Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

A SIMULATION-BASED PRODUCTION TESTBED

Albert Jones
Michael Iuliano

National Institute of Standards and Technology
Building 220 Room A127
Gaithersburg MD, 20899, U.S.A.

ABSTRACT

Researchers at the National Institute of Standards and Technology have been developing a simulation-based production testbed. This testbed contains continuous simulation models of production equipment and processes, and discrete-event simulation models of various production systems. In addition, it contains both commercial and prototype software applications that implement a number of production functions from order release to final inspection. This testbed provides an environment for the development and testing of interfaces which can provide an integrating infrastructure for these applications. These interfaces will be based on information models and exchange protocols, and will specify what information is shared across those applications and how it is exchanged. This paper describes the current state of the testbed, and the projects currently being conducted in the testbed.

1 INTRODUCTION

Since the mid 1980s, the National Institute of Standards and Technology (NIST) has been involved in research and development related to manufacturing systems integration. The Automated Manufacturing Research Facility (AMRF) was designed and built as a testbed production facility for a wide range of systems integration projects (Simpson, Hocken, and Albus 1982). Although it was highly innovative and immensely successful, the AMRF was subject to the same kinds of equipment problems faced by factories across the country. These problems increased the time and cost of integration testing.

NIST is collaborating with vendors, users, and university researchers to develop a simulation-based production testbed. This testbed will contain 1) both continuous and discrete-event simulation models of equipment, processes, and systems, 2) software applications to implement production functions *on top of* those models, and 3) a manufacturing data repository and network to provide an integrating infrastructure for both the models and applications

Construction of this testbed began about two years ago. In (Iuliano and Jones 1996), we described the initial configuration of the testbed and its long-range goals. In this paper, the authors describe the changes we have made and the current projects being conducted in the testbed. This paper is organized as follows. Section 2 provides a short description of the current testbed including software applications and hardware platforms. Sections 3, 4, 5 describe three current projects being conducted in the testbed. Finally, Section 6 briefly describes the project we hope to initiate next year.

2 TESTBED SOFTWARE AND HARDWARE

The original configuration of the testbed is given in (Iuliano and Jones 1996). It contained the following commercial applications: product data management, MATRIX®; operations planning, ICEM® PART; shop floor simulation, QUEST®; NC program simulation, VNC®; and shop floor scheduling, AUTOSCHED®. This year, we have added the following packages: ergonomic simulator, ERGO®; NC program simulation, NSEE ®; scheduling, FACTOR® ; shop floor data collection, JOBPACK® and INTRACK®; databases, Microsoft ACCESS® and ORACLE®; and, kinematics simulator, ADAMS®.

New prototype applications developed at NIST include a dispatcher and a status message handler. order entry, job routing, shop floor control, and machine control. Currently, there are three major projects are being conducted in this testbed: Manufacturing Engineering Tool Kit, Virtual Machine Tool, and Integration of Scheduling and Shop Floor Data Collection. The ensuing sections provide descriptions of those projects.

3 MANUFACTURING ENGINEERING TOOL KIT (METK) PROJECT

The main goal of the Manufacturing Engineering Tool Kit (METK) (McLean 1993) project is to identify generic interfaces, which can be used to demonstrate the integration of manufacturing engineering software applications. The initial focus of the project is the integration of operations planning and Numerical Control (NC) programs. This integration will be achieved using a generic specification for process plans. That specification will be based on the concept of a work element. A work element describes an activity to be performed and the information needed to perform it. A collection of related work elements constitutes the major part of this process plan. The generation and verification of these plans are done using both commercial and prototype software applications.

A system diagram for the METK project is shown in Figure 1. The software application which implements the product data management (PDM) function is called MATRIX®. MATRIX ® encapsulates a source file configuration management application and a workflow management application. MATRIX ® manages an object-oriented database of distributed files, the applications that create those files, and the process that governs their life cycle. This process is defined in a workflow management scheme. The CAD application, PRO/ENGINEER®, is used to create a product design file. This design file includes a solid model representation of the final geometry of the product, and a part blank, which represents the initial geometry. This design file is retrieved by the operations planing application, ICEM PART®. ICEM PART ® uses a computer automated feature recognition algorithm to detect machineable features from the solid model. This application then allows the user to specify a machine, the machine's tool set, and the required jigs/fixtures. The feature definitions, together with user-defined specifications are then used to create the necessary NC programs. These NC programs form the basis for the process plan file, which is used to drive the NC verification application, VNC®. VNC ® uses simulation models of the machine tool, fixtures, cutters, and the machine controller together with a part blank to generate and visualize the output of the NC program.

