2006-2513: THE DISTRIBUTED RECONFIGURABLE FACTORY TESTBED (DRFT): A COLLABORATIVE CROSS-UNIVERSITY MANUFACTURING SYSTEM TESTBED.

Jonathan Luntz, University of Michigan

E. Emanuel Almeida, University of Michigan

Dawn Tilbury, University of Michigan

James Moyne, University of Michigan

Keith Hargrove, Morgan State University

The Distributed Reconfigurable Factory Testbed (DRFT): A Collaborative Cross-University Manufacturing System Testbed

Abstract

As a collaborative effort between the University of Michigan (UM), and Morgan State University (MSU) in Baltimore, a Distributed Reconfigurable Factory Testbed (DRFT) has been developed as part of the NSF Engineering Research Center (ERC) on Reconfigurable Manufacturing. The testbed combines hardware and simulation components at both universities operating under common control using secure channels over the Internet, and is designed in such a way as to ease the addition and modification of its various components The original UM RFT comprises 1) a serial-parallel manufacturing line, 2) a Virtual Factory software component, 3) factory-wide open software integration platform and data warehouse, 5) modular logic control developed at the cell level, and 6) a multi-teir networked control and diagnostic structure. To this system, the MSU portion was added comprising an Automated Storage and Retrieval system and a conveyor, where communication for monitoring and control was achieved via the OPC protocol operating through a Virtual Private Network (VPN).

Students from both universities came together to plan the application and control structure for the combined system, giving students at MSU a chance to learn first-hand from the experiences of the students at UM. A demand-pull manufacturing application was built on the combined testbed where the MSU system functioned as a ”Supply Cell” operating under control of the System Level Controller and the Software Infrastructure at UM. A robust handshaking communication protocol was developed to ensure the correct ordering of events in the presence of Internet communication delays and uncertainties. The inventory of the Supply Cell was tracked at UM where decisions about part availability to meet orders were made by the Software Infrastructure. Once parts were retrieved form the MSU Supply Cell, identical parts were introduced to the UM manufacturing line and virtual factory for machining and assembly. Web-based Human Machine Interfaces (HMI's) enabled students at both sites to monitor and control the operation of the entire system, and students could communicate and view both systems over a set of webcams. The combined testbed was successfully demonstrated at the ERCs annual NSF site visit in May 2005.

1 Introduction

It is uncommon in undergraduate engineering programs, particularly at smaller teaching universities, to present students with educational experiences in manufacturing with a practical thrust. The resources involved generally prohibit such experiences, limiting manufacturing education to textbook-type learning. At large research universities, however, manufacturing research platforms are more common, and can be leveraged into part of the educational program. The National Science Foundation Engineering Research Center (ERC) on Reconfigurable Manufacturing at the University of Michigan (UM) is a unique facility with a number of full scale manufacturing research platforms. As part of the ERC, the Reconfigurable Factory Testbed (RFT) is a prime example of a manufacturing research platform which focuses on both device level and system wide control system research which has also been used as an educational platform. The RFT combines hardware, simulation, networking, logic control, and software components, all of which are part of an active research program, but has also been used as a platform for many undergraduate course projects and for educational outreach programs to other universities and to secondary schools. This paper describes a collaborative activity between the University of Michigan, a large research institution, and Morgan State University (MSU), a small, historically black teaching school.

MSU has joined the UM ERC as a partner school, and has been developing collaborative activities under the ERC in recent years. As part of this collaboration, students at MSU have been working with an existing smaller, older, factory testbed at MSU and learning from students at UM about operating more modern and comprehensive factory hardware, controls, and software in the context of the RFT. One major goal of this activity is to provide students at MSU with practical manufacturing experiences beyond what could be made available at such a teaching school, and to serve as a model for other similar collaborations. Through this experience, MSU students (primarily undergrads) were able to work as part of a large research team, and learn from the research experience of the graduate students at UM. Students learned and worked together both remotely and during extended site visits. The activity described in this paper was to build a combined system, operating under consolidated control over the internet, using the existing RFT at UM and the testbed hardware at MSU. In the development of this Distributed Reconfigurable Factory Testbed, students at both ends gained experience in a wide range of topics from working on large teams, to deployment hardware in device networks. The activity culminated in May 2005 in the NSF advisory board site visit at MSU where the distributed testbed was successfully demonstrated.

This paper is organized as follows: Section 2 describes the existing RFT system at UM before the collaborative activity. It is an updated version of the system as described in an earlier paper[1]. Section 3 lays out the combined system architecture, which was expanded significantly over the course of the effort. Section 4 describes how the collaborative activity was undergone, along with some of the details pertaining to the combination of the two systems. Finally, Section 5 makes concluding statements including enumerating some of the educational and technological outcomes of the activity.

2 Reconfigurable Factory Testbed Infrastructure at U of M

Prior to the collaborative effort, the Reconfigurable Factory Testbed (RFT) at the University of Michigan comprised the following: 1) a serial-parallel manufacturing line composed of two machining cells, 2) a virtual factory component including simulated versions of a third machining cell, an inspection cell, and an assembly cell, 3) logic control in the form of industry standard Programmable Logic Controllers (PLCs) and in-house Modular Finite State Machine (MFSM) controllers to control the real and virtual hardware components, 4) a factory-wide open software integration platform 5) a variety of data and field bus networks connecting the components, 6) a set of Radio Frequency IDentification (RFID) readers for part tracking, 7) an autonomously guided vehicle (AGV) for transporting parts from the Serial-Parallel Line to the Virtual Factory and 8) an established reconfigurable production system utilizing all components therein.

2.1 Serial-Parallel Line

The Serial Parallel line (SP line) is the central hardware component of the RFT and consists of a conveyor linking two machining cells, as shown Figure 1. The conveyor carries pallets which

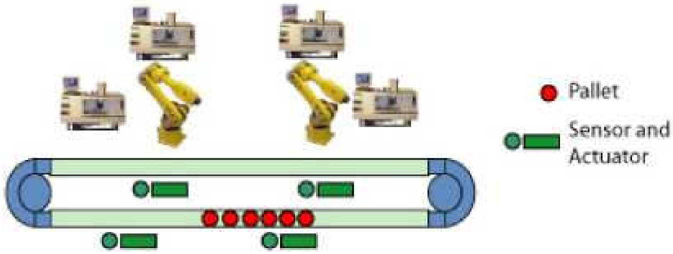


Figure 1: The Serial-Parallel Machining Line

Figure 2: The RFT Virtual Factory - from right to left, the machining cell, the inspection cell and the assembly cell

transport the parts to be processed. There are four pallet stops along the conveyor each with a proximity sensor and a pneumatic actuator to stop the moving pallets. The first stop is used to introduce parts into the system, the second and third stops are at the cell locations for introducing and retrieving parts from each cell and the fourth stop is used for transferring parts onto the AGV (that is, the point where parts exit the SP line). At each machining cell there is one robot and two milling machines. The robot picks parts from pallets and places them in machines for processing, and also returns fully machined parts to empty pallets on the conveyor stop.

