Hierarchical Structures Enabling Industry 4.0

Bernd Kast[1], Sebastian Albrecht[1] and Wendelin Feiten[1]

***Abstract*— Industry 4.0 requires the integration of still scattered and isolated solutions for automation and autonomous systems. For this purpose we propose an integrated development environment that enables autonomous systems ready for deployment.**

The system is based on a hierarchical modeling paradigm supporting flexible, easily extensible formal descriptions (concepts) of all relevant entities from the abstract semantic level down to physical detail. These concepts and their instances represent the declarative knowledge, i.e. they state what is known about the world. Complementary, the system also provides procedural knowledge, i.e. operators, which generate new declarative knowledge from given declarative knowledge. Both, concepts and operators, provide common interfaces, use strong typing and allow for simulating and controlling the system on various levels of abstraction.

Using these hierarchical structures, the planning tasks related to complex cyber-physical production systems (CPPS) are broken down into small, manageable parts. Each part of the task is then addressed by suitable planning techniques, which on their own would not be able to solve the overall problem. Steps in such a plan have to be refined one after the other in order to gather all physical details needed for execution. The proposed structure allows a seamless transition from abstract steps over simulation to real world execution.

We present an assembly use-case highlighting several key elements of our system. The task is to mount several modules on a hat-rail (common parts in a control cabinet) by using two robot arms with multi-purpose grippers and wrist-mounted cameras.

I. Introduction

Automation has come a long way. There are excellent solutions and systems that cover different aspects of automation and autonomous systems, but hardware and software of these systems are mostly engineered for specific tasks and predefined products. However, Industry 4.0 addresses small lot sizes and considers substantial variances in tasks, product properties and hardware setups. This yields the need to reduce the engineering efforts considerably and to enable an autonomous system to adapt fast. An integration of these currently often scattered and isolated solutions and systems in one common framework seems mandatory to cover the variety of tasks.

Independent specification of product and manufacturing hardware, which is vital for flexible production, requires abstractions of properties and capabilities. A common approach is to denote the abstract capabilities of an autonomous system as skills [1]. Such skills can then be fully described on a semantic level, hiding all sub-symbolic elements needed to actually deploy this capability on the real hardware. On the downside, those skills have to be defined on a rather high level of abstraction to achieve independence of sub-symbolic properties, which adds a lot of complexity to their implementation. Mapping continuous variables on discrete ones is no solution as well, since the planning task would run in problems due to exponential growth of complexity.

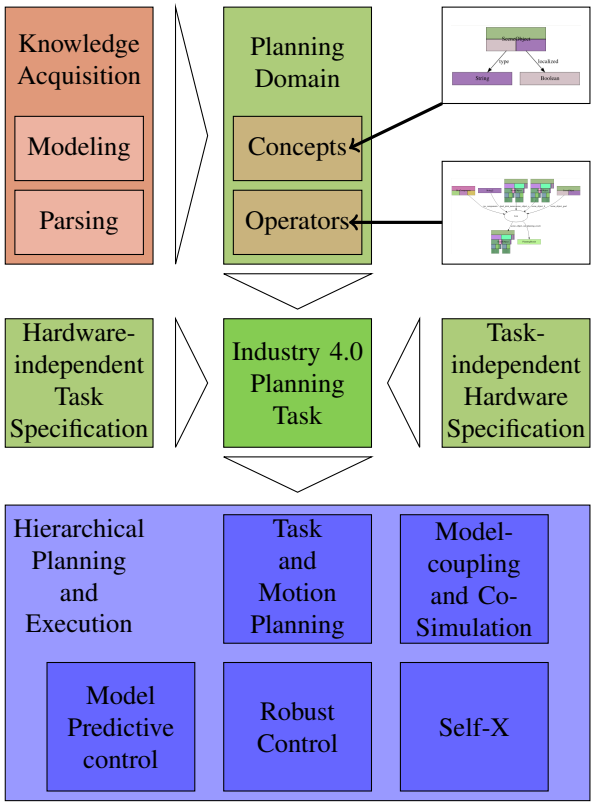

Fig. 1: Schematic overview of modeling and control elements needed for the autonomous system: Concept (declarative knowledge) and operator (procedural knowledge) modeling, planning task specification, hierarchical planning and execution on the hardware

Well-established methods of classical planning [2], [3] are applicable to fully symbolic domains (as they can be modeled in PDDL), while real-world examples of assembly processes link steps in the task sequence on a sub-symbolic level as well. For example, the grasp of an object has to be chosen in such a way that collisions can be avoided and the next assembly steps are possible. Thus classical planning could only be applied here if all relevant sub-symbolic aspects are captured by suitable models on the symbolic level. Such a lifting results easily in exponential growth of the problem

[1]Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany bernd.kast@siemens.com

size and demands problem-specific modeling, which is no option for Industry 4.0 setups. On the other hand, solving the full task and motion problem on the most detailed sub-symbolic level poses problems that also cannot be solved fast enough due to the curse of dimensionality.

