed to, but it may be useful to relook at the overall architecture, its assumptions and details.

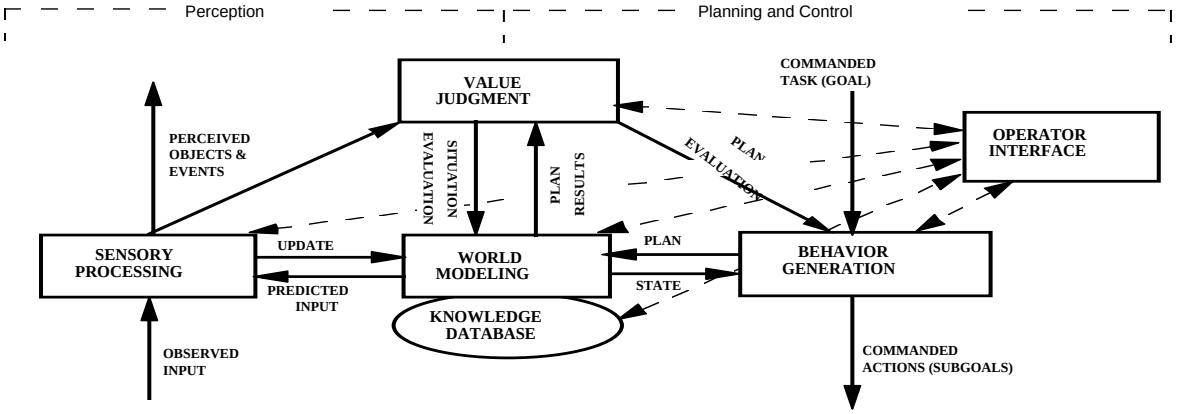


Figure 1 RELATIONSHIPS BETWEEN ALBUS' FUNCTIONAL ELEMENTS

A context for this discussion is a section from last year's PerMIS conference (Machine Intelligence: Measures & Issues II) which raised some issues relevant to the architecture and its assumptions. These included:

a. Going beyond sequential, functional, hierarchical architectural ideas to include how intentions and adaptive abilities emerge to provide the roots of "intelligent" agent actions (Berg-Cross, 2003).
b. How to understand intelligence performance when we consider how natural agents (individual humans and animals) readily adapt to rapidly changing and/or complex environments including rare, ambiguous, and unexpected events. An example of how might artificial intelligence might adapt was presented by Gunderson (2003) considering a robotic intelligence that has a robust ability to "develop" hypotheses to solve problems and thus starts chasing the family dog to more effectively (intelligently) perform housekeeping.
c. Moving towards "natural intelligence" might intelligent information-processing bridge between differently-scaled models by means of cross-scalar coherence and autonomy-negotiation to create a hyperscalar system (Cottam, 2003).
d. How valuable are functions of creativity and playfulness in intelligent systems when considered as part of an adaptive learning and developmental process? Is there a common architecture that includes interaction, adaptation, innovation and "immunity" (Arata, 2003).
e. Strong AI vs. Weak Ai asked the question on how important specific models found in knowledge bases versus general and abstract know. But much of the human knowledge seems inconsistent, loosely organized, and in perpetual flux rather than faithful copies of reality found in implementations of both strong or weak AI. The mind may neither be a highly organized knowledge base nor a large set of fuzzy images. A concept developed by John Sowa is *knowledge soup*. Knowledge is fluid, lumpy, with adherable chunks of Theories, Models and Hypotheses that float in and out of awareness. Is it useful to of considering agent knowledge as a fluid rather than precise system of "true" facts (Berg-Cross, 2003).

Taken together these question some of the foundational assumptions laid out by Albus in "The Engineering of Mind" and suggests modifications if not major overhaul to reference model architecture. We agree that the scope to resolve these issues will be broad requiring understanding and cooperative work from:

1. Neurosciences
2. Cognitive Sciences and cognitive information processing
3. Artificial Intelligence
4. Learning and Complex Adaptive Systems
5. Intelligent Control
6. Robotics and Intelligent Machines
7. Intelligent Manufacturing Systems
8. Game Theory and Operations Research
9. Image Understanding
10. Planning and Reasoning
11. Philosophy
12. Linguistics and Speech Understanding

As an example, much of neuroscience models (for example perceptual cognition) are network rather than hierarchical models. While it is good for engineering to ruese the same functional the basic "unit" of neural computation does not seem to share the same processing components as we move from one level to another.

Berg-Cross (2003) notes separate, but fundamentally different functioning to support the high-level belief-intention parts of intelligent behavior and, at least in human architectures, these seem significantly different from the lover level functions. This is related to the argument of how easy or practical it currently is to engineer intelligent systems (hierarchy simplifies the engineering) as opposed to letting them develop.

Another issue concerns the "meaning" and measurement of autonomy of a system that is learning, modifying its knowledge and adapting its behavior. Understanding the interaction of these components over time is a central part of understanding how an agent arrives at "appropriate action" to increase the probability of success in achieving high priority goals. How important is self-organization and development?

Finally it is worth noting that some "general principles" for complex systems may be useful to discuss for intelligence. The phenomena of pattern formation and self-organization found in nonequilibrium physical, chemical and biological systems may be governed by a number of general principles. This idea, arising in the natural sciences study of complex systems structureal development, has now been applied to discussion of the development of intelligence. It is a new way to bridge the gap between what individual element function, such as RCS nodes, and what many of them do when they adapt/learn/evolve to function "cooperatively" .

An issue then is how GOFAI and engineering approaches might be supplemented by research strategies and computational tools growing out of nonlinear dynamical systems. Numerical and analytical investigations of these systems lead to new mathematical results and problems, as well as to formal bridges to other biological and physical systems, notably dissipative systems that describe aspects of self-organization and nonequilibrium behavior. These formal investigations have already suggested new designs for computer vision, adaptive pattern recognition machines, and autonomous robots, and as an integrative approach may provided basic science with designs of adaptive intelligent systems. As an example, in the study of complex systems, especially natural ones, the idea of emergence is used to explain development of patterns, structures, and/or properties that do not seem reducable to a system's existing components and their linear interaction. Emergence becomes of increasing important as explanatory constructs for complex systems characterized when:

- Global, system organization appears to be of a different kind than functional components alone;
- Components can be changed or even removed without an accompanying loss of function of the higher functions of a system.

In concluding, adaptation and learning are seen as one of the central points of such IS research, but diverse approaches may be meaningfully integrated into thinking about the future of Intelligence Systems. We advocate an integrative approach - let us not be stymied about differences and together keep doing interesting things. In this panel, and in the track as a whole we hope to discuss core ideas and explore approaches that will possibly shape future work in the area of adaptive intelligence.

References

Albus, James.S. (1991). "Outline for a Theory of Intelligence". *IEEE Transactions on Systems, Man and Cybernetics*, **2 1** (3): p. 473-509

Albus, James S: (1999) "*The Engineering of Mind*", Information Sciences,v.117, pp. 1-18

Arata, L., **Interactive Measures and Innovation**, presented at PerMIS 2003.

Berg-Cross, G., **A Pragmatic Approach to Discussing Intelligence in Systems**, presented at PerMIS 2003.

Cottam, R., Ranson, W., Vounckx, R., **Abstract or Die: Life, Artificial Life and (v)Organisms**, presented at PerMIS 2003.

Gunderson, J., Gunderson, L., **Mom! The Vacuum Cleaner is Chasing the Dog Again!**, presented at PerMIS 2003.

Joslyn, Cliff: (2000) "**Towards Measures of Intelligence Based on Semiotic Control**", *2000 Workshop on Performance Metrics in Intelligent Systems*, ed. A. Meystel, NIST

Meystel, Alexander M and Albus **,** James**, Intelligent Systems: Architecture, Design, and Control** 2001

Developing Rational-Empirical Views of Intelligent Adaptive Behavior

Gary Berg-Cross, Knowledge Strategies
Potomac, Maryland 20854, USA

Abstract

A developmental perspective is useful to understand how intelligent human behavior comes to be performed because it combines insight of evolutionary factors that enable dynamic genetic-environmental interactions within individual humans. Such developmental adaptations may now be studied experimentally using developmental and epigenetic robots. Resulting insights is a useful step toward more complete, valid understanding of intelligent behavior, its adaptive nature and it structural roots. Taken together these broaden the concept of "engineering mind" to include the larger concept of "development". This paper overviews recent work of evolutionary and developmental psychology, epigenetic robots and cognitive science. A synthesis of these suggest means by which the fluid nature of adaptive knowledge arises developmentally within a heterogeneous architecture adapted for adaptation itself as part of a rational-empirical process. At this top-level of intelligence, situation-specific adaptive functions are processed using a dynamic mix of belief-based, rational-empirical cognitive processes and socialized methods adjusted within human cultures. General research goals of such an integrated, consilient view of intelligence are outlined for future research.

1. Introduction

In "The Engineering of Mind" (Albus, 1999) cited a range of research which, taken together, provided new insights into the nature of mind. These included work on learning & language, image understanding, rule based reasoning & planning, advances in real-time sensory processing using world modeling and data fusion, intelligent hierarchical controls, and effective use of parallel processing on focused attention as well as general progress in the cognitive/ neurosciences. Albus (1999) argued that what was lacking was synthesis of a "general theoretical model which ties all these separate areas into a unified framework that includes both biological and machine embodiments of the components of mind". Building on earlier work (Albus 1991) structured a reference model using widely discussed concepts of intelligence and mind – controlled behavior generation, via underlying knowledge bases and world models choosing goal appropriate action in uncertain environments. Berg-Cross (2002, 2003) noted that much of this underlying reference model represents a mid-level, plan based on a symbolic information processing view of intelligence which is amenable to hierarchical modeling and engineering, but which may not capture the full complexity of intelligence. For one thing, Albus reference architecture of mind is organized around a small number of general purpose mechanisms that are assumed to be content-independent. The functional modules have a variety of names such as "learning," "induction," "cognition," "world modeling," "or "sensor fusion". In Albus's information processing views these are organized around the world model, which combines with the other functional mechanisms to explain how an agent acquires a language or learns to navigate city driving.

Another way to view such phenomena is as a result of a rational process. Anderson (1990), for example, processes identifies 6 steps developing a theory of rational agents:

(1) Goals: specify precisely the goals of the cognitive system.
(2) Environment: develop a formal model of the environment to which the system is adapted.
(3) Computational limitations: make minimal assumptions about computational limitations.
(4) Optimization: derive the optimal behavior function, given 1–3 above.
(5) Data: examine the empirical evidence to see whether the predictions of the behavior function are confirmed.
(6) Iteration: repeat, iteratively refining the theory.

This paper addresses some aspects of an analysis of steps 1 and 2 recognizing the evolutionary roots of some goals and the essential importance of adaptation within the environment.

2. Unifying Views

As previously noted Berg-Cross's (2002) proposed a fundamentally different reference architecture than Albus (1999). A portion of this 3 part architecture is shown in the left portion of Figure 1. While not strictly hierarchical it is easier to introduce it in top, middle and bottom terms. There is an upper level for belief-intention driven pragmatic processes which provides intentional control and strategic guidance. In a lower level there are situated-reactive processes which also pass factual input to intermediate planning level. This architecture is beyond the traditional scope of engineering which attempts to decompose functions using mechanistic principles, but does reflect some pragmatic elements of the view of mind found embedded in Albus [1](1991). In particular the reference model provided a pragmatic view, shown in the middle portion of Figure 1, to the ubiquitous cognitive idea of a "rational agent' using 'knowledge' to 'succeed' in the world. Berg-Cross (2003) identified two key aspects that structure the new reference architecture:

[1] "Intelligence is a faculty of the system that provides an ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral sub-goals that support the system's ultimate goal."

- Unification of a belief-based, rational-empirical view of intelligence, reasoning & knowledge
  - for example, propositional representations are not primary knowledge but an ingredient to which becomes knowledge via interpretation. Thus there is a relation between propositions and the system of beliefs that cognitive agents know/use
- Characterization of agent knowledge as a fluid, inconsistent mix of approximate facts, hypotheses, goals and concepts that serve in a rational-empirical process (Sowa, 1999).

The current paper addresses a developmental perspective to further explain how pragmatic foundation for these constructs come into existence. It finds features of rational-adaptive nature of intelligence embedded in biological systems. This shown in the right side of Figure 1 which provides a very high-level view of the evolutionary and developmental influences to Sowa's (1999) rational-empirical knowledge cycle. In particular these help address the questions of primitives of cognitive processes within a rational-empirical cycle and how they might arise developmentally. These are not new questions, but discussion in the context of aspects of the reference architecture is. Both an evolutionary and epigenetic approach is introduced to provide a richer, plausible basis for fluid, adaptive knowledge structures and processes. Overlaying this are additional social processes which further shape adaptive structures and knowledge.

Discussion of the developmental nature of intelligence is in the spirit of unifying scientific approaches (Wilson, 1998) for biological, psychological and social science. A synthesis of these can enhance a strictly engineering view of mind. One initial goal is to add a biological frame to the information processing model and outline our understanding of intentions and beliefs that co-exist with rational and underlying reactive processes. A key idea uniting these is the adaptive nature of human intelligence and resulting human behavior which may be usefully viewed and discussed from 3 different points of view as a:

1. species evolutionary product (see Caporael, 2001 for an overview of Evolutionary Psychology - a distal explanation operating over evolutionary time),
2. result of broad epigenetic-environmental interactions within the lifespan development of individual humans/agents (such as discussed in Piaget 1950) and
3. bio-ecological phenomena based on situation-specific functions processed by a rational-empirical cognitive apparatus that is socialized and adjusted within human cultures (Bronfenbrenner & Morris,1998).

Each provides a useful, somewhat complementary, step to a more complete and valid understanding of intelligent behavior, its adaptive nature[2] and it structural roots. Taken together these broaden the concept of "engineering mind"

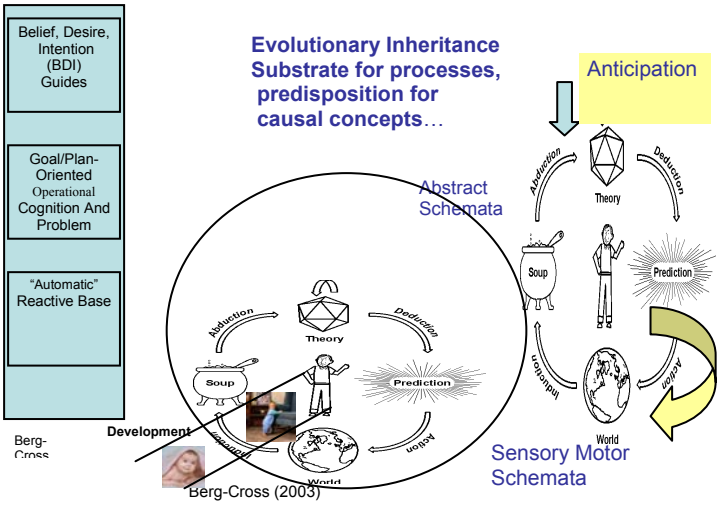


Figure 1: Three Level Hybrid architecture, Sowa's Rational-Empirical Knowledge Cycle, Evolutionary process articulated by Piagetian developmental processes

(Albus, 1991, 1999) to include the larger concepts of evolution, "development" and social influences on this development and to help understand the complexity of the issues.

3. Evolutionary Views and Levels of Explanation

The evolutionary perspective provides a biological view of our complex, characteristic intelligence in the form of a, neural system adaptive by evolutionary challenges. Thus, in an evolutionary perspective, such as posited by Evolutionary Psychology (EP), the human mind resides in a brain that is "a system of organs of computation designed by natural selection to solve the kinds of problems our ancestors faced in their foraging way of life" (Pinker 2000). Biology and thinking follow the same evolutionary path since long-term evolution has shaped all human minds as it has human bodies, around fitness structures and functions suited for hunter-gatherers. Current evolutionary neuroscience perspective on brain reorganization (left-right cerebral hemispheric asymmetries, human-like third inferior frontal convolution etc.) in light of brain imaging studies suggests the story that Homo intelligence arouse from a dynamically configured set of brain areas that collaborate adaptively to meet particular cognitive challenges. Evolutionary biologists (EB) do a form of reverse

[2] Following Pfeifer & Scheier (1998) adaptivity may be understood as an organisms' capability to modify its behavior so that they can more efficiently maintain their critical parameters in a "zone of viability". The pathway to this includes all of species, group and individual development. Piaget includes such ideas within his equilibration process.

engineering[3] when they hypothesize adaptive problems a species might have encountered during its evolutionary history, and then ask themselves, "What would a machine capable of solving these problems well under ancestral conditions look like?" Using this insight, EBs empirically explore the design features of the evolved "machines" that, taken together, comprise an organism. This is a one form of evolutionary explanation, but its goal is a more integrated explanation at several levels. A view of the explanatory role of evolutionary principles in the context of other ideas (e.g. brain structure) is represented in Figure 2 based on a framework found in the work of Cosmides and Tooby (1987). Their work uses an evolutionary focus to propose an outline the human mind's "design". Knowing what cognitive programs are posited to come into existence evolutionarily, in turn, guides the search for neural explanations as shown in the Figure. Taken together this theoretical outline of adaptive problems helps guide the integrated search for the cognitive programs that solve adaptive problems as part of integration of explanation. Wilson (1999) calls such integration across scientific disciplines consilience, a common ground of explanation that links scientific systems of thought. Belief in the possibility of consilience is a meta-assumption because systems of explanation do not spontaneously unify. Integration requires that a common body of abstract principles and related evidence be constricted and EP/EB suggests one such path for evolved intelligent agents.

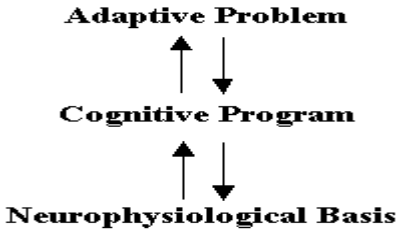


Figure 2: Three "complementary" levels of explanation in evolutionary psychology. Explanation at one level (e.g., adaptive function) does not preclude or invalidate explanations at another (e.g., neural, cognitive, social, cultural, economic). Constrained, plausible inferences between level s are represented by the arrows –after Cosmides and Tooby (1987).

Such definitions of adaptive problems may not uniquely specify the design of the mechanisms that solve them, because as a rule, there are multiple ways of achieving any solution. As result empirical, psychological studies are needed to focus in on which mechanism nature may have actually adopted. The engineering principle here is that the more precisely researchers define an adaptive information-processing problem -- the "goal" of processing -- the more clearly one can see what a mechanism capable of producing that solution would have to look like. This research strategy has been used in the study of vision, for example, so that it is now commonplace to think of the visual system as a collection of functionally integrated computational devices, each specialized for solving a different problem in scene analysis -- judging depth, detecting motion, analyzing shape from shading, and so on. This differs from the type of engineered "mind" described by IP models which are essentially unspecialized with general-purpose mechanisms that subjects all information to the same processes and stores it in the same places. Faces, words, geometric shapes, most other sensory information will be perceived, processed, rehearsed, and stored in the same fashion. An evolutionary perspective brings the unsettling notion that the evolution of cognition has produced memory/knowledge systems that specialize in the processing of particular types of information. It is well known that human perceptions and reasoning are biased - involving anchoring and stereotyping, and parceling out reality according to the limitations of experience and that they tend to focus on what is considered important by the person (see Dawes 1988 for a summary). This perspective provides one useful idea of the varied and inconsistent basis of an individual's knowledge – it is based on a mix of specialized modules. Seen at a distance the result is an intelligence that includes fluid, heterogeneous, ever changing, and often in-consistent knowledge leading to its characterization by Sowa (2000) as a knowledge soup. The soup metaphor serves to capture the idea of fluid material with solid chunks; corres-ponding to the pieces of alike kind of knowledge that have a type of internal consistency. Between chunks there may be inconsistency arising from various sources. This seems to be the case for the human visual system, for example. It is still early to say that a research program using this frame has yielded firm answers to many of the issues of intelligence. A number of issues revolve around the Tooby & Cosmides (1992) 6 meta-theory principles of adaptationism that may provide a coherent view[4]:

1. functional efficiency criteria for identifying adaptations shaped by natural selection,
2. the context-sensitive psychological adaptation (rather than the "instinctive behavior") as the appropriate level of analysis for human nature,
3. a highly modular view of the mind comprised of hundreds of domain-specific psychological adaptations,
4. a computational metaphor for the mind imported from cognitive psychology,
5. the universality of evolved human nature rather than the heritability of individual differences,
6. hominid small-group living in Pleistocene Africa as the most relevant ancestral environment for understanding most of human nature.

[3] Regular engineers figure out what problems they want to solve, and then design machines that are capable of solving these problems in an efficient manner.

[4] The last five of the six remain controversial, even among adaptationists

This meta-model remains controversial and it is not clear, for example, that point 3 is correct. It may be that adaptation has lead human mental abilities to use a "universal structure" or at least generalized/universal functions, which would simplify the measurement of intelligence. At this point we probably can say that the structure of computation-like neural networks has emerged in some form despite short-term influences which to Pinker includes culture, belief, or individual desires. But there is a large, interactive story to describe how these inherited structures serve an adaptive function to become phenotypic structures during development. I think it possible that an empirical-rational process is the general adaptive function. There is much to do to prove this hypthesis.

4. Dynamic Development and the Epigenetic Robotic Approach

It seems useful to consider intelligent performance as part of a dynamic system in which agent architecture/engineering and its environment join forces over time creating agent "development". A consilience-like convergence of data and theory from genetics, embryology, and developmental biology suggests to many the possibility of a more epigenetic, contingent, and dynamic view of how organisms develop (Lickliter & Honeycutt, 2003).

Developmental dynamics, properly formulated, can add to proximal details to the distal view of evolutionary psychology. Putting it simply, development and evolution can usefully be considered 2 views of intertwined phenomena. The standard model is to view mechanisms of evolution as essential to understand development, and the mechanisms of development are likewise essential to understand evolution (Griffiths, & Gray, 1994, 2001). This was changed by such things as Waddington's "Epigenetic Landscape" concept that " the degree to which a trait is innate is the degree to which its developmental outcome is canalized which is defined as the degree to which the developmental process is bound to produce a particular end-state in the face environmental fluctuations both in the development's initial state and during the course of development." Lickliter & Honeycutt (2003) hold the view that development is a self-organizing, probabilistic process in which order and pattern emerge to change as a result of transactions among developmentally relevant resources both internal and external to the organism. It follows that development should not be described as the result of the interaction between genetic and environmental factors, because neither operates as independent causes. In this dynamic view it is more accurate to say that development results from and bidirectional and dynamic transaction of genes, cells, tissues, organs, and organisms during the course of individual ontogeny. Lickliter & Honeycutt's (2003) argue that a study of developmental dynamics could reveal how underlying mechanisms unfold over time and provide insights that are complementary to, not mutually exclusive with, the functional explanations already provided by evolutionary psychology. This has now begun to be studied using "epigenetic robots" designed using Piagetian ideas of development. When applying these ideas to the development of intelligence these formulations parallel some of Piaget's (1950) views on adaptation to the environment. For example, Piaget introduced the term epigenesist to refer to such development, determined primarily by interaction rather than genes. Over the last few years biological thinking has emerged in a "new robotics" partly in response to lack of progress with the information processing paradigm which has proven ill-suited to come to grips with natural, adaptive forms of intelligence (Pfeifer et al 2004). One sub-set of this, called epigenetic robotics, focuses on experimental studies of Piagetian stage-theory processes[5] involving prolonged epigenetic development[6]. The approach is:

1. Start with some 'innate' components/substrate (as previously discussed)
2. Consider the nature and demands of the environment
3. Let development proceeds by an interaction between developing components & a dynamic environment
4. Along the developmental "path" temporary structures and processes may bridge to increasingly more complex cognitive structures (fitter ones) tuned to the environment by interactions with the environment (physical and social)

This developmental path to intelligence provides a substantial set of intermediate knowledge products for an agent. By Piaget's account, the sensori-motor stage in biological systems is a structuring process, where sensing mechanisms are gradually integrated with motor actuating mechanisms on the developmental path to a mature performing system. For human babies this takes lasts 2 years. The first four months organize a substrate of reflexive responses into more coherent motor strategies called physical schemas -scruffy knowledge structures proposed as the basis on which more abstract knowledge is built. In addition, sensory modalities are coordinated and attentional mechanisms begin to emerge – all satisficing environmental constraints and inherited motivators which are realized in a series of intermediate forms on the path to adult structures. From four to six months, reactions are "practiced" until an infant exhibits what seems like intentional

[5] Piaget theory of children's intelligence sees it is as controlled by construction of mental organizations, which he called schemes. These are used to represent the world and designate action. This adaptation is driven by a biological drive to obtain balance between schemes and the environment (equilibration).

[6] One interesting aspect of this process is its underlying ability to deal with and solve complex multi-level integration problems – unsolved challenges in engineering intelligent systems. The epigenetic robotics paradigm proposes that epigenetic development allows systematic exploration of this complex issue.

prolonging of special interactions. Taken together this path evidences the emergence of a belief, desire, intention structure such as described in Berg-Cross (2002). In Piagetian terms stable agent-world interactions patterns emerge that are evidence of satisficing, temporary cognitive structure built in a bottom up fashion. These are initially constructed as physical schemata, but may be used by an agent to handle other instances of this type of interaction. In a fluid model of knowledge, knowledge starts as physical schemata , is extended to other physical examples of a phenomena. These extensions are often tentative and prove less than optimum for situations and are thus highly modifiable as discussed in the Sowa (2000) model of a rational-empirical "cycle". Epigenetic robotic implementations can help clarify, evaluate, and even further such cognitive, developmental theories, which due to the complexity of the interactional processes involved have up to now remained somewhat speculative. An interesting line of psychological work suggests some direction for research into corresponding top-down processes, such as proposed in the 3 part reference architecture of Berg-Cross (2003). Gergely and Csibra's (1998) have proposed a theory of the one-year olds 'naïve theory of rational action' based on evidence for causal and other beliefs such as are hypothesized as belief-desire-intention processes. Infants seem to "comprehend" a goal-oriented aspect of agent behaviour of agents (see Berg-Cross, 1971 for an early experiment into a child's ability to detect intentions). This is also seen in the form and function of so called declarative pointing which is characterized by coordination of an extended arm and index finger intended to draw attention to a distal object. Unlike other pointing, it is not necessarily a request for an object. Instead, children often use declarative pointing to draw attention to objects when they are clearly outside their reach (e.g. the moon or a bird passing overhead). Further, declarative pointing only occurs under specific social conditions. Children do not point unless there is someone to observe their action. Other research on children's attentional responses (using habituation) shows that they take goals into account and anticipate future actions in coordination with these goals. For example, evidence of "teleological" interpretation by a 1 year old can seen in their attributions to an agent desires (e.g. to drink water) as an explanation of an abstract figure's jumping action over an obstacle. Controlled studies also show child beliefs (e.g. a belief that there is water in the bottle). Thus, there is developmental evidence for an early teleological model (see Figure 3) that over time can develop into an interpretive understanding of others as intentional agents (i.e., agents that have mental states such as goals and desires, and whose actions are related to their goals and desires). This is a rational model that helps one agent predict the behaviour of another. The second point leads to the constructive, interpretive aspect of intelligence with evolutionary roots. These concern the origins and development of an understanding of others as social/interactive agents that influence each other, interact with each other, and can have thoughts, emotions and dispositions about/towards each other. Developmentalists have long argued for its central role in child development.

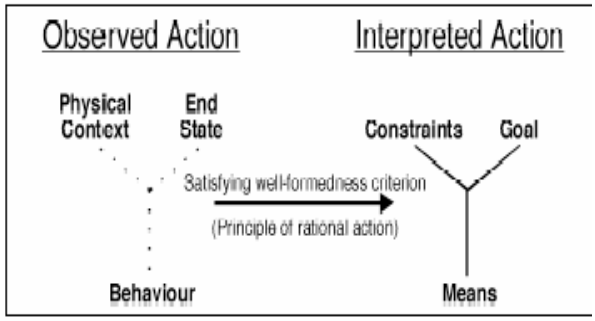


Figure 3 Teleological reasoning in infancy: the infant's naive theory of rational action (after Gergely. and Csibra,1997)

5. Social Roots of Rational Behavior

A standard principle of rationality is that it operates under normal circumstances to optimize the adaptation of the behavior of the organism. A long research line, growing out of Brunswick (see Oaksford and Chater 1996), shows that people behave irrationally with respect to logic, but that the behavior is rational when the context is broadened to encompass a person's '''normal'' life conditions'. One summary idea is that people can be seen to behave rationally with respect to the environment, where their behavior and underling intelligence has been structured and which evolution has provided a developmental substrate, but appear to be operating irrationally with respect to a artificial, experimental tasks. That is, adaptive behavior arises when intelligent-capable agents strive to maintain "ecological balance" of its relationship, both physical and mental, between their environment, material structure, and a degree of pragmatic control.

Tying some of the previous discussion together, a plausible story is that certain biological features found in normal human infancy were selected by a social process, during the stages of human evolution post-dating the invention of tools. These features might have been selected for their facilitative value in the process of what Vygotsky's colleague Leontiev (1981) called 'appropriation'. Infancy is then seen as a specific niche in which adaptive parameters are set by processes of individual appropriation. In this account, the biology of human infancy is a product of the co-evolution of culture and biology. Recent studies of infant cognition and social behavior lend support to such an account. Infancy, in this account, played a crucial role in the 'socialization' of human biology. Thus at any given point an agent has a degree of adaptive intelligence but as a developing agent it has a measurable "zone of proximal task intelligence" (after Vygotsky , 1978). Within this zone agent have not yet master autonomous skill but has a belief and intentions to do with tasks the guidance of humans or more skilled agents. Social explanations of adaptation

problem are detailed by Bronfenbrenner and Morris (1998) who point out how the human individual is an active constituent of any context it inhabits. This way of theorizing emphasizes that social context, including child rearing, is inherently relational and that human life is part of an open system, characterized by indeterminacy and the creation of novelty.

6. Summary

The views presented here provides an expanded view and framework of intelligence in the spirit of consilience. At a very high level, it represents a start on an integrated view of major aspects of human adaptation, including the development of approximate knowledge, the role of evolutionary substrate, intermediate structures and rational processes. A working hypothesis is that the proper use of epigenetic robotics may test developmental dynamic hypotheses and properly will over time add to the theoretical and empirical foundation of adaptive intelligence. Time will tell the merits of these concepts within a joint evolutionary and developmental. Relevant questions include those now attached to both evolutionary psychology and epigenetic robotic research:

- Does the theoretical perspective guide researchers to new and important domains of discovery?
- Does it lead to specific predictions about new, poorly predicted phenomena?
- Does it explain existing scientific findings in a parsimonious manner (better than current theories)?
- Does it yield significant new empirical data?
- Does it provide new insights in the frequently argued issues with adaptation[7]?