2.2 Virtual Factory

One component of the RFT is the virtual factory, which is a piece of software used to build simulations of factories. By using simulations, systems can be built and tested whilst the hardware is being installed, reducing system ramp up time. The solutions found to control the simulations can then be used to control the real hardware. This allows easy testing of reconfiguration solutions without having hardware, as the control logic does not differentiate between the real and virtual factory.

The RFT virtual factory consists of one machining cell which is a replica of the real machining cells of the SP line, one inspection cell and one assembly cell, as shown in Figure 2.

2.3 Logic Control

Logic control plays an important role in the RFT as it coordinates the discrete dynamics of the manufacturing system. There are two levels where logic control is used: the cell level and the system level. The cell level control coordinates hardware and software resources whereas the system control coordinates the cell control with the higher level software control.

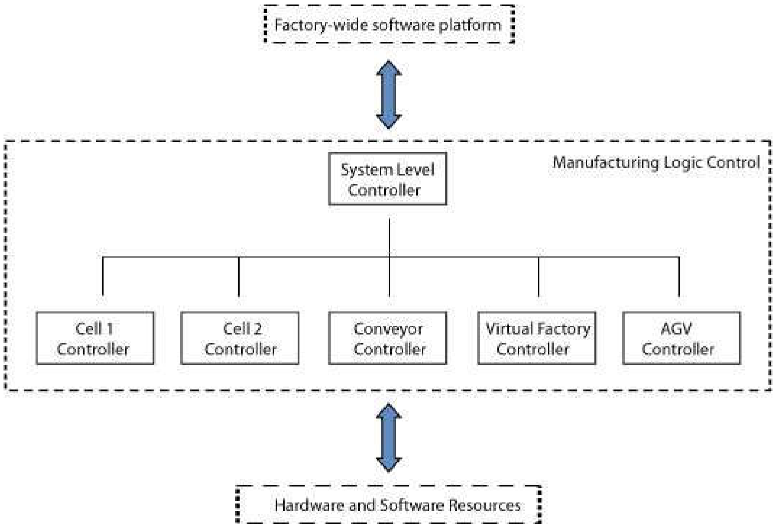


Figure 3: The RFT logic control hierarchy

At the cell level there are several logic controllers implemented in different ways. As shown in Figure 3, cell 1 refers to the first machining cell of the SP line, its controller has been built with Modular Finite State Machines (MFSMs)[2] and runs on a PC. The logic controller for cell 2 is a Win-LC Siemens controller whose control logic has been built as Sequential Function Charts with Siemens Step 7 software and the controller runs as a soft PLC. The third cell level controller is the conveyor controller which runs on a Allen Bradley PLC from Rockwell Automation and whose logic has been designed using ladder logic. There is also a controller for the virtual factory which, similarly to cell 2, runs on a Siemens Win-LC soft PLC, and a controller for the AGV which is run on a proprietary application.

To manage the entire manufacturing system, a higher level logic controller called System Level Controller (SLC) was built. The SLC communicates with all cell level controllers and with the factory-wide software platform. It has been built as an Event-Condition-Action (ECA) Modular Finite State Machine, a variation of an MFSM controller that is driven by ECA rules (for higher reconfigurability)[3].

2.4 Software Integration Platform

Higher level functionality of the RFT cannot be handled through logic control alone. To perform complex analysis and to make planning decisions involving data and complex logic, software modules are required which can run arbitrary code, potentially from a variety of different languages and software packages. An open software integration platform is required to manage software events in the system and the passing of information between dissimilar software modules and between the factory floor components.

The software infrastructure developed at UM is a data-centric, web-standards based system. The core control component is the Control Workflow Manager (CWM) which is responsible for maintaining and executing a set of control rules stored in the database. Upon events generated either by software modules or by hardware (sensors or logic controllers, for example) under what

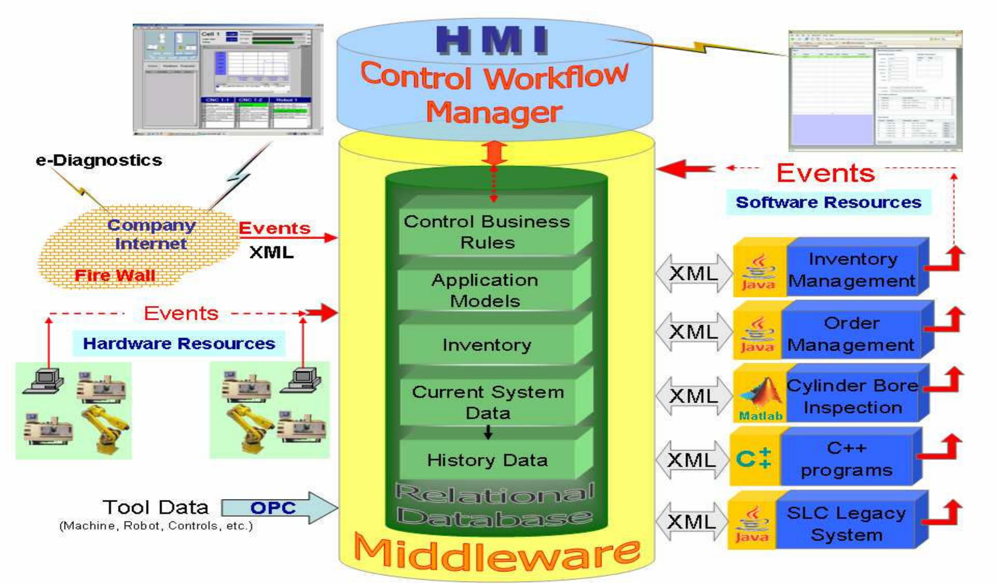


Figure 4: The software infrastructure architecture.

conditions various software modules should be executed. The execution of software modules and the passing of data is controlled with the middleware server which uses SOAP and XML to pass events and data to any of a large number of modules and applications including C++ or Java programs, Matlab, Labview, or commercial factory monitoring and control software. The operation of the software infrastructure system is centered on a database, which facilitates the passing of data between modules and contains the control rules. This architecture is depicted in Figure 4 and is described in detail elsewhere[4,5].