We propose to factorize the overall task into smaller planning problems in order to reduce complexity at the cost of optimality. However, this factorization is a hard task on its own. Without suitable structures and tool support the engineering process of modeling suitable hierarchies is challenging and poses the risk to link software and task to specific hardware again. Therefore, we introduce formal models for the declarative and procedural knowledge in the following, that allow determining suitable hierarchies of planning tasks automatically. Using hierarchical planning strategies it is then possible to solve the overall tasks and generate suitable control strategies for the hardware with reasonable complexity.

II. Main Contributions

- Flexible hierarchical modeling of procedural and declarative knowledge using common structures: Flexibility due an arbitrary number of levels and structures enabling reasoned abstractions from continuous properties to semantic levels.
- Automatic decomposition of planning domains: Factorization is key for scalable solutions in productions systems. Our hierarchical modeling paradigm supports automatic decomposition to generate smaller planning tasks, breaking the curse of dimensionality - possibly at the loss of optimality.
- Framework integrating domain-specific planners: Domains, that can be fed to domain specific planners (e.g. Pddl-planners, HTN-Planners, sampling based planners) can compiled from our factorized domains concentrating on relevant aspects.

III. Additional Contributions

- Modeling support: Graphical user interface to model all structural knowledge, declarative and procedural, in a single tool by making explicit usage of hierarchies.
- Providing a base to integrate various tools for analysis and optimization, e.g. data driven algorithms
- The framework is demonstrated in a two armed robotic use-case assembling modules on a top-hat-rail (see below).

IV. Assembly Use-Case

An assembly task is chosen as an example to test the proposed hierarchical modeling and planning framework: four modules have to be assembled onto a top-hat rail (common parts in a control cabinet). Following the line of hardware-independent product specifications, the input specification uses fully-symbolic concepts of these modules. The precise poses of all components are unknown in the beginning and have to be determined by suitable perception steps.

The autonomous system is a two-robot setup shown in figure 2. Both arms are equipped with a parallel gripper and a wrist-mounted camera for localizing parts.

Depending on the actual pose of a component, the part can either be picked and mounted by one arm alone or a handover to the other arm is needed to change the orientation in the gripper, such that the mounting motion is collision-free. Confer [4] for details on low-level task and motion planning to solve the hard task of choosing grasps and robot arm motions avoiding collisions while reaching the desired goal assemblies. Note that the actual clicking of the component onto the top-hat rail is a difficult process in the presence of measurement uncertainties. It has been solved by actively considering these uncertainties and by applying suitable planning techniques [5]. If more than one component is mounted, all elements have to be pushed together to make the needed connection between the components.

In our example the hierarchical planning evolves over the different levels of the hierarchy. The planner on the first planning level checks only if sampled start and goal poses exist such that no collisions occur. On the second planning level, coarse trajectories that are throughout collision-free are computed. The final level combines detailed planning with the actual execution: the trajectories are optimized here and the hardware is controlled accordingly. The following figure 2 shows some snapshots of such an assembly process. Confer [5], [4] for more details.

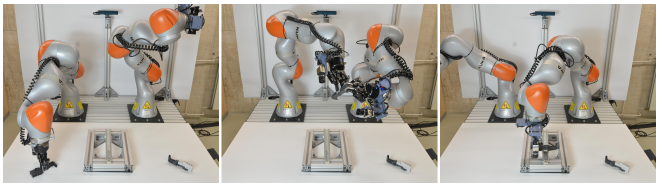

Fig. 2: Snapshots of an assembly done by the multi-robot assembly cell

References

[1] J. Backhaus, M. Ulrich, and G. Reinhart. Classification, modelling and mapping of skills in automated production systems. In *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 85–89. Springer, 2014.

[2] M. Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[3] D.S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J.W. Murdock, D. Wu, and F. Yaman. Shop2: An htn planning system. *Journal of artificial intelligence research*, 20:379–404, 2003.

[4] P. S. Schmitt, W. Neubauer, W. Feiten, K. M. Wurm, G. V. Wichert, and W. Burgard. Optimal, sampling-based manipulation planning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3426–3432, May 2017.

[5] F. Wirnshofer, P.S. Schmitt, W. Feiten, G. v. Wichert, and W. Burgard. Robust, compliant assembly via optimal belief space planning. In *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018.