A Final Technical Status Report for

Cost Effective Coordinated and Cooperative Robotics Enabled by Open Technologies

SwRI® Project No. 10.23202

Prepared for:

National Institute of Standards and Technology (NIST)

CFDA Number: 11.609

Prepared by:

Southwest Research Institute®
6220 Culebra Road
San Antonio, Texas 78238

Matt Robinson
Manager
Adaptive Technologies Section
matt.robinson@swri.org
(210) 522-5823

January 16, 2019

Approved:

Paul T. Evans, P.E.
Director
Manufacturing and Robotics Technologies Department
Intelligent Systems Division
Southwest Research Institute

A Technical Status Report for

Cost Effective Coordinated and Cooperative Robotics Enabled by Open Technologies

TABLE OF CONTENTS

TABLE OF FIGURES

1 Executive Summary

The United States' Manufacturing base is continually faced with cost-pressure, as a result of rising raw material, energy, labor and benefit costs. As a result, manufacturing companies need to become smarter and more resourceful in how they adopt, adapt and implement solutions for their manufacturing processes.

According to a 2015 U.S. Department of Commerce Economics and Statistics report (2012 Statistics of U.S. Businesses Employment and Payroll); almost 99% of all manufacturing firms in the U.S. are non-enterprise firms with fewer than 500 employees. While these very small to medium-size manufacturers account for just less than ½ of all manufacturing employment (45.5%), they still have a need for low-cost manufacturing solutions, in order to remain competitive in an increasingly competitive and global market. However, the high cost and level of engineering skill and expertise required to realize full automation potential is at odds with the lack of training, education and resources, required to adapt automated equipment rapidly and cost-effectively, and thus, meet the changing demands within smaller manufacturing environments.

Lowering the cost of automated solutions, by targeting areas where manufacturers have the most need to improve productivity, should be paramount for these smaller operations. For example, by automating quality control processes (replacing slow and costly manual inspection methods); through better equipment interoperability through flexible automation solutions; and in collection and analyzation of real-time production data, the result should be improved quality and increased output.

Proposed was an effort to investigate the use and bridging of open standards and technologies, and application of the results, within a flexible automation testbed that demonstrates lowering the cost of automating typical processes; such as in-process inspection; intelligent part management; and automated, just-in time servicing of machine and machine cell applications. Open standards, along with open source software, were used as a foundation to enable teams of mobile robots in the near-future to leverage the outcomes of the program to make dynamic decisions, based on the active needs of machine operations. Software and enhancements to standards were fully open-sourced and now available to NIST and to the public.

2 Project Scope

This project built upon the successful completion of Project Number: 17NCDMM12, Robot Control Integration Enhancements, using ROS-Industrial and MTConnect under NIST Grant Opportunity Number 2012-NIST-MSE-01. This initial project demonstrated the ability to implement ROS-Industrial to program a robot, and use MTConnect protocol for communications, between the robot and a CNC machine tool. Similar to the previous effort, this effort was primarily software based and used the open standard application level protocol, MTConnect, and the open source Robot Operating System (ROS) Industrial to enable facility-level interoperability between robot teams and machine-cell devices.

2.1 Background and Purpose

MTConnect is an open, royalty-free standard intended to foster greater interoperability between controls, devices and software applications, by publishing structured data over networks, using the Internet Protocol. ROS is an open source project that provides a common framework for robotics applications. ROS-Industrial is an extension of the open source ROS software stack (software suite), specifically for industrial robots and automation, to enable more advanced capabilities for manufacturers' future applications.

The expansion of the previous ROS/MTConnect solution, further enhances the viability of using industry-supported open source software for smart manufacturing applications. Open source software permits a continuation of free development, over a very large development workspace that ultimately solves complex problems, where the solution is free to the end user. The output from this project is intended to be an enabler for industry-wide adoption of open source technologies, by providing a use-case and testbed, showcasing lower cost solutions for comprehensive factory floor integration for the small and medium-sized manufacturer.

2.2 Technical Approach

The main technical thrust of this project was to advance the Interfaces standard, defined in Part 3.1 of the MTConnect standard. As defined therein, *Interfaces* are special types of machine components that represent a physical or logical connection, between two devices where *REQUEST* and *RESPONSE* types of messages are implemented. For example, an MTConnect-enabled robot may *REQUEST* an MTConnect enabled machine tool to open an access door, from which the machine would reply with a *RESPONSE*; a machine tool makes a *REQUEST* for material to be loaded to a robot to which the robot would have some RESPONSE.

Currently, this scheme has been tested between two devices paired directly. This project addressed the concept of extending the paradigm of supporting work cells with a single stationary device, to multiple interconnected devices, and potentially swarms of devices in a mobile environment in the future. One concept for investigation and evaluation was a messaging structure that allows for simplified failure recovery scenarios and reduced coupling of inter-related devices. Decoupling of devices then allows for replacement of parts of a system; a busy or faulted robot is replaced with another robot that can perform the required activity. The interface layer remains the same, since any device capable of handling material is capable of doing the work.

3 Tasks and Proposed Outcome

The following tasks detail the effort conducted by the team. SwRI managed the program and worked closely with AMT and System Insights on the technical activities. The combined team, along with participation from NIST, established the concepts for new extensions to the standards, the software architecture, and the interaction between ROS-I and MTConnect. SwRI led the ROS-Industrial activities, working closely to expand the initial implementation of the ROS-Industrial/MTConnect Bridge, one of the key outcomes of the program.

AMT and System Insights led the MTConnect activities, working closely with SwRI, on the interface standard with ROS-Industrial and enchantments of the MTConnect standard, with respect to interfaces. The team produced a prototype demonstration at the conclusion of the effort. The team presented an initial simulation to mimic the proposed enhancements and showcased new functionality. Once the simulation provided sufficient data from testing and evaluation, and goals are achieved, this was extended to a physical integration and demonstration of capabilities.

3.1 Project Management and Reporting

Southwest Research Institute has managed the effort of cost monitoring and schedule; performance and status was reported to NIST monthly. The team conducted regularly scheduled calls with NIST to discuss project tracking and activities. In addition to monthly status reports a mid-year technical update and this final report have been provided in conclusion of the effort.

3.2 Requirements Engineering

This task focused on interfacing with industry stakeholders to understand potential use-cases, the state of the art in fabrication work cell automation and in-process metrology, and elicit a guiding set of requirements. Documented was an estimate of lost opportunity within existing systems, along with loss root causes (e.g. limited integration between the CMM and CNC causing wasted out of spec parts). Through study and identification of actual manufacturing use-cases, the team will better understand gaps, and thus, where additional functionality should be targeted to advance the interconnection of ROS-Industrial and the MTConnect standard, for broader manufacturing-floor efficiencies.