References

1. Albus, James.S. (1991). "Outline for a Theory of Intelligence". *IEEE Transactions on Systems, Man and Cybernetics*, **2 1** (3): p. 473-509
2. Albus, James S: (1999) "*The Engineering of Mind*", Information Sciences, v.117, pp. 1-18
3. Anderson, J.R. (1990) *The Adaptive Character Of Thought*, Erlbaum
4. Anderson, J. R. & Milson, R. (1989). Human Memory: An Adaptive Perspective. *Psychological Review*, 96, 703-719.
5. Berg-Cross, L.G. (1971) Intentionality, degree of damage, and moral judgments. *Child Development*, 46, 970-974.
6. Berg-Cross, G. and Price, M. E. (1989). "Acquiring and managing knowledge using a conceptual structures approach: Introduction and framework." IEEE Transactions on Systems, Man and Cybernetics **19**: 513-527.
7. Berg-Cross, Gary (1992) "Can a large knowledge base be built by importing and unifying diverse knowledge? Lessons from scruffy work," *Knowledge-Based Systems*, vol. 5, no. 3, 1992.
8. Berg-Cross, G., A Pragmatic Approach to Discussing Intelligence in Systems, presented at PerMIS 2003.
9. Bronfenbrenner, U., & Morris, P. A. (1998). The ecology of developmental processes. In R. M. Lerner (Ed.), *Handbook of child psychology: Vol. 1. Theoretical models of human development* (5th ed., pp. 993c1028). New York: Wiley.
10. Caporael, Linnda Revolutionary *Psychology: Toward a Unifying Theory and a Hybrid Science,* Annual Review of Psychology February 2001, Vol. 52, pp. 607-628.
11. Cosmides, L. & Tooby, J. (1987). From evolution to behavior: Evolutionary psychology as the missing link. In J. Dupre (Ed.), *The latest on the best: Essays on evolution and optimality.* Cambridge, MA: MIT Press.
12. Dawes, Robyn *Rational Choice in an Uncertain World* (New York: Harcourt Brace Jovanovich College Publishers, 1988.
13. Gergely, G., Nádasdy, Z., Csibra, G., & Bíró, S. (1995). Taking the intentional stance at 12 months of age. *Cognition, 56*, 165-193.
14. Gergely, György (2003) What should a robot learn from an infant? Mechanisms of action interpretation and observational learning in infancy. In Prince, Christopher G. and Berthouze, Luc and Kozima, Hideki and Bullock, Daniel and Stojanov, Georgi and Balkenius, Christian, Eds. *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* 101, pages pp. 13-24, Boston, MA, USA.
15. Griffiths, P. E. and R. D. Gray (1994). Developmental Systems and Evolutionary Explanation. *Journal of Philosophy* XCI (6): 277-304.

[7] A core set of epigenetic research questions amenable to exploration via robotics includes:

- How much is inherited, the nature of motor primitives
- Models of planning and control, perceptual-motor mapping, equivalence of actions.

16. Griffiths. P.E & Gray, R.D (2001) Darwinism and Developmental Systems *Cycles of Contingency: Developmental systems and evolution* Oyama, S, Griffiths. P.E & Gray, R.D Cambridge, Mass., MIT Press: 195-218.

17. Grupen, Roderic A. (2003) A Developmental Organization for Robot Behavior. In Prince, Christopher G. and Berthouze, Luc and Kozima, Hideki and Bullock, Daniel and Stojanov, Georgi and Balkenius, Christian, Eds. *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* 101, pages pp. 25-36, Boston, MA, USA.

18. Leontyev, , A.N. (1981). Problems of the Development of the Mind. (Trans. M. Kopylova). Moscow: Progress Publishers.

19. Lickliter, R., & Honeycutt, H. Developmental dynamics: Toward a biologically plausible evolutionary psychology. *Psychological Bulletin,* 2003 *129,* 819 – 835.

20. Oaksford M, Chater N 1996 Rational explanation of the selection task. *Psychological R iew* 103: 381–91

21. Piaget, J. (1950). *The Psychology of Intelligence*. New York: Harcourt, Brace

22. Pfeifer, R., F. Iida and J. Bongard (2004) New Robotics: Design Principles for Intelligent Systems, *Artificial Life, Special Issue on New Robotics, Evolution and Embodied Cognition*, MIT Press.

23. Sowa, John F. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing, 2000.

24. Tooby, J., & Cosmides, L. (1992). Cognitive adaptations for social exchange. In J. Barkow, L. Cosmides, & J. Tooby (Eds.), *The adapted mind: Evolutionary psychology and the generation of culture* (pp. 163–228). New York: Oxford University Press.

25. Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press. (Original works published in 1930, 1933, and 1935)

26. Wilson, Edward O. *Consilience: The Unity of Knowledge* (Alfred Knopf, 1998)

Interactive Immunity and Adaptive Innovation

Luis O. Arata
Department of Fine Arts, Languages & Philosophy
Quinnipiac University, Hamden, CT 06518
E-mail: luis.arata@quinnipiac.edu

ABSTRACT

Autonomous systems must have the ability to operate on their own in dynamic, uncertain environments without breaking down. This paper presents concepts for the design and evaluation of self-repairing systems that draw on current work on immune mechanisms and artificial immune systems.

To survive in a dynamic environment we say that a constructed system must be able to adapt. The problem is how to implement the adaptive drive. This paper looks into one possible route: to model and implement features inspired by the immune system as a problem solving mechanism. This route does not exclude other adaptive mechanisms and can complement them.

The autonomous system detects malfunctions and tries to fix them on its own, tinkering with all it has at hand. Internal innovation happens when a new type of malfunction is fixed. An interesting aspect of this process is that what causes a new type of malfunction can be the result of the system's interaction with a new environment. Therefore this immune response mechanism functions as an adaptive drive. The system remembers the solution for future use and with quicker response. It has innovated with respect to its previous capabilities, and learns from this action. In this sense the system has adapted.

KEYWORDS: Interaction, innovation, immunity, imperfection, adaptation, evolution, repair mechanisms.

1. INTRODUCTION

A hurdle in the construction of an autonomous system is to find and implement design that would allow them to function in changing environments. The system has to operate according to its tasks and at the same time deal with unexpected situations that happen within its territory, for which it has no preset way to find solutions. If the system begins to malfunction while interacting with a new situation, it should be able to recognize that there is a problem, be involved in diagnosing the problem, and try to implement a solution.

Beyond its ability to carry out tasks, an autonomous system has to change in response to learning from unexpected situations. The system has to innovate at least with respect to itself. It has to be adaptable. This is one measure of its intelligence given the environment in which it is meant to perform.

Our own natural intelligent systems are a source of inspiration for designing autonomy. We appear to have two rather distinct systems tailored for different sorts of tasks. Our brain/nervous system is more for navigation and exploration. Llinás described our feeling of self as a way to help us have a sense of location in the maps of the world we create. This self we create helps us move as safely as possible in our environment [1]. Perhaps the most critical function of our brain is to help manage our body's interactions with the world.

Our immune system's task is more internal. It has to keep the body working well as it moves and interacts in its changing environment. Its function is mostly to manage internal interactions. Not surprisingly, its architecture is quite different from the nervous system. The architectures of these two intelligent systems are based on their tasks. We use both at the same time in a loosely coordinated way.

In this paper I will focus on intelligent features of our immune system that may be applied or emulated in the design of constructed autonomy. Immune mechanisms can be a vast source of inspiration for the design of autonomous systems for dynamic environments. Our natural immune system is an intelligent system in its own right, along with our brain/nervous system. It uses special procedures of its own that we can try to imitate to some extent in the design of artificial intelligence features. As we examine these features, it helps to keep in mind that they are not meant to replace existing architectures. What they could do is increase the robustness of current designs and therefore enable them to be more interactive and useful. We also have to realize that the drawback of such features is that they increase the complexity and cost of the system. The ultimate implementation of immune features would then depend on a balanced approach to the task at hand. This balance includes resources available and cost.

2. MODELS OF IMMUNITY

Alexander Tarkanov has done extensive work in mathematical modeling of immunocomputing. To help conceptualize this field, he first groups the various approaches to the construction of intelligent systems into two directions. One uses generative grammars and logic. The other is based on biological models. In this second approach there are three main directions of research: neural networks, cellular automata, and genetic algorithms [18]. Tarkanov indicates that his work belongs to a fourth direction that is starting to gain momentum: artificial immune systems (AIS), which includes immunocomputing [2].

Classifications come in all shapes and forms. This one helps locate AIS within the many areas of AI. In this paper I will review selected features of our immune system implemented in autonomous systems that operate in changing environments. I will argue that these features of the immune system help it adapt when confronted with new situations. In other words, these features are engines of innovation.

Models of the immune system fall within the range of two main philosophies. One presents immunity as a battle. The immune system detects and seeks to destroy intruders. The second view is that the immune system is an ensemble that seeks to stay in balance. Its response to whatever affects it is a rebalancing that in fact becomes adaptation [3].

In the first view, the immune system is inflexible in its identity. A successful cure leaves the system as it was before the attack, except that now it retains a memory of the aggressor and will be better able to respond to a similar attack in the future. This perspective goes back to the 1950's when immunology became known as the science of the self-nonself discrimination.

Stephanie Forrest and her group at the University of New Mexico developed a system of negative-detection of intruders in computer systems. Algorithms search the computer system for possible changes in specific areas. Then the system tags the change as an intrusion, deals with it, and retains a memory of the event for quicker future response to a similar intrusion. The immediate practical goal of this approach is to give computers a level of self-protection from unexpected viruses and other harmful intrusions. Forrest and Hofmeyr acknowledge that immunology is more than self-nonself discrimination. In a footnote to "Immunology as Information Processing" [4], they note, however, that their approach is an oversimplification. There are harmless intruders that the immune system tolerates, and there are also immune reactions against non-intruding self-cells as in some cancers and autoimmune responses to what should not be perceived as an attack.

The model that Forrest and Hofmeyr followed is non-adaptive. Although the environment can be dynamic and the attacks novel, the negative-selection response is always the same. Nonself is detected and attacked to destroy it. This is a war type model. It hinges on the destruction of the enemy. It is non-adaptive because in the end, there is nothing to adapt to. Nevertheless, the algorithm of negative-selection is considered a milestone in the development of AIS, and rightly so given the enormous complexity of immunity [1].

Tarakanov and Dasgupta used the approach of self-nonself recognition to develop a formal model of an artificial immune system as a highly distributed learning system [6, 1, 7]. They developed a very simplified artificial formal protein that includes torsion factors as a way to model the shape matching that is one tool for identifying non-self intruders. This is their attempt to model the fact that we have tagging proteins that learn by matching shapes with intruders and binding where a match occurs to act as a tag. Tarakanov and Dasgupta then added reproduction and death of the formal proteins to expand their search function. But this mathematical formalism has not yet been tested out. The problem is that it also needs an environment to function and evaluate the performance. In other words, it is not sufficient to formalize the search proteins; the searchable environment has to be also formalized. Or at least a suitable interface has to be developed to interact with the environment. This is still a difficult task to achieve.

Dasgupta and González compared negative selection and positive selection algorithms in the self-nonself approach to intrusion detection in computer networks [8]. They concluded that elimination of what is different is easier to implement and requires fewer resources than checking positive membership in the self-group. The reason is that positive selection is far more memory intensive, at least in the model they developed. The notion of self was not an abstraction but the membership itself. To detect membership in the set, it is necessary to compute the fitness with respect to each member of the self-set and a degree of abnormality is established. There is self-recognition if the abnormality is smaller than a set limit. Negative recognition is done, however, with respect to a single standard, and therefore is easier to implement but if offers less flexibility. It is interesting that in this approach non-self is identified as novelty and eliminated. I will explore how to take advantage of novelty. Rather than eliminate it, it can be used for learning and adaptation to the novelties of a changing environment. In this way a system can innovate.

Melanie Mitchell developed with Douglas Hofstadter an adaptive model based on analogy making [4, 17]. Although it predated the rise of AIS, it has a feature that can be applied to the construction of an immune system, and that is the development of a balance between exploration and attack. She based this on John Holland's concept of a balance between "exploration and exploitation" [5]. She notes that the immune system

exploits information from attacks and allocates resources, but it also continues to explore possibilities that might happen by continuously upgrading an enormous repertoire of cells used in recognizing intruders and initiating immune responses. In an actual response, the immune system tinkers with the intruder using detector or tagging cells until gradually the response becomes increasingly focused. Her constructed system uses random combinations and random mutations to perform such exploration and tinkering. This is in fact a blind search. It is feasible in small domains, but entirely random responses simply would take too long for our immune system to work consistently. Mitchell imagines that the next step is the development of more tailored explorations. What is needed is more preliminary interaction between the unknown organisms and the immune system to reach some sort of cooperative balance. After all, the most successful intruders are those that don't destroy the system, because they would also destroy themselves in the process.

At the opposite end of the immunology spectrum is the systemic model best represented by the initial work of Francisco Varela and then modified in collaboration with Antonio Coutinho. Varela did not follow the self-nonself procedure of identification of foreign agents invading the body. Instead he considered the body a semiotically closed system whose task it to maintain itself in balance. In such a system there are no intrusions, only perturbations to its balance. So the immune reaction is not to a foreign agent as such but to a disturbance. The striking difference between this model and the self-nonself one is that it does not follow the war metaphor. Instead it follows an interactive and cooperative approach. The system reacts to the disturbance and affects not just the agent but also the entire system. This requires the tinkering that Mitchell had noted, and the gradual learning of the situation in search for a solution. In this process, it does not matter what happens to the foreign agent as long as the system regains its balance. In other words, the agent may be incorporated into the system somehow, as long as it does not continue to unbalance it. But now the agent itself may be part of the new balance. In this way, the system adapts and evolves while following its task of maintaining a balance.

I can think of two examples to give a sense of how this systemic approach works. One is the population dynamics in U.S.A. We see that immigration rules act as a regulatory system for the country's semi-permeable borders. Immigrants may or may not enter according to rules, and the systemic reaction is not always one of elimination. Instead, the cultural disturbances of immigrants are transformed into the fabric of the country while at the same time the country readapts to the variations. This leads to the gradual series of adaptations that make the U.S.A. a self-similar nation yet at the same time one that has striking cultural differences from what it was before.

A second example is the Michael Crichton's technofiction novel *The Andromeda Strain*. An unknown strain from outer space has infected a small town and with a destructiveness that surpasses a plague. The zone is isolated and authorities try to figure out what to do to contain the threat, learn about it, and in the end eliminate it. Nothing seems to work. Towards the end, the containment perimeter is breached. An aircraft flies too low and undoubtedly carried the strain with it to its landing site. But surprisingly, no damage results. It turns out that the strain has mutated gradually into a form that is harmless to its new environment and is able to coexist on Earth. What Crichton showed is how the strain itself adapted to the new environment in a sort of highly intelligent systemic immunity in reverse. But we gather that this mutation responds to the devastation that the strain caused in its original form. In other words, the strain mutated from a non-cooperative configuration to a cooperative one, and we can imagine that it did not vanish but coevolved and thrived instead quietly in its new form.

3. IMMUNITY, ADAPTATION, AND INNOVATION

Let's consider balanced interactions in which there is possible co-adaptation of foreign agents, internal disruptions, and a system. Here conflict may be enhancing in some ways, and foreign agents, rather than being erased or assimilated, become contributors to the adapted fabric of the dynamic system.

What I would like to explore here is how immunity functions in this situation. This leads to the question of innovation, because the immune system has to deal with situations it did not encounter before and must balance them somehow without destroying itself. The solution would be an innovation for the system since it carried out a new procedure and since it recognizes the procedure as new and can implement it quickly the next time a similar situation happens.

Traditionally, intrusions of foreign agents and internal disruptions are imperfections in the system and called for corrections to restore the original sense of perfection. But now we know that such state of perfection can only be defined in a negative way and this leads to a non-interactive architecture, inflexible, and not suitable for operating in a changing environment.

As we saw before, a positive definition of self or system is redundant, in the sense that it hinges on membership [8]. But this is the type of sense of self we need because it is flexible. I can change in a gradual and somewhat controlled way. Membership hinges on a preset fuzzy distance to what we can imagine as the center of mass of the existing membership set. As new members join the set, the center of mass shifts. This in turn changes

membership acceptance. We can say that the system's self adapts to its environment this way.

If we place this dynamic sense of self as a model within the elementary loop of function, or ELF architecture, that Meystel and Albus developed [9], then we have the foundations for what could be the primary controller of an autonomous system capable of operating in a changing environment.

This is where immune concepts come in. Imagine the low resolution, control ELF as a cell with semi-permeable boundary defined by its membership class and the fuzzy measure that allows new members to enter the cell. Constructed immune mechanism must keep such self in balance internally and also with respect of the rest of the autonomous system that the ELF controls. This immune system then has to work within the top ELF, so to speak, as well as throughout the entire autonomous system, to make sure that shifts in self-identity at the top do not wreck the system. This should be achieved through immune mechanisms interacting through loops [10] at two levels: within the top ELF cell and throughout the entire autonomous system. Perhaps there could be a third loop involving the autonomous system and its environment.

In previous works I proposed that imperfections could fuels innovation through compensatory and repair mechanisms that function like an immune system of the imagination [11, 12]. Innovation could then be pictured as stemming from an immune mechanism coupled with a selection process and a well-tuned construction system. What we need to do is not filter out novelties that penetrate the control ELF, but somehow gauge them, and incorporate what works by turning it into a system innovation. As Dasgupta and González [8] noted, this requires more resources, particularly memory, but is also more effective than using the negative approach that eliminates all novelty.

If we try to incorporate immune capabilities into an autonomous system, the previous review suggests that to behave as an adaptive system in a changing environment, it would need special features in its architecture.

1 – It would need at least one control ELF that serves as self. As suggested before, the self-set uses a fuzzy measure for positive recognition of membership. The autonomous system is launched with an initial self-set. As the system interacts with the changing environment, it becomes infected, so to speak. The control ELF takes in new self-elements that fall within the fuzzy measure for positive recognition of membership. This process is Building blocks of elementary functioning loops that have memory and remodeling capacities. Neural nets can produce these capacities. A layered network of such blocks so that there are external and internal inputs at all levels of the system.

2 – To simplify the complexities of immune interactions, the system should use encoding or tagging mechanisms [13]. This helps integrate the many layers, hierarchies of ELFs and other mechanisms of the system. Such tags are like names in a language. They abstract features and allow operating directly on the tags. As Meystel indicates, this process of generalization and representation helps establish links among different levels of resolution in the system's architecture. It also reduces the complexity of the system because it can operate on compressed representations rather than on originals. The original can be decompressed when needed by calling the tag.

3 - Network loops can synchronize the entire system so that it can interact in various ways, especially in the formation of higher-level memories, remembrances, and remodelings [10]. For this we have the neurological model that Edelman calls reentry, which is the synchronous firing of widely dispersed neurons in the brain. Feedforward and feedback signals produce such reentry loop. We also have examples from the immune system. The intrusion into the system of infections puts into motion feedforward and feedback processes throughout the entire immune system until some final state is reached. The immune system acts as a network that interacts with the infection, learns from it, and readjusts. If all goes well, the system learns from the incident and is better prepared to deal with similar ones it might encounter.

4 - Redundancy [13]. This enhances the capacity of subsystems for self-repair or compensation. Meystel describes redundancy as excess in a system. He notes that although such excess may appear as a waste of resources, it is necessary for exploration. He links redundancy with a certain playfulness that operates in the excess. He links it also to a sense of desire or emotion that can serve as a vague guide in exploring by using excess resources.

Solé et al note that redundancy should be understood not as an excess of the same resources but of variant resources [16]. Edelman called this "degeneracy" [19]: structurally different components can yield similar results. Solé indicates that degeneracy is deeply related to tinkering in evolution in the sense that different systems can perform similar functions and therefore can be made available in non-linear ways to yield solutions that open the possibility of divergence of architectures. In a changing environment, the possibility to carry out a similar task using different architectures is a very robust and adaptive feature, since with changes new architectures would be more suitable to the shifting context and yet the original function can still be performed. Although I imagine that the original task would also be adapting. This dynamics is closely related to neutral development and play.

An adaptive autonomous system depends to some degree on being enabled to carry out the following two processes:

1 - Neutral development [14, 15]. This is a stand-by mode of search and development for no immediate use, but one whose task is to prepare the system for future imbalances by giving it more diversified resources.

Motoo Kimura is the architect of the notion of neutral evolution and he applied it initially at the molecular level. He notes that the overwhelming majority of changes in nature are not caused by natural selection but "by random fixation of selectively neutral or nearly neutral mutants" [14]. He adds that "although such random process are slow and insignificant for our ephemeral existence, in the span of geological times, they become colossal."

In constructed autonomous systems it is possible to speed up neutral search when necessary. This would be possible at moments the system is rather inactive and therefore has more resources available. Lobo, Miller, and Fontana picture neutral search as an aimless process that happens within a landscape of solutions that have optimal peaks [15]. Imagine a static environment for an autonomous system. The system has interacted with the environment and provided optimal solutions for the various regions of the landscape. After this optimal adaptation, the system has nothing else to do. It is stuck, so to speak, with the peaks. Should the landscape change, the system would have to start all over again. In neutral mode, the system does not rest even if the landscape remains static. The system roams around the peaks and develops imperfect solutions. Viability rather than optimization is what counts. In this way, the autonomous system engages in neutral development that later can impact its behavior when confronted with unexpected situations, and make it simpler to find new solutions since it has already tested out many. What fuels all this neutral development work at a time of system leisure is play.

2 - Tinkering and play [12, 16, 20, 21]. Solé et al noted that tinkering is an important process of evolution, and therefore of adaptation. They describe tinkering as the re-use of different parts of a system in order to achieve a given function. Those different parts are put to novel uses to have them do together what they were not really designed to do. Tinkering then is an imaginative use of resources, not really in optimal ways, but as back-ups or alternatives. This contributes to the robustness of the system because redundant subsystems evolve through tinkering, and they may come in handy one day when all normal channels fail. Internal tinkering can give a system more protection against random failure. This process is a clear example of neutral search.

Tinkering is a form of play restricted to a specific set of elements at hand. Play is more general. Piaget offered perhaps the most concise model of adaptive play as pure assimilation. That is to say, the player interacts and absorbs aspects of the world without changing its action schemata. The question is how to implement such concept with autonomous systems. Let's divide the process into two components. One is the unchanging schemata that provides equal weight to all choices available. The other is the individual player with preferences that mark individuality and that would prefer certain choices without clear justification beyond personal preference. Combining these two aspects, the simplest technical way to define play that I can think of, is to model it as a random search with preferences.

The practical advantage of having a play mechanism in an autonomous system whose tasks include exploration in a changing environment, is that a narrowed random search can save a great amount of testing and memory. The preferences arise from the interaction of the self with the local environment. It is a situated preference relevant to the moment. This is the engine that drives neutral development as well as recombinant process of exploration. It also helps the system with the overall capacity for self-repair or compensation. The system's degree of play can be controlled through resource allocation given its current state and task. This can be a control mechanism weighed internally. It could also be weighed externally but this reduces the autonomy of the system.

4. CONCLUSION

Natural immune mechanisms are a source of modeling inspiration for the design of autonomous systems capable of operating in dynamic environments. The construction of AIS processes is just starting. Most work is still at the conceptual level. We saw that natural immunity is an intelligent, learning ensemble of interacting mechanisms, and has a strong adaptive function. Modeling immunity can help make adaptive autonomous able to function in changing environments. We noted that to begin to tap this potential the AIS must be designed not in a negative reactive way following the war model, but in a positive way that is more like balanced trade across differences. As Holland noted, an adaptive system has to be able to exploit as well as explore. We saw that immunity is the part of the system that can assist with exploration as well as help maintain the system's integrity so that it can continue to carry out its tasks.

I focused specifically on two processes associated with immunity that could be implemented in autonomous constructions, although their importance is only indirectly related to achieving assigned tasks. These processes are neutral search, tinkering, and play. All are essential parts of natural immune systems, but are difficult to implement in constructed systems, and may seem to be too much idle luxury given the potentially high cost of making them work well.

But in the research and development phase, when discovery outweighs implementation, and when our imagination is freer in the use of resources, we do have

more room to play. We can tinker with models and architectures in unexpected ways as long as dogmas don't hold us back and we keep to the task of allowing innovation to happen. In the case of immune features, they might lead to surprising integrations with architectures we already have. Perhaps by enhancing the capacity of constructed systems to engage in neutral tinkering and play we may render them more curious and, paradoxically, more robust. This enactment of interaction may well mark the transition to a new generation of autonomous systems able to explore, innovate, and adapt while carrying out their tasks.

REFERENCES

[1] Llinás, Rodolfo. *I of the Vortex: From Neurons to Self.* Cambridge: MIT Press, 2001.

[2] Tarakanov, Alexander, Victor Skormin and Svetlana Sokolova, *Immunocomputing: Principles and Applications*. New York: Springer-Verlag, 2003.

[3] Coutinho, António, "A Walk with Francisco Varela from first- to second-generation networks: In search of the structure, dynamics and metadynamics of an organism-centered immune system." In *Biol Res* 36: 17-26, 2003.

[4] Segel, Lee A. and Irun R. Cohen. *Design Principles for the Immune System and Other Distributed Autonomous Systems*. New York: Oxford Univ. Press, 2001.

[5] Holland, J. H. *Adaptation in Natural and Artificial Systems.* 2nd ed. Cambridge: MIT Press, 1992.

[6] Tarakanov, Alexander and Dipankar Dasgupta. "A Formal Model of an Artificial Immune System." In *BioSystems* 55 (1-3), 151-158, 2000.

[7] Dasgupta, D. ed. *Artificial Immune Systems and Their Applications*. New York: Springer-Verlag, 1999.

[8] Dasgupta, Dipankar and Fabio González. "An Immunity-Based Technique to Characterize Intrusions in Computer Networks." In *IEEE Transactions of Evolutionary Computation*, 6 (3), 1081-1088, 2002.

[9] Meystel, Alexander M. and James S. Albus. *Intelligent Systems: Architecture, Design, and Control.* New York: John Wiley, 2002.

[10] Arata, Luis O. "Creation by Looping Interactions." *M/C: A Journal of Media and Culture* 5.4 (2002) http://www.media-culture.org.au/0208/creation.html

[11] Arata, Luis O. "Interactive Measures and Innovation." In *Measuring the Performance and Intelligence of Systems: Proceedings of the 2003 PerMIS Workshop*. NIST Special Publication forthcoming in 2004.

[12] Arata, Luis O. "Interaction, Innovation, and Immunity: Enabling Agents to Play." In *American Association for Artificial Intelligence 2004 Spring Symposium Series: Interaction Between Humans and Autonomous Systems over Extended Operation*, 41-46, 2004.

[13] Meystel, Alexander. "Evolution of Intelligent Systems Architectures: What Should Be Measured?" In *Measuring the Performance and Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop*. NIST Special Publication 970, 361-382, 2001.

[14] Kimura, Motoo. *The Neutral Theory of Molecular Evolution*. Cambridge: Cambridge University Press, 1983.

[15] Lobo, José, John H. Miller, and Walter Fontana, "Neutrality in Technological Landscapes." 2004 (soon to be published).

[16] Solé, Ricard, Ramon Ferrer-Cancho, Jose Montoya, and Sergi Valverde, "Selection, Tinkering, and Emergence in Complex Networks." In *Complexity* 8, No. 1, 20-33, 2002.

[17] Mitchell, Melanie. *Analogy-Making as Perception: A Computer Model*. Cambridge: MIT Press, 1993.

[18] Mitchell, Melanie and Stephanie Forrest. "Genertic Algorithms and Artificial Life." In *Artificial Life* 1 (3), 267-289, 1994.

[19] Edelman, Gerald and Giulio Tononi. *A Universe of Consciousness: How Matter Becomes Imagination*. New York: Basic Books, 2000.

[20] "Modeling Interactive Intelligences." In *Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop*. NIST Special Publication 990. September 2002. http://www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Metrics/PerMIS_2002_Proceedings/Arata.pdf

[21] "Can Your Autonomous Robot Come Out and Play?" In *Integration of Knowledge Intensive Multi-Agent Systems. KIMAS 03: Modeling, Exploration, and Engineering*. Institute of Electrical and Electronics Engineers IEEE, 2003.

[22] "Reflections on Interactivity." In *Rethinking Media Change: The Aesthetics of Transition*. David Thorburn and Henry Jenkins, eds. Boston: MIT Press, 2003.

Intelligence ≠ Autonomy ≠ Capability

J. P. Gunderson, L.F. Gunderson
Gamma Two, Inc.
{jgunders, lgunders}@gamma-two.com

ABSTRACT

In recent years, there has been a tendency to confuse the terms *Intelligence*, *Autonomy*, and *Capability*. In this paper we present the viewpoint that intelligence and capability are independent. These two factors describe an orthogonal design space that places upper bounds on the autonomy of the intelligent system. This design space for intelligent systems is illustrated by describing existing intelligent systems (some artificial, some natural) which demonstrate discrete points in the design space. Further, exemplars from biological systems indicate that there are natural constraints within the design space for intelligent systems, which suggest the need to balance the intelligence and capabilities of the designed systems. This design space is used to construct guidelines for the development of intelligent, capable, and autonomous systems.

KEYWORDS: *Intelligence, Autonomy, Capability, Intelligent Systems, Robotics, Autonomous Systems.*

1. INTRODUCTION

In recent years, there has been a tendency to confuse the terms *Intelligence*, *Autonomy*, and *Capability*. In this paper we present several arguments, drawn from research in biology, cognitive science, artificial intelligence, and psychology, that demonstrate that these concepts are far from equivalent. Further, we propose that, while these concepts are clearly related, they may be orthogonal. If this orthogonality can be demonstrated, then it could be leveraged to allow researchers to correctly categorize both the requirements of complex tasks, and the types of intelligent systems that are needed to achieve these tasks.

In this paper we present the viewpoint that intelligence and capability are independent – that one can have significant intelligence and lack capability, or vice versa. This is a critical design concept for engineers and scientists involved in the design and deployment of intelligent systems, since the design problem requires the careful application of the correct resources to solve specific problems. Autonomy is presented as an ability that is bounded above by the independent terms of intelligence and capability, however below this boundary, autonomy is shown to be conditionally independent as well.

This design space for intelligent systems is illustrated by describing existing intelligent systems (some artificial, some natural) which demonstrate discrete points in the design space. These exemplars are further used to extract design criteria for the needed levels of intelligence and capability, and the degree of autonomy which is necessary for an intelligent system to achieve its goals in its target environment.

2. DOMAIN

Before addressing the working definitions, it is necessary to develop a context. In this paper, the domain is considered to be goal directed behavior in dynamic and uncertain domains. Goal directed means that the intelligent systems deployed into these domains have goals that need to be achieved. This means that the systems are not behaving randomly, rather they are developing action sequences or selecting behaviors to change the state of the environment, and acting on these action sequences. The environment is dynamic, in that it is changing over time, in ways that are independent of the actions of the intelligent system. The intelligent system's perception of the environment is uncertain and the results of the actions taken by the intelligent system are non-deterministic.

These characteristics of the domain place significant demands on any intelligent system. In this domain, the intelligent system can only achieve probabilistic goal satisfaction; there is no guaranteed optimal performance. However, any system deployed in the real world must be able to deal with these problems.

3. INTELLIGENCE AND CAPABILITY

In order to have a common framework for discussion, it is necessary to have at least a working definition of the terms. While it would be nice to have precise definitions, these working definitions are meant to be tools for analyzing systems, not an absolute characterization that divides the world into classes.

In the area of measuring the performance of intelligent systems, a common approach is to measure the probability of goal satisfaction as a measure of intelligence. However, this results in confounding the effects of intelligence and capability. This confounding effect has contributed to the confusion of these two concepts, which has caused problems in the design and development of intelligent systems. The following sections present definitions for the terms intelligence and capability.

3.1 INTELLIGENCE

Intelligence has defied formal definition for as long as the concept has existed. There is a general consensus that intelligence is related to being able to solve problems, or to produce things that are of value to the society [1]. This

concept circles around the core idea that an intelligent system has the ability to achieve goals within an environment that is dynamic and uncertain. While intelligence is often characterized as a single entity, (e.g., Spearman's general intelligence, *g*), this theory has been criticized on a number of grounds [2]. Thomson argued in 1939 that there was no evidence that *g* represents any underlying structure in the nervous system of humans [3]. While Spearman proposed a two factor analysis, a combination of *g* and a collection of specific factors (*s*), recent work has proposed anywhere between seven and one hundred and fifty separate factors [4].