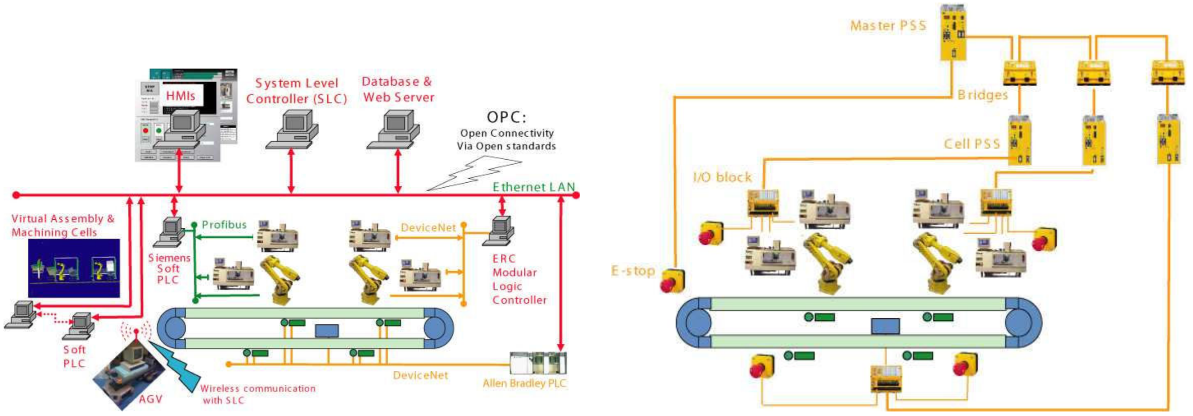
2.5 Multi-Network Connectivity

The RFT encompasses many network layers. The main layer is an Ethernet network which is used to network the SLC with all cell level controllers and software control. OPC (Open Protocol for Communication) is the technology that is used to establish the communication in between the different controllers across the network.

The control at the various cell levels is made across different field buses. At cell 1 and for conveyor control, a DeviceNet network is used whereas for cell 2 a ProfiBus network is used to connect the cell controller to the machines and robot. The control of the AGV is made using a wireless connection and there is also a completely separate SafetyBus p network from Pilz Co. to carry emergency signals from emergency stop buttons and sensors.

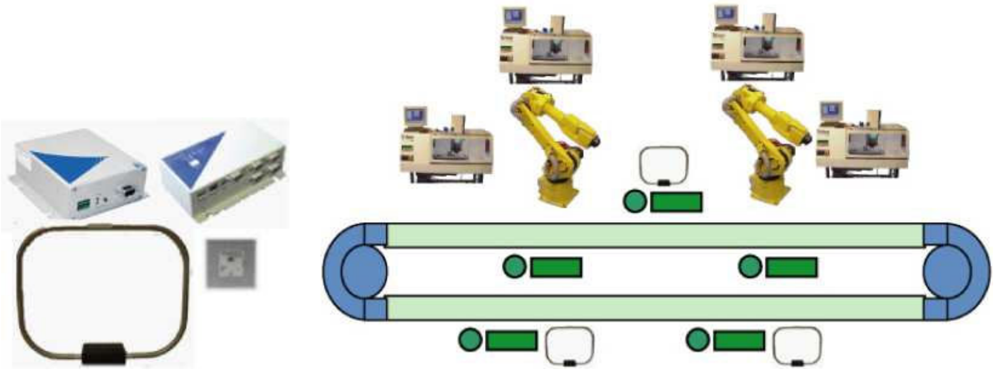
2.6 RFID

Radio Frequency Identification (RFID) is a technology where individual components (or parts in the case of manufacturing systems) are attached to high frequency read/write tags which allow the product to be tracked along a system. This way the history of every product can be traced and stored in a database, allowing for part, process and person tracking and web-based monitoring of



(a) (b)

Figure 5: a) The Ethernet network and the various field buses; b) SafetyBus p network



(a) (b)

Figure 6: a) An RFID antenna, a tag and hardware I/O devices; b) The layout of antenna's in the RFT

each product.

Before RFID was introduced into the RFT the part tracking process was hard coded into the SLC logic controller. Using RFID alleviates the logic controller from this memory function and allows for a much more robust and reconfigurable part tracking process using distributed sensors and databases.

The RFID implementation is divided into a hardware and a software component. The hardware consists of RFID tags which are placed on pallets and parts, of antenna's which are placed in strategic points along the RFT system for pallet and part detection and identification, and of the associated boards for the I/O as shown in Figure 6a.

The points where the antenna's are located will define where the product's tags can be read and written. In the RFT these antenna's were located at the start of the SP line where parts are placed on pallets, in between cell 1 and cell 2 (which forced a new pallet stop to be introduced to the conveyor system) and at the AGV pallet stop (which is where parts exit the SP line and are


(a) (b)

Figure 7: a) The real AGV; b) The AGV used in the virtual factory

taken by the AGV to the virtual factory for assembly). This configuration shown in Figure 6b was necessary as it was not possible to have an antenna at each cell stop location due to the maneuvers that the robot must undertake in picking and placing parts from pallets.

2.7 Materials Transport

The main material handler is the conveyor which carries pallets which in turn carry the parts to be processed. There is also an AGV as shown in Figure 7a which is used to transport parts from the fourth stop of the SP line to a virtual factory, where the parts can be virtually machined, inspected and assembled. The AGV is at the cell level and has its own proprietary cell controller, which communicates with the SLC through OPC (see Section 2.5).

As the AGV must go from the 'real' to the 'virtual' world, it has also been modeled in the virtual factory, as shown in Figure 7b. The real and virtual AGV's receive the same signals from the SLC – the virtual AGV is a synchronized duplicate of the real AGV.

2.8 System Operation

This section describes the system operation before the RFID or the Supply Cell of Section 3 were integrated. The product is a train as shown in Figure 8, which is assembled from two parts which are processed in the RFT.

The start or trigger to the system was when a part was placed on a pallet at the first pallet stop of the SP line. A button on the conveyor DeviceNet network was then pressed to release the pallet with the part to go to cell 1 (a different DeviceNet button was used to release empty pallets). At that point, the SLC would keep track of the parts and pallets traveling on the conveyor, such that when the pallet arrived at the cell 1 stop location, the SLC would immediately send an order to the cell 1 controller to process the part that had just arrived. Once the part was taken, the SLC would release the now empty pallet which would continue its path to cell 2. The processed part at cell 1 would wait for an empty pallet so that it could be taken to cell 2, where the same logical procedure would occur. When the finished parts finally reach the fourth pallet stop they would be taken in pairs (as a pair of parts are assembled to form a single product) on the AGV to the virtual inspection and assembly cell.