3.3 Architecture and Design of Software Systems

Based on the use cases established, and the physical systems specified in Task 1, the team evaluated the MTConnect and ROS/I standards for improvements and additions required. Anticipated high-level enhancements, identified during the proposal process, include:

1. Temporary pairing between devices for the duration of a single task. For example, the robot interacts (pair) with a single device (feeder, CNC, CMM) as part of load or unload operation.
2. Robot selection of tasks based on current needs of devices. Allows for dynamic decision making, based on active interfaces and workflows.
3. Identification and handling of multiple parts and managing multiple inspection levels. This included accounting of the final part status, rework, rejected, completed.
4. Addition of QMResults to the asset capabilities.

3.4 Prototyping the Systems Architecture

Prototyping is used frequently, during the requirements portion of a project as a way to elicit requirements. Some characteristics, such as screen or report formats, can be extracted directly from the prototype. Other requirements can be inferred by running experiments with the prototype.

The project iterated through two general prototype models (static and dynamic), in order to explore methods of coordinated activity. The static prototype model employed simple, low cost devices as proxies for actual robotic systems. Visual indicators and simple pushbutton input devices, enabled by existing device GPIO, will stand in for user interface and actual machine actions. Device movement, where required can be human assisted.

Once software systems were verified and desired functionality validated, the dynamic prototype module employed an actual robot to move and interact with simulations of other systems (carried over from the static model), as shown below in Figure 1. This was conducted in parallel with the buildout of the demonstration system.

Figure 1: Work cell prototype

3.5 Simulation of the System Architecture

This task focused on software adapters for the ancillary components and how to conduct a full workpiece cycle with and without faults. This included development for proposed extensions to each standard to support the new capabilities and interfaces. In addition, existing state machines were improved to include intelligent routing and rerouting of raw material and parts. As it relates to the demonstration, one goal was to develop error-handling-logic, so that fault tolerance was provided, as it relates to method of recovery.

The team created a kinematic (no physics) simulation of the work cell to mimic production flow and cell activities The simulation of the work cell (robot, machine tool, and accessory) used the ROS robot visualization and MTConnect simulators for each piece of equipment in the prototype (See Figure 1). The simulation demonstrated the temporary pairing of the devices and the coordination of activities with multiple devices all sharing a central resource. The software programs prepared for this effort were thoroughly debugged during the development of the simulation. The simulation was driven, using a representative production cycle (i.e. multi-part, multi-hour run), from Task 1. Part variation was injected into the dynamic model in order to evaluate the effectiveness of different strategies. These strategies included different robot strategies, as well as different inspection levels. The results of this simulation included metrics (throughput, cycle time) for the various employed strategies.

3.6 Physical Testbed Deployment

In order to validate the results from the simulation, the team installed and operated a pilot work cell at a partner facility. The software was deployed and tested in a controlled environment to demonstrate functionality. Both positive and negative functional tests were performed to highlight concepts, such as recovery from exception conditions, like service interruptions and component failures for example. The platform was tested for performance and stability, with data being logged for later analysis.

The Machine Tool partner, Hurco, supported the team to ensure that material loading logic was added to the machine tool's PLC and NC configuration. The CNC was able to report on its controller state, as well as its current health and operation. The programmable interface retrieved information from the numeric control system, as well as the programmable logic controller to determine its operational state. From the material loading aspect, the Machine Tool had the ability to request material to be loaded and unloaded from a remote device and report on critical components states, such as chucks and spindle interlock.

The robot controller coordinated by SwRI, provided by Universal Robots, was ROS-capable and had the ability to read or accept data from an outside PC running the Linux operating system (Ubuntu distro). The robot had the ability to programmatically read/write data generically over a TCP or UDP socket connection, made with the Linux PC. The data passed between the PC and the controller was used to pass state information, as well as command motion. Motion was commanded in trajectory streaming, where points are streamed to the controller and executed as received.

3.7 User Community Communication and Transfer to Industry

SwRI led and coordinated enhancements to open source ROS-Industrial code-bases, provided communications to the community, and discussed the outcomes in the Developers Meetings and via the ROS-Industrial Blog. AMT and System Insights led the modifications and enhancements to the MTConnect Standard and presented to the TSC and the TAG. These efforts served to transfer knowledge to industry for their use and adoption of the outcomes of the program.

4 Progress Towards Task Completion

This section describes the progress on the program.

4.1 Project Management

Per the stated objectives, Southwest Research Institute has led the project relative to cost monitoring and schedule. Performance and status have been reported to NIST stakeholders on a monthly basis, via provided monthly status reports. Team meetings were open to NIST stakeholders and have been documented, via the team's Confluence project website [1].

4.2 Requirements Engineering

Upon kicking off the project, and the team meeting face-to-face with the NIST and partner stakeholders, consensus was set to move to an agile approach. Requirements were driven by scenarios, and as scenarios are tested this drove requirements.

The requirements that were determined were separated by requirements relative to ROS and requirements relative to MTConnect. They were listed at the final stages of this work as:

ROS Requirements to realize stated capability:

- Respond to Request - A machine will make requests that ROS must respond to
- Prioritize Tasks - Tasks that are available are prioritized by ROS prior to binding
- Bind Status - ROS reports binding status (unbound, binding, bound)
- Observation - ROS checks for available tasks
- Sub-tasks - ROS creates necessary sub-tasks to complete bound task

- Task Complete - ROS provides information that task is complete

MTConnect Requirements to realize stated capability

- Many-to-many - The MTConnect interface shall enable multiple robots to bind with one of many CNC machines or CMM systems.
- Dynamic resource allocation - The MTConnect interface shall enable dynamic allocation of resources.
- Standards - MTConnect shall use QIF standards Must Have
- Error Recovery - MTConnect shall facilitate error recovery

4.3 Architecture and Software Design

An established architecture, that was tested in simulation, and has been demonstrated in a physical demonstration, has been developed. The state models have been documented, as they have been conceived, and the exercising of these state models was performed over the duration of the project.

A summary of the architecture has been included herein, which remains largely unchanged, since the one year project update, and is also maintained within the project documentation site.

Overview

The architectural concerns for device orchestration and collaboration, provide the framework for allowing multiple manufacturing processes to coordinate their activities to complete a task. The central axis of this model will be the abstract task, which will need to be defined in the following cases, described in [2]:

- Single-task robots (ST) vs. multi-task robots (MT): ST means that each robot is capable of executing at most, one task at a time, while MT means that some robots can execute multiple tasks simultaneously.
- Single-robot tasks (SR) vs. multi-robot tasks (MR): SR means that each task requires exactly one robot to achieve it, while MR means that some tasks can require multiple robots.
- Instantaneous assignment (IA) vs. time-extended assignment (TA): IA means that the available information concerning the robots, the tasks, and the environment permits only an instantaneous allocation of tasks to robots, with no planning for future allocations. TA means that more information is available, such as the set of all tasks that will need to be as- signed, or a model of how tasks are expected to arrive over time.