In the transition from natural intelligence to artificial intelligence, there is less emphasis on deriving the structural factors and more emphasis on the functional aspects. In part, this is due to the constructive nature of artificial intelligence as an engineered product, which is designed to meet specific requirements. Therefore, for this paper, the focus is on measuring intelligence via factor based performance metrics – how well does the intelligent system *develop* solutions to problems in a dynamic and uncertain environment. This is more closely allied with Newell and Simon's "intelligence as problem solving." [5] Albus and Meystel have proposed defining intelligence as "the ability to behave appropriately in an uncertain environment," where appropriate behavior will maximize the likelihood of goal satisfaction [6].

A critical aspect of intelligence is that it is based on the ability to determine or develop a good solution to a problem, not necessarily in the ability to execute it. In the human realm, a skilled mechanic, who can quickly and accurately diagnose a complex problem, does not become less intelligent because an injury prevents her from physically manipulating the wrench needed to execute the repair. Nor does she regain intelligence when the injury heals. Any working definition of intelligence must not fall into the trap where breaking one's leg makes one stupider. The definition used here is:

> **Intelligence**: the ability to determine behavior that will maximize the likelihood of goal satisfaction in a dynamic and uncertain environment.

This definition meets the criteria that an incidental change that impairs the ability of the system to execute the behaviors does not alter the intelligence of the system. Clearly however, the ability to successfully execute the appropriate behavior does affect the system's ability to satisfy goals in the real world. This successful execution is the capability of the system.

3.2 CAPABILITY

Execution capability has been less studied. Capability describes the ability of the intelligent system to successfully execute behaviors. An intelligent system may be able to correctly determine a valid course of action to achieve a goal, but be incapable of executing that course of action, while another system (such as a teleoperated robot) might be incapable of developing any 'intelligent' solution, however, given one, it can execute it and respond to minor failures during execution. An example of this would be our injured mechanic. She can correctly determine both the cause of a problem and the necessary repairs, but lack the capability to execute the repair. Her assistant, who may not have the intelligence or experience to solve the problem, can follow her instructions to effect the repair. The capability of her assistant does not imply intelligence.

> **Capability**: the ability to successfully execute behaviors or actions in a dynamic and uncertain environment.

There is one thing that should be noted about this definition. Unlike intelligence, capability is not goal oriented. This means that doing the wrong thing successfully does not imply reduced capability. However, selecting the wrong behavior to execute does imply reduced intelligence.

3.3 INDEPENDENCE OF INTELLIGENCE AND CAPABILITY

No attempt will be made in this paper to present a mathematical proof of independence. Rather, examples of intelligent systems which span the range of intelligence and capability will be presented. It is clear from the working definitions presented above, that intelligence is defined by the determination of behaviors, and has no definitional component related to execution. Likewise, capability is defined without any reference to the appropriateness, or correctness, of the actions, and is dependent only on the successful execution of the required actions.

Consider the example of vacuum cleaning robots and a two story home. The vacuum cleaning robots are available in several versions. They can be intelligent, with detailed maps of the furniture, and an understanding of traffic patterns; or they can be simple reactive systems that bump into things and vacuum at random. Second, they can be equipped with simple wheels, or they can be equipped to climb stairs as well. The goal is to keep all the carpets in the home clean.

Table 1 - Goal Satisfaction for vacuum cleaning robots

		Stairs	
		Low	High
Intelligence	Low	0.25	0.50
	High	0.50	1.0

In this case, the intelligent vacuum cleaners do a much better job of cleaning the corners, and getting the high-traffic areas, so the goal satisfaction is higher for the areas that they can reach. The stair climbing robots can reach more of the carpets, regardless of how good a job they do, so the goal satisfaction is higher overall. In effect, for all four of these robots goal satisfaction is a function of both intelligence and capability – but high intelligence can exist independently of high capability, and vice versa. This implies that:

$$g = f(c,i), \quad [1]$$

where g = goal satisfaction,
c = capability, and
i = intelligence.

4. RELATIONSHIP OF AUTONOMY TO INTELLIGENCE AND CAPABILITY

In the previous section, working definitions for intelligence and capability were presented. These definitions offer several advantages in that they are independent, measurable, and are derived from the common usage of the words. As mentioned in the introduction, these are intended as tools to analyze systems, not necessarily to characterize them.

In this section autonomy is addressed, building from the previous definitions. The intent is to delineate the scope and relationships between these terms.

4.1 DEFINING AUTONOMY

The concept of autonomy, like that of intelligence, is controversial in the artificial intelligence and robotics communities. However, it is somewhat less controversial in everyday usage. According to one dictionary definition (in all cases the dictionary used is the 1969 American Heritage [7]). Autonomy is defined as:

> **Autonomy**: 1.The condition or quality of being self-governing. 2. Self government, or the right of self-government; self-determination, independence. 3. A self-governing state, community, or group

The common term in all these definitions is *self-governing* or *self-government*. Clearly, to be autonomous is to be able to govern oneself, but what does this mean when applied to an intelligent system?

Govern also has an everyday meaning, which, while not commonly applied to a robot, has direct applicability to intelligent systems.

> **Govern**: 1. To control the actions or behavior of; guide; direct. 2. To make and administer public policy for (a political unit); exercise sovereign authority in. 3. To control the speed of magnitude of; regulate. 4. To keep under control; restrain. 5. To decide; determine.

And the list goes on to grammatical uses. However, the clear thread in the definition of govern is the ability to decide and implement decisions. By extension, the sense of self-governing is the ability to decide and implement decisions for and by oneself. Hence, autonomy is the ability of a system to make choices and enforce its decisions. While this is consistent with the common usage, and has been supported by many researchers, this definition can be considered to conflict with a definition that limits the autonomy to be with respect to some goal or task assigned by an outside agency [8]. This definition addresses the issue raised by researchers such as Alan Schultz that a truly autonomous robot would be sitting on a beach somewhere, drinking motor oil, not slaving away on some human assigned task [9].

As an interesting side note, the etymology of govern is from the Greek *kubernan*: to steer, guide. This is the same root work that gives us cybernetics, coined by Norbert Weiner to refer to the theoretical study of control systems. In effect, an autonomous intelligent system is simply one that has the capability to control itself, to make decisions and implement those choices. Where, then, do these choices come from? Certainly, if the system is incapable of generating multiple options to achieve a goal, it cannot decide which behavior to undertake. With no viable behaviors, or a single solution to a problem, there is no choice – and therefore no ability to decide. While a system that can produce fifty possible goal satisfying solutions may be more intelligent that one that can only produce ten such solutions, it has no more autonomy. A system that can successfully execute more complex behaviors may be more capable than another, but that does not make it more autonomous, if an outside agency can override the chosen behavior and force an alternate behavior.

4.2 ESSENTIAL RELATIONSHIP

Notice there is nothing in the definition of autonomy that addresses the quality of the decisions made by the autonomous system. An entity can be autonomous and stupid, or be autonomous and smart. A system can make good decisions and implement them poorly, or implement them well. Autonomy is independent of intelligence (the ability to select appropriate behavior) and independent of capability (the ability to successfully execute an action or behavior.

Given the definition that autonomy is simply the ability of an entity to decide its own behavior and to execute that behavior, how does that relate to an intelligent system such as a robot? One critical factor is that the system must have the ability to select between options. If a system has no choice in the behaviors that it exhibits, then it cannot be

autonomous. In addition, if the choices made by the system can be overridden by an external agency, then the system is not autonomous. If a system has options to select from, and the ability to select and implement an option, then it is autonomous. Thus, a bi-metallic strip thermostat has no autonomy. The user selects the set point, and physics define the only possible action at any time. The thermostat is not free to say "Well, I know my set point is 72 degrees Fahrenheit, but I won't kick over until it gets down to 68." On the other hand, a simple random walk robot, with no goals except to keep moving, may have complete autonomy in the choice of direction and distance, and can autonomously fall down a staircase if it selects the wrong option.

Since the inability to produce a course of action precludes its use, and the inability to execute a course of action prevents it from being valid; it follows that intelligence and capability act as upper bounds on the autonomy of a system. However, while these two capacities limit the maximum autonomy of a system, below this limit the system can have a much or as little autonomy as the designer (in the case of an artificial intelligent system) allows.

Based on the definitions, it seems that intelligence and capabilities define an action space, and autonomy is bounded by this space (See Figure 1).

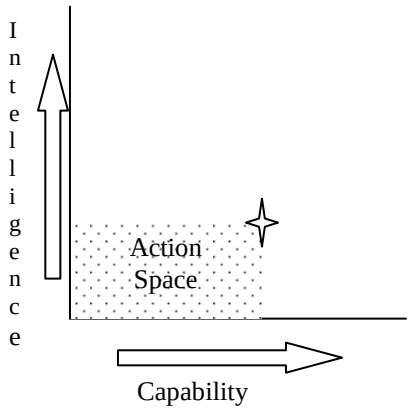


Figure 1: Effect of Intelligence and Capability on the range of autonomy. As intelligence and capability increase, the range of available options increase. The system can be autonomous, in which case it can select from these options. Or the system may not be autonomous, in which case its action space collapses to a single point for any goal/environment combination.

The available actions may be discrete options such as 'open a door' or 'send an email'. In the case of a mobile robot selecting a new heading or selecting a distance to travel the options may be drawn from an effectively continuous range. In terms of autonomy, it does not matter how many options are available, or even if there are an infinite number of options. All that matters is that there are options, as defined by the intelligence and the capabilities of the system, and that the system can select between them.

This leads naturally to the following question: Given a fixed system that can produce multiple feasible solutions to one problem, but can only produce a single solution to another problem in the same domain, is the system autonomous in the former case, and not autonomous in the latter?

Unfortunately, this leads to a typical white box versus black box problem. What if the system has multiple options, but always selects the same one? How does this differ from a system that only has that single behavior available to it? Is one autonomous and the other not – and how can one tell from the outside? This is clearly another variation on the 'strong' versus 'weak' question that has plagued artificial intelligence, cognitive science, and philosophy since their inceptions. The focus of this paper is not to answer such a complex conundrum, but to look at the implications of the relationship between autonomy and intelligence and capability.

Simply put, if the system cannot generate options for behavior, then the autonomy of the system is compromised; if the system cannot select between options, then the autonomy of the system is compromised; and if the system can select one of several options, but the execution of the option is controlled by an outside agency, then the autonomy is compromised. Thus, given that the system is enabled to choose between options in the method of achieving its goals, the intelligence and the capability of the system act as bounds on the autonomy of the system.

These two controlling factors, intelligence and capability, are independent in the abstract sense presented so far, and as such there are no constraints on either range. In theory, an intelligent system could be awesomely intelligent, yet totally incapable of achieving anything. Alternatively, a system could be a dumb as a box of rocks, yet have the capabilities of the most advanced robot ever imagined. These do not seem to be reasonable combinations. Examining of the existing intelligent systems, one does not see either extreme; rather the exemplars seem to be grouped with roughly balancing capabilities and the intelligence to use those capabilities. In the next section, examples of successful intelligent systems, deployed into harsh, dynamic, and uncertain domains are examined, drawn from biological systems.

5. CONTROLLING FACTORS ON INTELLIGENCE, AND CAPABILITY.

Brains are expensive. [10] This means that, for a species to compete effectively, the increased intelligence must covey an increased survival advantage. In the following discussion, the relationship of intelligence and capability in natural intelligences will be examined. In order to talk

about animal intelligence, some discussion is needed of the types of intelligence. Animal intelligence has been partitioned into two major classes 'menu-driven' intelligence and 'social' intelligence [11].

The 'menu' in menu-driven intelligence does not refer to pull-down option selection in a Graphical User Interface. Rather it is based in the concept that biological organisms must eat to survive, and therefore those that can find and acquire more varieties of food (their menu) more effectively have higher survival rates. Since finding food in a dangerous, harsh environment is extreme problem solving, it drives the development of intelligence.

Social intelligence is intelligence that is oriented towards communicating between individuals of the same species. The need for cooperation in hunting and defense require the ability to both develop team-based solutions to problems, and the associated skills of communication and maintaining group dynamics. This type of intelligence corresponds to Gardner's Lingustic and social intelligences.

These two types of intelligence will be treated separately. Since most organism can not describe their thought processes, all experimental evidence can show is the combination of intelligence and capacity. Starting with 'menu driven' intelligence, it has been shown that distribution of foods acts as a stimulus for mental development in primates [12]. In effect, the need to cognitively maintain and track the recognition patterns, locations, acquisition techniques and risks of additional types of food, increases the cognitive demand on the system, and requires allocation of resources to cognition, hence more intelligence is needed.

From this, it can be argued that in a highly competitive environment, any increase in intelligence that is not matched by an appropriate increase in capability could lead to the extinction of a species. The argument follows thusly. Let us consider two species in the same ecosystem. Both of these species are confronted with a new food source. One of the species uses its expensive brain power to construct a feeding strategy that it is capable of implementing. The other species constructs a feeding strategy that it is not capable of implementing. Clearly they have both exerted energy coming up with a solution, but only one has received a reward. In an evolutionary setting the species that came up with an implementable strategy has a major advantage. If the advantage is great enough, the losing species may face extinction. It is an interesting side note on the nature of biological systems, that the idea of an organism coming up with an unworkable strategy is almost unthinkable. This is the result of living in a harsh evolutionary system, where unsuccessful adaptations die out quickly.

'Social' intelligence is one that humans are most familiar with. Certainly for our species, it may be one of the most important of the many types of intelligences. However, even in this type of intelligence, capability plays an important role. In order for this type of intelligence to be expressed, there must be both a sender with an idea and the capacity to transmit this idea, and a receiver with the capacity to receive the idea. By the same argument as above, if an organism spends energy developing a social intelligence, but it lacks the capacity to share, that organism will be out-competed by another organism that does not expend the energy to develop the social intelligence, or has the capacity to communicate the ideas.

The notion of a species having a capability without the corresponding intelligence to use that capacity is hard to imagine. Try to envision a bird, for example, perfectly capable of flying, but unable to 'think' of flying away when a predator attacks. Such a situation could, perhaps, arise but it would not exist for very long under evolutionary pressures. Either a ground based species would out compete the bird, since it would not need to maintain the expensive, but useless, flight equipment; or a subset of the birds would develop the intellect to use flight, and out compete their stupid brethren, Clearly there are birds which do not fly, but those that are capable of flying have the intellect to use the capability when needed.

For both of these classes of intelligence that have been established for animal minds, the intelligence and the capacity to use that intelligence must develop together. In the competitive natural environment, an imbalance between these factors would lead to competitive failure of the species.

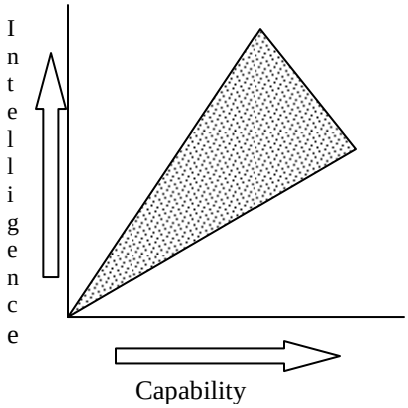


Figure 2: Apparent range of the balance between intelligence and capability in biological systems.

In Figure 2 is a rough representation of the existing biological intelligent systems, indicating the approximate parity between the capabilities of the system and the intelligence to use those capabilities. In general, the practical space is a close approximation to the abstract line where intelligence is precisely balanced by capability. However, there is a certain amount of variation as a result of the dynamic nature of biological systems. The region above the shaded area would correspond to species which had

intelligence in excess of their capabilities – resources invested in brains that could produce theoretic solutions to problems, solutions which could not be implemented with the existing capabilities. The region below the shaded area would correspond to species which had capabilities that it was impossible to figure out how to use.

In an environment with resource limitations and a high cost for failure the angle of the shaded region is fairly small. However, in the design space of engineered intelligent systems it is possible to create systems which are not bounded by the harsh realities of life. Researchers routinely develop intelligent systems that have capabilities which far outstrip their intelligence, and, less frequently, systems are designed and built that have the intelligence, but lack the capability to achieve their goals. In the following section, examples of these systems are presented and some approximate design rules are suggested to attempt to balance the allocation of resources between capability and intelligence.

5.1 Law of the Minimum

In ecologic systems there is the notion that every species in an environment has a limiting factor. Leibig's Law of the Minimum states that for every species, there is a single limiting factor that controls growth of that species in that system [13]. For desert species, the limiting factor is the available water. For a rain forest species, the limiting factor is the amount of sun energy. This paper argues that, in an autonomous system, the limiting factor will either be intelligence or capability. If intelligence is the limiting factor, then increasing the capability will not significantly improve the performance of the system or allow for greater autonomy. Conversely, if capability is the limiting factor, then increasing the intelligence will not yield a significant improvement.

6. Guidelines for Design of Autonomous Systems

The current state of the development of intelligent systems abounds with examples where the intelligence is out of balance with the capabilities of the system. In many cases these systems are built as research tools, and the focus is on exploring only one of the two aspects. However, there are also examples of systems which are deployed into dynamic and uncertain domains and which are intended to achieve specific goals.

The domain abounds with tele-operated systems where the capabilities of the system far outstrip the available intelligence. Systems which were intended as 'force multipliers' require three to five human operators. In part this is due to investing in capability rather than intelligence, and then falling back onto humans to do the 'hard part.'

6.1 Balance Capability and Intelligence

This will sound trivial, but if the system is not meeting its performance goals, determine what the problem is before attempting a fix. The reasoning above demonstrates that there are at least two possible, independent causes for reduced performance metrics. Either could be the limiting factor that is restricting performance. Fixing the wrong one won't help as much as fixing the right one.

Determine the limiting factor and address it, do not just throw more resources at the easy factor. In dynamic domains, where the intelligent system must deal with uncertainty, it is critical to limit the intelligence to that which is sufficient to solve the problem. Adding more capability to an intelligence limited system will show an improvement, but adding more intelligence will show a greater improvement.

6.2 Don't Make it Autonomous, Unless That is What You Need.

Autonomy is problematic. If the system should be tightly controlled, then it needs to be tightly controlled. Autonomy is not a 'magic bullet' that will make an ineffective system work in a dynamic, uncertain domain. If the problem domain requires a specific response to specific inputs, then the designer must provide that mapping. Rather than adding autonomy, focus on building the correct mapping.

If the system needs to be autonomous, and in dynamic, uncertain domains most successful systems must be autonomous[14], the intelligence and capability must be sufficient to support the autonomy. It is pointless to give the system autonomy if it has neither the brains nor the capability to use the autonomy successfully. In addition, making the system autonomous will not magically enable it to do its job. Autonomous stupidity is easy.

7. Conclusions

One drawback to performance based metrics of intelligence is the confounding effects of intelligence and capability. No one would argue that a physicist such as Steven Hawking is not intelligent. Yet if he were put into a room and given a battery of intelligence tests, his physical disabilities would impact the measure of his intelligence – unless the tests were specifically designed to correct for the confounding effects.

Recently, it has been the practice to measure the performance of intelligent systems and label the result 'intelligence'; however, the tests measure the combination of intelligence and capability. This has resulted in the merging of these two very important, very different terms. It is not uncommon to hear researchers say that capability is the same as intelligence. This paper is an attempt to clarify these two terms, so that they can be used to more effectively describe the complexities of intelligent systems.

Intelligence is a cognitive process that allows a system to propose a viable solution to a problem or task. Capability is the ability to implement or execute a proposed solution in a dynamic, uncertain environment successfully. Both of these abilities are necessary for goal satisfaction and both are very difficult. However, by conflating the two terms, researchers run the risk of having a system that fails to achieve its goals due to insufficient intelligence, but increasing the capacity of the system in an attempt to improve it. This can be both fruitless and frustrating for the researcher.

By keeping the concepts of capability and intelligence orthogonal, the system designer has a design tool that will allow her to determine the correct area to focus attention, and the correct type of improvements that will result in a better system.

8. REFERENCES

[1] Gardner, H. *Frames of Mind: The Theory of Multiple Intelligences Tenth anniversary edition*, New York: Basic Books, 1993, ch.4, pp. 60-62.

[2] Richardson, K, *The Making of Intelligence*, New York: Columbia University Press, 2000, ch.1, pp. 8-10.

[3] Thomson, G.H. *The Factorial Analysis of Human Ability*, Boston: Houghton Mifflin, 1939, ch.1, pp.4.

[4] Gardner, H., Kornhaber, M.L., and Wake, W. K. *Intelligence: Multiple Perspectives*, Fort Worth, TX: Harcourt Brace College Publishers, 1996, ch.3, pp. 70-75.

[5] Newell, A. and Simon, H. A. *Human Problem Solving* Englewood cliffs, New Jersey: Prentice –Hall, Inc., 1972, ch. 1, pp. 6–9.

[6] Albus, J. S. and Meystel, A. M. *Engineering of Mind*, NewYork: Wiley Interscience, 2001, ch.1, pp. 6-7.

[7] *American Heritage Dictionary of the English Language*, New York: American Heritage Publishing Co., 1969.

[8] Barber, K.S. and Martin. C.E., "Agent Autonomy: Specification, Measurement, and Dynamic Adjustment," in Proceedings of the Autonomy Control Software Workshop, part of Autonomous Agents, 1999, pp.8-15.

[9] Schultz, Alan, Panel Discussion, AAAI Spring Symposium on Interaction Between Humans and Autonomous Systems over Extended Operation, 2004.

[10] Gunderson, J. P. and Gunderson, L. F. "Mom! The Vacuum Cleaner is Chasing the Dog Again," Proceedings of the 2003 PerMIS Workshop. 2003.

[11] Yoerg, S. I., *Clever as a Fox*, New York: Bloomsbury, 2001, ch. 8, pp. 162-175

[12] Milton, K. "Distribution Patterns of Tropical Plant Food as a Stimulus tp Primate Mental Development," *American Anthropologist*, vol. 83, pp. 534-548.

[13] Barbour, M.G., Burk, J.H., and Pitts, W.D. *Terrestrial Plant Ecology*, Menlo Park: CA, 1987, ch.3, pp.29-30.

[14] Gunderson, J.P. and Martin, W. N. "A Probaility-Aware Planning Architecture to Support Variable Autonomy," *American Association for Artificail Intelligence Technical Report SS-03-04*, Menlo Park, CA: AAAI Press, pp. 83 – 88.

Intelligence?

Ron Cottam, Willy Ranson and Roger Vounckx

In a very specific way, intelligence and information-processing are synonymous.

Information-processing takes place between differently-scaled models of a relevant context. The way in which these models relate to each other, and therefore the result of an inter-scalar processing, depends on the constraints which are imposed on inter-scalar computation, and on the manner in which those constraints are applied.

Mono-scalar 'systems' can always be deconstructed more or less precisely to a set of rules: a single non-fragmented scale corresponds to a single formalization. Any consequent 'systemic' complexity is the result of formal incompleteness or of our lack of understanding.

Multi-scalar 'systems' constitute artificially- or naturally-constrained hierarchies, where the style of processing depends on whether the inter-scalar constraints are externally imposed or internally recursively generated by the information-processing itself. These two types of hierarchy, artificial and natural, have very different properties, and exhibit completely different styles of 'intelligence'.

The unification of a processing assembly into a 'system' is *always* through our intervention, whether at a single scale or across multiple scales of operation. If an artificial hierarchy is not to be scale-fragmented, it must possess some kind of cross-scalar coherence, imposed through our manipulation of the inter-scalar constraints. A natural hierarchy generates this cross-scalar coherence itself, through an autonomy-negotiation between its various scales, creating a hyperscalar system.

This appears to be the 'meaning' of intelligence: it enables a multi-scalar system to operate as if it were simultaneously multi-scalar and mono-scalar. Different individual scales of operation retain a context-dependent degree of autonomy, but the entire assembly is unified at a hyperscalar level.

A 'system' is *always* hyperscalar, whether through artificially- or naturally-imposed constraints. It is questionable whether it would be possible to generate sufficient cross-scalar correlation in an artificial information-processing assembly to generate an interesting or useful degree of independent 'intelligence'. In any case, any attempt to do so in a definably-operating assembly such as a Boolean logic computer could only succeed if the logical definability were relaxed, either by intention or through the relativistic isolation of different parts of the assembly. Even then, current artificial information-processing mechanisms would be incapable of dealing non-catastrophically with the unpredictable structural incompletenesses.

Application of a Crash Prevention Boundary Metric to a Road Departure Warning System

S. Szabo
National Institute of Standards and Technology
Gaithersburg, MD 20899
B. Wilson
Volpe National Transportation Systems Center
Cambridge, MA 02142

ABSTRACT[1]

The U.S. Department of Transportation is actively involved in assessing the benefit of road departure warning systems (RDWS). A crash prevention boundary (CPB) metric has been proposed as one means of objectively measuring system performance. This paper presents the results of applying the CPB metric to data collected during the validation of an experimental RDWS. Two types of road departure warning scenarios are examined: curve speed and lateral departure.

KEYWORDS: *road departure, crash warning system, metrics, crash prevention boundary*

1 INTRODUCTION

The U.S. Department of Transportation is actively involved in assessing the benefit of road departure warning systems (RDWS) that may help reduce the number of collisions and deaths. Many types of metrics designed to objectively quantify warning system performance are being considered. The crash prevention boundary (CPB) metric can be used to determine the amount of acceleration required to avoid a crash as a function of the timing or location of a warning [1][2]. A series of tests consisting of potential crash scenarios designed to elicit a warning [3] were conducted using an experimental RDWS. Data was collected during the test using an independent measurement system. The data was analyzed using the CPB metric, with a focus on evaluating the timeliness of the warning. The warning system provided a warning in almost all cases, but a determination of whether the warning was late or early relied upon a pass/fail criterion established by the warning system's designers. The government is interested in establishing a baseline of acceptable system performance, which is intended to help convince the driving public that these systems provide a worthwhile benefit. The CPB metric may prove useful in establishing a performance baseline. This paper shows how the CPB metric can be used to analyze two types of road departure warning scenarios: curve speed and lateral departure. For this particular warning system, the results of the CPB analyses indicate that the deceleration required to negotiate a curve safely has a markedly different profile than the lateral acceleration required to avoid running off the road.

A measurement system for analyzing warning system performance is also described. The paper closes with some recommendations for future warning-algorithm metrics and design guidelines.

2 MEASUREMENT SYSTEM

An independent measurement system (IMS) was developed by National Institute of Standards and Technology (NIST) for evaluating warning system performance. The IMS allows evaluators to measure performance without relying on the system under test for sensor data. In addition, the IMS provides redundancy, quality control and an opportunity to collect miscellaneous data for additional analysis (e.g., data for validating the CPB metric was collected using the IMS). The IMS consists of three video cameras mounted on a detachable roof rack (Figure 1) and a fourth camera pointed at the warning system's dash display. A microphone is located in the cab to capture the audible warnings issued

---

[1] No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. This publication was prepared by of the United States Government employees as part of their official duties and is not subject to copyright.

by the system and to capture comments from the driver. The videos from the four cameras are fed into a multiplexer producing the image shown in Figure 2. The output of the multiplexer is recorded onto a digital video recorder (DVR). A GPS antenna is mounted on the roof and the raw output of the GPS receiver is recorded on the second audio channel of the DVR at a one Hz rate, ensuring that the GPS data is synchronized with the output from all four cameras. The GPS data is post-processed using National Geodetic Survey Continuously Operating Reference Stations (http://www.ngs.noaa.gov/CORS). The side and forward cameras are calibrated so that pixel coordinates can be transformed into ground coordinates either in a GPS-referenced coordinate system or in a local-vehicle coordinate system. Analysis software is used to view video and the vehicle's trajectory, for making measurements to lane markers and obstacles, and for calculating a curve's location and radius.

Figure 1. Three weatherproof cameras mounted on roof rack.

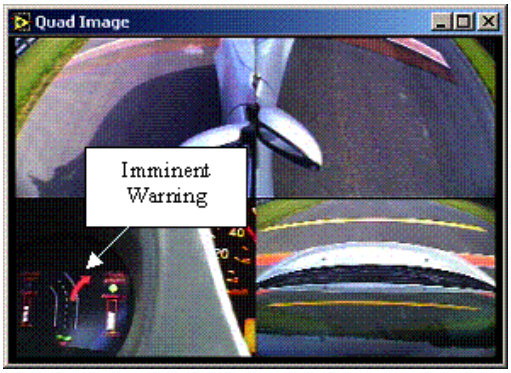


Figure 2. Images from roof-mounted cameras and dash-view camera are multiplexed into a single quad image. The dash-view camera shows an active imminent warning icon.

3 CURVE SPEED CRASH PREVENTION BOUNDARY

Curve speed tests consist of a vehicle traveling straight toward a curve at an excessive speed. Figure 3 shows the geometry used for the CPB analysis. The curve shows a critical point (CP) a short distance into the curve. The warning is given at a distance of x (meters) prior to the CP.

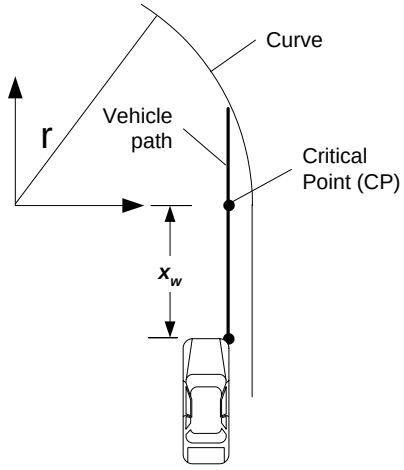


Figure 3 Geometry of curve speed test

Drivers who receive a legitimate warning (a true positive) have a range of reaction times and decelerations levels available to them sufficient to avoid a collision. The locus of the required deceleration versus the reaction time is known as the CPB. The following equation provides the CPB for the curve speed-warning situation depicted in Figure 3. This version of the CPB equation relates warning location to the required longitudinal deceleration for negotiating the approaching curve using the following equation:

$$d_{req} = \frac{v_o^2 - v_s^2}{2(x_w - t_r v_o)} \quad \textbf{(1)}$$

Where:

v_0 = vehicle initial forward speed
v_s = safe speed for the curve
t_r = driver reaction time
x_w = the distance between the warning location and the CP
d_{req} = the deceleration required to reach the safe speed at the CP

Although details of the RDWS warning algorithm are proprietary, a public algorithm under development by the Applied Physics Laboratory and NHTSA [4] provides insight into the general warning process. The algorithm assumes a warning system has different sensitivity settings that will affect the location of a warning. The settings use values for the

desired limit of lateral acceleration (A_s) in a curve and longitudinal deceleration before the curve (D_W). The lateral acceleration limit is used to determine the safe speed going into the curve as follows:

$$v_s = \sqrt{A_s r} \qquad \textbf{(2)}$$

The values for each setting used in this analysis are given in Table 1. A near setting means that the driver would like a warning closer to the curve's entry, which means that the driver is comfortable applying a greater deceleration before the curve and carrying a higher lateral acceleration through the curve. One can think of these values as defining a typical driver's projected velocity profile for the curve-warning algorithm. The first phase of the profile has a constant velocity equal to v_0. The warning occurs and the velocity is maintained for 1.5 s. Afterwards, the velocity ramps down based on D_w until v_s is reached.

Sensitivity Setting		A_s	D_w
1	Near	0.42 g	0.70 g
2	Near-mid	0.36 g	0.60 g
3	Mid	0.30 g	0.50 g
4	Mid-far	0.24 g	0.40 g
5	Far	0.18 g	0.30 g

Table 1 Sensitivity settings for APL/NHTSA CSW public algorithm.

The curve speed test was conducted on the Transportation Research Center (TRC) winding road course (Figure 4) on October 2, 2003. The vehicle starts at point E, reaches a speed of 50 mph (22 m/s) by point D and travels through curve C. The vehicle should provide a warning before reaching curve C. The radius of curve C, measured using the IMS, is 115.7 m ± 0.5 m. The vehicle speed during each test, measured using the GPS, was 48.8 mph ± 0.05 mph (21.8 m/s ± 0.02 m/s). The lateral acceleration (v^2/r) in the curve if the speed is maintained is 0.42 g.