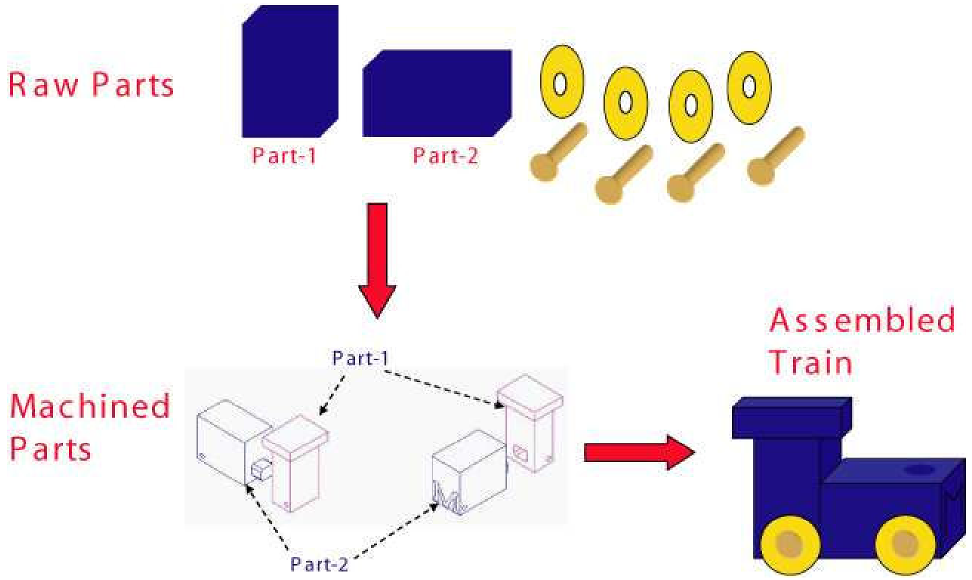


Figure 8: From raw parts to the RFT train

3 Combined System Architecture and Manufacturing Execution

The collaborative effort between the University of Michigan and Morgan State involved combining the existing UM RFT system with the existing MSU system. In a general sense, the UM RFT is a manufacturing line with the corresponding control systems, while the MSU system (described below) is a part storage and handling system. Therefore, the combined system uses the MSU system as a *supply cell* which feeds parts the UM RFT. This section describes the new architecture and controls involved in the combined system.

3.1 Demand-Pull Manufacturing

With the addition of a supply cell, the combined testbed could employ a more realistic and pedagogically interesting scheme for the management of parts in the system. A demand-pull architecture was developed where orders for particular assemblies are placed, and parts are only introduced into the system when they are part of a placed order. The purpose of a demand-pull architecture is to reduce inventories and warehousing costs of both of raw materials and of finished assemblies, and to provide products on a made-to-order basis. It is practical in a real manufacturing context when assembly time is short enough to satisfy customer orders in a reasonable time. The combined UM-MSU system has the following demand pull architecture shown in Figure 9:

- An ordering system accepts customer orders for assemblies and converts them to a list of ordered parts.
- The parts orders are sent sequentially to the supply cell.
- The MSU supply cell provides the parts which are transfered (virtually) to the UM serial-parallel line.

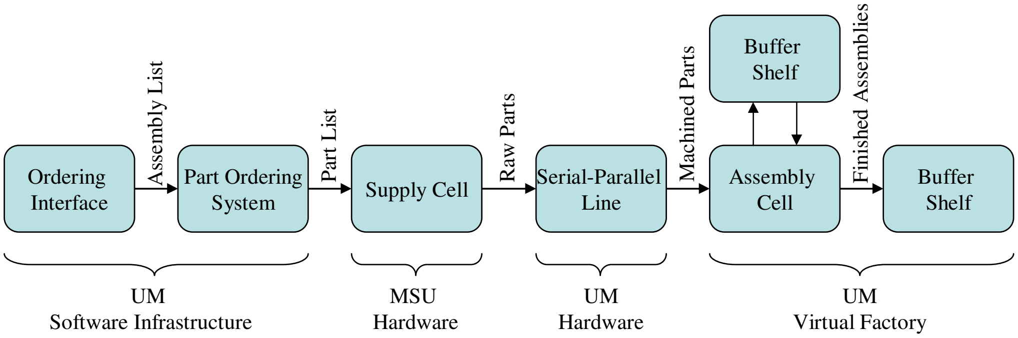


Figure 9: Demand pull architecture showing flow of information and parts through the system. The flow crosses boundaries between the University of Michigan and Morgan State University, and from the real to the virtual world.

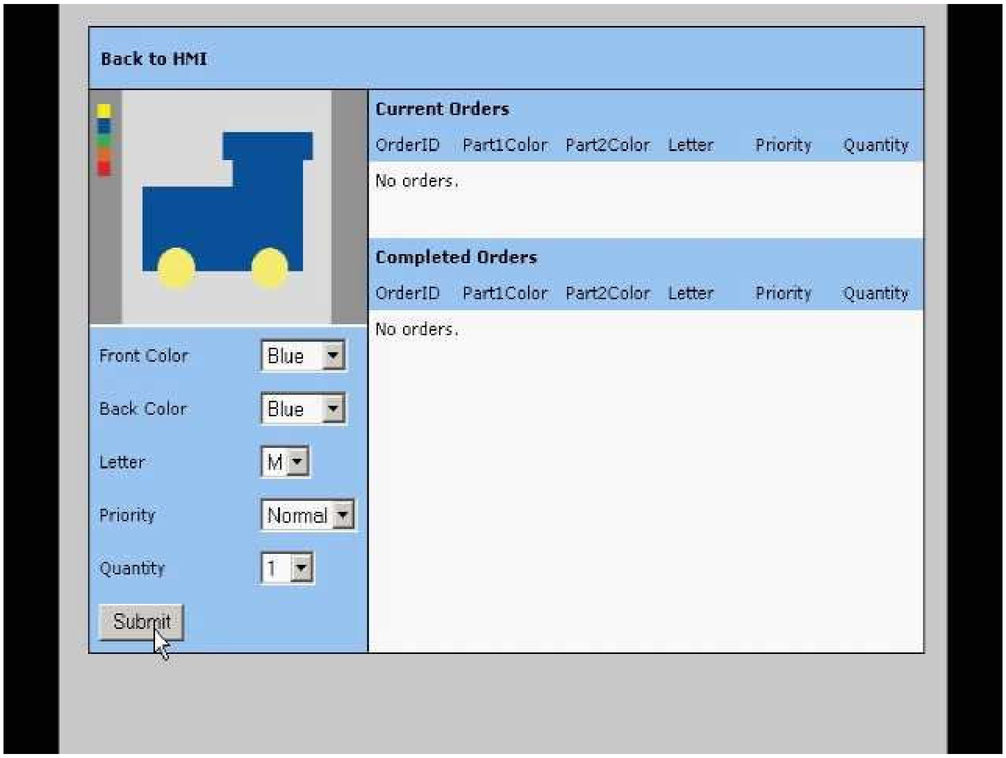
- After machining processes are performed on the serial-parallel line, the parts are transferred (virtually) to the updated assembly cell in the virtual factory portion of the UM system.
- Upon arrival to the assembly cell, the parts are placed on a buffer shelf.
- When two parts belonging to the same assembly are present on the buffer shelf, the parts are assembled.
- The completed assembly is placed in a virtual outbox, and the order marked as completed in the database.

During this process, parts throughout the system are tracked by a set of part tracking software modules which get information from the RFID system as well as the ordering and assembly systems to maintain information about which parts exist in the various portions of the system.

3.2 Assembly Ordering System

The ordering of parts is done on a web-based Human-Machine Interface (HMI) screen developed at UM. The ordering HMI allows the ordering of one or more train assemblies where the color of each part can be chosen from a list of available colors (blue and green). This HMI is shown in Figure 10.