Collaboration Models and Binding

A task is the unit of work expressing a desired goal that devices can collaborate for some productive end. The unit of work is defined by the task, with the necessary collaborators and their required capabilities. The association of tasks with devices was being referred to as binding–what is happening is collaboration, since it is a peer-to-peer process. The requestor is the organizer, but each player is equal in the exchange.

Information Models

As with all MTConnect information models, there are be two levels of abstraction, archetype and the instance models.

The archetype represents the common task information that serves for all tasks of that type. The archetype pertains to a specific type of collaboration that requires a set of devices, with a given set of capabilities. The capabilities model is shared with the process model, since they are ostensibly the same information.

This frees the instance to only include the information about the individual collaborators and the state of the individual task being performed.

The instance of the task contains the state of the task, and the state of the currently available collaborators that have interest in the task. The instance is monitored by all the devices wishing to participate allows the devices to track the status of the task. Task coordination is paired with the real-time data, coming from the agent, to allow for device tracking as well as the asset.

Only the coordinator/requestor can modify the instance, so the task requestor is required to monitor the state of all participants. There is some issue about how to limit the number of devices that need to be watched in a swarm scenario, but in a cell, it should not be an issue.

Devices

After reviewing a number of scenarios of material handling and transfer, it became imperative that all steps of the manufacturing process have representation. This means that if there is inventory between the machine tool and the CMM, the inventory or collector of parts will need to be an intelligent device that can create load and unload tasks. This architecture simplifies the temporary storage of parts and allows for tasks that move from point to point to be managed.

An unload task from a machine tool is really unload from the machine tool, and loaded into inventory before it is placed in the CMM, unless there is no buffer. Regardless, the task specifies a source and a destination. This means that one device in this scenario is passive, and one is active, or there are two tasks that are related and coordinated. The other side of the material handling, unload from the machine and load to the inventory could also be seen as a subtask, but would bind lock the inventory during the unload, which would be unnecessary.

Otherwise, the load to the inventory is effectively passive or a completely separate task that naturally follows the material flow from the CNC. To follow this, there is a flow of material that follows the process steps, and should not duplicate the material flow in the process. But, there needs to be a way to state that a task involves two steps, unload and load.

Another option is that they are both subtasks of the parent. This would necessitate delayed binding; could be done with a single subtask that is related to the parent task. If task pairing is required, then the initial bind of the parent will only include the parent device capabilities or the specific task capabilities. The issue is that if the device representing the inventory is currently bound to a task, it will be unavailable for any other tasks. In a multi-robot scenario, this could be an issue.

The explicit nature of the movement of the material to the inventory is a good thing, since it will allow for more complex and deterministic material handling that can be coordinated by self-aware devices.

One other architecture is to have the tasks as associated peer tasks that allows the inventory and the CNC to associate them as sequential. Types of associations can be sequential and parallel, depending on the requirements. Associated tasks will allow for handoff from one to the other and when one retires the other picks up if sequential. They will share at least one device that is going to be the collaborator and can be associated, once the other task begins.

State Models

Below are Figure 2, Figure 3, Figure 4, the working state models, as they were presently conceived to be tested via the primary scenario, the robot servicing the CNC and the CMM, via an observational state machine process, as described. However, to enable extensibility to the multiple mobile robot scenarios, where multiple resources could possibly respond to requests, the model needs to account for these use cases.

Coordinator Only Tasks

It became evident, as the design of the physical system and testing of the architecture moved forward with Scenario 1, the CNC, CMM, one robot scenario, that an additional task type was required. This is the Coordinator Only Task, and it is required when there are cases where a task will be created as a placeholder for an activity that must be performed by a device, but does not require collaboration. The purpose of this is to allow other devices to track the completion of these intermediary tasks for sequencing purposes. Examples of these include, TakeMaterial, PlaceMaterial, (or Object Target), OpenDoor (if door is not automatic), etc. These tasks help to inform the other devices when they can perform the next activity or when the control should switch.

These tasks indicate that they do not involve interfaces by the absence of any collaborator. If a collaborator is required, then the task will be paired with an interface, otherwise they are informational. Since the tasks must be executed in the order as prescribed in the parent task, the next sequenced task cannot begin until these tasks are completed. Though this can be orchestrated between the individual devices, it is not imperative that the tasks are present and therefore they are optional.

The coordinator is be advised that the tasks are added to make the structuring of higher level tasks more implicit and deterministic. When a task is not included, it is assumed that the device is responsible for the task and will handle it internally, for example, picking up a part or taking/placing a part.