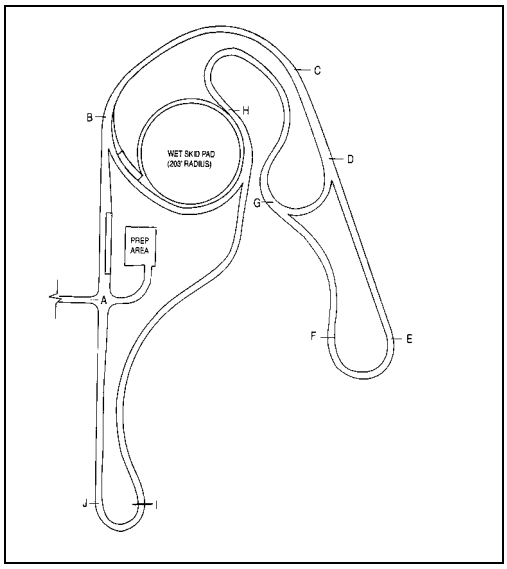


Figure 4. TRC winding road course. Analysis was performed on data collected from test runs where vehicle travels from E into curve C.

The experimental warning system produces two levels of warning: cautionary and imminent. As a driver approaches the curve at an excessive speed, the system will first issue a cautionary warning. If the driver fails to reduce the vehicle's speed, the system will then issue an imminent warning. Figure 5 shows test results for cautionary warnings and Figure 6 shows test results for imminent warnings. Each crash prevention boundary curve in the figures is plotted using equation (1), with the mid-level sensitivity setting used to determine the safe speed and x_w set to the location of the warning for a given run. Based on the location of the warning, the curves describe the required longitudinal deceleration to reach the safe speed for a given driver reaction time.

The plots should *not* be interpreted as classical CPBs. In a classical CPB, a driver reacting to a warning must decelerate at a level *above* the level shown in the curve that corresponds to the specific lapse between the warning and the driver's response. For example, CPBs for slower and stopped lead vehicles [5] specify the required decelerations for cases in which a following vehicle is approaching a lead vehicle and receives a rear-end collision warning. Drivers decelerating at lower levels than those provided by the curves will crash. In the longitudinal CPBs in Figure 5 and Figure 6, in contrast, drivers that take longer to react and decelerate less will not necessarily crash. Rather, they will need to negotiate the curve, at

least initially, faster than the safe speed in equation (2). Section 5 will comment on this.

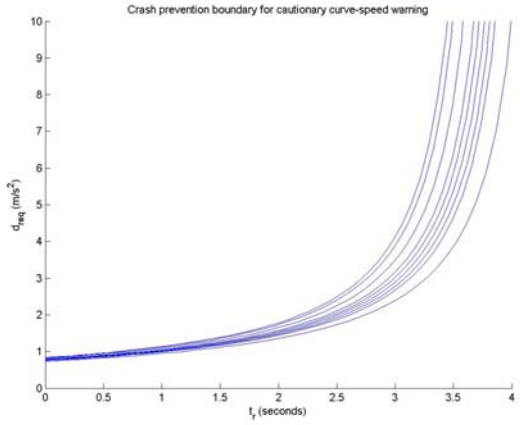


Figure 5. Longitudinal CPB for series of cautionary warnings. Each curve corresponds to the location of a warning during test runs.

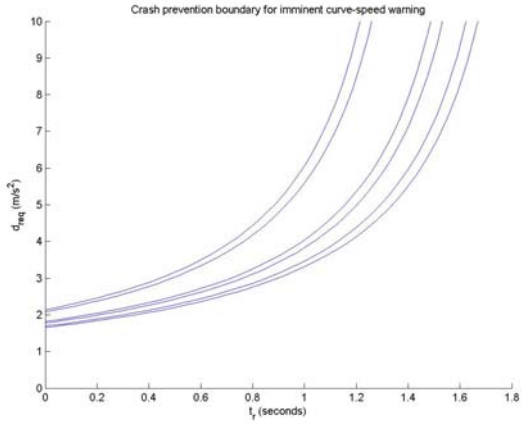


Figure 6. Longitudinal CPB for series of imminent warnings

4 ROAD DEPARTURE CRASH PREVENTION BOUNDARY

Lateral drift tests consist of a vehicle traveling at a constant speed and departing the road at a constant angle. Figure 7 shows the geometry for a lateral drift into a jersey barrier. The location of a warning, y_w, is the distance from the vehicle to the road edge (or other boundary) at the time of warning. A warning should provide the driver time to react and steer away from the road boundary.

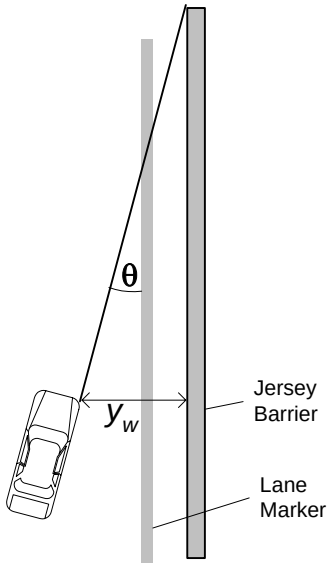


Figure 7. Geometry of a lateral drift test. In this test, the Jersey barrier is the road boundary.

Reference [2] also provides a CPB equation that can be used to relate warning location to the required lateral acceleration to avoid a lateral departure using the following equation:

$$a_{lat} = \frac{(v\theta)^2}{2(y_w - t_r v\theta)} \quad \textbf{(3)}$$

Where:

v = vehicle forward speed
θ = departure angle
a_{lat} = lateral acceleration used to avoid departure
t_r = driver reaction time
y_w = the distance between the warning location and the road boundary

The validation tests for the warning system examined for this paper specify a lateral velocity during a departure as opposed to a departure angle. The following equation is used to relate lateral velocities measured during a test to departure angle required in equation (3) (this relationship is based on the tangent of the θ, which for small angles is equivalent to θ):

$$\theta = \frac{v_{lat}}{v_{long}} \quad \textbf{(4)}$$

Where:

v_{lat} = lateral velocity
v_{long} = longitudinal velocity (i.e., forward speed or just v)

Figure 8. Test set-up for road departure toward jersey barrier.

One type of test used to analyze the warning system is the continuous obstacle test, which consists of a straight lateral departure onto a shoulder with a jersey barrier offset 1 meter from the inside edge of the lane boundary. Tests were conducted at the TRC skid pad using a water-filled jersey barrier (Figure 8.). Cruise control was used to set the forward speed during the test at 40 mph (18 m/s). The specified lateral velocity for the test (i.e., the velocity toward the barrier) is 0.4 m/s ± 0.1 m/s. The velocities measured using the IMS ranged from 0.3 m/s to 0.53 m/s.

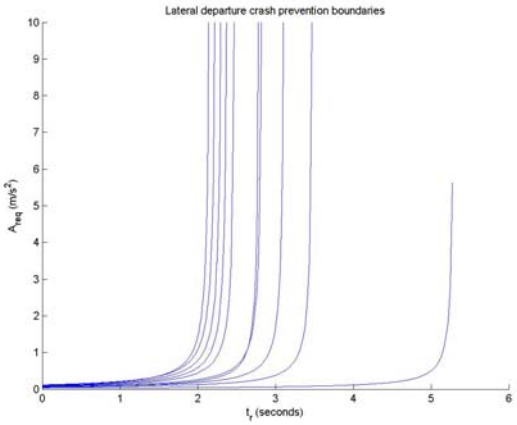


Figure 9. Lateral CPB for series of imminent warnings

Figure 9 shows the test results plotted using the CPB metric. Note that in this case the curves are legitimate CPBs: a driver who receives a warning, steers at some t_r second(s) later, and does not steer with a lateral acceleration greater than that corresponding to the value at this t_r in the appropriate plot will crash.

5 Future Metrics and Design Guidelines

Having developed longitudinal and lateral CPBs in sections 3 and 4, we will now comment on them and their utility as a metric. As indicated, the longitudinal CPB in section 3 is not a true *crash* prevention boundary, because a driver who brakes above the boundary will not necessarily crash. Rather, the driver will be forced to negotiate a curve's critical point at a speed greater than a predetermined safe speed. A driver's skill level and the available traction will determine if a collision occurs.

Despite this limitation, the longitudinal CPB remains a useful metric for both RDWS developers and analysts. A developer can readily understand the implications of a more aggressive warning algorithm, e.g., shown in Figure 6 versus Figure 5. We see in Figure 5, whose CPBs originate from a mid-level (cautionary) warning setting, that approximately 0.1 g of deceleration is required to negotiate the curve safely when the warning is issued. If a driver requires 1.5 second to respond, a deceleration of approximately 0.15 g is required.

In Figure 6, we see that a more aggressive warning algorithm assumes drivers will react more quickly and decelerate at higher levels. Indeed, the change in the CPBs between the cautionary and imminent CPBs in Figure 5 and Figure 6 indicates that in the latter case drivers must react in approximately half the time and decelerate at approximately twice the level as in the former case.

The lateral CPBs shown in Figure 9 all show quite low initial values for the *lateral* acceleration that is required to avoid a road-departure collision (colliding with the barrier). Even 1.5 second later, the required acceleration is still quite low. At reaction times greater than 2 seconds, however, the required acceleration increases markedly.

The different kinematics between curve speed warnings and lateral departure warnings merit comment. We see in equation (1) that the square of vehicle forward speed minus the square of the safe speed composes the numerator of the term that determines the required deceleration. In SI units, 50 mph equals 22.5 m/s and its square is 506. A safe speed of 35 mph results in a numerator of 258, a significant number. The lateral CPB counterpart, equation (3), has the lateral speed as its sole term in the numerator. A lateral speed, even in a departure situation, of 1 m/s is rare, so the numerator of equation (3)

typically equals unity or less. Only when the actual reaction time starts approaching the reaction time programmed into the warning algorithm will the denominator of (3) approach 0 and the required lateral acceleration increase rapidly. Thus for curve-speed CPBs, Figure 5 and Figure 6, we see higher initial values and a continuous increase in the required deceleration with reaction time. The lateral departure CPB in Figure 9 shows a lower initial value in the required lateral acceleration and a very flat curve until the reaction time times the lateral speed approaches the alert distance (i.e., until the vehicle is near the road edge or barrier).

Although not technically part of the CPB, we can also use the CPB equations to determine the sensitivity of the required deceleration or lateral acceleration to sensing errors or uncertainty. By taking the derivative with respect to the warning distance, x_w, in equation (1), we can determine the sensitivity of the required acceleration to uncertainty in the warning distance:

$$\frac{d_{req}}{dx} = -\frac{d_{req\_nom}}{x - t_r v_o} \quad \textbf{(5)}$$

where d_{req_nom} is the nominal required deceleration from equation (1). An initial warning distance of, say, x = 50 m decreases the sensitivity of the required deceleration to errors in the warning distance by a factor of 50. The effect of these errors, however, becomes more pronounced as the reaction distance (initial speed times the time required to decelerate) approaches the warning distance.

The sensitivity of the required lateral acceleration to errors in the lateral distance has a form identical to equation (5):

$$\frac{a_{req}}{dy} = -\frac{a_{req\_nom}}{y - t_r v \theta} \quad \textbf{(6)}$$

In this case both the lateral warning-position errors and the warning position *(y)* will be smaller than their longitudinal counterparts, so the sensitivity of the required lateral acceleration to lateral sensing errors (or uncertainty) is low, at least when the warning is issued. As the denominator in equation (6) decreases, the sensitivity, of course, increases.

5.1 *Lateral Sensing*

The lateral CPB curves in Figure 9 are highly dependent on the estimated lateral speed and the estimated distance from the road edge (or barrier). The vertical asymptotes occur when the product of the reaction time and lateral speed equals the warning distance. Although there are a number of sensing issues, we will call attention to only one of them: the lateral distance used to issue a warning. The CPBs in Figure 9 result from using the *actual* distance between the vehicle's front bumper and the barrier in the CPB equation (3). The variation in the CPBs results from variations in the initial conditions as well as variations in the lateral-distance estimates. We believe, however, that a reasonable range of initial conditions should not produce variations in CPBs. Drivers should be warned consistently, with the understanding: "You need to steer out of this with a timing and level consistent with how you adjusted the sensitivity." A wide variation in the required timing and level, which will result from improper sensing or processing, violates this understanding and may lead to driver dissatisfaction. The CPB is a useful tool for showing this variation, or lack thereof, for a series of warnings and conditions.

5.2 *Extensions*

As indicated in Section 3, the longitudinal CPB in this paper is not a true CPB because it merely represents the required deceleration to negotiate a curve at a predetermined safe speed. Beyond this issue, however, lies the larger problem of describing a general CPB for negotiating a curve safely. In our tests, we observed that after receiving the alert the driver decelerated and continued decelerating while turning. Other efforts (and expected behavior) demonstrate this same trend: drivers brake before the curve and continue to brake while negotiating curves, at least at the curve entrance. In addition, drivers often "cut" the curve, further complicating efforts to model vehicle motion and crash boundaries. Given that (1) the longitudinal CPB in Section 3 is not a true crash prevention boundary, (2) drivers may exceed the safe speed when entering a curve without necessarily losing control, and (3) drivers cut curves, we see some limitations in applying the existing analyses to describing the CPB of a given curve speed warning. The limitation suggests the need for a more comprehensive CPB, an effort we are pursuing.

6 References

[1] Burgett, A., Gunderson, K., “Crash Prevention Boundary for Road Departure Crashes-Derivation”, DOT HS 809 399, September 2001.

[2] Wilson, B., Burgett, A, “Crash Prevention Boundaries for Road Departure”, Ninth World Congress on Intelligent Transportation Systems, Chicago, IL.October, 2002.

[3] Szabo, S. Norcross, R., “Recommendations for Objective Test Procedures for Road Departure Crash Warning Systems”, NIST report under Agreement Number DTFH61-00-Y-30132, February 18, 2003.

[4] Deal, F.C., Men, H., Phamdo, N., “Single Vehicle Roadway Departure Prevention Program – Public Curve Speed Warning (CSW) and Lateral Drift Warning (LDW) Algorithms Design and Performance”, Draft report prepared by Johns Hopkins Applied Physics Laboratory, April 2004.

[5] Wilson, B., “How Soon To Brake and How Hard To Brake: Unified Analysis of the Envelope of Opportunity for Rear-End Collision Warning Systems,” 17th International Technical Conference on the Enhanced Safety of Vehicles, Amsterdam. June, 2001

PRIDE: A Framework for Performance Evaluation of Intelligent Vehicles in Dynamic, On-Road Environments

Craig Schlenoff, Jerome Ajot, and Raj Madhavan
National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

We are developing a novel framework, PRIDE (PRediction In Dynamic Environments), to perform moving object prediction for unmanned ground vehicles. The underlying concept is based upon a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. The lower levels of the framework utilize estimation-theoretic short-term predictions while the upper levels utilize a probabilistic prediction approach based on situation recognition with an underlying cost model.

In addition to predicting the location of moving objects in the environment, we have extended PRIDE to generate simulated traffic flow during on-road driving. In this paper, we explore applying the PRIDE-based traffic control algorithms for the purpose of performance evaluation of autonomous vehicles. Through the use of repeatable and realistic traffic flow simulation, one is able to evaluate the performance of an autonomous vehicle in an on-road driving scenario without the risk involved with introducing the vehicle into a potentially dangerous roadway situation. In addition, by varying a single vehicle's parameters (e.g. aggressivity, speed, location) with the traffic flow, we can show how the entire traffic pattern is affected. We will show the successes that have been achieved to date in a simulated environment, as well as enhancements that are currently being researched and expected in the near future.

KEYWORDS: *autonomous vehicle, on-road driving, traffic simulation, performance metrics, PRIDE*

1.0 Introduction/Problem Statement

The field of autonomous systems is continuing to gain traction both with researchers and practitioners. Funding for research is this area has continued to grow over the past few years, and recent high profile funding opportunities have started to push theoretical research efforts into practical use. Autonomous systems in this context refer to embodied intelligent systems that can operate fairly independently from human supervision.

Many believe that the DEMO III Experimental Unmanned Vehicle (XUV) effort represents the state of the art in autonomous off-road driving [10]. This effort seeks to develop and demonstrate new and evolving autonomous vehicle technology, emphasizing perception, navigation, intelligent system architecture, and planning. It should be noted the DEMO-III XUV has only been tested in static environments. It has not been tested in on-road driving situations, which include pedestrians and oncoming traffic.

There have been experiments performed with autonomous vehicles during on-road navigation. Perhaps the most successful has been that of Dickmanns [4] as part of the European Prometheus project in which the autonomous vehicle performed a trip from Munich to Odense (over 1,600 kilometers) at a maximum velocity of 180 km/hr. Although the vehicle was able to identify and track other moving vehicles in the environment, it could only make basic predictions of where those vehicles were expected to be at points in the future, considering the vehicle's current velocity and acceleration.

What is missing from all of these experiments is a level of situation awareness of how other vehicles in the environment are expected to behave considering the situation in which they find themselves. When humans drive, we often have expectations of how each object in the environment is expected to move based upon the situation. When a vehicle is approaching an object that is stopped in the road, we expect it to slow down behind the object or try to pass it. When we see a vehicle with its blinker on, we expect it to turn or change lanes. When we see a vehicle traveling behind another vehicle at a constant speed, we expect it to continue traveling at that speed. The decisions that we make in our vehicle are largely a function of the assumptions we make about the behavior of other vehicles.

To date, the authors are not aware of any autonomous vehicle efforts that account for this information when performing path planning. To address this need, we have developed a framework, called PRIDE (PRediction in Dynamic Environments) that provides an autonomous vehicle's planning system with information that it needs to perform path planning in the presence of moving objects [8]. In this paper, we describe how we leveraged the algorithms in the PRIDE framework to simulate traffic patterns during on-road driving. We can then use these simulated traffic patterns to control vehicles in the environment in an on-road driving arena being developed at NIST [9] in a way to assess the performance of autonomous vehicles being tested within these arenas.

In Section 2, we survey some related work in traffic simulation. In Section 3, we describe the various components of the PRIDE framework. In Section 4, we show how the PRIDE framework can be applied to traffic simulation for on-road driving. In Section 5, we explain how this traffic simulation can be used to apply

performance metrics for autonomous vehicles. Section 6 concludes the paper.

2. RELATED WORK

Most of the work in the literature dealing with drivers' actions and predicted behavior has been performed by psychologists in an attempt to explain drivers' behaviors and to identify the reasons of certain dysfunctions.

There have been a few efforts that have tried to simulate traffic patterns. One of more prominent ones in the literature is ARCHISM [3,5], but even this effort is based upon driving psychology studies. These traffic simulations use laws that can be applied for a specific environment or a specific situation. Some of those postulates can be expand to generic situations but still attached to a kind of situation.

3. THE PRIDE FRAMEWORK

We are using the 4D/RCS (Real-Time Control System) reference model architecture [1] as the basis in which to apply the representational approaches that are being developed in this effort. 4D/RCS was chosen due to its explicit and well-defined world modeling capabilities and interfaces, as well as its multi-resolution, hierarchical planning approach. Specifically, 4D/RCS allows for planning at multiple levels of abstraction, using different planning approaches as well as utilizing inherently different world model representation requirements. By applying this architecture, we can ensure that the representations being developed for representing moving objects can accommodate different types of planners that have different representational requirements.

The RCS architecture supports multiple behavior generation (BG) systems working cooperatively to compute a final plan for the autonomous system. The spatial and temporal resolution of the individual BG systems along with the amount of time allowed for each BG system to compute a solution are specified by the level of the architecture where it resides. In addition to multiple BG systems, multiple world models are supported with each world model's content being tailored to the systems that it supports (in this case the BG system). As such, it is necessary for moving objects to be represented differently at the different levels of the architecture.

To support this requirement, NIST has developed the PRIDE (PRediction In Dynamic Environments) framework. The underlying concept is based upon a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. This framework supports the prediction of the future location of moving objects at various levels of resolution, thus providing prediction information at the frequency and level of abstraction necessary for planners at different levels within the hierarchy. To date, two prediction approaches have been applied to this framework.

At the lowers levels, we utilize estimation theoretic short-term predictions via an extended Kalman filter-based algorithm using sensor data to predict the future location of moving objects with an associated confidence measure. Estimation-theoretic schemes using Kalman Filters (KFs) are well established recursive state estimation techniques where estimates the states of a system are computed using the process and observation models [6]. The recursive nature of the algorithm utilizes the system's CPU more uniformly to provide estimates without the latency resulting from batch processing techniques. The (linear) KF is simply a recursive estimation algorithm that provides minimum mean squared estimates (MMSE) of the states of a linear system utilizing knowledge about the process and measurement dynamics, process and measurement noise statistics subject to Gaussian assumptions and initial condition information. When these assumptions are satisfied, the estimates provided by the Kalman filter are *optimal*. The extension of the linear Kalman filtering ideas to a non-linear system is termed extended Kalman filtering.

The Extended Kalman Filter (EKF) is a linear estimator for a non-linear system obtained by *linearization* of the nonlinear state and observation equations. For any non-linear system, the EKF is the best linear unbiased estimator with respect to minimum mean squared error criteria. Within the on-road driving hierarchy, short-term prediction of objects moving at variable speeds and at given look-ahead time instants are predicted using the EKF. More information about this approach can be found in [7].

At the higher levels of the framework, moving object prediction needs to occur at a much lower frequency and a greater level of inaccuracy is tolerable. At these levels, moving objects are identified as far as the sensors can detect, and a determination is made as to which objects should be classified as "objects of interest". In this context, an object of interest is an object that has a possibility of affecting our path in the time horizon in which we are planning. At this level, we use a moving object prediction approach based on situation recognition and probabilistic prediction algorithms to predict where we expect that object to be at various time steps into the future. Situation recognition is performed using spatio-temporal reasoning and pattern matching with an *a priori* database of situations that are expected to be seen in the environment. In these algorithms, we are typically looking at planning horizons on the order of tens of seconds into the future with plan steps at about one second intervals. At this level, we are not looking to predict the exact location of the moving object. Instead, we are attempting to characterize the types of actions we expect the moving object to take and the approximate location the moving object would be in if it took that action. More information about this approach is included in the follow section.

Active research is exploring the integration of these two prediction approaches in a way that the predictions from one can help to enforce or not enforce the predictions of the other.

Both of these prediction methods have been implemented in two different simulation environments. The EKF approach has been implemented in the OneSaf testbed (www.onesaf.com). OneSaf is a composable, next generation computer generated forces (CGF) that represents a full range of operations, systems, and control process from individual combatant and platform to battalion level, with a variable level of fidelity. OneSaf is able to represent moving objects and provide the object's location and velocity at any point in time, through custom-developed Application Programmers' Interface (API) calls. A user-interface was built on top of OneSaf to display the predicted locations of the moving objects.

In Figure 1, the triangle represents the moving object whose future location is to be predicted. The large circle in front of the triangle is the area in which we are 99 % confident that the object will be in two seconds and the small shaded circle is the area in which we are 50 % confident that the object will be in two seconds. For our implementation, we found that the EKF provided reasonable predictions within a two second horizon. A horizon greater than two seconds introduced too much uncertainty to be useful for our autonomous driving scenarios.

The situation-based probabilistic prediction approach has been implemented in the AutoSim simulation package developed by Advanced Technology Research Corporation[i]. AutoSim is a high-fidelity simulation tool which models details about road networks, including individual lanes, lane markings, intersections, legal intersection traversibility, etc. Using this package, we have simulated typical traffic situations (e.g., multiple cars negotiating around obstacles in the roadway, bi-directional opposing traffic, etc.) and have predicted the future location of individual vehicles on the roadway based upon the prediction of where other vehicles are expected to be.

At the point this paper was written, we have simulated a handful of driving situations and have used approximately a dozen costs to determine the probabilities of one action over another. In this context, a cost is a penalty that is incurred by performing a maneuver or occupying a state. Current costs are incurred based on: 1) proximity to other objects in the environment as a function of necessary stopping distance, 2) exceeding or going below the speed limit by a given threshold, 3) changing lanes, 4) not being in the rightmost lane, 5) rapidly accelerating or decelerating, and 6) changing lanes where double yellow lines in the road exist, among other costs.

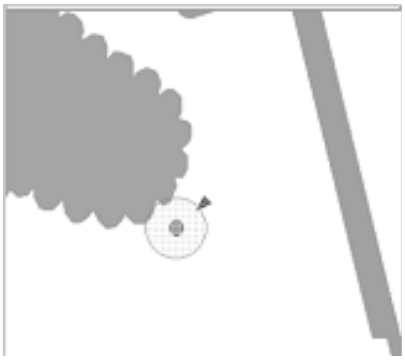


Figure 1: Short-term Prediction.

It should be emphasized that costs are not static numbers. The cost that a vehicle incurs by taking an action is heavily a function of the perceived personality and intention of the moving objects. Using these costs, we are able to predict up to ten seconds into the future at a rate of two predictions per second. A snapshot of the implementation is shown in Figure 2 (a) and (b).

4. APPLYING THE PRIDE FRAMEWORK TO TRAFFIC SIMULATION

Although the PRIDE framework was originally developed to inform a planner about the future position of moving

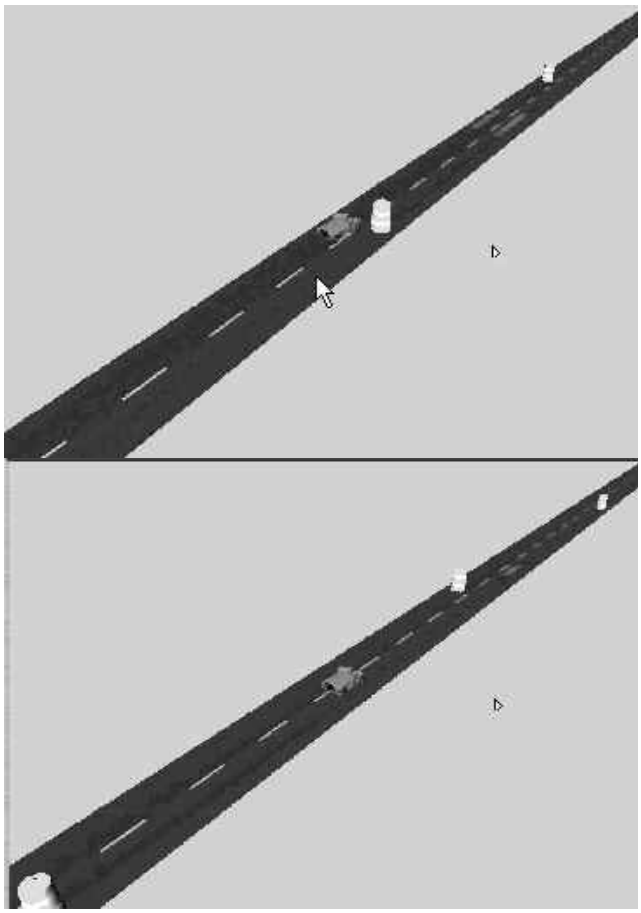


Figure 2: Situation-based probablistic prediction. (a) above and (b) below show a vehicle performing a passing operation around stationary obstacles.

objects for the purpose of path planning and collision avoidance, we have found that the same set of algorithms could be applied to simulating traffic patterns during on-road driving. More specifically, we applied the situation recognition and probabilistic algorithms to determine the likely actions that a vehicle in the environment would take when confronted with a specific situation, and then command that vehicle to perform that action. By doing this with multiple vehicles, we are able to simulate fairly sophisticated traffic situations in which vehicles behave in a way that is very similar to how a human would behave. Vehicles will slow down and/or pass when approaching a stopped or slow object in their lane, they will typically only change lanes when the next lane is clear and they are going slower than desired, they will keep a safe follow distance, etc. By providing realistic simulations of traffic situations, we are able to test the autonomous vehicle during realistic on-road driving situations, without having to place the vehicle on a potentially dangerous city street or highway.

The vision is that we will use these algorithms originally in simulated environments (such as the OneSaf and AutoSim simulation environments discussed above) to test out the planning algorithms in the presence of moving objects. Then, when the NIST On-Road Driving Arenas (which are described in another paper in this conference), are completed, these algorithms will be used to control "environmental" vehicles in the arena to simulate on-road traffic.

The remainder of this section describes the details of how the situation recognition and probabilistic algorithms are used to simulate on-road traffic.

The basic assumption behind this situation-based probabilistic prediction approach is a driver's behavior can be quantified using costs. In general, a driver will prefer an action that minimizes its cost. With this assumption, the cost can be converted to probabilities, where the higher the cost, the lower the probability that the driver will execute that action.

4.1. POSSIBLE VEHICLE ACTIONS

For the purpose of the algorithms, we have discretized the possible actions that a vehicle can make at any given time. Note that for this exercise, we have not accounted for intersections. All of the examples occur on a continuous stretch of roadway.

A vehicle can execute two types of actions. The first type of action is regarding its velocity, namely, quick acceleration (QA), slow acceleration (SA), constant velocity (CV), slow deceleration (SD) and quick deceleration (QD).

The second type of action is regarding lane changing, namely, changing to the left lane (CL), staying in the same lane (SL) and changing to the right lane (CR). These are shown in Figure 3.

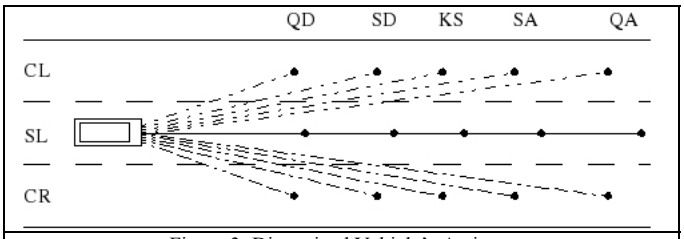


Figure 3: Discretized Vehicle's Actions.

As shown, at any point in time, the vehicle can have up to 15 possible future actions.

4.2. THE PRIDE ALGORITHMS

In this section, we will use the following scenario as shown in Figure 4 to explain the concepts in the algorithms. This scenario is composed by three vehicles, two (A and B) on the same lane (L1) and another one (C) on the opposite lane (L2) and a static obstacle (D) on the lane L1.