Once an assembly order request is made, the HMI sends an event to the Software Infrastructure for processing. This event contains information about the requested parts. The event is passed to the CWM through the middleware server where it is sent to the part ordering software module. This module is responsible for converting the assembly order into a list of parts. The database stores a table of parts available in the supply cell's inventory, and the list of ordered parts is checked against the database table to see if the order can be fulfilled. If so, the corresponding parts are marked as assigned to an order in the inventory, the assembly order is placed in a database table of past and current orders, and the parts are placed on a part order queue as a database table, with the ASRS bay number listed on the table. The part ordering module then sends a status message back to the ordering HMI whether or not the order can be fulfilled, and it also alerts the supply cell tracking



Current Orders

OrderID	Part1Color	Part2Color	Letter	Priority	Quantity
No orders.					

Completed Orders

OrderID	Part1Color	Part2Color	Letter	Priority	Quantity
No orders.					

Front Color: Blue
Back Color: Blue
Letter: M
Priority: Normal
Quantity: 1

Figure 10: Assembly Ordering HMI allows part colors and carved letter to be selected.

module of the number of new pending parts on the order queue. This architecture is depicted in Figure 11.

3.3 Morgan State University Supply Cell

The testbed facility at Morgan State University is over 12 years old and included a conveyor system, an Automated Storage and Retrieval System (ASRS), and a robot. The robot was not used in the development of the combined system, but the ASRS and the conveyor together with a PLC controller provide a *Supply Cell*: a cell as part of the factory testbed responsible for sourcing parts into the manufacturing process. Thus, parts requested by the higher level logic of the system could be served automatically from the inventory of the supply cell and transfered (virtually) to the University of Michigan system. In reality, two identical sets of parts are necessary, one at MSU, and one at UM, such that at the time a part is removed from the MSU supply cell, the corresponding part is placed in the UM system.

3.3.1 Hardware

The ASRS comprises a 9x8 shelf-like rack with 72 slots each of which can hold a pallet. A 2-axis x-y servo platform carries a manipulator over the rack and also to a pallet stop on the conveyor system. The special-purpose binary-actuated manipulator is capable of grasping specially shaped pallets and pulling them out of the rack. A photograph of the Morgan State University supply cell is shown in Figure 12.

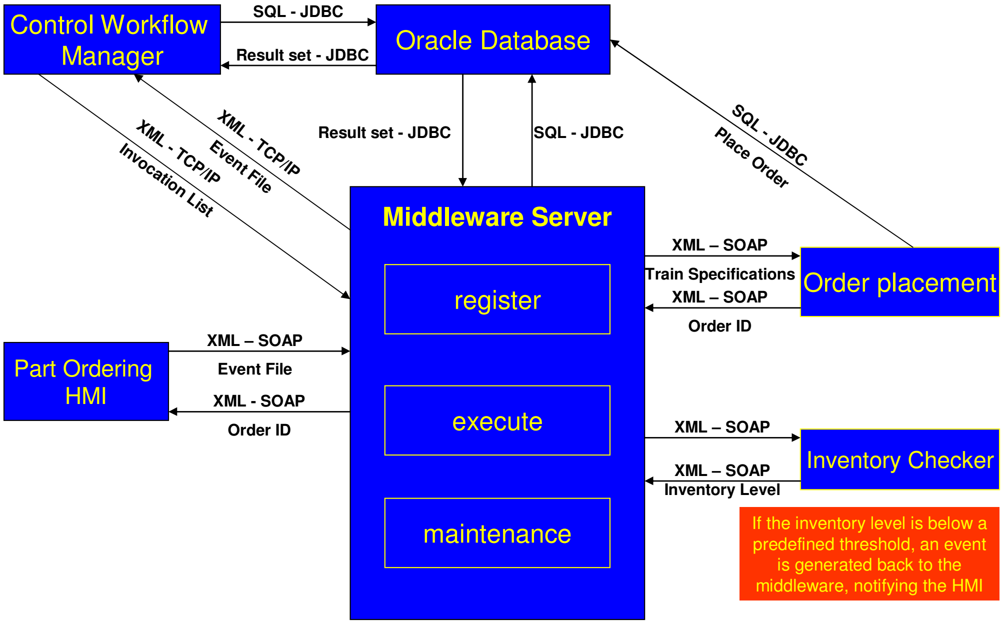


Figure 11: Software infrastructure architecture for handling part orders.

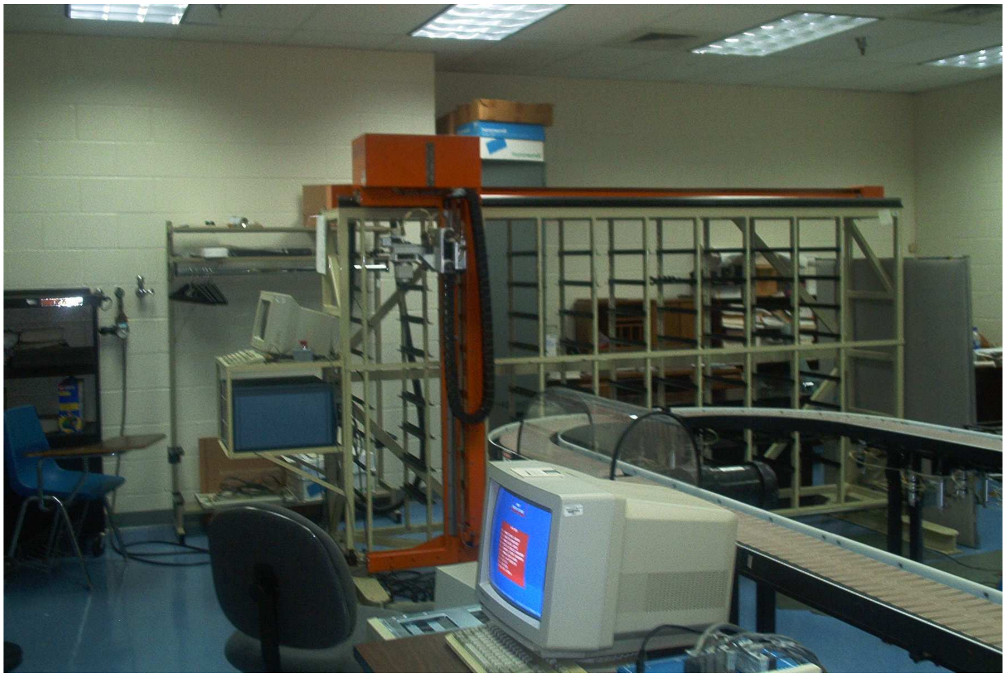


Figure 12: Photo of the Morgan State University Supply Cell. The ASRS serves parts on to the conveyor which get transferred (virtually) to the University of Michigan. The original PC driving the entire cell is in the foreground, and the PC interface to the servo controller of the ASRS is in the background.