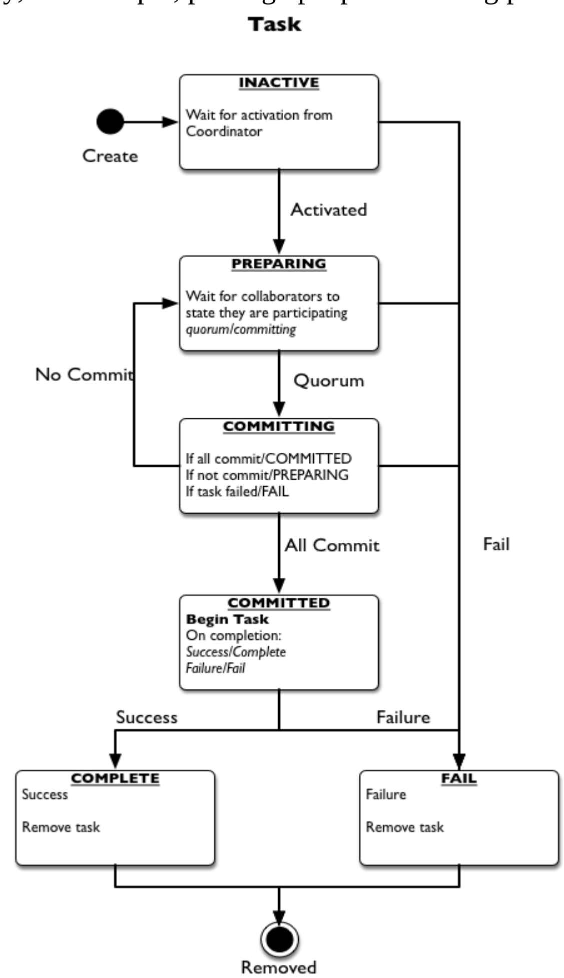

Figure 2: Task State Model

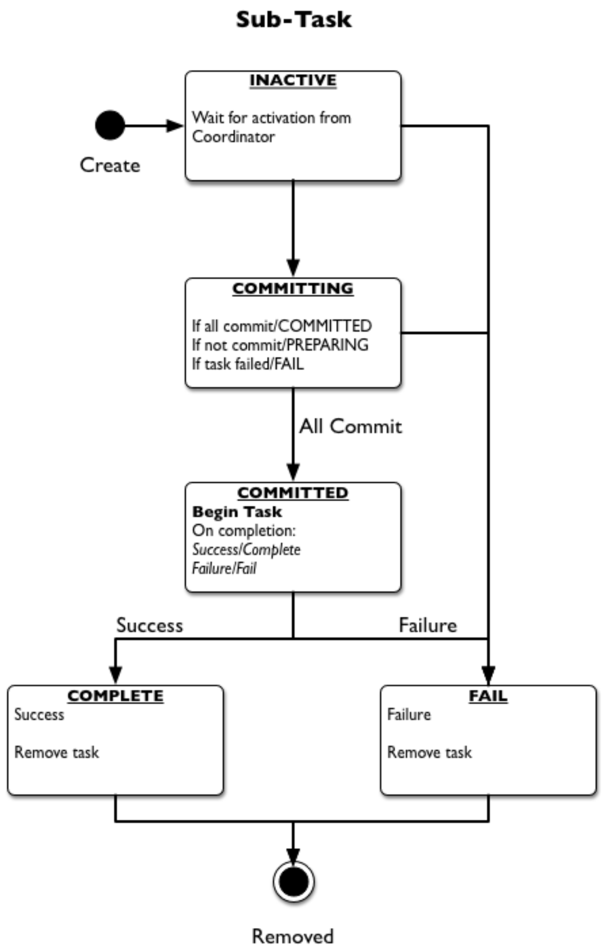

Figure 3: Sub-Task State Model

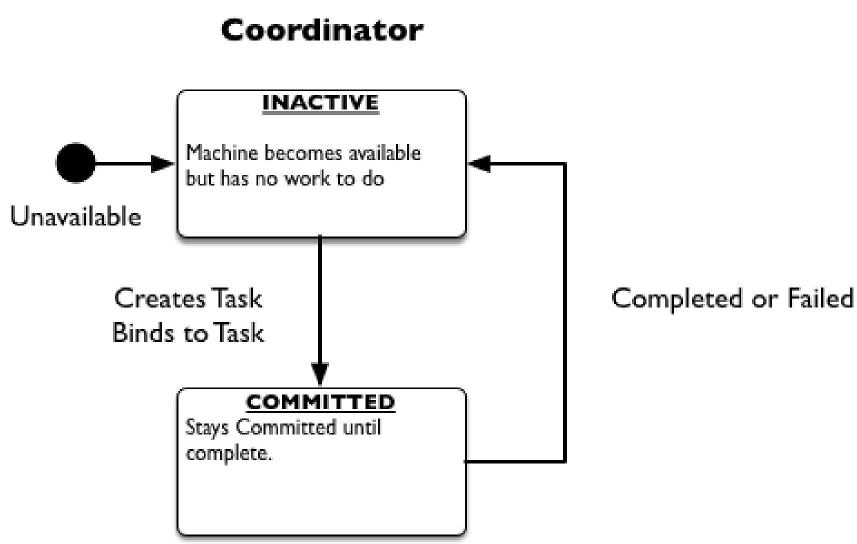

Figure 4: Coordinator State Model

4.4 Prototyping of the System Architecture

The current status of implementation of the system architecture exists in two primary parts. The first is a version of the MTConnect ecosystem, where the architecture state models exist and respond to simulated resource state signals. The second part of the prototype architecture allows for state machines to exist as well as a visualization environment, within the ROS kinematic simulation environment RVIZ.

There is a working RVIZ environment that can host and visualize the state of each asset and represent the assets, with collision zones/collision models, with the ability to incorporate multiple robots. There is also a Python-based MTConnect ecosystem that enables the creation of state machines and the tuning of the architecture for the specific scenarios. The following, Figure 5 and Figure 6, are examples of the latest instance of the visualization environment of what was demonstrated at the recent International Manufacturing Technology Show in Chicago, in September 2018. These simulation tools are available now on the MTConnect Github site for full evaluation and use.