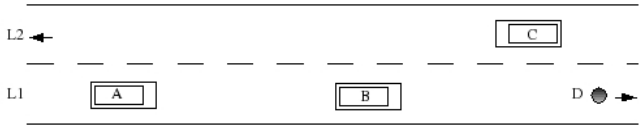
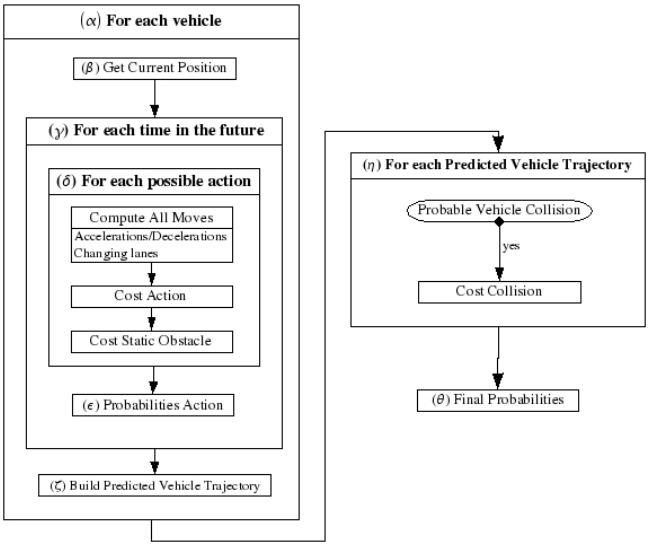


Figure 4: Scenario

Figure 5 shows the overall process for the algorithms (a) graphically and (b) in pseudo-code. The algorithm proceeds as follows:

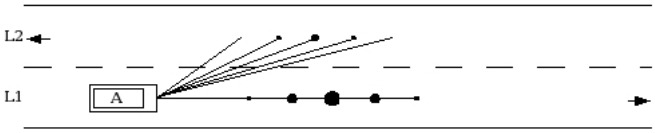
1. For each vehicle on the road (α), the algorithm gets the current position and velocity from external programs/sensors (β).
2. For each time in the future and for each starting position (χ), the algorithm creates a set of next possible positions in the future and assigns a cost to each of them corresponding to the action performed and the state occupied. At the end of the first prediction, each ending position is set at the starting positions at time *t* are used as "starting position" to build the next set of future position for time t+1. This loop (δ) is performed t_{final} iterations, where t_{final} is the predetermined time in the future that we wish to predict.
3. Using the costs found in step 2, the algorithm computes the probability for each movements of the vehicle (ε). In Figure 6, ten possible positions are shown for the vehicle (a) A, (b) B, and (c) C at time t_{final}.
4. The width of each dot in Figure 6 represents the relative probability of each action with respect to the others, where the bigger the dot, the higher the probability.




```
Begin
  Loop
    For each vehicle (α)
      Get Current Position (β)
      For each time in the future (χ)
        For each possible actions (δ)
          Compute All Moves
          Calculate Cost Action
          Calculate Cost Static Obstacle
        End for
        Calculate Probabilities Action (ε)
      End for
      Build Predicted Vehicle Trajectory (ξ)
    End for
    For each Predicted Vehicle Trajectory (η)
      If Probable Vehicle Collision
        Then Calculate Cost Collision
      End if
    End for
    Calculate Final Probabilities (θ)
  End Loop
End.
```

Figure 5 (a) (above) schematic of overall process, (b) (below) pseudo code

5. The algorithm then builds the Predicted Vehicle Trajectories (ξ) which will be used to evaluate the possibly of colliding with another vehicle. The Predicted Vehicle Trajectory notion is explained later in this document.
6. For each pair of Predicted Vehicle Trajectory (η), the algorithm checks if there is a probable collision and assigns a cost to the collision.
7. In the example, for the vehicles A and C, Figure 7 shows that one possible trajectory for A intersects a possible trajectory for C, thus resulting in a collision.



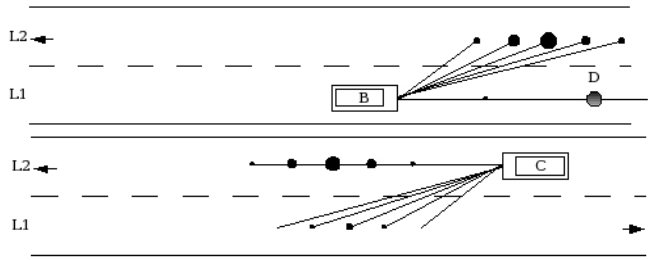


Figure 6: Predicted Positions of (a) (top) Vehicle A, (b) (middle) Vehicle B, and (c) (bottom) Vehicle C.

8. The probabilities (θ) based on the costs of each action are recalculated based upon the collision information. Figure 8 shows the final vehicle probabilities for Vehicles A, B, and C. Notice how the size of the dots have changed from Figure 5, thus accounting for the cost due to possible collisions.

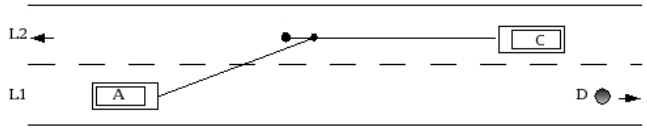


Figure 7: Possible Collision Between A and C

At the end of the main loop, the path with the highest probability for each vehicle represents the most likely location where the vehicles will be in t_{final} in the future.

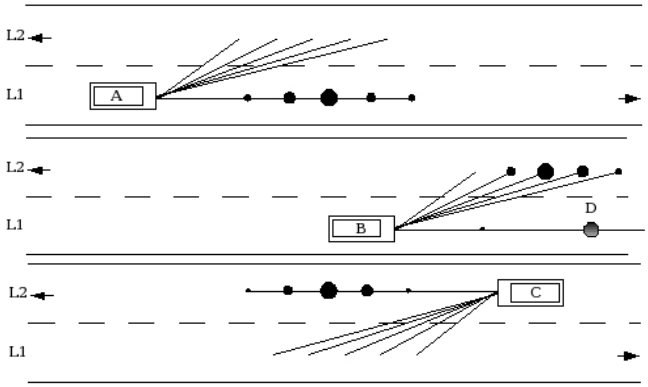


Figure 8: Final Vehicle Probabilities for (a) Vehicle A, (b) Vehicle B, and (c) Vehicle C.

4.3. COST MODEL AND PROBABILITY

The moving object prediction (MOP) algorithms can be separated in two parts, the first one is the creation of a set a predicted positions for each vehicle and the second is the evaluation of the interaction between each vehicle on the road. Every evaluation is based on costs that are converted to probabilities.

The Cost Model (CM) assigns danger ratings (cost) to each action that a vehicle can perform and the states that it

occupies after performing that action. This exact cost assigned to each action and state depends upon the aggressivity of the driver.

The cost of an action is the sum of the costs that are encountered by performing that action, which could include a cost for changing lanes, a cost for accelerating, a cost for going over the speed limit, etc. The cost of the state is described in the next section. This approach for building costs is based upon work performed in [2].

4.4. PREDICTED VEHICLE TRAJECTORY

The Predicted Vehicle Trajectory (PVT) represents the possible movements of a vehicle throughout the time period being analyzed. The PVT is a representation of the trajectory, as shown in Figure 9.

Figure 9: Predicted Vehicle Trajectory.

The PVT is built with the origin position $\{x_{IP}, y_{IP}, t_{IP} = 0\}$ at time = 0 and the predicted position $\{x_{PP}, y_{PP}, t_{PP} = t_{final}\}$ where t_{final} is the predetermined time in the future for the prediction process. It also contains the action-cost and action-probability information.

The PVT is used to determine possible collisions. Because the PVT represents a trajectory of one vehicle (origin to predicted), we can determine possible collision between any two vehicles by determining if two PVTs cross.

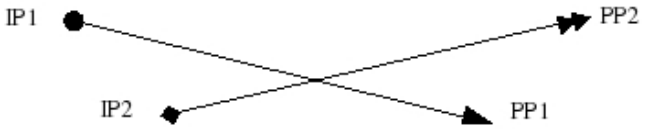


Figure 10: Two PVTs Crossing.

When two PVTs are crossing each other (Figure 10), it is important to know where are they crossing. This information can be obtained by using a parametrization of each PVT.

The parametrization:

$$\begin{aligned} x_1(t_1) &= x_{PP1}t_1 + x_{IP1}(1-t_1) \\ y_1(t_1) &= y_{PP1}t_1 + y_{IP1}(1-t_1) \end{aligned} \text{ where } t_1 \in [0,1]$$

$$\begin{aligned} x_2(t_2) &= x_{PP2}t_2 + x_{IP2}(1-t_2) \\ y_2(t_2) &= y_{PP2}t_2 + y_{IP2}(1-t_2) \end{aligned} \text{ where } t_2 \in [0,1]$$

where t_1 and t_2 are the parameters of each PVT.

By using the Theorem of Cramer, t_1 and t_2 can be determined:

$$\begin{aligned} x_{PP1}t_1 + x_{IP1}(1-t_1) &= x_{PP2}t_2 + x_{IP2}(1-t_2) \\ y_{PP1}t_1 + y_{IP1}(1-t_1) &= y_{PP2}t_2 + y_{IP2}(1-t_2) \\ (x_{PP1} - x_{IP1})t_1 + (x_{IP2} - x_{PP2})t_2 &= x_{IP2} - x_{IP1} \\ (y_{PP1} - y_{IP1})t_1 + (y_{IP2} - y_{PP2})t_2 &= y_{IP2} - y_{IP1} \end{aligned}$$

$$t_1 = \frac{\begin{vmatrix} x_{IP2} - x_{IP1} & x_{IP2} - x_{PP2} \\ y_{IP2} - y_{IP1} & y_{IP2} - y_{PP2} \end{vmatrix}}{\begin{vmatrix} x_{PP1} - x_{IP1} & x_{IP2} - x_{PP2} \\ y_{PP1} - y_{IP1} & y_{IP2} - y_{PP2} \end{vmatrix}}$$

$$t_2 = \frac{\begin{vmatrix} x_{PP1} - x_{IP1} & x_{IP2} - x_{IP1} \\ y_{PP1} - y_{IP1} & y_{IP2} - y_{IP1} \end{vmatrix}}{\begin{vmatrix} x_{PP1} - x_{IP1} & x_{IP2} - x_{PP2} \\ y_{PP1} - y_{IP1} & y_{IP2} - y_{PP2} \end{vmatrix}}$$

So the two vehicles will cross each other at two different times $t_1 t_{final}$ for the first vehicle and $t_2 t_{final}$ for the second one. For a small difference between the two time, the collision is highly likely. Conversely, if the difference is large collision is improbable.

The inverse of the difference between the two collision times represents the coefficient of the collision cost:

$$CollisionCost = \frac{M}{t_{final}\left|t_1 - t_2\right|}$$

where M is a preset of the Cost Model for collision costs.

5. APPLYING TRAFFIC SIMULATION TO PERFORMANCE METRICS

Now that we've described how we can simulate traffic patterns, we will discuss how this could be used to associate performance metrics to an autonomous vehicle. In evaluating how an autonomous vehicle is performing during on-road driving, we need the ability to test that vehicle in various driving situations. Those situations could be a function of the environment (e.g., winding roads, steep slopes, traffic signals, intersections), weather conditions (e.g., rain, fog, ice on the roadway), and static and dynamic objects in the environment (e.g., traffic barrels, pedestrians, other vehicles). The traffic simulator allows us the ability to dynamically change information about static and dynamic objects in the environment in order to introduce a variety of situations that we can evaluate the autonomous vehicle against.

The traffic simulator allows one to have repeatable, realistic traffic patterns that only vary in response to the autonomous vehicle motions. As such, one would be able to place two different autonomous vehicles in an identical traffic environment to evaluate how each performs. If the two autonomous vehicles behaved in identical fashion, the entire flow of traffic would be identical. Conversely, if the two autonomous vehicles' behaviors differed in any way, the flow of traffic would most likely differ (since other vehicles in the traffic pattern may be reacting to the actions of the autonomous vehicle). Metrics could be assigned to the autonomous vehicle's actions, based on a number of criteria, including proximity to other vehicles, staying within the speed limit, number of lane changes, obeying traffic signs and signals, etc.

The traffic simulator also allows for the ability to vary the aggressivity of drivers on the road. Using this capability, we could have different difficulties of driving scenarios, based, in part, on the aggressivity of the drivers on the road, where the higher the aggressivity, the harder the course. One could even imagine situations where an accident among a pair of traffic vehicles is imminent. The autonomous vehicle could then be evaluated based upon its ability to predict this accident and take precautions in time.

6. CONCLUSION

In this paper, we have described the PRIDE framework, which was developed to perform moving object prediction for unmanned ground vehicles. PRIDE is based upon a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. The lower levels of the framework (not discussed in detail in the paper) utilize estimation-theoretic short-term predictions while the upper levels utilize a probabilistic prediction approach based on situation recognition with an underlying cost model. We showed the results achieved from applying PRIDE in a simulated environment.

We have then shown how PRIDE can be extended to simulate traffic patterns during on-road navigation. The PRIDE algorithms are used to provide the underlying logic to control vehicles in the environment, thus generating a realistic flow of traffic. We use the prediction algorithms within PRIDE to determine the probability that a vehicle will exhibit a certain behavior given a set of environmental conditions, and then command the vehicle to perform the action which has the greatest probability. By doing this, we are able to create realistic, repeatable, and non-scripted traffic patterns that closely mimic the types of traffic flow expected to be encountered during on-road driving.

We then explore how the PRIDE-based traffic control algorithms can be applied to performance evaluation of autonomous vehicles. Through the use of repeatable and realistic traffic flow simulation, one is able to evaluate the performance of an autonomous vehicle in an on-road driving scenario without the risk involved with introducing the vehicle into a potentially dangerous roadway situation. In addition, by varying a single vehicle's parameters (e.g. aggressivity, speed, location) with the traffic flow, we can show how the entire traffic pattern is affected.

The goal of this paper was to describe how the work to date in moving object prediction could contribute to performance metrics for autonomous systems. Though the algorithms described in this paper have been developed and implemented, there is still much work to be accomplished. For one, the metrics that should be applied when evaluating an autonomous system in the presence of traffic situations being generated by these algorithms need to be determined. This will be the topic of future work and will be realized in the NIST On-Road Driving Arenas mentioned earlier in the paper. Also, these set of algorithms have been shown to work successfully on straight roads, but have not been fully tested on intersections. This will also be a topic of future research.

REFERENCES

1. Albus, J. and et.al., "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," NISTIR 6910, National Institute of Standards and Technology, Gaithersburg, MD, 2002.
2. Balakirsky, S., *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Spaces*, IOS Press, Berlin, 2003.
3. Champion, A., Espie, S., and Auberlet, J., "Behavioral Road Traffic Simulation with ARCHISM," *Proceedings of the Summer Computer Simulation Conference, USA*, 2001.
4. Dickmanns, E. D., "An Expectation-Based Multi-Focal Saccadic (EMS) Vision System for Vehicle Guidance," *Proceedings of the 9th International*

Symposium on Robotics Research (ISRR'99), Salt Lake City, 1999.

5. Espie, S., Saad, F., and Schnetler, B., "Microscopic traffic simulation and driver behaviro modeling: the ARCHISM project," *Proceedings of the Strategic Highway Research Program and Traffic Safety on Two Continents*, Lille, France, 1994.
6. Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35-45.
7. Madhavan, R. and Schlenoff, C., "The Effect of Process Models on Short-term Prediction of Moving Objects for Unmanned Ground Vehicles," *Submitted to the 7th International IEEE Conference on Intelligent Transportation Systems*, Washington DC, 2004.
8. Schlenoff, C., Madhavan, R., and Barbera, T., "A Hierarchical, Multi-Resolutional Moving Object Prediction Approach for Autonomous On-Road Driving," *Proceedings of the 2004 ICRA Conference*, 2004.
9. Scrapper, C., Balakirsky, S., and Weiss, B., "Autonomous Road Driving Arenas for Perfomance Evaluation," *To be published in the Proceedings of the Perfomance Metrics for Autonomous Systems (PerMIS) 2004 workshop*, 2004.
10. Shoemaker, C. and Bornstein, J. A., "Overview of the Demo III UGV Program," *Proceedings of the SPIE Robotic and Semi-Robotic Ground Vehicle Technology Conference*, Vol. 3366, 1998, pp. 202-211.

---

[i] Certain software tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

Autonomous Road Driving Arenas for Performance Evaluation

Chris Scrapper, Stephen Balakirsky, Brian Weiss
National Institute of Standards and Technology
Intelligent Systems Division
Gaithersburg, Maryland 20899
Email: {scrapper, balakirsky, bweiss}@nist.gov

ABSTRACT— The development of performance metrics is critical in the evaluation and advancement of intelligent systems. Obtaining the pinnacle of intelligence in autonomous vehicles requires evolutionary standards and community support. In order to analyze and compare competing implementations of intelligent systems, the critical components of the system must be decoupled to facilitate repeatable trials that target specific aspects of the system's overall task. This paper presents a framework for a real virtual simulation environment that provides the facilities and tools to formally test the limitations and capabilities of autonomous road driving vehicles.

Keywords:

Performance Metrics, Autonomous Road Driving, Real Virtual Simulation, MOAST, RCS

> "A major barrier to the development of intelligent systems is the lack of metrics and quantifiable measures of performance. There cannot be a science of intelligent systems without standard units of measure." [1]

1. Introduction

Autonomous vehicles have made remarkable advancements during the past few decades. Researchers have produced many systems in the quest to develop intelligent control architectures capable of various autonomy levels. These advancements have elevated the capabilities of autonomous driving and have been showcased in many competitions and events recently. One instance of such competitions is the Defense Advanced Research Projects Agency's (DARPA) Grand Challenge [2], which seeks to advance a variety of architectures and implementations by luring researchers to demonstrate their autonomous systems in a timed, autonomous ground vehicle "race". However, a task's parameters and constraints may dictate that high speed is not always indicative of successful performance by an autonomous vehicle (e.g., military operations may encourage stealth over speed). Thus, autonomous systems should produce a multitude of behaviors within the same dynamic environment based upon a particular set of constraints. The success of the architecture relies heavily on the ability of the intelligent system to be cognizant of its environment. To accomplish this, intelligent systems must use perception systems to gather situational information from non-deterministic, stochastic real world environments. Following detection and classification of the environment, the autonomous system selects an appropriate mode of operation. On-road driving is the mode of operation that is targeted in this paper.

As the pace of this technology continues to develop and even accelerates, performance metrics must be designed and refined to measure the effectiveness of autonomous systems. An archetypical system will emerge from the application of these metrics as the eventual pinnacle of autonomous road driving achievement. By this definition, an autonomous system will have the ability to produce rational behaviors allowing it to successfully achieve its goals in any dynamic, unstructured, and complex environment. In order to realize this archetypical system, critical architecture components must be de-coupled for independent evaluation. The evaluation of these components will allow evolutionary standards to be applied as necessary to help define the performance metrics, direct the focus of the community, and encourage collaborative efforts. Without community-acknowledged performance metrics and a complete comprehension of capabilities and limitations, it will be very difficult to assess and compare competing methods [3].

This paper describes the testing, development, and evaluation of autonomous vehicles. In Section 2, there is a brief discussion on elements of autonomous road driving and the critical components of the intelligent system that governs the vehicles behavior. In Section 3, concept for the testing and evaluation of autonomous road driving systems is presented. Section 4 briefly summarizes the technical rationale behind this effort.

2. Autonomous Road Driving

As mentioned above, the archetypical autonomous vehicle is capable of operating within its domain for extended periods of time without human intervention, acting in a manner that escalates the likelihood of successful completion of the system's intent or goals [4]. However, as [5] expresses, designing an outdoor mobile robot to follow a straight line is one thing; operating in a realistic environment is another.

Real outdoor environments are stochastic, non-deterministic domains that contain countably infinite combinations of percepts or sequences of situations [6]. Even road networks, which provide some structure in the form of lane markings, signs, and rules that govern the road, are still extremely complex [7]. The road networks can be seen as multi-agent heterogeneous environments that contain both static and dynamic obstacles.

The rationality of autonomous vehicles within any domain depends on the performance measures delineating the criteria for success, *a priori* knowledge of the domain, plausible actions that can be performed, and *in situ* knowledge perceived from the environment [6], [8]–[10]. This requires that an autonomous road driving vehicle be a goal-directed agent that is capable of applying situational awareness to these multi-agent environments. For example, the vehicle must be able to navigate intersections, avoid obstacles, and follow the rules of the road in any weather or environmental state. This highlights the need for an autonomous system to contain a general theoretical model of intelligence that permits the integration of knowledge, perception, and behavior generation into a unified framework [11].

Albus *et al.* [12, Chap. 2] provides a comprehensive overview of reference architectures for intelligent systems. Yet, the realization of an autonomous vehicle operating safely and efficiently in a realistic on road driving domain has not occurred. The ability for the mobile robot to "act rationally" [6] in dynamic non-deterministic environments is a central aspect to the evaluation of autonomous road vehicles. [13] attributes the formulation of behaviors or actions to the convergence of the ontological models, or knowledge, with sequences of percepts. Thus, posing the ability to gather knowledge in dynamic environments that is accurate, reliable and current is necessary for the development of rational behaviors [12], [14], [15]. This has led to the identification of several key subsystems of an autonomous road driving system that will serve as the initial test elements to foster evolutionary standards and performance metrics to achieve the archetypical autonomous system. The components of the autonomous system that will initially be targeted are high-level control, low-level control, and perception.

High-level control is an aspect of an autonomous vehicle that provides the system with long-term goals given constraints on the system. Constraints are parameters that are incorporated while generating behaviors that dictate the selection of the appropriate routes and maneuvers on the road, e.g., find the shortest path while obeying the rules of the road. Therefore, high-level control is comprised of a set of functions that use artifacts derived from a knowledge base concerning the rules of the road and the degree of aggressivity to formulate coarse path plans and elementary maneuvers that the vehicle must perform to achieve its ultimate goal. When an obstruction or obstacle is encountered that renders the existing path plan obsolete, the high-level control system must be able to re-evaluate the situation to replan alternative routes and/or maneuvers.

Low-level control is an aspect of an autonomous vehicle that provides the system with more immediate actions or short-term goals. The low-level control system uses a coarse path plan that is received from the high-level control system to generate trajectories. The trajectories provide the system with fine tuned aspects of autonomy such as lane control and obstacle avoidance. In short, low-level control provides the actuators of the vehicle with kinetically viable trajectories that do not endanger the vehicle or other elements within the environment [16]. Therefore, low-level control must account for the presence of both static and dynamic obstacles and/or changing conditions in the environment when generating these trajectories.

Outdoor environments in the real world present an enormous perception problem, given varying weather conditions, shadows, lighting conditions, etc. [5]. Perception systems give the autonomous system the ability to detect the edge of roads, read traffic signs, and track objects within the environment. Sensory information is processed by the perception systems in order to create a dynamic internal representation of the real world environment. This representation is combined with ontological models to extract knowledge to be used in the generation of appropriate actions. This extracted knowledge is then combined with *a priori* knowledge of the mode of operation to identify, recognize, and predict objects within a time horizon of the vehicle [12].

3. Program Concepts

The Intelligent System Division (ISD) at the National Institute of Standards and Technology (NIST) has long been a proponent of performance metrics for intelligent systems. Under the DARPA Mobile Autonomous Robot Software (MARS) program [17], [18], ISD is proposing to construct a set of Autonomous Road Driving Arenas (ARDA) to encourage the development of autonomous road driving vehicles and the creation of community recognized performance metrics for autonomous road driving. The sharing of knowledge by the community has facilitated rapid advancement of technologies and performance metrics within its domain. Therefore, this effort will attempt to gather the participation of academic, government, and commercial institutions through publications, technical reports, data sets, competitions, and accessibility to the arenas themselves.

This effort will draw on experience from the development and proliferation of the NIST Reference Test Arenas for Autonomous Mobile Robots [19] and the associated formal testing of urban search and rescue robots. The Urban Search And Rescue (USAR) arenas originally provided physical arenas to conduct controlled repeatable scenarios that challenge the different facets of the USAR robots. Since the inception of the USAR arenas, virtual components

and tools have been developed to simulate these arenas [20]. Under the Mobility Open Architecture Simulation and Tools (MOAST) [21], efforts are being made to seamlessly integrate the real and virtual simulation environments into a single simulation environment and toolset. In recent years, the arenas have proven to be an invaluable resource, lifting the standards of development within the community by providing comprehensive data sets, publicly accessible USAR arenas, and competitions such as RoboCup Rescue.

The Autonomous Road Driving Arenas (ARDA) will provide an environment that will support the cost effective development and testing of critical components of autonomous road driving vehicles. The ability to measure performance in terms of efficiency, effectiveness, precision, and reliability requires that the testing environment have the ability to independently evaluate the various aspects of autonomous road driving through various methods as detailed by [22]. MOAST provides the reference framework for the development of the facilities and tools to gather vital information during repeatable trials, which can be used to measure the performance of the autonomous vehicles both quantitatively and qualitatively.

ARDA will provide a realistic, scaled world that will evolve from simple lanes to a controlled dynamic environment that will exploit the capabilities and limitations of autonomous road driving vehicles. Therefore, a standardized platform will also be developed that will support these testing efforts by providing researchers with identical physical capabilities for their algorithms to perform against. It is important that ARDA provides the assurance that all participants experience the same degree of difficulty in order to measure and compare the performance of intelligent autonomous vehicles [1].

The capabilities of ARDA will emerge in three phases. Phase I will consist of a 2D planar arena with simple controlled intersections and scripted traffic. In Phase II, more complex environmental conditions, such as dynamic changing lighting conditions, and more challenging traffic conditions, such as construction zones, will be integrated into ARDA. Phase III will complete ARDA by adding non-planar road surfaces and other obscuring features such as bridges, hills, and tunnels. At the end of each phase, a technical report will be issued detailing the developmental life cycle of each particular phase, including feasibility analysis and technology assessment studies.

4. Technical Rationale

The rich world that exists within virtual simulation environments has long been an invaluable resource for the testing and development of intelligent systems. Traditionally, these environments were developed in-house for specific purposes [23]. More recently, there has been an emergence of game-based simulation environments that capitalize on the optimized rendering facilities and rich dynamic physic models contained in professionally developed game engines [24]. The Serious Games Initiative, USC's Institute for Creative Technologies, the MOVES Institute sponsored by the US military and many others, illustrate the rapid growth of the interest in the gaming community for robust simulation packages.

Though the rich virtual worlds contained within these simulation environments are invaluable, they are still not the "magic bullet". Algorithms that have been trained or developed within the virtual environments can become dependent on synthetic data cues. The majority of the virtual simulation environments are built on deterministic models that do not provide false alarms or missed detection and employ unrealistic mobility models. However, virtual environments do provide the flexibility and control in the environment to evaluate the capabilities and limitations of autonomous systems. This facilitates the ability to target specific aspects of autonomous systems by decoupling critical components and logging vital data of an autonomous system in repeatable trials.

Russell *et al.* [6] describes the environments contained in the real world as stochastic, non-deterministic, dynamic, multi-agent environments that are only partially observable to any one agent. When implementing autonomous systems on the actual platforms, the system must contend with real mobility characteristics of the vehicle's platforms. The system must also rely on real sensors that provide noisy data and missed detection. Thorpe *et al.* [5] express that these environments force the development of algorithms that are more robust and reliable in order for the system to contend with the realistic mobility characteristics of the actual platform and imperfect sensory data. However, the real world environments also have drawbacks. Since the planner relies heavily on the information obtained by the perception system, the performance of the system is greatly limited by what is known as the myopic planning effect [25]. In short, the myopic planning effect is the limitation of the planner to anticipate objects beyond the range of the vehicle mobility sensors, e.g. LADAR (laser radar). The inherent stochastic nature of real world environments makes it difficult to perform repeatable trials that have the same environmental state and conditions. Lastly, there is a great concern to adhering to Isaac Asimov's "Three Laws of Robotics" (safety to itself and others) when using the real world environments for the testing and evaluation of autonomous vehicles.

The technical rationale behind this effort is to create an environment that benefits from the best of both real and virtual worlds and to create an array of incremental steps that provide a smooth gradient to transition an autonomous system from a purely virtual world to a entirely real world. By allowing the virtual components and the real components to function where they perform best, ARDA seeks to achieve an optimal testing environment that seamlessly integrates both real and virtual components [21], [26], [27]. For example, Barbera *et al.* [28] describes a methodology

of reverse engineering performance metrics for sensors and perception systems by evaluating the world model and knowledge requirements needed for the behavior generation systems. Therefore, in the ARDA environment the use of virtual sensing can give us the opportunity to understand what is better needed from a perception system. ARDA will use the MOAST general-purpose framework to provide this effort with a developmental reference model during implementation.

MOAST manages the fidelity of the simulated worlds in order to reduce the computational overhead that is inherent in dynamic simulation environments. This is accomplished by the concurrent use of the real vehicle platform, a high-fidelity simulator and a low-fidelity simulator. An example of the implementation of the MOAST framework is to use the vehicle platform to run the low-level mobility, planning and low-level perception, using a high fidelity simulator to accurately simulate objects in the vehicles immediate sphere of influence, and using a low-level simulator for tracking, and simulating distant objects. Objects are automatically transferred between simulators as their proximity to the vehicles changes within the MOAST framework. This framework also permits the transparent transference of data between real and virtual components, which gives the developer the ability to toggle a "switch" between the two components. For example, virtual sensor that simulate raw sensory data or real sensors can be used as input into a perception system which in turn can either be an actual functioning subsystem or a virtual system that simple places preprocessed knowledge into the *WM*.

ARDA will provide an environment, with extensive high-resolution data sets of annotated maps, features, and *a priori* knowledge to implement the MOAST framework. This will provide ground truth and a benchmark for the testing, development, and evaluation of the intelligent systems in this specific domain. A scaled physical model of a road network will be used to initially evaluate high-level planning, low-level mobility, and low-level perception. Later, this environment will be used in the testing and development of multiple agent systems (MAS) [24], essential components such as various levels of Situational Awareness [29], and axioms of cooperation such as distributed artificial intelligence, resource conflicts, learning, group architectures, and geometric problems [30] .

ARDA will leverage existing technologies and on-going efforts in simulation to construct a simulation environment based on the MOAST framework consisting of four basic components: Real World Arenas, Virtual Simulation Worlds, Vehicle Platforms, and a Tracking System. The real world arenas will be the physical world where the vehicle platforms will be used to test various aspects of autonomous systems. The virtual simulation environment will reflect the real world environment and will utilize information captured by the tracking system in order to provide ARDA with an array of visualization tools and services. Tracking will also be used to provide navigation and other vital data to the agents operating within the arenas. These four components will be detailed further in the following subsections.

4.1. Real World Arenas

ARDA implements the MOAST framework in a one-tenth scaled replica of a real environment that includes objects and features such as plants, buildings and roads. The arena design will provide an equal level of difficulty for every autonomous system operating in ARDA. The arenas will target specific aspects of the autonomous system by mimicking the environmental conditions found in the on-road driving domain. These environmental conditions include static and dynamic objects, variable weather and lighting conditions, and features associated with road networks, e.g. controlled intersections and lane markings. The initial configuration of the arena will consist of a planar world with limited environmental conditions. More complex environmental conditions will be incrementally introduced in response to the escalating capabilities of the autonomous road driving systems.

Essential design factors that will continually be considered will be the use of commercially available components, modularity, reconfigurability, and durability. Commercially available components are standardized parts that are readily available and relatively inexpensive. The use of these components will assist in the proliferation of the arenas by allowing participants to construct or repair their own replicas according to ARDA's specs. The consistency of the standardized components expedites rapid repairs while maintaining the arena's integrity. A modular design also assists in efficient repairs through the easy replacement of damaged components. Reconfigurability is the ease with which environmental features can be added, modified, or deleted from the arenas. This gives ARDA the versatility to alter the configuration of its constituent components in order to target different aspects of the autonomy. Reconfigurability facilitates different road network implementations ranging from small demonstration configurations to large competition networks. Durability refers to the resilience of the arenas to the continual weathering due to the environmental conditions and the stress of assembly/disassembly and packing/unpacking associated with frequent travel.

With the design criteria in mind, ARDA will be composed of inter-locking rectangular frames with attached panels to create the base, upon which the road networks will be placed. The primary members of the road network frames will potentially be aluminum extrusion. Several types of commercially available aluminum extrusion are easily machined to provide unlimited attachment solutions. Plastic panels that are durable and lightweight will be tailored to conform to the frame and will provide the road surface and the foundation upon which terrain features will be placed. Upon the construction of the base, various terrain

objects and artificial turf will be used to populate the world. Lane markings will be painted on the road surface to create passing and non-passing zones. Controlled intersections will be composed of traffic signs and/or traffic lights. Traffic lights will be controlled through basic transistor logic or through interfaces to the virtual simulation controller.