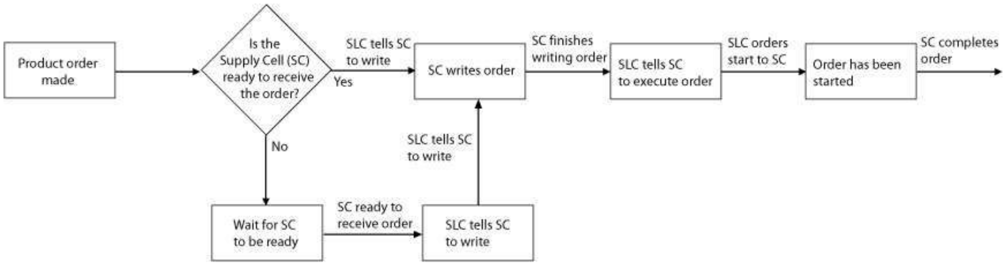


Figure 13: The communication sequence between the SLC and the SC for part retrieval from the ASRS.

The ASRS is driven by a servo box controlled directly by an 8086-based PC allowing manual control of the ASRS as well as providing configuration and calibration functions. The conveyor system PLC handled the logic of the pallet sensors and pallet stops on the conveyor. Each pallet stop has one switch sensor which detects the presence of the pallet a short distance before it reaches the pallet stop position. The pallet stop has two pneumatically actuated stops: one in front of the pallet to stop the pallets motion along the conveyor, which runs continuously, and one behind, which stops additional pallets, keeping the stopped pallet separated from incoming pallets.

As most of the existing equipment at Morgan State, particularly the control equipment was of an earlier generation, to enable the exchange of control signals between system, and to provide a more up-to-date experience for the students, the control hardware needed to be upgraded. This upgrade process is described in Section 4.3

3.3.2 Operation

When the supply cell is attached to the rest of the system, it acts between the ordering system and the serial parallel line. When the supply cell tracking software module (SCTM) receives the message from the part ordering module that a list of parts are on the order queue, it initiates the serving of parts, one at a time, from the ASRS. The SCTM retrieves the bay number of the next part from the order queue and then initiates a communication sequence between the system level controller (SLC) and the supply cell (SC) PLC controller, as laid out in Figure 13. In this sequence, the SLC waits until the ASRS is not busy at which point it tells the SCTM to write the ASRS bay number to the SC. The SCTM must write the bay number since the SLC, running under the MFSM engine, can only handle binary events, and not data. Once the bay number is written, the SLC tells the SC to execute the order. The SC then pulls the part off the ASRS, after which, it is ready to receive another order. After each part is placed on the conveyor, the SCTM is also alerted, and that part's information is placed on a tracking queue table in the database to keep track of the parts on the SC conveyor.

The second stage in the SC is the virtual transfer of parts from the SC to the serial-parallel line. Parts from the ASRS travel along the conveyor at MSU until they reach a pallet stop. When this event occurs, the SC initiates the transfer of the part according to the communication sequence laid out in Figure 14. In this sequence, the SC alerts the SLC that a part is waiting to be transferred, and sends a message to the SCTM. The SCTM looks up the next part in the SC conveyor queue and causes the part type (front or rear of the train) and color to be displayed on an operator HMI at UM,

Figure 14: The communication sequence between the SLC and the SC for part retrieval from the ASRS.

and a similar operator HMI is displayed at MSU. At the same time, the SLC checks the availability of a pallet at the serial-parallel line loading station (pallet stop 1) and when one becomes available, it lights a light at the loading station. At this point, an operator at MSU removes the part from the pallet and presses a button on their HMI. An operator at UM looks at the HMI and takes a part from a local stock identical to the part removed from the pallet at MSU according to the HMI and places it on the available pallet on the serial-parallel line. Thus, the part is *virtually* transferred from MSU to UM.

3.4 Part Tracking and Control

Each part introduced at UM each has a unique radio frequency identification (RFID) tag attached as does each pallet. These part and pallet tags are used by the system not only to record the progress of parts through the system, but also to provide information for making control decisions. The RFID readers are read continually and the read events, including which tags were read, are passed to the RFID event listener module (RFIDELM). Normally, this module ignores these events unless told to record them by another system component.

The tracking process is initiated when the operator places a part on a pallet at the loading station and presses a hardware button to indicate that the transfer is complete. The SLC then sends a message to the cell 1 tracking module (C1TM) that a part was placed. The C1TM tells the RFIDELM to start reading the RFID antenna at the loading station. For robustness reasons, the RFIDELM takes five consecutive readings to determine what tags are present at the loading station. From these five, if any two readings return a particular part tag number, then that tag is assumed to be present. This redundancy was required since multiple part types, each with different RFID tag placement and orientation are read by the same antenna. In a real manufacturing environment, a separate antenna would be used for each tag orientation, but our environment was cost-limited to a single antenna per station. Thus, when reading tags at non-ideal locations relative to the antenna, approximately 20% of the reads were lost. By requiring only two out of five readings to be successful, the chance of a failed read is reduced to well below 1%, providing sufficient robustness for our system.

When the RFIDELM gets a positive read on a part and pallet, it responds to the C1TM with the tag numbers which are added to the cell 1 queue database table and the SLC releases the pallet onto the conveyor. This first-in, first-out queue tracks which parts are on the conveyor until they are received (or rejected) at cell 1. When a pallet arrives at the cell 1 pallet stop, the SLC requests from the C1TM the nature of the part on the pallet: whether it is of type 1 (rear of train), type 2 (front of train), or empty (no part). If the part is of a type for which a corresponding machining resource is available in the cell, the SLC tells the cell to take the part (otherwise the part is rejected), the part and pallet are removed from the cell 1 queue, and the pallet is released. The part taken into the cell is added to the process history list in the database for that cell (information that is used further along for decision making purposes). Note that it was not possible to add an RFID antenna at the robot loading pallet stop at the cell since it would block the motions of the robot picking and placing the parts on the pallet, thus the need for the tracking module.

When the cell is done machining a part, it alerts the SLC. The SLC contains logic for releasing empty pallets onto the conveyor when they are needed. When an empty pallet arrives at the cell (as determined by the C1TM), the SLC commands the cell to place the part on the pallet, and the pallet is released. At this point, the part on the pallet is not tracked until it reaches the pallet stop and RFID station preceding cell 2. From the cell 2 tracking station, the same tracking and control process is performed. Note that "forgetting" the parts between cells allows additional reconfigurability since additional operations could be performed between cells without requiring any changes to the tracking and control logic.