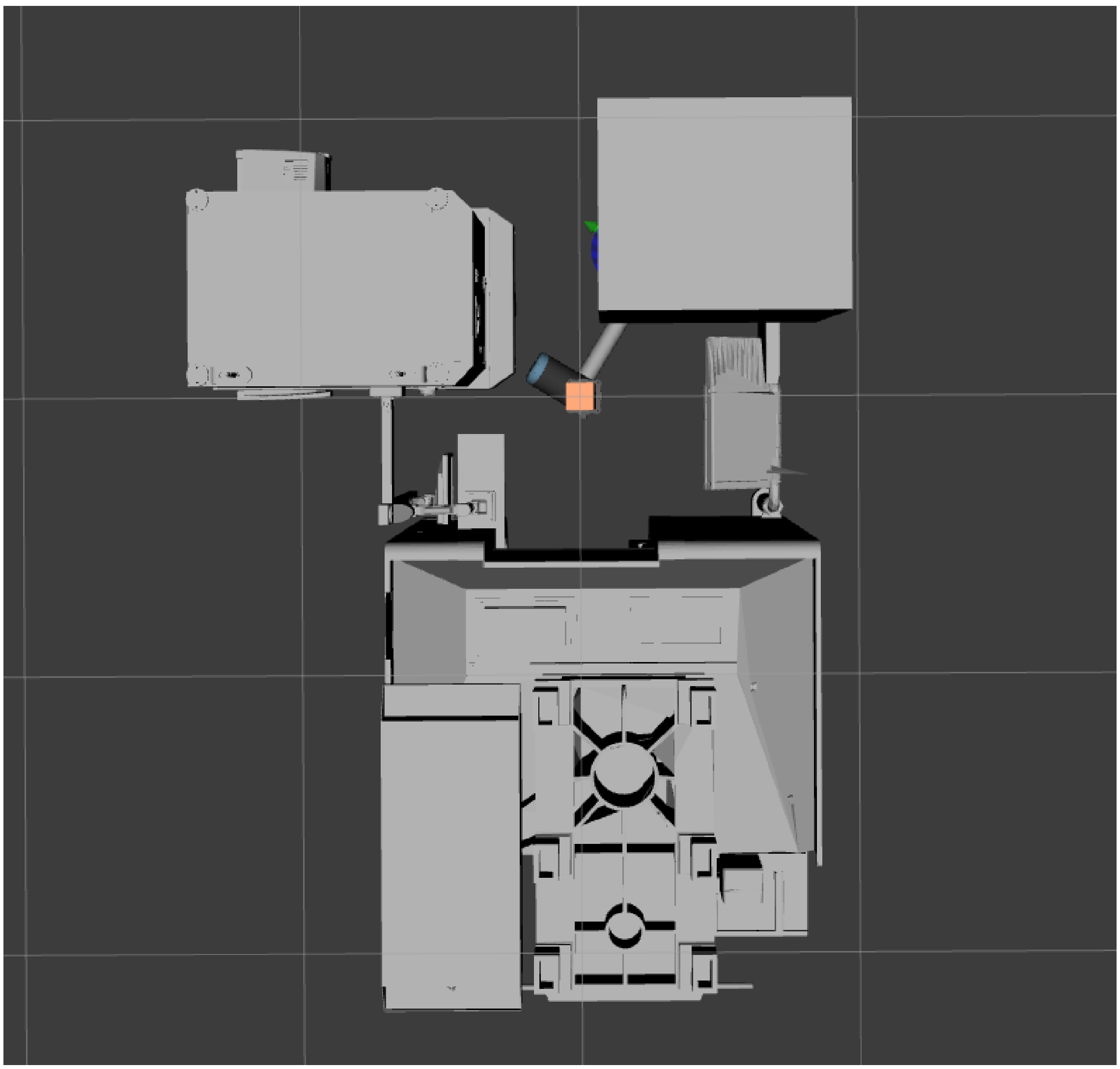

Figure 5: Layout View within RVIZ environment for eventual demonstration cell.

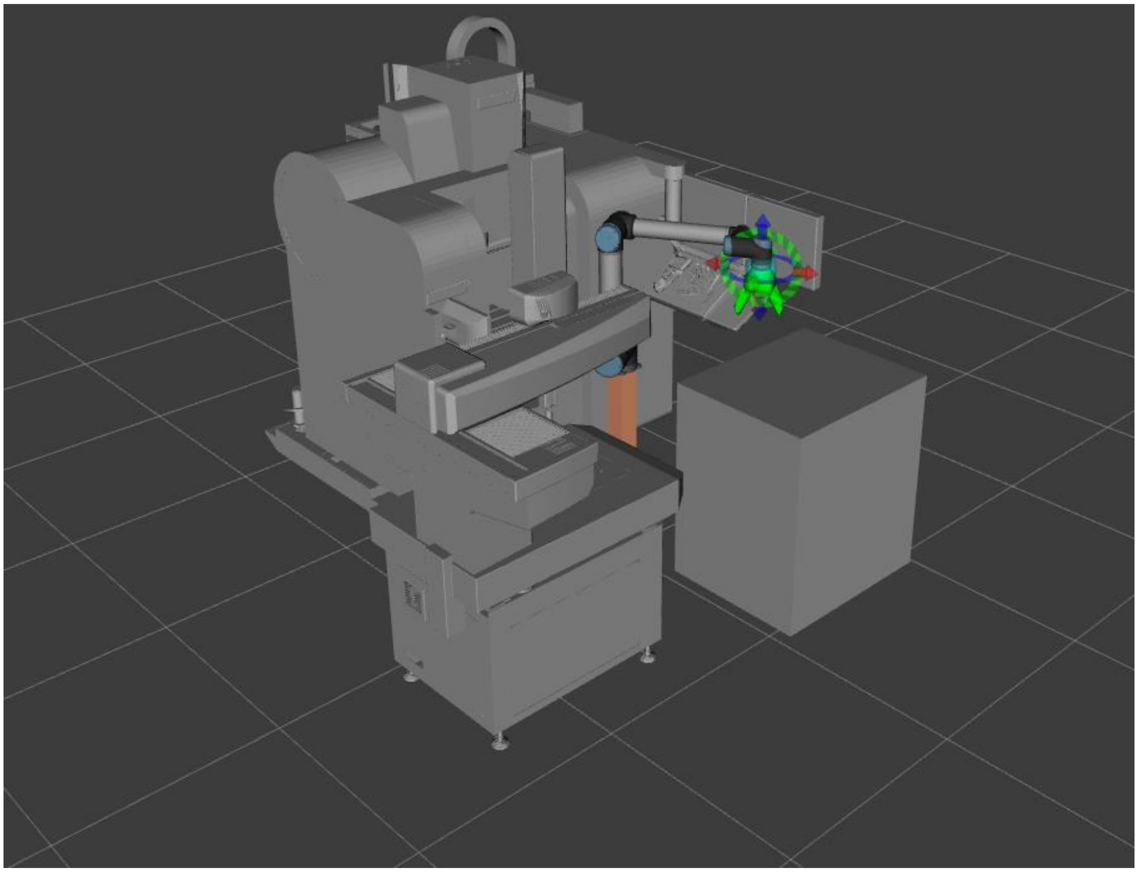

Figure 6: Isometric view of demonstration cell.

4.5 Simulation of the System Architecture

Detailed simulation was performed for, scenario 2, detailed below, of the robot working between a machine tool and a measurement device. This resulted in the development of the Collaborator Only task type described in Section 4.3.

To enable additional testing of the System Architecture and the MTConnect to ROS Bridge, a Test Environment was created and the details for setting up the environment have been developed on the project website. [3] The setup includes detailed instructions, including launch commands that launch a graphical environment to show a (kinematically) simulated robot and an example workcell. If RViz starts up, and the user sees a 3D visualization of a robot and the other specified hardware, the user is ready to begin testing.

Per previous status reports, additional scenarios were considered. These scenarios, 3 through 11, were attempted to be tested virtually in the environment, however updating of the environment needs to occur to enable collision checking, between the multiple robots that may be added to the environment. While this was not a significant addition to the environment, timing and resource limitations did not allow this update to be completed. Additionally, instructions/documentation for the creation of the state machine within the MTConnect ecosystem was added within the last reporting period to support follow on testing of these additional scenarios.