4.2. Vehicle Platforms

All vehicle platforms that conform to the one-tenth scaled environment will be permitted to compete in ARDA. However, standardized platforms will be developed for use in ARDA that are akin to modified remote control cars (popular with hobbyists). For these platforms to behave rationally, the platforms must extend their computational capabilities to achieve the capabilities and computational specs currently available in an implementation of an actual autonomous vehicle, such as the NIST High Mobility Multipurpose Wheeled Vehicle (HMMWV) [31]. This requires that the platforms contain sufficient computational power, control architecture, sensors, and communication bandwidth. Initially, a few criteria will be used to evaluate the platform's design. The criteria identified to maximize the platform's capabilities are:

- Reference Control Architecture
- Standardized Parts
- Computational Power
- Battery-Life vs. Power Consumption
- Communication
- Sensors
- Closed-Loop Controller for Actuators, Sensors, and Communication

1) Reference Control Architecture: It is critical to identify a reference model architecture to organize the hardware and software within an intelligent system. The Real-Time Control System architecture, RCS, was chosen for the platforms in the ARDA because it uses a modular, hierarchical control architecture as an efficient way to manage the system's complexity. RCS is a real-time, distributed, hierarchical architecture that consists of computational nodes with well-defined functional components and clearly defined interfaces. RCS contains a systematic regularity so that each control node in the hierarchy performs the same general type of functions: sensory processing (*SP*), world modeling (*WM*), value judgment (*VJ*), and behavior generation (*BG*).

The principal difference between control nodes at the same level is in the set of resources managed, while the principal difference between nodes at different levels is in the knowledge requirements and the fidelity of the planning space. This regularity in the structure enables flexibility in the system architecture that allows scaling of the system to any arbitrary size or level of complexity. [1]

Each level in RCS has a characteristic range and resolution in space and time, determined by the specific implementation requirements. Each level has characteristic tasks and plans, knowledge requirements, values, and rules for decision-making. Every module in each level has a limited span of control, a limited number of tasks to perform, a limited number of resources to manage, a limited number of skills to master, a limited planning horizon, and a limited amount of detail with which to cope [32].

2) Standardized Parts: The standardized vehicle platforms developed for ARDA will be modified one-tenth scaled model remote control (*RC*) cars. Standard off-the-shelf parts, e.g. chassis, wheels, suspension and servos/actuators, will provide the base hardware for the vehicle platforms. These parts for these RC cars are widely available, inexpensive, and standardized allowing for vehicles to be modified easily. This facilitates efficient repairs and construction of vehicles platforms, which is ideal for development and testing of this nature. This effort will capitalize on available internal programs, competitions, and partnerships to develop a standardized platform that will provide the optimal functionality within ARDA.

3) Computational Power: Each platform must have sufficient computational power to handle the computational load of the autonomous systems, e.g. low-level planner, high-level planner, and low-level perception systems. The MOAST framework and modular hierarchical design of RCS allows for standard low-power single board core-processing units to be coupled through the designated interfaces with off-board processing power to meet any computational load needed by the vehicle. This provides the environment the flexibility to distribute the autonomous system's components or subsystems between the onboard and off-board processing units.

4) Battery-life vs. Power Consumption: The vehicle platforms will not be tethered; therefore, their electrical systems must operate on battery power. This bounds the time that the vehicles will be able to operate in ARDA. This requires that battery packs provide a significant number of milli-amp hours (mAH) to effectively operate the vehicle's electrical system, without signifigant power loss, for a duration of time that is appropriate for testing. For example, the core-processing units available are 12 W systems, drawing 1.75 V. RC battery packs are available in a 6-cell battery back (9 V) that provide 3000 mAh, which equate to 3 Ah. Given the power requirements of the core-processing unit and additional TTL logic required, the system would draw one amp from the battery pack. This only gives the vehicle three hours of operation before the power supply would need to be recharged. This does not take into account the power requirements to operate the motor, servos, or additional hardware that may need to be added for sensing. Multiple batteries can be mounted on the platform that provides ample power for the vehicle's electrical system.

5) Communication: Again, the combination of the MOAST framework and RCS lends flexibility to distribute the components of the autonomous system between the onboard and off-board processing units. The interfaces between the distributed subsystems, command and con-

trol channels, require enough communication bandwidth to accommodate a wide range of command messages. The communication infrastructure of ARDA must also be able to handle the command and control interfaces for multiple vehicles and broadcast global information, such as navigation data.

6) Sensors: Intelligent systems gather knowledge about the state of the world and its relation to the world through sequential percepts captured through sensors. Sensing capabilities within the MOAST framework can either be obtained virtually or through real onboard sensors. These capabilities include the collection of navigation data, color video, and range finding information. In the scaled environment of ARDA, objects will remain proportionally scaled to their real world counterparts, requiring sub-centimeter resolution in the sensory information. During the first phase of the development, the vehicle platforms will rely heavily on virtual sensing. The vehicle platforms will be designed with appropriate interfaces that will facilitate the addition and/or re-configuration of real onboard sensors.

7) Closed-Loop Controller for Actuators, Sensors, and Communication: The architecture controlling the sensors, actuators and communications will rely on a core-processing unit for its computational requirements. Closed-loop adaptive controllers will provide the core-processing unit with control interfaces and added computational power to regulate its facilities. This provides each distinct facility with a closed communication control loop that is comprised of a command sent to the controller, feedback received from the controller, and data transference between the core-processing unit and the controlled facility. Each control loop will consist of a stacked I/O card that contains additional computational power to run small control programs.

4.3. Virtual Simulation Envirnoment

Within the MOAST architecture, virtual and real components are combined transparently to mutually augment each other in order to achieve an environment conducive for the evaluation of autonomous road driving vehicles. This effort requires the virtual simulation environment to be capable of handling the overhead of simulating complex, unstructured environments contained in road networks. The virtual environments will be required to provide multiple levels of resolution, e.g., in the sphere of influence around the vehicle the virtual simulator must have high fidelity, while the distant objects require less fidelity.

The virtual simulation environments must provide the facilities and interfaces to allow for the developer to create new worlds, provide virtual sensing capabilities, modify objects, and appropriate command and control channels to interface with all agents in both the real and virtual environments [33], [34]. The virtual world must also provide the facilities to log vital information in real-time and visualization tools to access and display the internal state of the vehicle during operation. These tools provide developers with a level of control that permits for the internal state of the vehicle to be modified. For example, a vehicle will perceive virtual objects as actual objects in the real world, where the virtual world will view the vehicle as another agent within the virtual world. Having these tools to visualize the internal state of the autonomous system in real-time gives the developer more insight into the decision making process and the causality of behaviors being produced.

The virtual simulation environments must also be able to provide realistic mobility characteristics, sensory data, and tactical behaviors for each vehicle. Dixon *et al.* [26] points out that a signifigent amount of low-level infrastructure is required to support the communication, control, and simulation of multiple agents in high-fidelity virtual environments. The low-level infrastructure is time-consuming and requires that the developer have a high-level of expertise to maintain the infrastructure [35]. It appears that there is a positive, non-linear correlation between the number of objects in the world and the computational load to simulate these objects. An effective way to manage the computational complexities within these virtual environments is to couple a high-fidelity simulator with a low-fidelity simulator. The high-fidelity simulation provides very accurate information about objects in a bounded region, where as the low-fidelity simulation simulates the behavior and sensing of a larger area of distant objects at a lower resolution. This provides the autonomous system with *in situ* knowledge at an appropriate resolution, while orchestrating the entire scenario at a lower, more efficient resolution.

4.4. Tracking System

A tracking system will be integrated in the arenas to collect real-time navigation data of all the subjects in the real world environment and broadcast this knowledge to all the constituents that comprise ARDA. This knowledge can serve as a virtual positioning sensor for each vehicle or can be coupled with *a priori* ground truth knowledge to provide other virtual sensing capabilities. It can provide redundant navigation information for each platform by being coupled with wheel encoders or other positioning sensors located on the actual vehicle. The tracking system logs this data so that is can be used for the evaluation of the behaviors generated by the autonomous vehicles. This positioning knowledge can be compared with positioning information logged from the vehicle to validate trajectories and coarse path plans. These options are being explored based upon the available platform sensors and current resolution, accuracy, and update rate limitations of available tracking systems.

Overhead, two-dimensional camera tracking is being considered as the initial tracking system due to the planar configuration in ARDA's real world environment. The technologies available in commercially available color video cameras/camcorders provide sufficient resolution and updates rates to serve as the overhead visual sensor for the

tracking system. Many publicly available vision algorithms can be mated with most overhead cameras to provide tracking update rates of 30 Hz or slightly greater. One such available software package is Mezzanine [36]. This highly competent overhead visual tracking software is publicly available at no cost and may be used with almost any video camera/camcorder. The software enables the camera to track objects by following color-coded fiducials placed on top of each autonomous vehicle platform. Mezzanine provides real-time translation of pixel location into world coordinates.

Other software is widely available for camera tracking including algorithms developed by the RoboCup Soccer Small-Size Teams for use in their annual competitions [37]. This league features teams of five robots playing soccer against one another where all necessary sensing and processing takes place off-board. Each team sets up an overhead camera and places their own color-coded fiducials on their robots. Teams utilize their own tracking algorithms so that they may locate the position and orientation of their robots, their opponents, along with the position of the ball. The off-board computational facilities process this data and send the appropriate commands back to their robots. Several teams have extensively documented their vision algorithms that have the potential to be used with ARDA [38]–[40].

5. Summary

Under the DARPA MARS project, the Intelligent Systems Division of the National Institute of Standards and Technology is proposing the construction of a real/virtual simulation environment for the performance evaluation of autonomous road driving vehicles, ARDA. The effort is seeking to coalesce together an autonomous road driving community that will support the development, testing, and evaluation of various implementations of autonomous systems. ARDA will be designed to support the MOAST framework, which will give the developers the ability to target specific aspects of autonomous road driving in repeatable scenarios. This effort hopes for the realization of an intelligent autonomous system that will service as the archetypal system, which all other systems will be modeled and compared.

6. Acknowledgement

This work was supported in part by the Army Research Lab's program in unmanned vehicles (PM. C. Shoemaker) and DARPA's Mobile Autonomous Robotic Software program (PM. D. Gage).

7. Product Disclaimer

Commercial equipment and materials are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the material or equipment identified are necessarily the best available for the purpose.

8. REFERENCES

[1] J. Albus, "Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems Workshop* , 2002.

[2] DARPA, "Grand Challenge," 2004. [Online]. Available: http://www.darpa.mil/grandchallenge

[3] J. Evans and E. Messina, "Performances Metrics for Intelligent Systems," in *Proceedings of the 2000 Performance Metrics for Intelligent Systems Workshop* , 2000.

[4] H.-M. Huang, "Terminology for Specifying the Autonomy LEvels for Unmanned Systems: Version 1.0," NIST, NISTIR 1011, 2004.

[5] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Toward Autonomous Driving: the CMU Navlab. I - Perception," *Expert, IEEE [see also IEEE Intelligent Systems]*, 1991.

[6] S. Russell and P. Norvig, *Artificial Inelligence: A Modern Approach*. Prentice-Hall, 1995.

[7] D. Pomerleau and T. Jochem, "Image Processor Drives Across America," *Photonics Spectra*, 1996.

[8] S. Balakirsky and A. Lacaze, "World Modeling and Behavior Generation for Autonomous Ground Vehicles," in *Proceedings of the IEEE Conference on Robotics and Automation*, 2000.

[9] S. Balakirsky and A. Lacaze, "Value-Driven Behavior Generation for an Autonomous Mobile Ground Robot," in *Proceedings of the SPIE 16th Annual Symposium on Aerospace—Defense Sensing, Simulation and Controls*, 2002.

[10] J. Albus, "Outline for a Theory of Intelligence," *IEEE Transaction on Systems Man and Cybernetics*, 1991.

[11] A. Meystel and J. Albus, *Intelligent Systems: Architecture, Design, and Control*, ser. Wiley Series on Intelligent Systems. John Wiley and Sons, Inc., 2002.

[12] J. Albus and A. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*, J. Albus and A. Meystel and L. Zadeh, Ed. John Wiley and Sons, Inc 2001, 2001.

[13] N. Howard, *The Theory of Intention Awareness in Tactical Military Intelligence: Reducing Uncertainty by Understanding*. First Book Publisher, 2002.

[14] S. Balakirsky, *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Space*. IOS Press, 2003.

[15] B. Krogh and C. Thorpe, "Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles," in *Proceedings of 1986 IEEE International Conference on Robotics and Automation (ICRA '86)*, April 1986, pp. 1664 – 1669.

[16] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Toward Autonomous Driving: the CMU Navlab. II - Architecture and Systems," *Expert, IEEE [see also IEEE Intelligent Systems]*, 1991.

[17] D. Gage, "Mobile Autonomous Robot Software (MARS)," 2003. [Online]. Available: http://www.darpa.mil/ipto/programs/mars/index.htm

[18] NIST, "DARPA Mobile Autonomous Robot Software," 2003. [Online]. Available: http://www.isd.mel.nist.gov/projects/darpa_mars

[19] A. Jacoff, B. Weiss, and E. Messina, "Evolution of a Performance Metric for Urban Search and Rescue Robots," in *Proceedings of the 2003 Performance Metrics for Intelligent Systems Workshop*, 2003.

[20] J. Wang, M. Lewis, and J. Gennari, "USAR: A Game Based Simulation for Teleoperation," in *Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society*, 2003.

[21] S. Balakirsky and E. Messina, "A Simulation Framework for Evaluating Mobile Robots," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems Workshop* , 2002.

[22] NIST, "DARPA Mobile Autonomous Robot Software," 2003. [Online]. Available: http://www.isd.mel.nist.gov/projects/darpa_mars/Perf_Metrics.html

[23] R. Sukthankar, D. Pomerleau, and C. Thorpe, "SHIVA: Simulated Highways for Intelligent Vehicle Algorithms," in *Proceedings of Intelligent Vehicles '95*, September 1995, pp. 332 – 337.

[24] R. Adobbati, A. Marshall, A. Scholer, S. Tejada, G. Kaminka, S. Schaffer, and C. Sollitto, "Gamebots: A 3D Virtual World Test-Bed For Multi-Agent Research ," in *Proceeding of the 2nd Workshop on Infrastructure for Agents, MAS, and Scalable MAS at Autonomous Agents*, 2001.

[25] B. Nabbe and M. Hebert, "Where and When to Look," in *IROS 2003*. IEEE, October 2003.

[26] K. Dixon, J. Dolan, W. Huang, C. Paredis, and P. Khosla, “RAVE: A Real and Virtual Environment for Multiple Mobile Robot Systems,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’99)*, vol. 3, October 1999, pp. 1360–1367.

[27] R. Vaughan, B. Gerkey, and A. Howard, “On Device Abstractions for Portable, Reusable Robot Code,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, ser. IROS 2003, 2003.

[28] A. Barbera, J. Horst, C. Schlenoff, E. Wallace, and D. Aha, “Developing World Model Data Specifications as Metrics for Sensory Processing for On-Road Tasks,” in *Proceedings of the 2003 Performance Metrics for Intelligent Systems Workshop* , 2003.

[29] M. Endsley and R. R. Hoffman, “The Sacagawea Principle,” *Intelligent Systems, IEEE*, 2002.

[30] Y. Cao, A. Fukunaga, and A. Kahng, “Cooperative Mobile Robotics: Antecedents and Direction,” *Autonomous Robots*, vol. 4, pp. 7–27, 1997.

[31] K. Murphy, M. Abrams, S. Balakirsky, D. Coombs, T. Hong, and S. Legowick, “Intelligent Control for Unmanned Vehicles,” in *Proceeding of the 2000 World Automation Congress Conference*, 2000.

[32] J. Albus, H. Huang, E. Messina, K. Murphy, M. Juberts, A. Lacaze, S. Balakirsky, M. Shneier, T. Hong, H. Scott, F. Proctor, W. Shackleford, J. Michaloski, A. Wavering, T. Kramer, N. Dagalakis, W. Rippey, K. Stouffer, S. Legowik, R. Bostleman, R. Norcross, A. Jacoff, S. Szabo, J. Falco, B. Bunch, J. Gilsinn, T. Chang, A. Meystel, A. Barbera, M. Fitzgerald, M. DelGiorno, and R. Finkelstein, “4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems,” NIST, NISTIR 6910, 2002.

[33] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems,” in *Proceedings of the International Conference on Advanced Robotics*, ser. ICRA 2003, 2003, pp. 317–323.

[34] G. A. Kaminka, M. M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. N. Marshall, A. Scholer, and S. Tejada, “Gamebots: A Flexible Test Bed for Multiagent Team Research,” *Communications of the ACM*, vol. 45, no. 1, pp. 43–45, January 2002.

[35] A. Jacoff, E. Messina, and J. Evans, “Performance Evaluation of Autonomous Mobile Robots,” *Industrial Robot: An International Journal*, 2002.

[36] USC Robotics Resesarch Lab, “Mezzanine: An Overhead Visual Object Tracker,” 2003. [Online]. Available: http://playerstage.sourceforge.net/mezzanine/mezzanine.html

[37] B. Browning, “Official RoboCup Small Size League,” 2003. [Online]. Available: http://www-2.cs.cmu.edu/ brettb/robocup

[38] A. Egorova, A. Gloye, A. Liers, R. Rojas, M. Schreiber, M. Simon, O. Tenchio, and F. Wiesel, “FU-Fighters 2003 (Global Vision),” in *RoboCup-2003 - Proceedings of the International Symposium*, 2003.

[39] J. Bruce, T. Balch, and M. M. Veloso, “Fast and inexpensive color image segmentation for interactive robots,” vol. 3, October 2000.

[40] J. Loomis, J. Palmer, and P. Pandit, “Performance Development of a Real-Time Vision System,” Master’s thesis, Cornell University, 2003.

Software Tools for Design and Performance Evaluation of Intelligent Systems

S. Joshi, M. Slicker, A. Surianarayan
Pathway Technologies Incorporated
Blue Bell, PA 19422

ABSTRACT

The objective of this effort was to leverage the principles of knowledge engineering in the ongoing development of a software tool for rapid design, simulation, prototyping and performance evaluation of *Intelligent Systems* in the Matlab/Simulink environment. In this paper, we have demonstrated the use of this software tool to design a cable robot, automatically generate the kinematic and dynamic relationships for this robot and develop an automatic on-line calibration scheme for this cable robot where traditional one-time or periodic calibration methods do not provide adequate measures of performance. Hence, from the standpoint of intelligent system design and performance metrics, through this example, we demonstrate the usefulness of leveraging the principles of knowledge engineering to develop domain specific knowledge. The models so developed can be used to evaluate the performance of such robotic systems and modified to improve the performance. For example, kinematic errors such as assembly errors are likely to be introduced in the construction; faults such as joint failures are likely to be introduced in the operation. Hence, automated on-line calibration of intelligent systems (such cable robots) becomes particularly important for continuous performance evaluation (positioning accuracy) and enhancement.

Keywords: *Intelligent system design, Cable robot, Kinematic modeling, optimization*

1. INTRODUCTION

The NIST intelligent systems division (ISD) has employed a generic shell approach to facilitate the development of intelligent real-time control (RCS) systems. The RCS approach/architecture organizes the elements of intelligence to create functional relationships and information flow across the various levels, of a hierarchy, that have assigned responsibilities. In this context, knowledge is one of the cornerstones of intelligent real-time control system design and implementation. In this work we have shown how knowledge can be used for performance evaluation and enhancement of intelligent real-time systems (such as robotic systems).

The objective of this effort was to build upon previous work done by the authors on calibration of cable robots and conduct case study to demonstrate the use of robotics toolkit software developed at Pathway Technologies Inc to rapidly design a cable robot, transition seamlessly to controller design, target implementation and set up an automatic on-line calibration scheme for this cable robot where traditional one-time or periodic calibration methods do not provide adequate measures of performance. Hence, from the standpoint of intelligent system design and performance metrics, through this example, we demonstrate the need for automated on-line calibration of intelligent systems (cable robots) so that repetitive manual calibration can be minimized. A flowchart summarizing our approach is shown in Figure 1. Starting out from the development of a graphical model of the cable robot in a GUI, we generate the governing kinematic equations of the cable robot and then automatically generate a SimMechanics model of this system, then a set of kinematic parameters is selected for identification. It is well known that for some calibration methods, all the kinematic/geometric parameters cannot be identified: some of them have no effect on the calibration model, and some others are grouped together [2]. Hence, a parameter identifiability analysis is performed to make sure that all geometric / kinematic parameters can be identified uniquely. Once the parameter identifiability is ascertained, a set of configurations to be used for calibration and validation are selected. Note that the kinematic parameter identifiability Jacobian can also be used for the purpose of optimal pose selection in calibration of the cable robot. The calibration problem is then set up as a nonlinear optimization problem and is solved by using the nonlinear least squares estimator available in the robotics toolkit. The corrected kinematic model obtained from the optimization procedure is then validated using simulation data.

The 6-DOF cable robot, whose calibration we have studied in detail in this report is a closed-chain mechanism in which the mobile platform is connected to the fixed base by six variable length cables. Such cable robots offer the advantages of a larger workspace and low weight with the disadvantage that the cables can apply forces only when in tension. The join of each of these cables with the fixed and the moving platforms are kinematically equivalent to and are modeled as passive spherical joints. Whereas the cable can be modeled as a prismatic joint that can apply forces in one direction only. A typical control strategy for such cable robots is to specify the pose of the moving platform in some world coordinate frame and then to use the inverse kinematics relationship to solve for the cable lengths. The accuracy of the moving platform location critically depends upon the kinematic model of the cable robot that resides in the robot controller. The kinematic calibration of such parallel mechanisms improves the accuracy of the moving platform through modification of the manipulator kinematic model.

Calibration of a general parallel manipulator normally encompasses the following tasks:

1. Kinematic modeling of the platform to account for major error sources.
2. Measurement of platform poses
3. Identification of the kinematic error parameters of the platform by use of the measurement data.
4. Accuracy compensation of the platform by use of the identified error parameters.

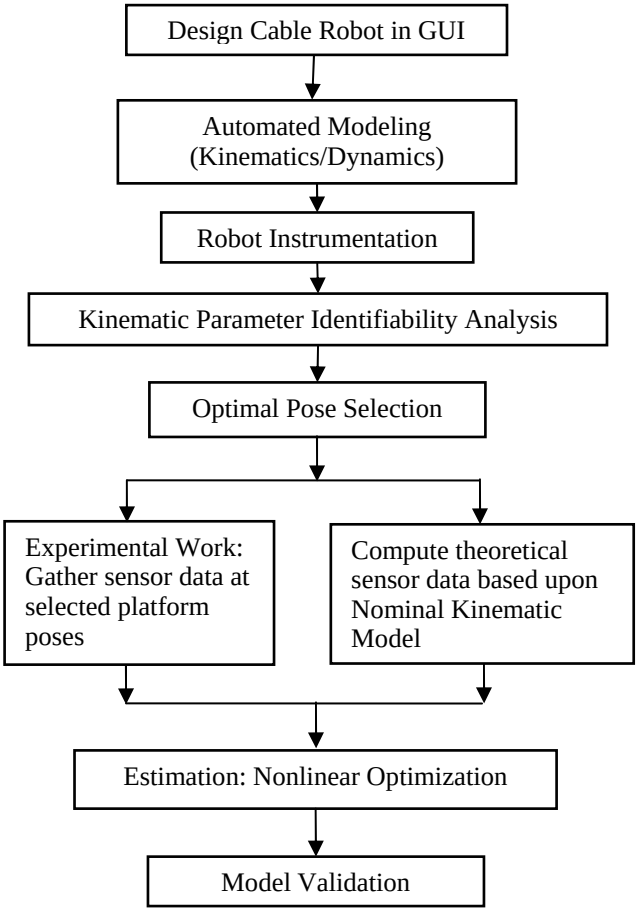


Figure 1: Cable robot calibration process

The kinematic calibration procedure presented in this paper can be classified as a classical calibration can be classified as a classical calibration technique [1, 2, 3, 4, 9, 10] as opposed to self-calibration methods [6, 7, 8]. The method of kinematic calibration used in this work was presented earlier in detail by the authors [11]. It requires the measurement of moving platform orientation by using two inclinometers and the measurement of cable lengths. The authors had considered the following geometric parameters in their calibration: coordinates of the cable joins on the fixed and moving platforms (36 parameters), offsets of the cable lengths (6 parameters) and error on the perpendicularity of the two inclinometers (1 parameter). However, the coordinate systems on the base and moving platforms are placed in such a fashion that 8 out of the 36 geometric parameters are equal to 0. Remaining 28 parameters are constant and may not be equal to zero in general. Aim of the calibration process is to compute the exact values of these 28 parameters, those of the 6 offsets of the prismatic joints, and the error angle on the perpendicularity of the two inclinometers.

Organization of the technical part of the paper is as follows: In the next section we will talk about knowledge representation and the significance of domain specific knowledge in the design and performance evaluation of intelligent systems, followed by the software architecture and then the specifics of the calibration and optimization problems studied here. Finally we present some results obtained from simulation studies.

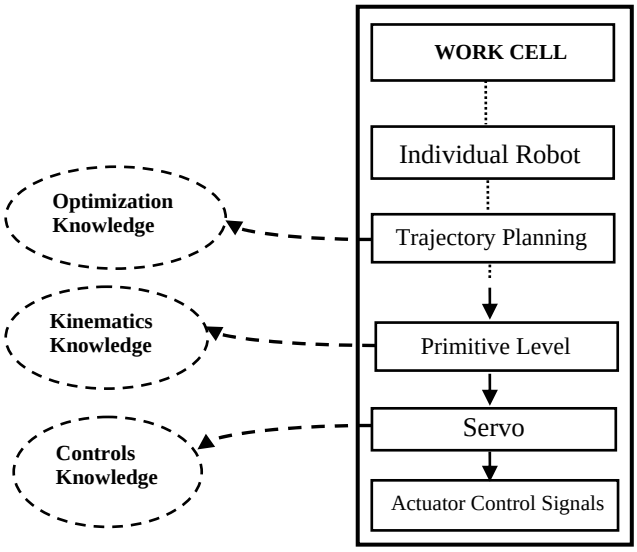


Figure 2: Various activities related to robotics domain

2. KNOWLEDGE REPRESENTATION

Large-scale intelligent systems design can benefit from existing domain specific knowledge from various disciplines through knowledge engineering processes. In this context, based on the broad definition of knowledge, components of knowledge can be broadly classified in to symbolic, iconic, and parametric [12]. For example in the area of parametric knowledge, system models are generated based upon analytical principles such as physics based equations of motions, empirical modeling based upon maps, and system identification based on non-linear input-output system relationships which are difficult to quantify using analytical relationships. In the area of symbolic knowledge, system models are generated based upon mathematical logic, frames, rules, and semantic nets and such. In traditional AI systems there is a larger emphasis on symbolic knowledge whereas in traditional intelligent control there is a greater emphasis on parametric

knowledge. In the context of 4D/RCS, based on the four key paradigms, this would mean that in the lower levels of the hierarchy, such as, servo there is a greater emphasis on parametric knowledge whereas higher level planning nodes the emphasis is on symbolic knowledge. Figure 2 shows the hierarchy of various activities carried out in the operation of a robotic work-cell and the types of knowledge associated with these tasks. In the robotics toolkit we deal mostly with parametric knowledge such as that required for kinematics, dynamics, and optimization.

3. DOMAIN SPECIFIC KNOWLEDGE: KINEMATIC MODELING

Numerous definitions of intelligence found in literature make references to knowledge. However, very little literature is available which addresses the questions what is the needed knowledge in intelligent systems? and, how to generate and represent this knowledge?[12]. In this context, one of the key objectives of this work was to develop or build domain specific knowledge in the areas of (i) Kinematics, (ii) Dynamics, (iii) Path Planning, (iv) Control System Design and Implementation, (v) System Identification and Adaptive Behavior, and (vi) Numerical Optimization techniques for real-time implementation. So far kinematics, dynamics, and planning, knowledge has been developed.

Another important aspect of the development process is a need to have structured approach to developing domain specific knowledge. Figure (3) shows an example wherein under the category kinematics, we define class consisting of different types of dynamical systems such as sceleronomic and rheonomic systems. Under each classification, it is possible to specify subclasses, namely, in the case of *Sceleronomic Systems*, the different types include holonomic and non-holonomic systems. Such a structured approach to system classification leads to an evolutionary approach to building knowledge base. In the following paragraph we have outlined the development of kinematics related domain specific knowledge and the development of kinematic models of the cable robot under study.

Kinematics deals with the constraints on the spatial motions of various bodies within a system. One possible classification of kinematic systems is presented in Figure 3. This classification will enable us to design and develop apriori knowledge repositories for various classes of kinematic systems. At Pathway Technologies we have done some preliminary research in this area. We have developed a unified methodology to generate the kinematic maps for a class of robots that can be classified as sceleronomic, holonomic systems. By making use of the D-H and Klienfinger notations we have been able to develop algorithms for generating the kinematics of such systems on the fly. We have thus succeeded in capturing some expert knowledge in this domain area. A user of our system who wants to design a robot can have all kinematic relations of the robot generated for him without his writing a single equation. In the event of situations like joint failures, we can recompute the kinematics for the robot with one less degree of freedom as compared with the normal operating conditions. The kinematic planners can then plan trajectories accordingly. During hardware operation we can lock the failed joint and continue operations with reduced degrees of freedom. Figure 4 shows a screenshot of the cable robot designed with the robotics toolkit.

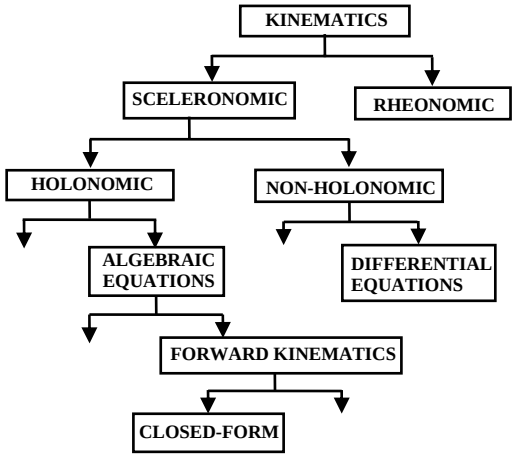


Figure 3: Classification of kinematic systems

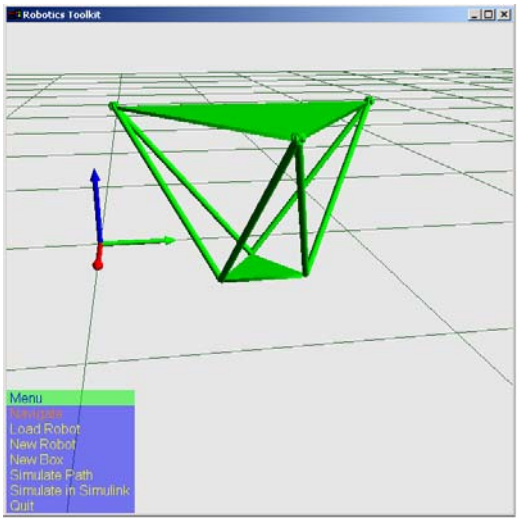


Figure 4: The robotics toolkit generates kinematic maps on the fly.

A schematic of the 6-DOF cable robot studied in this report is shown in Figure 5. A coordinate system A: *xyz* is attached to the fixed platform and another coordinate frame B: *uvw* is attached to the moving platform. The inverse kinematic model (IKM) which calculates the leg lengths vector for a given T_r, which is the homogeneous transformation matrix from frame B to frame A, is easy to obtain. On the contrary, the direct kinematic model (DKM) which calculates the moving platform location T_r as a function of given cable lengths vector, is difficult to obtain analytically. A numerical iterative method based on the inverse Jacobian matrix of the cable robot is used to find a

local solution to the direct kinematics problem. For the purpose of solving the direct kinematics problem we have used a general algorithm which can solve the direct kinematics of any general robot manipulator (serial, parallel, or hybrid). This algorithm converges rapidly and can be summarized as follows:

1. Input the actuated joint variable values, $\mathbf{q}_a$, the initial guess values on the moving platform location, T_r, and the passive joint variable values, $\mathbf{q}_p$.
2. Solve the kinematic constraints for the manipulator to compute the actual passive joint variable values, $\mathbf{q}_p$.
3. Calculate the corresponding moving platform location by solving the forward kinematics of any limb of the parallel manipulator and update the initial guess on the moving platform location.