3.5 AGV Loading and the Reconfigurable Virtual Machining Cell

After parts have passed through the serial-parallel line, they are ready to be transferred to the virtual factory. The parts arrive at the AGV loading station pallet stop, and once again, an RFID antenna reads the tags. At this point, a decision regarding the next destination of the parts is made based on the processing history of the part. In particular, if a part has been processed in both cell 1 and cell 2, it is ready to move on to the assembly cell. If not, it must first visit the virtual machining cell (cell 3) to receive any remaining machining operations. This virtual machining cell is assumed to be a reconfigurable machine tool (RMT) of the sort developed in the ERC[6] such that any machining operation can be performed. The idea is that by having a single reconfigurable machining cell, many part flow reconfigurations and error recovery situations can be handled without requiring more complex RMT's at all cells in the factory, allowing the system to easily adapt to reconfigurations and other events such as the shutting down of a cell for maintenance.

When the part tag is read, the AGV tracking module (AGVTM) looks up in the process history lists for cells 1 and 2 to determine if the current part needs further machining. After the part is read, a light lights up indicating to an operator that the part should be placed on the AGV. Once the operator does this and presses a hardware button, the pallet is released. Simultaneously, a virtual part is placed on the virtual AGV in the simulation. Once two parts are transferred (generally, but not necessarily part of the same order), the real AGV and the virtual AGV are told by the SLC to move. The destination of the AGV's depends on the AGVTM's decision: either to the assembly cell or to the virtual machining cell. The real AGV moves to a predetermined location in the lab, while the virtual AGV moves to the appropriate location in the virtual factory. After both AGV's arrive at their destination, the parts can be processed. At the virtual machining cell, one or both parts are machined as required and placed back on the AGV which is then sent to the assembly cell.

3.6 Virtual RFID

Once the parts are on their way to the assembly cell, the part information is again "forgotten". The assembly cell contains a virtual RFID station (VRFID) which operates and is used analogously

Figure 15: Updated assembly cell in the virtual factory. The robot (center) places parts on a shelf (right) before assembly. Finished assemblies are placed on the table to the left and accepted by the human.

to a real RFID. To implement VRFID, when parts are introduced to the virtual world, their tag numbers are stored as a property in the simulation. Note that this is different than tracking the tag numbers in a controller or in the database; the part numbers in this case are attached (virtually) physically to the parts, rather than as a data parameter in the control system. This mimics as closely as possible how information is located in the real world. When a part is read by the VRFID reader, the VRFID reader portion of the simulation reads the tag number from the part property, and returns it to the control system.

3.7 Assembly Cell

The final stage of the system is the virtual assembly cell (AC), shown in Figure 15. When the AGV arrives at the AC, the SLC directs the AC to pick up the parts one at a time. As each part is picked up, the AC robot picks it up and holds it in front of the AC's VRFID antenna, which reads the tag number. The robot then places the part on an available space on a buffer shelf and writes the tag number and its location on the shelf into a database table for the AC buffer. After each part is placed on the shelf, the tags are compared to the order table in the database, and if two parts exist on the shelf that correspond to the same order, they are assembled and placed on the outbox table. Once this is done, the order in the database order table is marked as completed.

4 Collaborative System Development

To realize the combined system as described above, a combined effort between students at the University of Michigan and students at Morgan State University was required. Most of this effort took place between January 2005 and May 2005, although some of the groundwork had been laid earlier. In summer 2004, a team of MSU undergrads came to Michigan to work with the team of

UM grad students for ten weeks. These students worked to learn skills required to implement a factory testbed including modeling a virtual factory, building HMI's, configuring device networks and PLC's, and working with factory hardware. The students came back to work for two weeks in January 2005 to discuss the development of the combined system. During this time, the two student teams worked together to develop the plan for the combined system, taking into account the current capabilities of the two systems, decide upon required upgrades, and to design and agree upon networking and communication protocols for the connection of the two systems. From this meeting came a development plan for both teams. This section describes some of the collaborative efforts particular to upgrading and combining the systems.

4.1 Control Connectivity: OPC via VPN over Internet

Essential to the combined operation of the systems was the network connection over the Internet between the control systems at each site. Most of the control signals on the network on the UM RFT were transmitted over Ethernet using the OPC protocol. There are two major fundamental problems using OPC over the Internet: first, since OPC uses the DCOM communications channels built into Microsoft Windows, there are serious security risks in leaving these channels open to Internet access, and second, MSU has a strict university-wide firewall preventing all but the most common communication ports from access.

Therefore, to allow free communication between sites, a Virtual Private Network (VPN) was established at UM. One PC at UM was set up as a firewall, using Microsoft Windows Server 2003, for the rest of the RFT computers such that all the computers and PLC's operating at UM were protected from the Internet and had their own local IP addresses. This gateway computer, installed with two network cards, one connected to the Internet, and the other connected to the local RFT network. The OpenVPN application, an open source VPN client-server solution, was used to build the VPN. OpenVPN uses secure SSL communications along a single port to allow computers outside the local network to pass encoded Ethernet messages of all types over a single communications port. Thus, computers outside the network transparently appear as if they are part of the local network when connected in. Since clients connecting in (using the OpenVPN application) initiate the connection, they can do so even if those computers are behind another firewall. In this way, control signals could be passed between computers at each site.

In addition to OPC communication, once the MSU computers were connected to UM's network, other network resources became available. For example, webcams at each site were used to ease communication and collaboration between sites. Also, Windows Remote Desktop Connections were used to aid MSU students in the setting up of hardware and software remotely, as students at UM could operate computers at MSU.

4.2 Communication Protocol

Even though the VPN connection was very robust, delivery times or even the ordering of control signals could not be guaranteed over the Internet. Therefore, particular care was taken into the development of handshaking protocols using OPC signals to ensure that the systems would remain synchronized, guaranteeing that one system would not move on to the next step in a sequence of actions until explicit acknowledgment was received from the other system that the previous

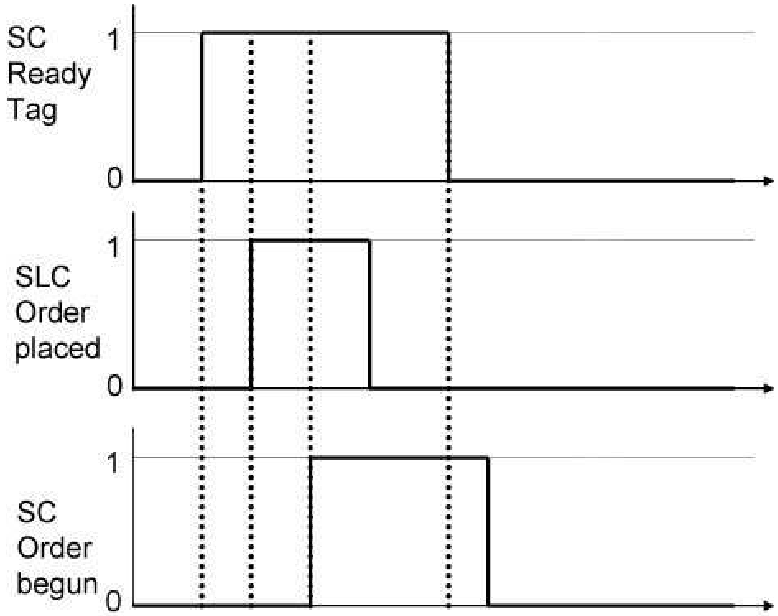


Figure 16: An example of the handshake of events between the SLC and the Supply Cell

communication was received.