Currently, all of these METK applications reside and execute on a single Silicon Graphics ONYX workstation running the IRIX5.3® operating system. This workstation is in the Advanced Manufacturing Systems and Networking facility (AMSANT) facility at NIST. AMSANT was established to support testing of high performance computing and networking hardware for next generation manufacturing systems. All workstations within AMSANT are connected to the Internet and capable of file transfer from collaborator sites external to NIST. The METK project intends to make use of this capability to demonstrate plug compatibility with similar software applications at project collaborator sites.

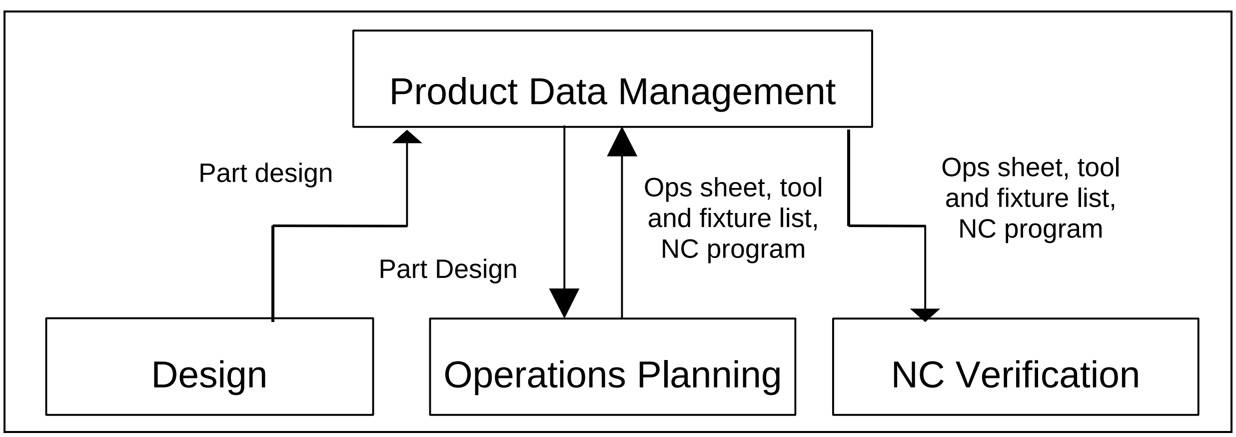

Figure 1. Systems Architecture for METK Project

The integration of ICEM PART® and VNC® will be coordinated through MATRIX®, using a process plan (McLean 1987). This process plan must specify all of the resources, potential alternatives, and all of procedures needed to machine a part. Furthermore, it must specify precedence relations that may exist among some of those procedures. An example of the kind of plan executed in the METK is given in Fig 2.

Because the METK project intends to simulate the execution of the plan, all resources will be files. They include a tooling file, a fixture file, an NC file, a part blank file, machine model file, and a machine controller model file. In the current demonstration, the tooling file and the NC file are created in ICEM PART®; the remaining files are created in VNC®. This plan is stored as a file in PART 21 format (ISO /IS 10303-21 1994).

The actual creation of this file is carried put using any text editor. At execution time, this file is parsed and a series of commands are sent to VNC® which simulates the execution of the NC program on the specified machine tool and reports back any errors (See Figure 3).

The METK project is also developing a validation methodology, which can be used to test the validity of engineering data before it is sent to the floor. To implement this methodology, we are extending the notion of a process plan to include those highly manual operations that take place at the tool crib and the material preparation room (Figures 4 and 6). To simulate those operations, we are using an ergonomic simulator, ERGO®.

We are also using ERGO® to simulate various manual operations at the machine including part and tool handling (Figures 5 and 7). Using the ergonomic capabilities of a simulator such as ERGO® will provide a means for both validating data and analyzing the movements of the human operators in the system. It will now be possible to evaluate operator movements to detect such things as fatigue, interference, and over extension.