Scenarios to be considered

1. Existing work: Two devices collaborate to load and unload material from a piece of manufacturing equipment.
2. A robot, machine tool, measurement device collaborate to make a “good” part that passes in process inspection.
3. A robot, and a pair of machine tools, and a measurement device collaborate to make a part.
4. Two robots, a machine tool, and two CMMs collaborate to make a part.
    a. Robots can attend to any piece of equipment.
    b. Robot 1 loads machine tool, Robot 2 unloads machine tool and attends CMM.
5. Mobile platform retrieves material from stock room and delivers to cell.
6. Mobile platform retrieves material from stock room and attends machines in cell.
7. Kanban–robot takes parts from the bin and follow dynamic process plan to make the part.
8. Multiple identical machines.
9. Multiple heterogeneous machines with different capabilities – milling, turning, etc.
10. Multiple collaborative devices working to solve a problem.
11. Multi-Robot coordinated resources
    a. Of multiple robots, two robots need to load a machine tool in concert, they must retrieve the appropriate EOAT and retrieve the correct material. and then, properly interact with the machine tool. Points of failure: EOAT not available, Material Not Available, Appropriate for task robot not available, failure of “loading” process (chuck not engaging, work holding on machine tool not confirming part loaded), robot 1 not in final position when interference/interaction needed/sensed.
    b. Of multiple robots, two robots need to retrieve/support/manipulate a part/assembly for another robot to execute a process. Points of Failure: EOAT for handling robots, material availability, EOAT for Process Robot not available, fault around being in proper position when expected, Process fault related to coordination (i.e. handling robots manipulate, while process robot executes process, but process feedback indicates distance between process EOAT and part is too great, this could be represented as simply material variation).
    c. Of multiple robots, a part requires a series of process steps. These process steps can occur in parallel with three unique EOAT configs. Three robots “pick up” the proper EOAT and begin their processing. Some sequential processing could be incorporated, such as in a sealing operation there could be “prep” process, followed by application of sealant, then final robot passes over with a curing EOAT process. Points of Failure: EOAT availability, Right robot with right EOAT, executing process flow, faults related to process, particularly if sequential operations, as described above, faults related to material variation.
    d. Multiple Robots need to execute the same process in parallel to meet a Takt Time. This use case could be multiple robots approach a car frame and perform spot welding. All have same EOAT, but need to coordinate to realize a time-based optimum. Points of Failure: EOAT availability, robot availability, process fault, optimization of coordination/process execution.
    e. Multiple robots deliver add-on components to facilitate a robotic assembly operation. A robot assembler requires a flow of material and multiple robots deliver these “just in time” as the assembler needs them. Points of Failure: Material Availability, synchronization of the delivery robots, availability of deliver robots, part to part variation impacting time of assembly and impacting coordinating deliver robots.
    f. Multiple Robots operating on a low-lot, high mix, multi-process line. Four large and flexible robots have access to an array of EOAT. The robots move along the value stream

stations executing process steps as dictated. When required, two robots will work cooperatively. These processes can be, but not limited to assembly, light duty fixing of members to each other (tack welding), executing fastening, subsequent assembly, NDI, and sealing. Coordination with delivery robots is in scope as well. Points of Failure: Material availability, variability, scheduling of these robots, timing of the execution as variation is managed, coordination of delivery, NDI.

g. A mobile robot interacts in-process, with a station based robotic system, within the process flow, within a standard fabrication robotic cell, the station robot will stop on part 1 to index to a part 2 to continue process, as the station robot is working on part 2, the mobile robot adds parts that station robot will come back and process when it is complete. This is common operation, today with an operator, in large fabrication facilities. Points of Failure: Material Availability, process execution on time, variation of the fabrication process impacting ability of mobile robot to assemble.

h. Multiple Robots converge on a part to execute an assembly process. This will be a mix of type of robot, large payload to manipulate the base component, and multiple process robots doing assembly (coordinated between handlers and other process robots) and additive or subtractive process robots. This could be defined as a high mix swarm. The use case could be an agile automotive assembly process, where larger robots manipulate the chassis as other process bots come and do various processes on the manipulated chassis. Points of Failure: EOAT, robot availability by type, by process requirement, part variation, time to execute, coordination, material availability, quality fault, collision recover, assessment of damage, particularly in automotive or aerospace, where high value part could be damaged.

4.6 Physical Testbed Deployment

Physical Testbed Deployment, per the last status report, was considered a preferred path forward. At the time of this writing, a partnership is in progress, but final details are not available at this time. Additional funding requests through the Advanced Robotics for Manufacruting (ARM) Institute have also been submitted that would leverage Manufacturing Innovation Institute (MII) project calls, to further the capabilities and migrate the capability to more robust transport protocols (DDS and OPC-UA), as well as test the functionality on ROS2. A number of Original Equipment Manufacturers (OEMs) have expressed interest in supporting the testbed, and the deployment will continue beyond the conclusion of this particular funded initiative.

4.7 Demonstration at IMTS

The International Manufacturing Technology Show (IMTS) is one of the largest manufacturing technology shows in the world. With the support of the Association for Manufacturing Technology (AMT), and via collaboration with Hurco, Hexagon Manufacturing Intelligence, Universal Robot, and with support from Robotiq a demonstration cell was developed leveraging:

- Hurco VM5i 3-axis CNC
- Hexagon Tigo Shop Floor CMM
- Universal Robots UR 10
- Robotiq S-Gripper

An integration session was hosted by Hurco, where all the hardware was integrated, along with the software, and testing commenced. Various modes of operation were developed and tested, including a response to a bad part, which led to the robot doing a tool change, and sending the non-compliant part back to the CNC for rework. During the demonstration phase, a number of technical challenges were solved, including limitations in the accuracy of the path planning, due to the UR10 ROS package, and issues around the ethercat interface for the Robotiq gripper.

The demonstration took place September 10-15, at Chicago's McCormick Place in IMTS' Emerging Technology Center, Figure 7. The demonstration at IMTS was intended to show the type of operation or intelligence that may be deployed, by leveraging this new approach to interoperability. As seen in Figure 8, the intent is to enable a robot, leveraging ROS/ROS-Industrial, to communicate with other types of manufacturing equipment that already take advantage of the MTConnect standard, as far as communicating what they are doing.