Where T_r is a homogeneous matrix that defines the location of the moving platform with respect to the base coordinate frame and $\mathbf{q}_a$ is the given vector of cable lengths.

4. SOFTWARE ARCHITECTURE

We have developed an interactive 3D environment for viewing serial and parallel robot models described in text form, constructing robot models and robotic workcells, positioning and posing robots, specifying tasks and end-effector trajectories, and visualizing robots performing tasks and trajectories through time-based animation. Figure 4 illustrates the view of the workcell as seen by the user.

For the purpose of simulating the dynamics of robotic workcells, and implementing robot controllers, the resulting models built using the interactive 3D environment can be exported to the Simulink environment. Robots are modeled in terms of SimMechanics blocks, trajectories as a sequence of end-effector positions/orientations, kinematics as S-function blocks written in the C language. Models of joint control and actuation can be further elaborated by the user or can be chosen from models already developed.

Synchronization is provided between the model built in the interactive 3D environment and model elaborated in Simulink environment. The tight integration of the Simulink environment and the interactive 3D view enables the rapid development of robotic workcells.

In addition to automatic generation of kinematic models for serial and parallel robots, our tool supports the automatic generation of calibration models. The position/orientation of the end-effector is estimated in the controller, and if the difference from the expected position/orientation is significant the robot controller can automatically recalibrate itself, and derive a new kinematic model. As the recalibration computation in non-trivial this part of the model can be partitioned and implemented on a host PC connected to the robot controller.

Performance of the model can be evaluated by taking the difference between the actual path taken in both simulation and experiment and the path that is specified in the 3D environment. Experiments can be constructed such that the performance can be measured over a range of working conditions. One expected source of decalibration is assembly errors that modify the kinematics of the robot. The results of the simulation and can be stored on the permanent storage of system for further study or they can visualized in through 3D animation.

5. CABLE ROBOT CALIBRATION MODEL

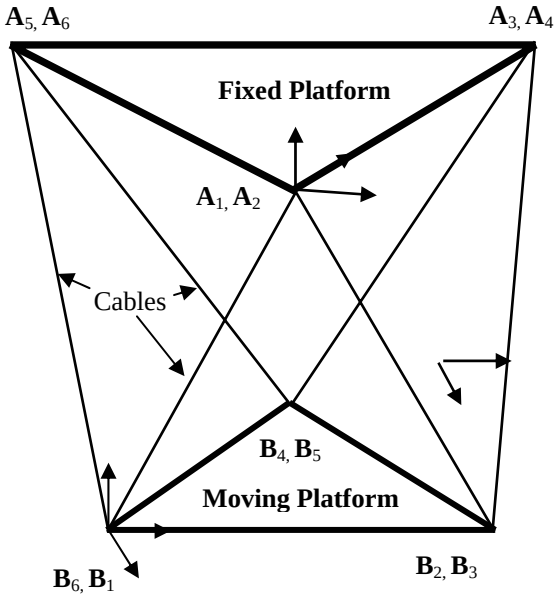


Figure 5: A Six DOF cable robot

By making use of the direct kinematic model of the cable robot we can calculate the orientation of the moving platform with respect to some coordinate frame A: *xyz* as a function of cable lengths and the nominal values of the geometric parameters. In this work we will demonstrate an approach to calibrate this 6-DOF cable robot using two inclinometers mounted on the moving platform of the cable robot. This calibration procedure follows the approach proposed by Besnard and Khalil [1] for the calibration of Stewart Platform using two inclinometers.

Following Besnard and Khalil [1] we consider that there is an error angle γ on the perpendicularity of the two inclinometers. Hence then inclinometer angle values are:

$$\alpha_1 = \sin^{-1}(T_r(3,1))$$

$$\alpha_2 = \sin^{-1}(T_r(3,2) - T_r(3,1) \sin\gamma) \qquad (1)$$

The angle γ is unknown and it can be included in the parameters to be calibrated. In the calculations presented in this paper we have assumed the two inclinometers to be perfectly perpendicular and hence γ is identically equal to zero.

6. OPTIMIZATION

The inclinometer values are calculated for each of the *k* robot manipulator configurations using Equation (1) as function of the nominal geometric parameters and the cable lengths:

$$\Phi_m^n = f(q^n, \zeta_m) \qquad (2)$$

where $\Phi_m^n = [\alpha_1^n, \alpha_2^n]^T$ is the vector of computed inclinometer values at the n^{th} moving platform location, ζ_m is the 35x1 vector of nominal values of the robot geometric parameters.

Similarly we define the vector Φ_r^n of the measured inclinometer values (real) for the n^{th} manipulator pose. If the model is exact, the angles calculated and measured must have the same values at any arbitrary moving platform pose:

$$\left\| \Phi_m^i - \Phi_r^i \right\| = 0 \quad for\ i = 1, 2, ... n ..... k \qquad (3)$$

Using *k* configurations, the geometric parameters are identified such that $\|F\| = \min$, with

$$F = \begin{bmatrix} \Phi_m^1 - \Phi_r^1 \\ \vdots \\ \Phi_m^k - \Phi_r^k \end{bmatrix} \qquad (4)$$

This least squares estimation based nonlinear problem was solved by an estimation algorithm that is a part of the robotics toolkit software.

7. SIMULATION PROCEDURE AND RESULTS

We simulated the calibration method on a cable robot whose nominal parameter and real parameters are given in Table 1 through Table 4. At present we do not possess any consistent (real) sensor data to validate our calibration model. We hope to acquire some field data in the near future. This will enable us to check the accuracy of calibration and also validate the calibration procedure.

Table 1 Nominal values of cable attachment points at the Base platform in the fixed coordinate frame.

	x	y	z
${}^{base}A_1$	0	0	0
${}^{base}A_2$	0	0	0
${}^{base}A_3$	0.8189	0	0
${}^{base}A_4$	0.8189	0	0
${}^{base}A_5$	0.4095	0.7092	0
${}^{base}A_6$	0.4095	0.7092	0

Table 2 Real values of cable attachment points at the Base platform in the fixed coordinate frame.

	x	y	z
${}^{base}A_1$	0	0	0
${}^{base}A_2$	0.01	0	-0.009
${}^{base}A_3$	0.8312	-0.01	0.0043
${}^{base}A_4$	0.7989	0.03	0.0147
${}^{base}A_5$	0.4275	0.6792	0.004
${}^{base}A_6$	0.3858	0.7302	-0.03

Table 3 Nominal values of cable attachment points at the Moving platform in the Moving coordinate frame

	x	Y	z
${}^{moving}B_1$	0	0	0
${}^{moving}B_2$	0.2572	0	0
${}^{moving}B_3$	0.2572	0	0
${}^{moving}B_4$	0.1286	0.2228	0
${}^{moving}B_5$	0.1286	0.2228	0
${}^{moving}B_6$	0	0	0

Table 4 Real values of the cable attachment points at the Moving Platform in the Moving coordinate frame.

	x	Y	Z
${}^{moving}B_1$	0	0	0
${}^{moving}B_2$	0.2770	0	-0.052
${}^{moving}B_3$	0.2472	0.05	0.02
${}^{moving}B_4$	0.1716	0.2028	0.1
${}^{moving}B_5$	0.1076	0.2398	-0.03
${}^{moving}B_6$	0.04	-0.03	-0.05

The procedure used to generate the simulation configurations can be summarized as follows:

1. Generate *m* random sets of cable lengths.
2. Select *n* sets of cable lengths from the *m* for which the moving platform lies within the manipulator workspace. Further, select *k* sets of cable lengths from the *n* that have that lowest condition numbers for the Identification Jacobian matrix.
3. Compute the inclinometer values using the forward kinematics solution.
4. Add some random numbers on the nominal inclinometer values to generate a set of data that we would term as the real sensor readings.
5. Compute the objective function for the purpose of optimization using these sets of the so-called real and computed inclinometer values.

6. Solve the nonlinear optimization problem using the Robotics Toolkit.
7. Compute the real geometric parameters for the robot under consideration.

6. SUMMARY

In summary, we present the features of our robotics toolkit software and the support it provides for the performance evaluation and performance enhancement of intelligent systems. This toolkit allows designers to build robots and robotic work-cells rapidly in a graphical environment, automatically generates the kinematic model and dynamics (using SimMechanics) of such intelligent systems and provides support for performance evaluation and enhancement. In addition, the use of Matlab/Simulink environment helps control designers to transition seamlessly from design to hardware. We also present a case study to illustrate the use of this software to rapidly deploy an automatic kinematic calibration method that allows the intelligent system (cable robot) to precisely manipulate its surroundings. Simulation studies can be conducted off-line and sensitivity of the positioning performance to various geometric parameters can be studied in a virtual environment. Also, the presented continuous on-line calibration approach provides performance improvement over the conventional periodic calibration methods as it continuously compensates for mechanical changes to the system. This improved system behavior provides quantitative measures of performance improvement.

ACKNOWLEDGEMENTS

Development of the robotics toolkit was funded in part by NIST under an SBIR. The authors wish to thank Elena Messina of NIST for valuable comments.

REFERENCES

[1] Besnard, S., and Khalil, W., “Calibration of Parallel Robots Using Two Inclinometers,” Proceedings of the IEEE International Conference on Robotics and Automation, May 1999.

[2] Besnard, S., and Khalil, W., “Identifiable Parameters of Parallel Robots Kinematics Calibration,” Proceedings of the IEEE International Conference on Robotics and Automation, May 2001.

[3] Yang, G., Chen, I. M., Lim, W. K., and Yeo, S. H., “Self-calibration of Three-Legged Modular Reconfigurable Parallel Robots Based on Leg-End Distance Errors,” Robotica, Vol. 19, pp. 187-198.

[4] Irascu, C. C., and Park, F. C., “Geometric algorithms for Closed Chain Kinematic Calibration,” Proceedings of the IEEE International Conference on Robotics and Automation, May 1999.

[5] Zhuang, H., and Liu, L., “Self-calibration of a Class of Parallel Manipulators,” Proceedings of the IEEE International Conference on Robotics and Automation, April 1996.

[6] Besnard, S., and Khalil, W., “Self-Calibration of Stewart-Gough Parallel Robots Without Extra Sensors,” IEEE Transactions on Robotics and Automation, Vol. 15, No. 6, December 1999.

[7] Zhuang, H., and Roth, Z., “Method for Kinematic Calibration of Stewart Platforms,” Journal of Robotic Systems, 10(3), pp. 391-405, 1993.

[8] Zhuang, H., Yan, J, and Masory, O., “Calibration of Stewart Platforms and Other Parallel Manipulators by Minimizing Inverse Kinematic Residuals,” Journal of Robotic Systems, 15(7), pp.395-405, 1998.

[9] Lintott, A. B., and Dunlop, G. R., “Parallel Topology Robot Calibration,” Robotica.

[10] Vischer, P., and Clavel, R., “Kinematic Calibration of the Parallel Delta Robot,” Robotica, Vol. 16, pp.207-218, 1998.

[11] Joshi, S.A., and Surianarayan, A., “Calibration of a 6-DOF Cable Robot Using Two Inclinometers,” Proceedings of the PerMIS 2003 Conference, Gaithersburg, MD.

[12] Messina, E., Albus, J., Schlenoff, C., and Evans, J., “Knowledge Engineering for Real Time Intelligent Control”, Proceedings of the International Workshop on Knowledge Management Techniques, Krema, Italy, 2002.

Adaptive Problem Solving with Particle Systems

Alejandro Rodríguez and James A. Reggia
Dept. of Computer Science and UMIACS
University of Maryland
College Park, MD 20740
{alejandr,reggia}@cs.umd.edu

ABSTRACT

Self-organizing particle systems ("swarms") consist of numerous autonomous, reflexive agents ("particles") whose collective movements through space are determined primarily by local influences. We are currently extending particle systems so that they not only move collectively, but also solve simple problems. This is done by giving the individual particles/agents a rudimentary intelligence in the form of a limited memory and a top-down, goal-directed control mechanism. Such enhanced particle systems are shown to be able to function effectively in performing simulated search-and-collect tasks. However, measuring the effects on particle collective performance of different design choices for individual agents proved to be difficult. We resolved this issue by allowing different agent teams to compete with one another under a variety of controlled conditions. This allowed us to demonstrate clearly how different agent features (independent vs. coordinated movement, exploratory vs. protective behaviors) impacted the behavior of the collective as a whole.

1 INTRODUCTION

Recent work creating computational models of coordinated movements by collections of locally interacting individuals includes, for example, models of bird flocks [12, 13], fish schools [6, 16], social insect swarms [2], and self-assembling molecules [4]. All of these systems consist of numerous autonomous "particles" (birds, ants, vehicles, etc.) whose movements through space are governed by primarily local "forces" exerted on them by other nearby particles or the environment. Methodologically-related approaches such as particle swarms [8], cultural algorithms [3], and bacterial chemotaxis algorithms [10] have generalized this idea to abstract, n-dimensional "cognitive spaces". Here we will refer to all of these past models as *self-organizing particle systems.*

Typically, these systems consist of mainly reactive particles whose behavior is completely determined by reflexive movement dynamics. Interactions between these particles result in remarkably complex global behavior which emerges from the joint actions and relatively simple behaviors of the individual particles, thus exhibiting self-organization. These properties have led to applications in computer graphics [12, 13], multi-robot team control [1, 5, 11, 17, 18], and numerical optimization [8]. The simplicity of the particles makes their use difficult for tasks other than the simulation of movement patterns. However it has been proposed in a few studies that it is possible to extend the paradigm towards autonomous problem solving, for example using multi-robot teams for pushing objects [9] or foraging [7] or more general tasks as the Robocup competitions [17].

The ultimate goal of our research is to extend particle systems to act as more general problem-solving entities that not only move in a coordinated fashion, but also collectively set goals and make decisions, thereby forming a *self-organizing collective intelligence* [15, 19]. By this term we mean a self-organizing particle system where the individual particles are no longer mindless, but have at least a limited goal-driven intelligence and decision making ability. To achieve this, the low-level movement dynamics used in past particle systems is complemented by a top-down, goal-driven control mechanism capable of simple inferences and of autonomously switching between different formulations of its movement dynamics. Since such enhanced particles have simple inference abilities that influence their movements, we will also refer to them as *agents* that form a *team.* In this paper, we describe our first steps in this work: combining a finite state machine (FSM) control process that selects goals and switches between alternate movement dynamics as the agent collective solves "search and collect" problems.

Given the self-organizing and interactive nature of such systems, measuring the impact of design choices for individual agents on the performance of the system as a whole poses a particularly difficult problem since the contribution of the individual components to global behavior cannot be easily determined. We successfully addressed this problem by systematically varying specific agent features and comparing the results through a series of competitions between different agent teams.

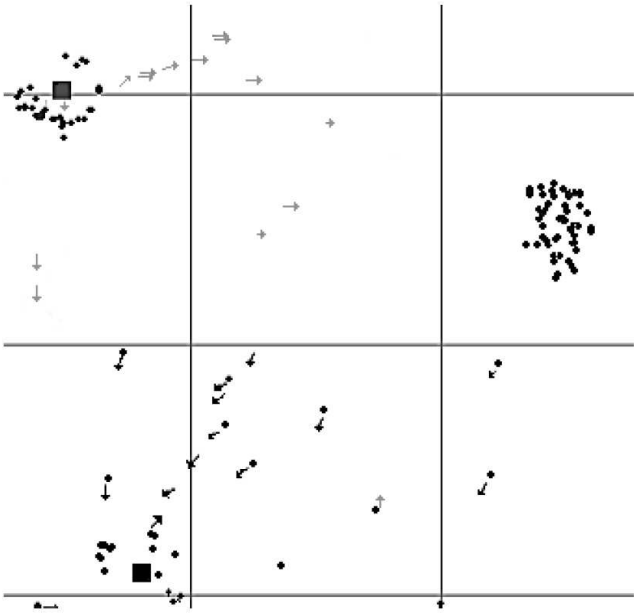


Figure 1: A small section of a 3000 x 3000 continuous world with a single "mineral deposit" on the right and two different teams (dark and light arrows) exploiting it. Teams' homes are denoted by small solid squares and mineral units as spots. The dark team agents are returning home carrying minerals (spots adjacent to arrows). Most agents of the light team are returning to the deposit after unloading minerals at home, but some of them are simply exploring.

2 METHODS

As a first test of the benefits of collective rather than individual problem-solving, we used a resource collection task. A team of homogeneous agents is initially deployed at a random location inside a 2D world with periodic boundary conditions (see Figure 1). In this world, units of some resource that we call minerals (but which could also represent any other valuable material) are located randomly. A few areas are heavily dense in mineral, thus serving as "deposits", while the rest of the minerals are scattered throughout the world. The task of the agents is to collect the minerals and deposit them in a designated "home" location. The success of a team is measured in terms of the amount of minerals accumulated at its home over time, which advances in small discrete steps. Other teams may be present in the world, and members of one team can potentially hinder rival teams by different means, such as blocking them (agents are not allowed to collide with others), or even stealing from their home.

Agents have a two layer architecture. The bottom level layer (local information plus movement dynamics) controls the reactive behavior of the agent, making instantaneous decisions about the actions to be performed. It takes input solely from the local environment at a given instant, and outputs a corresponding action based on the immediate goals of the agent and current (local) state of the environment; this is similar to what has been done in past particle systems. The top layer, not found in past particle systems, consists of a very limited memory and a finite state machine (FSM) that directs agent behavior in a top-down fashion, modifying the movement dynamics used over time. For example, if the FSM decides that it is time for the agent to go home, it will switch to the state *carrying* and provide the bottom layer with the target location of its home location. The bottom layer will then determine at each step the steering actions needed to properly navigate from the current location to the home. Since the bottom layer is mostly reactive it can temporarily override the long term goal of going home for a more pressing need, such as avoiding a competing agent or obstacle.

Other agents and objects in the world are visible to each agent as long as they are within its local neighborhood. The neighborhood of an agent is defined as a circular segment of radius r and angle α in front of the agent. An agent senses the relative position, orientation and current velocity of every object in its neighborhood, in addition to its own current position. Although agents from the same team are homogeneous and interchangeable, they are able to distinguish between members of their own team and other agents. Agents may also know the state of members of their own team in their neighborhood. Agents are able to pick up, carry and put down a single resource at a time.

At any time, each agent is in one of several possible mutually exclusive states. For example, an agent could be *searching for minerals* or *carrying minerals home*. These states and the transitions between them are represented with a finite state machine (FSM). The agent's current state determines which set of parameters for the low-level reactive controller currently drives its movements and behavior. Figure 2 shows the FSM employed in the experiments reported here and the corresponding movement behaviors associated with each state. Agents are initially in an idle state. Once they receive a "start signal", they begin *searching* for resources. When they find some, they choose between *picking up* the minerals or *guarding* the deposit from other teams, depending on whether a group of guarders is already formed. When an agent detects five or more agents guarding in its neighborhood, it decides the deposit is already guarded. Agents recognize a deposit when they detect a certain amount of minerals in their local neighborhood. This implies that homes of other teams are interpreted as deposits given that they have accumulated enough minerals. If the agent succeeds in picking up a unit of mineral, then it starts *carrying* the minerals home. After arriving home with a unit of mineral and depositing it, if the home is unprotected agents go to *guard-*

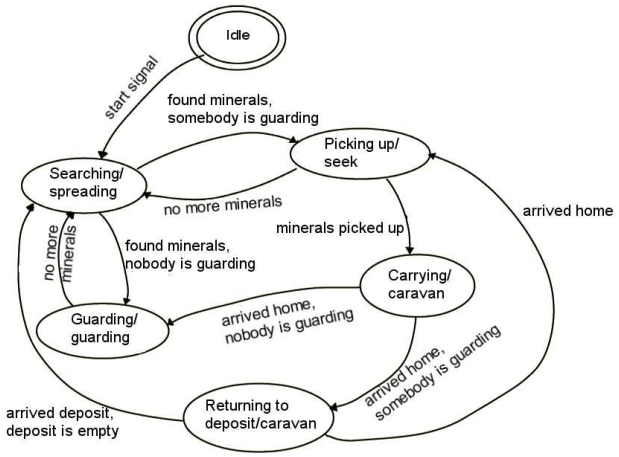


Figure 2: FSM of an agent showing its states and the movement behaviors associated with each state. States are represented by circles labeled by <State/associated controller>, while arrows represent transitions labeled by the triggering event. The initial state is marked by a double circle.

ing to protect it, otherwise they return to the last deposit they were exploiting (*returning to deposit*). If the flock arrives at its (unprotected) home, the first five agents will become guarders and the rest will return to the deposit. When an agent is guarding, it will remain so unless all of the mineral in its neighborhood is taken away (the deposit is exhausted), in which case the agent returns to *searching*. When agents are *returning to deposit*, it could happen that the deposit is already exhausted, in which case the agent goes into *searching* for another deposit, otherwise it tries *picking up* more minerals.

Agents have a very simple memory that allows them to recall the location of the deposits that the agent has found and the location of the home of the agent. The precise current goal of the agent is thus represented as a combination of the current state and the contents of this memory. For example, the state could determine that the agent is going back to a deposit to exploit it, while the memory determines which deposit the agent is to exploit.

The low level movement dynamics that guide the agent through the environment are inspired by earlier work [12, 13]. Movements are governed by a set of individual influences (avoiding an obstacle or competing agent, staying with the flock, keeping the same distance from other agents from the team, etc.) that produce an instantaneous acceleration determined by a desired velocity vector. The individual influences are combined in a non-linear fashion. By changing which individual influences are combined and their relative weights (done by the FSM), a large variety of movement behaviors can be implemented, each one associated with a different state or goal.

Seven individual influences act on each agent and are computed based on the position and velocity of neighboring agents. For instance, a *cohesion velocity* $\vec{v}_c$ tends to move the agent toward the center of the flock and is computed as

$$\vec{v}_c = \left(\frac{||\vec{p}_n - \vec{p}||}{r}\right)^2 v_{max} \frac{\vec{p}_n - \vec{p}}{||\vec{p}_n - \vec{p}||}$$

where $\vec{p}$ is the current position of the agent, $\vec{p}_n$ is the average position of its neighbors and $\vec{v}_{max}$ the maximum speed of the agent. The direction of this velocity is directly toward the center $\vec{p}_n$ of the agents in its neighborhood, while its magnitude is a fraction of the maximum velocity that increases quadratically with the distance from that center. The remaining six velocities are explained in more detail in [15]. An *alignment velocity* tends to move the agent in the same direction that its neighbors are moving. An *avoidance velocity* tends to move the agent away from other agents. A *separation velocity* tends to move the agent away from neighbors by steering away from the center of the flock. A *seeking velocity* influences an agent to move toward an observed unit of mineral. A *clearance velocity* influences an agent to steer toward the side when there is an agent in front of it and tends to align a group of agents side by side. Finally, a *homing velocity* drives the agent to a point in the space such as the home of the agent or a remembered deposit. The individual velocity influences are combined as a weighted sum at each time step to update each agent's resulting velocity $\vec{v}$. The summation process is prioritized, with the terms being added in a fixed order, and when the sum exceeds a certain threshold v_{max} the term is dropped. Different prioritizations and weight values are used to produce each specific behavior of the agent. Once the new velocity of the agent has been computed, its new position is updated.

By prioritizing and weighting each of the seven velocity components differently, four different *movement behaviors* are implemented. Full details are given in [15]. Briefly, *spreading agents* tend to form a flock slightly similar to the broad, V-shaped formations that migratory birds adopt (Figure 3a). This formation is ideal for searching for deposits, since upon seeing a deposit, an agent will tend to "pull" the rest of the flock to it via local interactions (Figure 4). *Seeking agents* go after a target, for example a unit of mineral, while avoiding obstacles and other agents (Figure 3b). *Caravaning agents* move in a pattern similar to spreading agents, but they are homing on a particular point in space (e.g., a previously visited deposit) and thus have a narrower front (Figure 3c). Finally, *guarding agents* patrol a deposit (Figure 3d), distributing themselves about evenly through the deposit. The interaction between nearby guarding agents and even non-guarding agents make them keep moving, effectively orbiting the

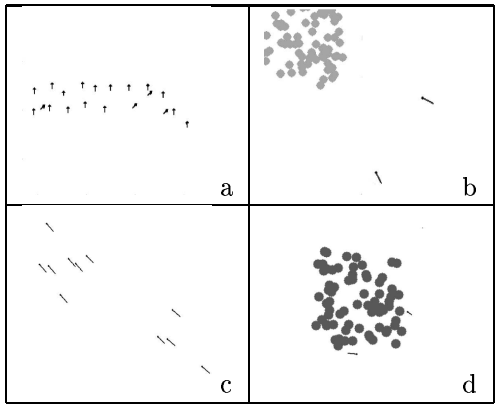


Figure 3: Collective movements. (a) Spreading agents (arrows), are located roughly side by side. (b) Two seeking agents head to the closest mineral unit (gray spots). (c) Caravaning agents tending to align. (d) Two guarding agents patrolling a deposit.

deposit. Since agents cannot bump into each other, this helps to keep intruders away by repelling them from the deposit.

Each computational experiment below consists of letting one or more simulated teams search for the minerals in the world for a limited time. To compare the effectiveness of collective movement behaviors versus independently moving agents, a set of independent agents have been implemented. These agents follow the same model explained above, with the important exception that they do not take into account other agents in their dynamics. The controllers for these agents replace dependent steering behaviors (cohesion, alignment, separation, clearance) with random wandering. Collision is not replaced. Otherwise the FSM of the non-flocking (independent) agents is the same as for the collectively moving agents (Figure 2).

The results reported below are the average over 20 runs for each experiment, using 50 agents per team (unless indicated otherwise) in a continuous world of size 3,000 x 3,000 and one team for every type of agent (described below). During each run, 12 deposits were randomly located and independently created, each deposit consisting of 80 units of minerals. These parameters were chosen so that it is valuable for a team to go back to a deposit repeatedly. Also, the number of deposits and size of the world make it easy for agents to find the deposits (and other agents' home), but still force agents to compete for the resource (minerals). Teams' homes were also independently located at random in each run and agents were initially randomly positioned. In each run, the simulation lasted 40,000 time-steps (iterations), which was adequate for the average amount of mineral in teams' home to converge to some value.

Multiple teams of agents were often used in an experimental run, with each team being of a different type. Which teams were involved could vary from experiment to experiment. The six different types are:

- Full-guarding flock. Collectively moving agents which guard home and any deposits that they find.
- Home-guarding flock. These agents will not guard a deposit, but will still guard their own home.
- Non-guarding flock. These agents are the same as above except they do not have a guarding state.
- Full-guarding, home-guarding and non-guarding independent teams. These three types of agents correspond respectively to the three types of flocking agents above, but do not undertake collective movements; they move independently. They search through random wandering and see other agents only as obstacles to be avoided.

3 RESULTS

In the first experiment, a team is present in the world without any other competing teams. The experiment was repeated for each team. Figure 5 shows the amount of minerals collected over time, averaged over 20 runs. After 20,000 iterations most teams have succeeded in collecting almost all of the minerals available in the world, with the exception of the full-guarding teams that are still slowly collecting resources. This is due to the fact that after the full-guarding teams split off members to guard every deposit found, not enough agents remain to collect the minerals rapidly. The non-guarding flocking team seems to be the fastest in collecting minerals, which is consistent with the fact that all team members are actively searching for and collecting minerals, and that in the absence of other teams, guarding mineral deposits has no value. In this world where agent teams do not need to compete with other teams, there is no clear difference between the other four teams.

In the second experiment, all six types of agents (one team per type) are present simultaneously in the same simulation. The number of agents per team was decreased to 30 to avoid overcrowding. Figure 6 shows the amount of minerals in each teams' home over time, averaged over 20 runs. Most striking is that the home-guarding, flocking team's collected resources increase monotonically, with this team clearly outperforming all others. Early in simulations (during the first 5,000 iterations), both this and the non-guarding flocking team collect minerals faster than any other team. After the first few thousand iterations, the explored area of each team is wide enough for teams to find each others' homes. Accordingly, the amount of minerals

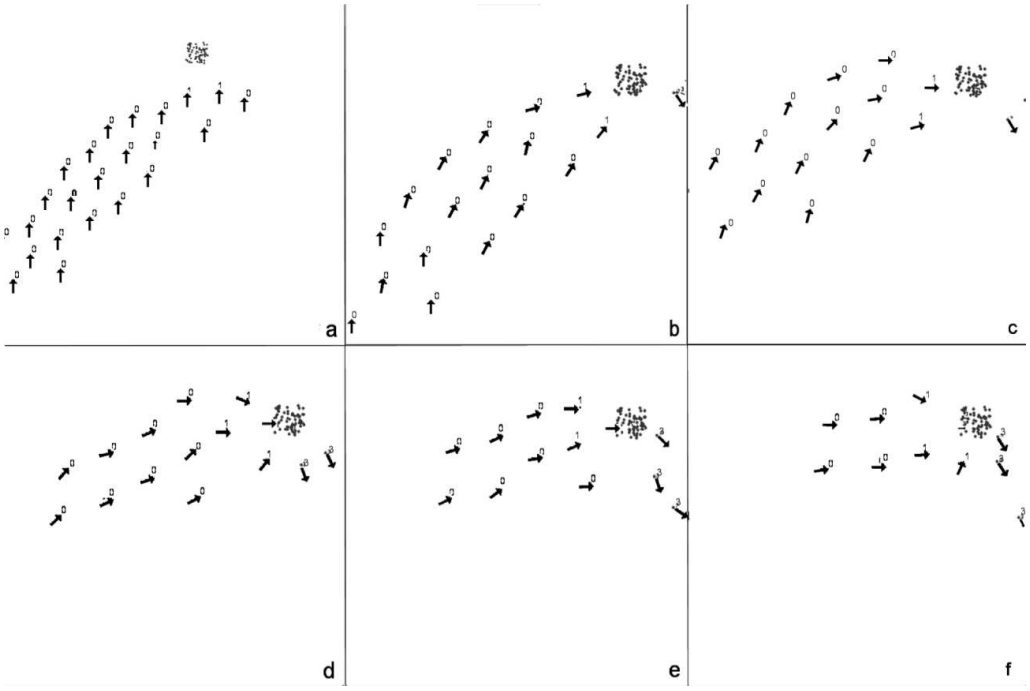


Figure 4: Agents in a flock being pulled toward a deposit. The number on top of each agent represents its current state (0 for Searching for a deposit, 1 for Picking up). Only agents in state 1 actually detect the deposit. At a, only two agents have located the deposit, while the rest of the flock moves northward. At b and c, agents that are near the deposit but that do not yet see it turn toward those agents that have seen the deposit and are already going toward it. From d to f, the whole flock gradually turns toward the deposit and collects minerals. Such behavior indicates an advantage of collective movements in recruiting other agents to carrying a resource when it is discovered by just a few.