Figure 4 shows an operator in the tool crib setting tools into their adapters. When completed, he will place the tool into the carrier shown at the bottom of the Figure. This carrier will be transported to the machine, either by another operator or by a transport device such as a forklift or and AGV. The machine attendant will remove the tools from the carrier and place them (see Figure 5) into the slot specified in the process plan (see Figure 2).

Figure 6 shows the operator placing a piece of bar stock onto a table. This will be taken to the machine and loaded into the fixture using the instructions in the process plan. Following the simulation of the NC program, the finished part is removed from the fixture (see Figure 7). It is then taken to the next machine.

```
HEADER Section
plan_id = P12345
part_name = Air_frame_test_part
Creation_date = 10/24/96
Planner = Mike Iuliano

RESOURCE Section
machine_id = CINC_MILA_T30
tool_name = 1/4" TWIST_DRILL
tool_name = 1/2" CENTER_DRILL
tool_name = 1/8" BALL_NOSE_END_MILL
tool_name = SHANK_END_MILL
fixture_name = vise
workpiece_name = Air_frame_blank
nc_program = Air_frame.cnc

PROCEDURE Section
Step 1   LOAD_MACHINE
      machine_id = CINC_MILA_T30
      Machine_controller = GE2000
      end_step
Step 2   LOAD_TOOL
      tool-name = TWIST_DRILL
      tool_id = T266
      magazine_slot = 1
            end_step
Step 3   LOAD_TOOL
      tool-name = CENTER_DRILL
      tool_id = T271
      slot = 2
      end_step
Step 4   LOAD_TOOL
      Tool_name =BALL_NOSE_END_MILL
      tool_id = T268
      magazine_slot = 3
      end_step
Step 5  LOAD_TOOL
      tool-name =SHANK_END_MILL
      tool_id = T234
      magazine_slot = 4
      end_step
Step 6   LOAD_FIXTURE
      fixture_name = vise
      fixture_id =  V178
      ref_frame = x_axis
      x,y,z_offset = 152.4, 101.6, 44.45
      units = inches
      end_step
Step 7   LOAD_WORKPIECE
      workpiece_name = Air_frame_blank
      workpiece_id = W123
      ref_frame = fixture_name
      x, y,z_offset =  0, 0, 0
      units = inches
      end_step
Step 8  LOAD_NC_PROGRAM
      nc_program = Air_frame.cnc
      end_step
Step 9  RUN_NC_PROGRAM
      nc_program = Air_frame.cnc
       end_step
```