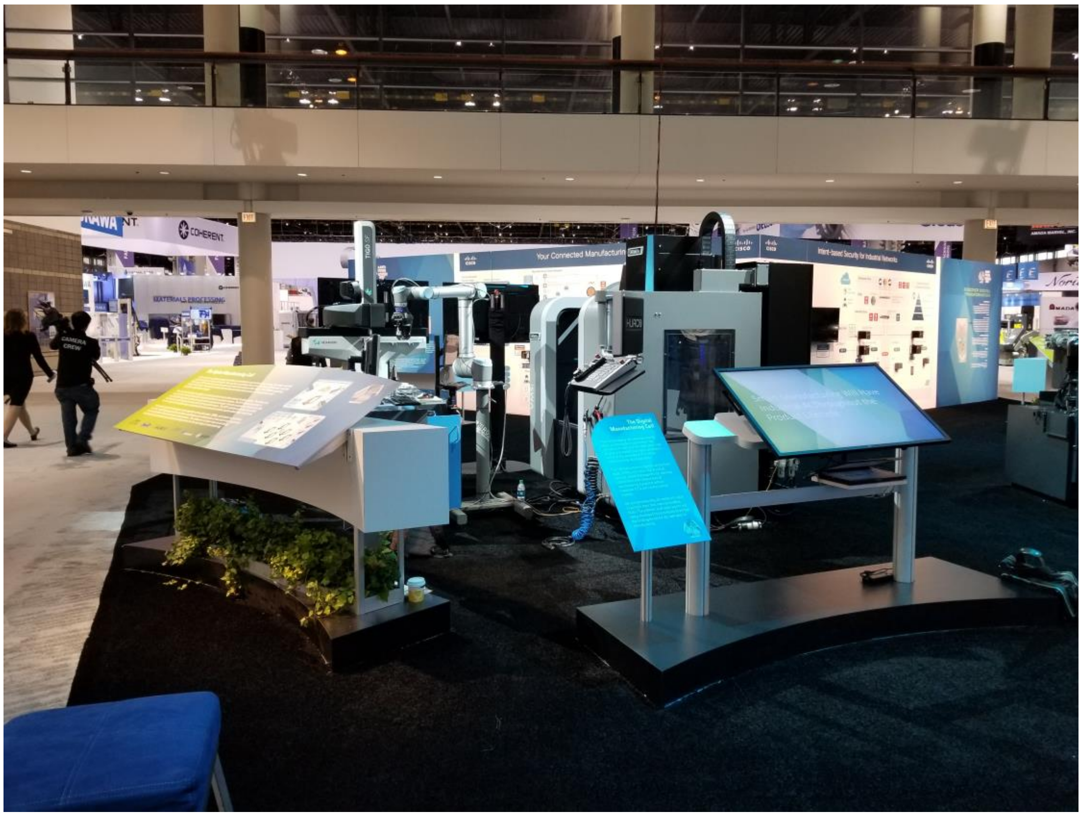

Figure 7. Demonstration at IMTS within the ETC.

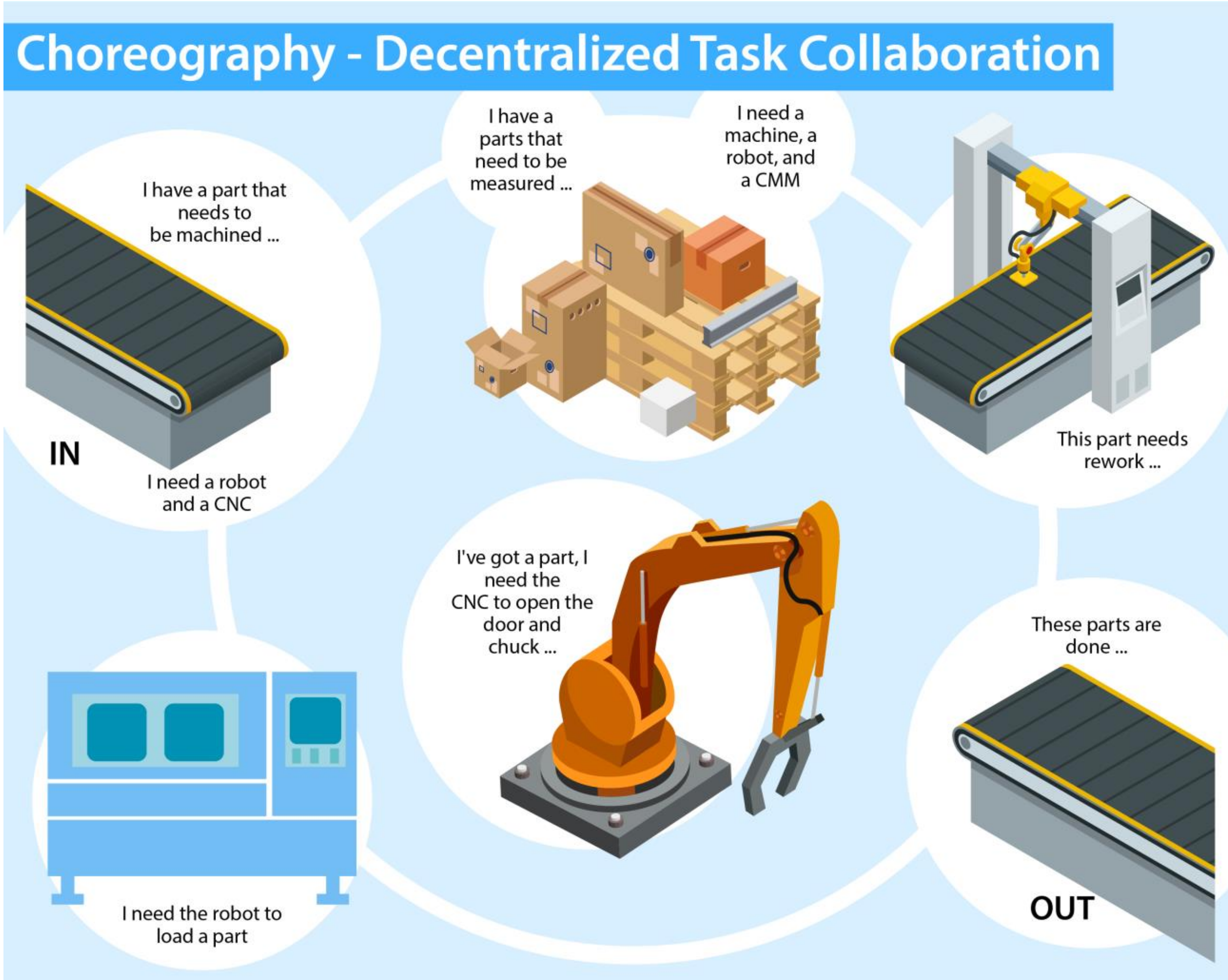

Figure 8. Collaboration Demonstration at IMTS

The demonstration conveyed the principles, such as enabling dynamic response to a non-compliant condition. Tool change was held, due to some issues with the gripper, but overall, through interaction with the attendees, the team was able to convey the principal. Both physical motion/response, within the demonstration and the state views, along with the RVIZ visualization, enabled a thorough understanding for attendees that had questions. The larger goal was to communicate a future state.

This future state leverages the developed framework that is inherently extensible. For instance, industrial AI will be an essential addition to future capability, enabling the notion of autonomous, continuous improvement or dynamic optimization through learning. Plans can be previewed, as conditions change, and subject matter experts that choose to intervene to ensure consistency in value stream performance can also be additional input into this Industrial AI capability. A graphical representation of this future state was presented at the demonstration and can be seen in Figure 9. [4]