Built around the sequences of communications events described in Figures 13 and 14, these protocols are enforced through a handshaking of events that occurs between the SLC and the Supply Cell. Figure 16 shows an example of how events are exchanged in between the SLC and the Supply Cell to guarantee the correct execution for the communication. In this case, there is a Supply Cell tag which indicates that the Supply Cell is ready to retrieve parts from the ASRS. The SLC will then place an order, which is seen by the Supply Cell as the raising of a tag – the Supply Cell will start the order and signals this to the SLC (which then lowers its previous tag). The Supply Cell will lower the ready tag to indicate ASRS unavailability. When the order is complete, the process can repeat itself. In this example, the extra layer of communication ensures both that a second order cannot be placed until the first is processed and that the first order cannot be mistaken for a second order if the falling edge of the order placed tag get delayed over the Internet.

4.3 MSU System Upgrades

As most of the control hardware at MSU was of an earlier generation technology, it was not designed to be operated by remote control signals over a network. In upgrading the system, an Allen-Bradley PLC was selected, similar to PLC's used at UM, to control the supply cell. OPC communication to and from the PLC was provided by RSLinx Gateway Edition software, which provides an OPC server mirroring data in the PLC (this is also the way that communication is performed with PLC's at UM). Thus, even though the PLC can not sit directly under the VPN, the OPC server on a regular PC at MSU can, and remote communication to the PLC is possible.

The new PLC replaced the function of the former PLC, controlling the pallet stops on the conveyor, as well as holding control logic to operate the ASRS. To connect the pallet stops, MSU students wired the pallet sensor switches and the pneumatic valve solenoids to an I/O relay block on a Devicenet network connected to the PLC. Control logic was written to drive the pallet stops in response to the sensor input and commands from the SLC system.

Connection to the ASRS was significantly more difficult. Previously control of the ASRS was only possibly through an 8086-based console computer, which connected to the ASRS control box using a proprietary, undocumented serial port connection, and only a limited set of commands was available along this connection. UM personnel along with students at MSU had to reverse engineer this connection so that the new PLC could send equivalent commands to the ASRS control box using a programmable serial port module. By operating the ASRS with the old console computer and listening in on the serial port with another computer, commands to retrieve parts from specified bay numbers and to drop parts on the conveyor were deduced. These commands were then duplicated in the new PLC such that parts could be requested by the SLC and served by the ASRS. By connecting to the ASRS at this level, direct control of the ASRS beyond these simple commands was not possible, and neither was it possible to execute high level functions built into the control box such as inventory management. In the combined architecture, however, these high-level functions were handled by the SLC and the software infrastructure, so were not necessary.

A complete control logic program was developed and debugged for the MSU PLC with the help of UM students. The communications were first tested at UM, and then tested and debugged remotely at MSU in conjunction with the operation of the whole system.

4.4 Demonstration

In May 2005, the combined system was demonstrated to the National Science Foundation's ERC advisory board as part of their annual site visit. This visit was conducted at MSU for the first time, such that the demo of the entire system, including both MSU and UM components had to be performed from MSU. Presenters at MSU gave the demo while students at UM were present to operate the system there and load the parts onto the serial-parallel line. The UM components were displayed remotely on a computer at MSU connected to the VPN. A webcam was displayed on this computer which was aimed and controlled by a student at UM. The web-based HMI's controlling and monitoring the system were displayed and operated on this computer. The virtual factory was displayed in a remote desktop window on this computer. In this way, the advisory board could view the entire system as the demo was performed.

The demo went smoothly, and the combined system operated successfully. Two assemblies were ordered and processed through the system, one going through the entire serial-parallel line and then to the assembly cell, and the other bypassing cell 1 which was manually shut down, and going to the virtual reconfigurable machining cell before going to assembly. All portions operated correctly from supply to assembly.

5 Conclusions and Educational Outcomes

The development of the University of Michigan / Morgan State University Distributed Reconfigurable Factory Testbed was the result of a successful collaboration between students at UM and students at MSU. This was a particularly interesting collaboration because due to the coupled design and testing of the control connections, many of the development tasks could not be strictly divided between the two sites. Thus, close cooperation was required between the student teams.

The development of the testbed presented a tremendous learning and growth experience for students both at UM and MSU, where the following is a sampling of the many educational outcomes of the collaboration:

- Students at MSU learned and developed the following set of skills and subjects
  - Working as part of a large team on a large scale research project.
  - Development of virtual factory simulations.
  - Design and development of Human-Machine Interfaces (HMI's).
  - Programming control logic in a PLC.
  - Difference between legacy and current control equipment, and how to interface them.
  - Configuring field networks and devices.
  - Wiring sensors and actuators to control devices.
  - Developing robust communication protocols.
  - Operation of factory systems.
  - Communication skills including report writing, presentation, and poster development.
  - Team organization and timelines.
- Students at UM learned and developed the following set of skills
  - Tutoring and training other students.
  - Leading sub-projects as part of a bigger project.
  - Design of a Manufacturing Execution System.
  - Remote collaboration skills.
  - Implementation of remote networks and secure VPN's.

There were also some significant technological and methodological developments and milestones as part of this experience, including the development of a Virtual RFID system, reconfigurable part tracking and control, and the operation of OPC over VPN via the Internet. It also provided the opportunity to further test and develop research methodologies developed at UM including Modular Finite State Machine controllers using Event Condition Action Rules, and the Open Architecture Data-Centric Software Control System.

The final, and most essential outcome of this collaborative effort was to prove that an educational and research platform could be successfully developed and operated between universities separated by a significant distance. It is hoped that this project serve as an example for collaborative education and research between universities.

References

1. J. Moyne, J. Korsakas, and D. Tilbury. Reconfigurable factory testbed (rft): A distributed testbed for reconfigurable manufacturing systems. In *2004 Japan USA Symposium on Flexible Automation*, July 2004.

2. E.W. Endsley, M.R. Lucas, and D.M. Tilbury. A logic control programming framework using modular finite state machines. In *American Control Conference*, 2002.

3. E. Almeida, J. Luntz, and D. Tilbury. Modular finite state machines implemented as event-condition-action systems. In *16th IFAC World Congress*, July 2005.

4. K. Sukerkar, H. Wijaya, J. Moyne, and D. Tilbury. An integrated distributed software system for reconfigurable manufacturing. In *Advanced Manufacturing Technologies 2004*, June 2004.

5. J. Moyne, D. Tilbury, and H. Wijaya. An event-driven resource-based approach to high-level reconfigurable logic control and its application to a reconfigurable factory testbed. In *3rd CIRP Reconfigurable Manufacturing Conference*, May 2005.

6. Y. Koren and S. Kota. Reconfigurable machine tool. u.s. patent #5,943,750, August 1999.