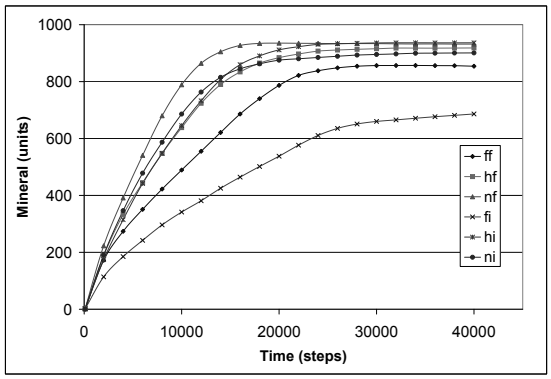


Figure 5: Mineral collected over time by each team alone. ff: Full-guarding flocking, hf: home-guarding flocking, nf: non-guarding flocking, fi:full-guarding independent, hi:home-guarding ind., ni: non-guarding ind.

decreases in subsequent iterations for most teams, especially the non-flocking teams. The differences in the mean amount of collected minerals by each team after 40,000 iterations over 20 runs are statistically significant at the level of 95% according to a two-way ANOVA, both in sociality (flocking vs independent) and guarding strategy (full-guarding, home-only and none). These data suggest two main hypotheses. First, teams of collectively moving agents are more effective at this task than corresponding teams of independently moving agents. With collectively moving agents, whenever a deposit was discovered by an agent, numerous other agents were immediately nearby and thus pulled in by local inter-agent influences to help collect the discovered minerals (e.g., see Figure 4). Second, for both collectively and independently moving agent teams, agents that guarded only their home did better than non-guarding agents, who in turn did better than full-guarding agents. Presumably allocating agents to guard resources, especially multiple deposits, has a large cost: it removes these agents from collecting minerals, and this loss is not adequately compensated for by any protective influences they exert through their blocking actions.

The impact of collective versus independent movements on agent teams can be clarified by varying just that factor

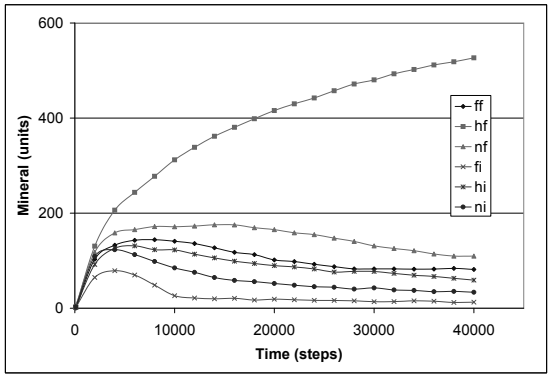


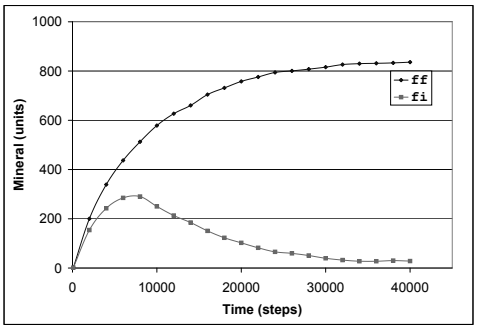
Figure 6: Mineral present in teams' home per unit of time when all teams compete simultaneously in a single world.

between two competing teams. Figure 7 shows the mean amount of mineral saved at home over time for pairwise competitions of collectively moving versus independently moving teams of agents. These results are significant at the level of 95%. It is clear that the flocking teams are always faster in accumulating minerals. Even more striking, the independently moving teams are not sufficiently effective in protecting its home from being looted by the flocking team, and their collected minerals considerably decrease during the following iterations.

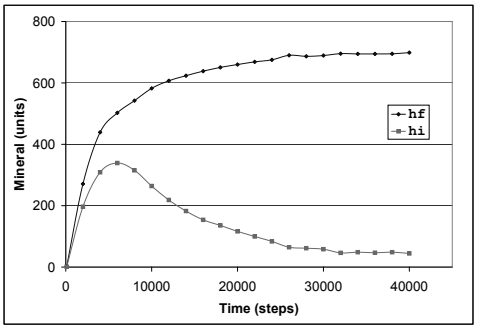
Finally, experiments to compare the matched pairs of guarding versus non-guarding teams were performed. Figure 8 shows the mean amount of mineral saved at each team's home over time. Results are significant at the level of 95%. Early on in the simulation (about iteration 5,000) pairwise teams have similar performance, but after this guarding teams show a clear advantage, as their amount of minerals saved at home keep increasing, while the amount of minerals in the home of non-guarding teams decreases, probably as it is taken by the opposite team. Again, home-guarding teams perform better than full-guarding teams.

4 DISCUSSION

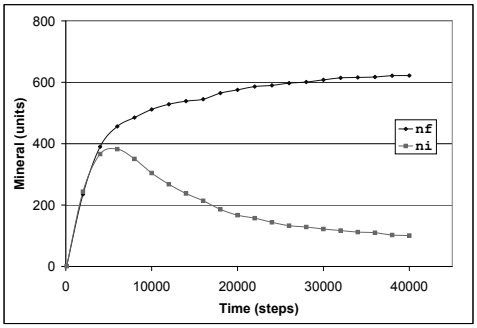
In this paper, we have examined the question of whether self-organizing particle systems can be extended to exhibit behaviors more general than just collective movements. Specifically, our hypothesis was that by giving the normally purely reflexive agents found in particle systems a few behavioral states, a simple finite state transition graph that governs state changes, and a simple memory of the locations of significant objects that are encountered, the resulting agent team would have the ability to collectively solve resource locate-and-collect problems. Individual behaviors are implemented by letting each state of an agent be associated with both a different goal and with a corresponding set of parameters that influence the agent's



(a)



(b)



(c)

Figure 7: Mean mineral collected over time by full-guarding teams. (a) ff: full-guarding flocking versus fi: full-guarding independent agents. (b) hf: home-guarding flocking versus hi: home-guarding independent agents. (c)nf: non-guarding flocking versus ni: non-guarding independent agents.

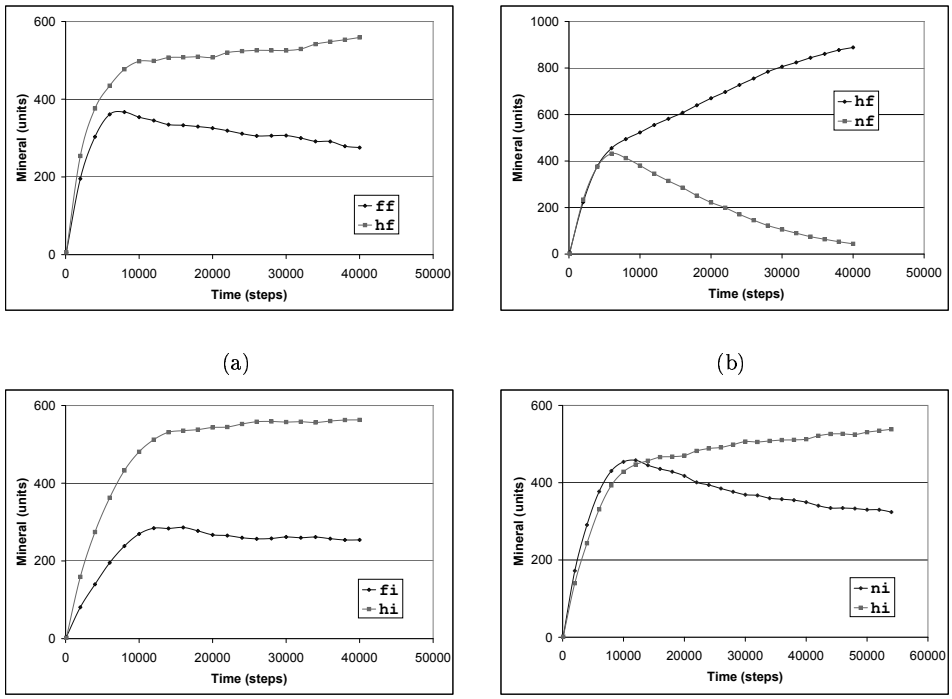


Figure 8: Mineral present in teams' homes per unit of time. (a) ff: full-guarding flocking versus hf:home-guarding flocking. (b) hf: home-guarding flocking versus nf:non-guarding flocking. (c) fi: full-guarding independent versus hi:home-guarding independent. (d) hi: home-guarding independent versus ni:non-guarding independent.

movements. This effectively couples the collectives' goals to different movement dynamics. Under such conditions, where state changes are triggered by environmental events and the states of other nearby agents in a way that retains the local nature of information processing in particle systems, one would anticipate the emergence of problem-solving abilities by an agent team as a whole.

The simulation results presented in this paper and our earlier work [15, 19] provide substantial support for our hypothesis. As state changes occurred and spread throughout a collection of agents via local interactions, the group's motion as a whole was influenced and shifted to provide collective problem-solving. An agent team could routinely search for, collect, and return discovered resources to a predetermined home location, all the while retaining movement as a "flock" of individuals. Further, it was found in simulations that a team of agents that moved collectively was more effective in solving search and collect problems than very similar agents that moved independently. This was because when one or a few agents on a collectively-moving team discovered a site with plentiful resources, they would automatically pull other team members toward that site, greatly expediting the acquisition of the discovered resource. Thus, a benefit of underlying collective movements of particle systems which, to our knowledge, has not been appreciated in past work, is that they have the potential to automatically recruit additional agents to complete a task when one agent detects the necessity of that task.

We also undertook computational experiments in which multiple teams with somewhat different behaviors simultaneously competed to find and collect the resources that were present. Regardless of whether the teams competed two at a time or all at once, we found consistently that collectively moving agents were superior to independently moving teams of matched agents in collecting resources. Further, and regardless of whether agents moved as a team or independently, we found that those that were allowed to guard only their home base did best, those that tried to guard both home and discovered resources did worst, and those that guarded nothing were in between. This finding reflects a kind of exploration/exploitation trade off: guarding has a protective value for preserving located and collected resources, but also a cost in that fewer agents are available to continue searching for and collecting new resources. Most importantly, these simulations exhibited

group-level decisions not just about which type of movements to make, but also about when it was appropriate to split into groups, with one smaller group remaining to guard resources. They exhibited decentralized cooperation without explicit coordination, such as when a wandering agent would follow another agent that knew the location of uncollected resources, simply because wandering agents tended to follow other agents. Our results, as well as related ongoing work [19], show that the reflexive agents of contemporary particle systems can readily be extended to successfully support goal-directed problem solving while still retaining their collective movement behaviors.

Acknowledgments: This work was supported by NSF ITR award ITS-0325089.

REFERENCES

[1] Balch, T., & Arkin, R. (1998). Behavior-based Formation Control for Multi-robot Teams. *IEEE Robotics and Automation*, 14(6),926-939.

[2] Bonabeau, E., Dorigo, M., & Theraulz, G. (1999). *Swarm Intelligence*. Oxford Univ. Press.

[3] Chung C. & Reynolds R. (1996) A Testbed for Solving Optimization Problems Using Cultural Algorithms, *Evolutionary Programming*, 225-236.

[4] Edwards, L., Peng, Y., & Reggia, J. (1998). Computational Models for the Formation of Protocell Structures. *Artificial Life*, 4, 61-77.

[5] Fredslund, J. & Mataric M. (2002). A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication. *IEEE Transactions on Robotics and Automation*. 18(5), 837-846.

[6] Huth, A., & Wiesel, C. (1992). Simulation of Movement of Fish Schools. *J. Theoret. Biol.*, 156, 365-385.

[7] Jones, C. & Mataric, M. (2003). Adaptive Division of Labor in Large-Scale Minimalist Multi-Robot Systems. *Proc. IEEE International Conference on Intelligent Robots and Systems*, 1969-1974.

[8] Kennedy, J., & Eberhard, R. (2001). *Swarm Intelligence*. Academic.

[9] Kube, C., & Zhang, H. (1992). Collective Robotic Intelligence, *Second International Conference on Simulation of Adaptive Behavior*, 460-468.

[10] Muller, S., Marchetto, J., Airaghi, S., & Kourmoutsakos, P. (2002). Optimization Based on Bacterial Chemotaxis. *IEEE Transactions on Evolutionary Computation*, 6, 16-29.

[11] Parker, L. (1993). Designing Control Laws for Cooperative Agent Teams. *Proc. IEEE International Conference on Robotics and Automation*, 3, 582-587.

[12] Reynolds, C. (1987). Flocks, Herds and Schools. *Computer Graphics*, 21, 25-34.

[13] Reynolds, C. (1999). Steering Behaviors for Autonomous Characters. *Proc. Game Developers Conf.*, 763-782.

[14] Reynolds, C. (2000). Interactions with Groups of Autonomous Characters. *Proc. Game Developers Conf.*, San Francisco: CMP Game Media Group, 449-466.

[15] Rodríguez A. & Reggia J. (2004) Extending Self-Organizing Particle Systems to Problem Solving, *Artificial Life*, 10(4), 379-395.

[16] Tu, X. & Terzopoulos, D. (1994). Artificial Fishes: Physics, Locomotion, Perception, Behavior. *Proc. ACM SIGGRAPH*, 42-48.

[17] Vail, D., & Veloso, M. (2003). Multi-Robot Dynamic Role Assignment and Coordination Through Shared Potential Fields. In *Multi-Robot Systems*. Schultz, A., Parker, L., & Schneider, F. (editors). Kluwer.

[18] Verth, V., Brueggemann, V., Owen J., & McMurry P. (2000). Formation-Based Pathfinding with Real-World Vehicles. *Proc. Game Developers Conference*.

[19] Winder R. & Reggia J. (2004) Distributed partial Memories to Improve Self-Organizing Collective Movements, *IEEE Trans. SMC B*, 34, 1967-1707.

Negative Knowledge and Rational Creativity

Mark H. Bickhard

Rationality cannot be restricted to the following of logical rules. Such a restriction encounters foundational problems - for example, What is the rational warrant for those rules? And it also encounters the following conundrum: No logic can construct a more powerful logic than itself. Therefore, if rationality is equivalent to honoring the rules of logic, the history of logic (and mathematics), which consists, among other things, of the construction of more and more powerful logics, is arational.

I propose that the core of thought is not rationality, opposed by the creative passions. Instead, thought is fundamentally an internal variation and selection process, an evolutionary epistemology, involving creativity at its core. But a crucial aspect of the development of this process is the learning of selection criteria, of what constitutes error. Of course, such negative knowledge of error can itself be in error, so criteria of potential error form a hierarchy, with some criteria applying to constructions concerning interactions with the world, and other criteria applying to lower level criteria. Because such criteria, when expressed in words, form the grounds for criticism, I call them *critical principles*.

The positive knowledge that is generally identified as rational is the knowledge of how to avoid error. Rationality, then, is a natural developmental tendency of this variation and selection creative process. One very general, and therefore important, kind of rationality focuses on relations among extensions of representations, independent of the specific natures of the elements in those extensions. For example, if the extension of A is included in the extension of B, then All As are Bs. This view yields a reconstruction of logic along the lines of Tarski, Mostowski, and Sher, and, thus, a recovery of logic as an aspect of rationality, though not its center, and rationality an aspect of creative variation and selection processes. Rationality and creativity, therefore, constitute aspects of an integrated developmental tendency, not opposing forces. The key to this integration is the negative knowledge of critical principles.

Agent and Multi-Entity Systems Modeling

C. Hein, H. Zebrowitz

Abstract

This paper describes a method for studying the functional and temporal performance of systems controlled by distributed autonomous agents through simulation of the agents within larger systems.

There are many challenges to developing successful agent systems. Although there are multiple agent architectures or approaches for a given application, it is difficult to predict which is best. The behavior of complex distributed autonomous agent communities is notoriously difficult to predict or certify prior to the final implementation of an approach. Agent architecture development is quickly burdened with implementation details before major questions are answered, such as:

- stability, - convergence, - negotiation logic
- efficiency, - robustness, - agent architecture
- speed, - resilience, - agent class structure

New capabilities are needed to predict and analyze agent architectures early, prior-to and throughout detailed development. It will assist in guiding optimization, and simplification of implementations. It can provide quick "what-if" analysis.

A comprehensive, flexible agent modeling technique is needed which must model the outer systems that agents operate within, and which must resolve functional, temporal, and spatial aspects. Ideally, there should be an automated flow from modeling and development to fielded systems. Such capability would Support early demonstration, to incorporate feedback from users within time to accept it.

A brief survey of previous agent simulation environments is presented, covering such tools as U. New Mexico - SWARM, and Brookings Institute/Nutech - Ascape, U. Chicago - Repast.

Finally, an initial experimental model with two scenarios is described to add some concreteness to the discussion by example. A hierarchical agent-based simulation environment called ATL-CSIMis introduced. The modeling work done with this tool demonstrates agent operations within larger systems while revealing timing performance/response and assessing functional quality of the collective agent solutions. The preliminary results show emergent multiple-level hierarchical organization. The first scenario involves agent navigation in a two dimensional space with a series of progressive challenges provided by a mixture of agent roles. The second scenario operates in a text-based mathematical equation space for a non-linear dynamic system. The two distinctly differing scenarios were chosen to demonstrate the generality of the approach, and to show that it can be applied to many domains. The work is contrasted with the previous efforts.

References:

[1] R. Pfeifer, C. Scheier, "Understanding Intelligence",
MIT Press, 1999, ISBN 0-262-16181-8.

On Knowledge Representation issues

Alex Abramovich
webdao@yahoo.com

ABSTRACT

Knowledge Representations issues take on special significance in the light of development of the novel Web's reality that involves the Semantic Web, GRID, P2P and other today's ITs. In contrast to the previous IT evolution's stages, the recent one utilizes ontology as separated resource. An elaborate knowledge representation approach implies an efficiency of knowledge-based systems and their interoperability. This paper deals with Ontology Engineering approach that allows both build and generate the consistent dynamic autonomous knowledge-based systems.

Keywords: *Knowledge Representations (KR), reasoning, human activity, domain world, private world, ontology*

1. Introduction

Range of Knowledge Representations' issues, include, but are not limited to:

1. measure of KR approach's adequacy to the represented knowledge
2. measure of knowledge role with respect to the goal that is trying to be achieved
3. measure of overall quality of knowledge within the knowledge representation
4. measure of knowledge uncertainty for the knowledge utilization by the autonomous system
5. measure of the consistency of knowledge that is provided by the autonomous software agents or by the service providers
6. measure of the ontologies' role in autonomous systems

Proceeding from the assumption that human behavior is defined by his knowledge, we have a right to expect a successful evolution of autonomous systems only under the stipulation that it exists a reliable KR foundation.
Unfortunately, underdetermined system of KR's terminology itself produces numerous problematical KR approaches.
In this paper we will attempt to look at aforesaid KR issues as at reasoning's problems and to subordinate knowledge representation to reasoning one.

1.1 What do we mean by knowledge

In order to assess which types of knowledge representation are appropriate for which type of information, including corresponding performance measures as well as to consider other KR issues, it is necessary to define what we mean by knowledge.
Consider some knowledge definitions from Google:
The act or state of knowing; clear perception of fact, truth, or duty; certain apprehension; familiar cognizance; cognition.
"Knowledge, which is the highest degree of the speculative faculties, consists in the perception of the truth of affirmative or negative propositions." *Locke.*
That which is or may be known; the object of an act of knowing; cognition; -- chiefly used in the plural.
"There is a great difference in the delivery of the mathematics, which are the most abstracted of knowledges." *Bacon.* "Knowledges is a term in frequent use by Bacon, and, though now obsolete, should be revived, as without it we are compelled to borrow "cognitions" to express its import." *Sir W. Hamilton.* "To use a word of Bacon's, now unfortunately obsolete, we must determine the relative value of knowledges." *H. Spencer.*
"That familiarity which is gained by actual experience; practical skill; as, a *knowledge* of life. "Shipmen that had knowledge of the sea." *1 Kings ix. 27."*
As we see, knowledge is one of those concepts, concerning which everybody has own opinion. Nevertheless, the last one seems the most operable. Practically, it equates knowledge with an activity representation. In any case, (since the practical skills is used by human in his activity) it means that knowledge is a mental instrument, with is used for the human activity achievement.
Thus it is possible to say that knowledge is an instrument of reasoning.

1.2 Why does a human think?

Before we will define the reasoning model, it is appropriate to put a question – Why does a human think?
"Reasoning is a mediate generalized reflecion of appreciable and regular dependences of reality."[1]
As such it is an instrument of human life cycle providing.
"Thinking and acting are the specific human features of man. They are peculiar to all human beings. They are, beyond membership in the zoological species homo sapiens, the characteristic mark of man as man."[2]

Since a human life cycle is constituted by set of profession/living activities, reasoning serves these activities' achievement.
At that, knowledge is used as the human activities awareness.
To Ludwig Edler von Mises [2] "human activity is a goal-seeking behavior" and "human action is necessarily always rational".
And so by human activity we mean:

Definition 1. Human activity is time-, place-, state-, and event- ordered set of multidisciplinary actions aimed to achievement of socio-specified goal.

1.3 Activities' types

In spite of the obvious differences of social institutions and persons, their life cycle as set of activities, on closer examination, seems in the following way:

- An activity (activities) that provides the means of subsistence (both profession and other socially specified activities),
- Properly living activities, namely, learning, execution, repair, protection, *an advancement of results, supply, an analysis and control*.

The first activity (activities) belongs to certain area (areas) of expertise (domain). As domain activities we differentiate *domain generic activities* and *private activities.*
The properly living activities we designated by the common name of *generic living activities.*
Domain generic activity is a basis framework of actions, operations and/or activities aimed to achieve one or more *domain specific goals*, where domain goal is a socio-claimed product or service.
Private activity is an adapted domain/living generic activity provided by a social unit, where by *a social unit* we mean a government, an enterprise, a community and a person.
Thus we differentiate the following activity's types: generic living activities, domain generic activities, and private activities.

1.4 A human mental activity

Now, there is time for *correlate with each* other a social unit's life cycle, activities, knowledge and reasoning:

Definition 2. Reasoning is a human mental activity that operates with human activities knowledge for the purpose of the social unit's life cycle providing.

At that, on the *level of the social unit's life cycle organization* reasoning operates with activities as with data type, and on the *level of the activities' implementation* it operates with an activity's components (see below) as with data types.

It is important to note that reasoning trace is a certain algorithm, and its data types' names constitute *a reasoning ontology*.
We differentiate life cycle of domain (*domain world*) and life cycle of a social unit (*private world*).
Reasoning's algorithm of domain world we denominate as *domain world activity*, and, analogously, reasoning's algorithm of private world we denominate as *private world activity*.
We emphasize the domain world activity and the private world activity, since, as a matter of fact, they define *behavioral/management models* of domain or of a social unit.

Definition 3.

a) Domain world activity (Adw) is a resultant activity of the domain community, composed of domain generic activities (owned by domain experts) and private activities (owned by the other domain community's members), aimed to the domain life cycle providing;
b) Private world activity (Apw) is a resultant activity of the private profession/living activities, owned by social unit and aimed to the private life cycle providing.

2. THE Reasoning

The suggested Ontology Engineering approach forms a core of THE (Total Human Experience) Web project. In the network of THE Web project it is proposed to build an integrated Web knowledge resource (THE KB) with the purpose of the exhaustive Web service providing of the profession/living activities. THE Web service will be realized by an integrated multi-agent system (THE MAS) under multilevel dispatching.
THE KB is constituted by human activities' representation and derived ontological as well as causal environment.
Human Activity is represented in form of Activity Proposition (AP) on the Reasoning Language (RL). RL is THE Web's internal language that data types are represented by Core Ontology (CO), *Domain Ontology (DO) (as CO* extension), Private activity's ontology (PO) as a certain DO extension, Domain World Activity's Ontology (DWO) as DO extension, Private World Activity's Ontology (PWO) as CO and a certain DOs extension that are derived from corresponding activities' propositions (see below).
AP represents an algorithm of the activity performance' steady states transformation. So called Steady Reasoning (SR) serves (validates and directs) this algorithm performance. SR operates the following knowledge types:

- A private activity's *initial state* (AIS)
- A state *transforming* private activity (STA)

- Set of *possible STA effect* states as result of STA,

where a state knowledge includes

- a state ontology,
- a state determinant,
- determinants of state's components;

and a private activity knowledge includes

- an activity ontology,
- an activity's states,
- an instrumental private activities toolkit and
- an activity's determinant.

At that, an activity's and activity state's determinant is a semantic framework of its ontology's components that is a mandatory for inheritance at all generations.

RL provides also transient reasoning's means for the purpose of Transient Reasoning (TR) achievement. In addition to above mentioned, TR operates the following knowledge types, derived from THE KB:

- Network of generalized causalities,
- Generalized cause (that is, set of causes that derive from the same state the same effect),
- Causality determinant.

RL is interpreted by THE MAS reasoning framework (THE Reasoning).

THE Reasoning process is provided by the following agents:

1. *Recognizer* that recognizes determinants of activities and activities' components,
2. *Executor* that executes the AP's sequence of operations,
3. *Predictor* that predicts an eventual course of events,
4. *Reason _detector* that detects a reason of deviation from the specified steady state and generates a target setting,
5. *Activity_generator* that derives from the KB a new activity proposition as a discovered (or received) problem solving.

3. Activity Proposition

This paper is not RL presentation. Therefore we will consider RL features that concern Knowledge Representations issues only. RL is a procedural, a markup, an ontology language as well as an action language, destined for the description of reasoning that required for the activities' performance.

As a procedural language it allows to describe an activity's algorithm.

As an action language it represents a causality in the form of a triplet {I,C,E}, where I is an initial condition, C is a cause, E is an effect.

As an ontology language it allows to input both concepts and concepts' relations.

As a markup language it provides a semantic marking of AP text that allows the ontology mapping.

Activity Proposition plays a part of a canned program and at the same time it is considered as knowledge module. At once on completion of AP design, it occupies THE KB position in compliance with its causal interpretation.

At that, it is necessary to note that we extend concept of an activity actor beyond the social units. We mean by Activity Proposition (AP) a semantically marked description of *purposeful system of operations that producible by human(s) and/or by service provider(s) and/or by apparatus(es) and /or by software applications.*

At that,

- *Activity's ontology* is AP text's remainder of deletion both RL's terms and lexical forms as well as semantic tags (that is, a semantic ordered set of words (*ontology units*) used for AP representation).
- *Ontology unit's semantics* is fixed by the nearest semantic tags (opening and closing) and
- *Ontology unit's meaning* is Web, THE Web or private resource.

3.1. Personal world

Private world (PW) is constituted by set of actual private profession/living (p/l) activities derived from Basis and Domain generic activities. At that, ever it remains the PW composition, namely, *learning, practice (*that is, an execution of a socially specified activity(ies) that provide(s) the livelihood), *repair, protection, an advancement of results, supply, an analysis and management*.

Every p/l activity is correlated with others by time, by place, by preferences and by cost. Space of correlated p/l activities is rank-ordered by APpw that represents a scenario of parallel/sequential executable private p/l activities, which are marked by a special set of tags. RL keeps *AP special sets of semantic tags* that define an activity's position in the personal world. APpw provides a semantic sharing of private p/l activities as well as of private p/l ontology. Private world's activity represented by APpw is aimed to the achievement of it's owner p/l goals with a cost minimization.

A priority of APpw's performance produces *a particular causal stipulation* of private activities as well as *particular reasons of response* to external occurrences (*a private logic*). A corresponding APpw ontology has, therefore, *private semantic features.*

A private logic induces interoperability issues both on the profession and on the living level that must be considered as an operation problems both of PW management and of PWs interaction. In case that a

response to an external occurrence is not contradict APpw performance's logic it will be executed. If not, a response's execution will hurt the PW.
The response's motivation takes on special significance for reasoning, particularly, for Reason _detector and Activity_generator.

3.2 Domain world

Domain world (DW) is constituted both by domain generative activities and by private activities of professional communities, of enterprises and of specialists. THE Web engine keeps *AP special sets semantic tags* that define a profession position of all domain world participants. A corresponding domain world AP (*APdw*) provides a semantic sharing of domain activities as well as of domain ontology.
Domain world activity, represented by APdw, is aimed to the achievement of domain socioeconomic, sociopolitical and socio-productive goals with cost minimization. APdw performance is achieved via domain Web portal.

3.3. THE self-organization

Ontology constitutes the external level of human experience's knowledge representation. Every Onto-unit has THE KB's multi-semantic position represented by set of DW related triplets (APdwName, APName, SemanticTag) as well as by PW related triplets (APpw, APName, SemanticTag). At that every Onto-unitName is accomponied by links to DOName or CO (that is, to Onto-unit parent's name).
This Ontology organization grounds an opportunity of the interoperability issues' solving.
Recognition of an activity's determinant in the current input activates THE Reasoning process.

3.3.1 Target setting's processing

A target setting as an output of Reason_detector or due to a customer's initiative is sent to Activity_generator in form of an initial and a finite state.
Using knowledge of activities states' determinants, Activity_generator search the corresponding THE KB nodes and AP paths between them.
The next problem is a correction of one of this paths with the purpose of utilization it by PW owner.
This correction is a type of a semantic translation that represents a sequential revision of the inter-tags' spaces.
An impossibility of the inter-tags' spaces filling is fixed as a problem that involves a target setting for Activity_generator.
As result of this recursive procedure is a new AP.

4. Measure of KR approach's adequacy to the represented knowledge

Suggested Ontology Engineering approach deals with unified model of above mentioned knowledge types (see paragraph "THE Reasoning") representation. AP representations of existing software tools/agents/applications utilization's procedures will extend THE KB. Representations ones will be used as procedures of access to these resources. In the same way it represented an implement's, an apparatus', equipment's, a sensor's (and so on) utilization as an activity states' components representations. At that, principles of operation of above-named devices are represented by means APs too.

Thus THE Web operates with active knowledge forms, for which AP representation is adequately.

5. Measure of knowledge role with respect to the goal that is trying to be achieved

According to M. Polanyi [3], the components of an optimally organized system must not be further partible in the certain, defined for this system, ratio. M. Polanyi made out of a system's components at a ratio of their contributions to the goal achievement. A component's position in the system's organization defines its semantics. Its contribution defines the component's significance.
Due to RL notation, semantic tags define an ontology unit's contribution to the AP, and an ontology unit is utilized as a pointer of a related resource that details an access procedure (or this knowledge principle of operation). Thus THE KB represents knowledge system, ordered in M.Polanyi sense, and THE KR approach provides a contribution of every knowledge unit to the goal's achievement.

6. Measure of overall quality of knowledge within the knowledge representation

Since an ontological design is provided by domain expert or by APpw owner, the overall quality of knowledge within the knowledge representation is depended on its author's skill level or on the APpw owner's preferences that always may be submit for consideration of new customer. THE engine provides the AP designers' rating and chooses (in the presence of choice) the best AP version.

7. The ontologies' role in autonomous systems

Among a manifold of an ontology definitions the Protégé' one is the most close to RL notation:
"Ontologies are explicit specifications of the types of resources that exist and possible relationships between them, and specific instances of concepts in the ontologies" (http://protege.stanford.edu/).
THE Reasoning utilizes an ontology as a semantically ordered set of Web resources' pointers. Similarly, a human operates on concepts. At that, as concepts it is used both scientific/technical/common terms and arbitrary identifiers of arbitrary objects sets as well as of various process' parts, of states, of situations and so on.

Therefore in THE notation the problem of primary importance is *a reconstruction of the individual conceptual system* (that is, the private ontology mining). A discovery of corresponding DO/CO terms grounds a semantic translation of private situation to the DO/CO specification. Only after that it is possible generating for customer a personalized Web service. Remind that in the previous chapters we considered an ontology as data type names' space. Thus, since reasoning process is grounded by conceptual schemes, an ontology plays a part of primary importance for all knowledge based systems include autonomous ones.

8. Conclusion

We considered a particular Knowledge Representations' approach. We simplified a problem by consideration unified KR form called Activity Proposition. We consider that it optimally satisfies both human and machinable reasoning and that in such a way we are able to build of a personalized Web service.

Reference:

[1] "Reasoning is a mediate generalized reflecion of appreciable and regular dependences of reality." (http://azps.ru/articles/proc/proc9.html)
[2] Ludwig Edler von Mises, "*Human Action: A Treatise on Economics*",The Foundation for Economic Education, Inc. Fourth revised ed., 1996, printed 1998
[3] M. Polanyi, Personal Knowledge, Harper & Row, New York, 1958