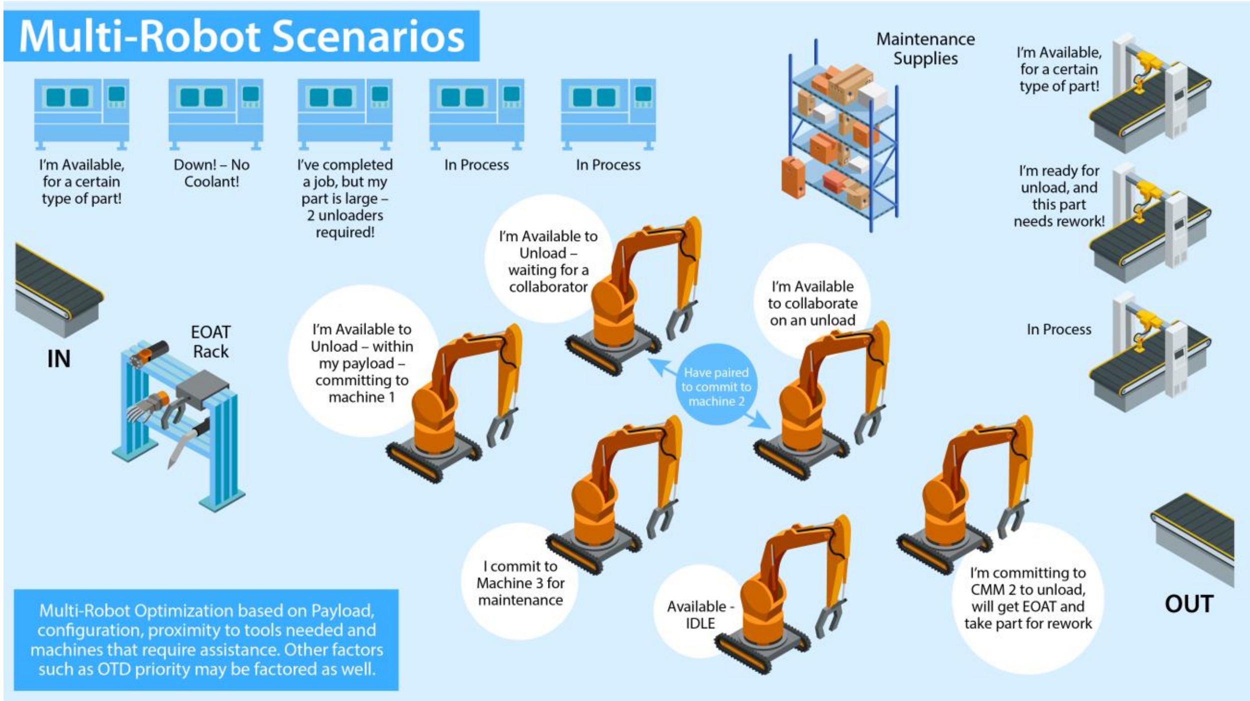

Figure 9. Future state enabling dynamic collaboration

5 Remaining Schedule

This project has now concluded, and this status report serves as the final report for this project. There is currently no funding or tasks remaining to complete. The full bridge code and the simulation environments have been open-sourced, within the MTConnect GitHub organization, for reference and leverage by NIST, our commercial partners, and the broader public.

5.1 Project Status

There were originally proposed two phases around requirements, gathering and specification development/amendment. However, once the team decided to adopt an agile approach, realizing requirements through performance, around end user scenarios, the phased approach was no longer. Below is the program status representing the agile methodology employed.

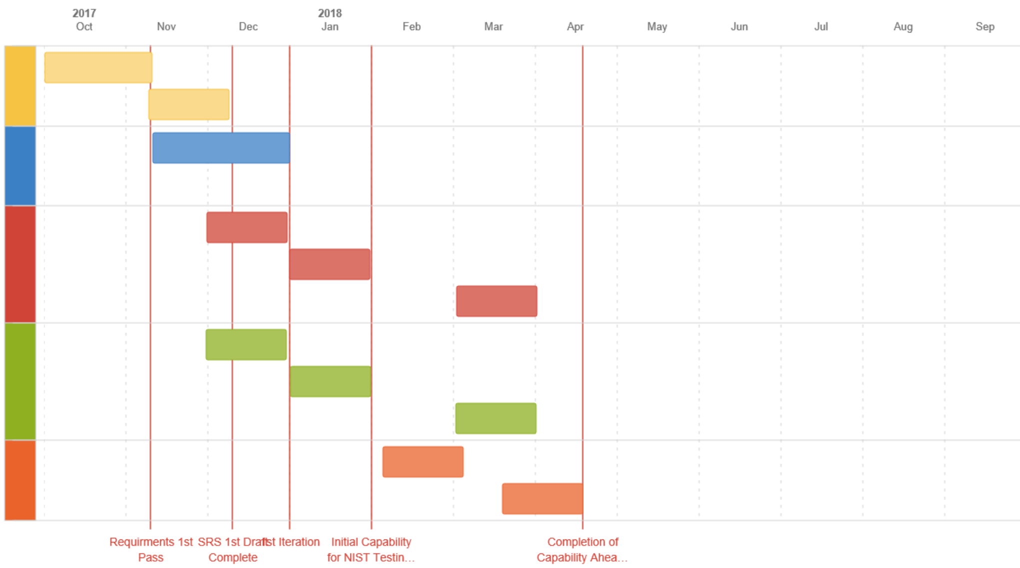

Figure 10: Project Status Chart

As can be seen in Figure 10, the project tracked behind, due to delays in securing hardware and delays in finalizing the software. However, interfacing with the hardware was efficient, due to their existing support of MTConnect. Iteration and interfacing with hardware was also parallelized with simulation and testing against scenarios indicated in Section 4.5. The extensive testing within simulation environment and the full team being able to work on-site, as the demonstration was being assembled, enabled for rapid capability refinement at the demonstration cell was completed.

An unplanned follow-up visit was required to refine performance, relative to the UR10 motion planning, and the performance of the ethercat communication with the Robotiq gripper. These were both resolved, and the demonstration testing and validation was completed, prior to shipment to the IMTS show location.

6 Program Budget Summary

The budget to perform all tasking as detailed in this proposal is $322,170. This includes costs to cover supplies and materials, all team members' participation on this program, travel and associated expense. To date $183,011 has been consumed. Minus commitments to our partners, a budget of $135 remains at conclusion of the project

A no-cost extension, till November 15, 2018, was approved to enable further testing of the multi-robot scenarios, described in Section 4.5. As noted previously, some improvements to the simulation test environment were required, including documentation for the MTConnect ecosystem for the creation of additional state-machines for the various types of equipment that were to be supported. Unfortunately, all the improvements to the simulation environment were not achievable, but documentation and refactoring into libraries were completed prior to the November 15, 2018 deadline.

Furthermore, full open-source, once MTConnect ecosystem was refactored into libraries with appropriate documentation, along with the specified ROS packages at https://github.com/mtconnect occured on November 15, 2018.

7 References

1 *Cost Effective Coordinated and Cooperative Robotics Enabled by Open Technologies* . (n.d.). Retrieved from https://confluence.datasys.swri.edu/display/CECCREBOT/Cost+Effective+Coordinated+and+Cooperative+Robotics+Enabled+by+Open+Technologies

2 *ROS Environment Setup, CECCREBOT Project Communication Site, Josh Langsfeld and Matthew Powelson,SwRI*, https://confluence.datasys.swri.edu/display/CECCREBOT/ROS+Environment+Setup

3 *IMTS 2018 – Leveraging Open Standards and Technologies to Re-Imagine Interoperability within Factories*, https://rosindustrial.org/news, Matt Robinson, September 2018, https://rosindustrial.org/news/2018/9/28/imts-2018-leveraging-open-standards-and-technologies-to-re-imagine-interoperability-within-factories