Figure 2. Example of a Simplified Process Plan

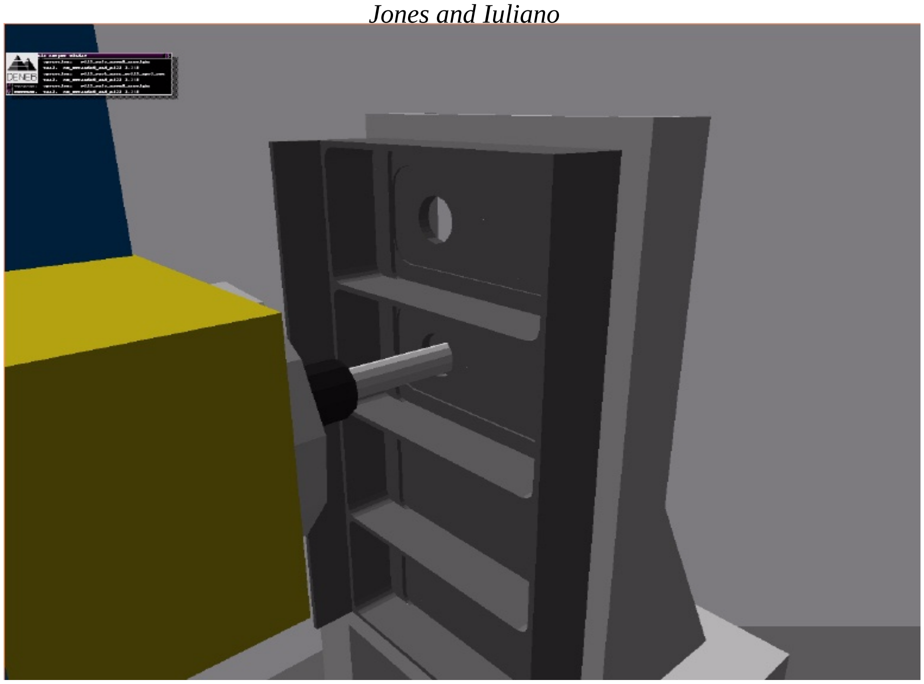

Figure 3. VNC® Simulation of NC Program Execution

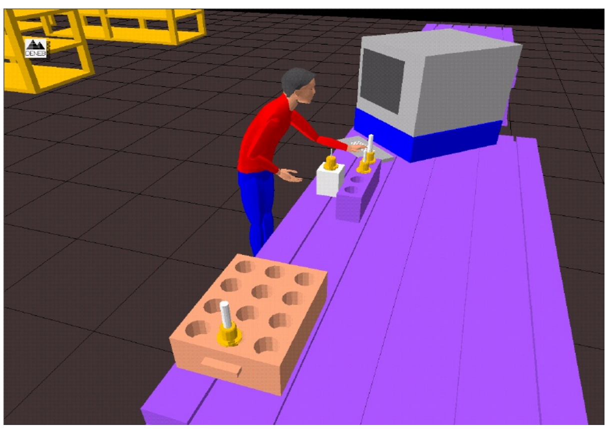

Figure 4. Tool Setting Operation

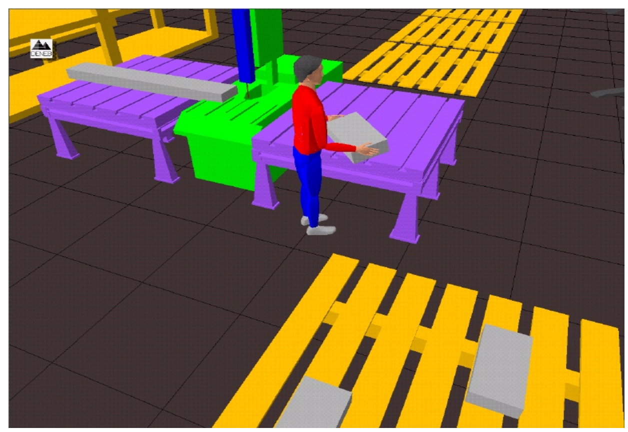

Figure 6. Material Preparation

Figure 5. Operator Loads Tools

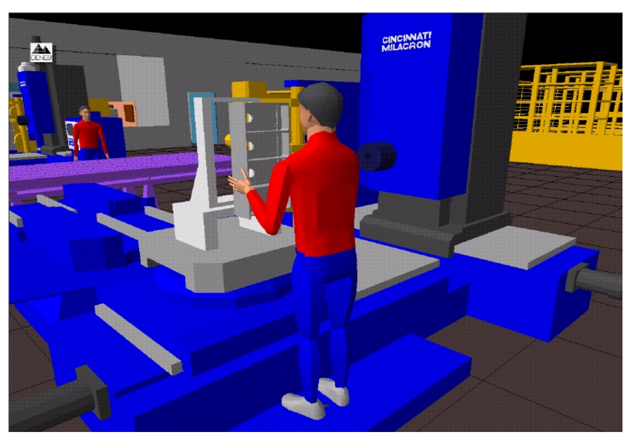

Figure 7. Operator Mounts Finished Part

4 VIRTUAL MACHINE TOOL PROJECT

It is assumed in the METK project, that the simulation models accurately reproduce the structural and behavioral characteristics of the machine tools they represent. If this were true, then manufacturers could use the interfaces defined the METK project to test and optimize the contents of process plans before any machining resources are committed for actual production. In addition, they could, given the entire profile of production costs, determine which of a number of potential machines should be used for the production of a particular part. This would reduce the amount of scrap and the amount of time that machines are used for prototyping and first article production.

The fact is, however, that the simulation tools on the market today model only "perfect" machines. These tools allow users to define machine geometry and ideal motions of structural components (slides and spindles). Based on these definitions, users can visualize a machine as a solid model and emulate the motion of the slides during machining. There is no provision, however, to incorporate a machine's actual behavior and to predict the impact of that behavior on the finished part. Real machine tools exhibit many unintended error motions. Machine slides have error motions in six degrees of freedom: linear displacement, horizontal straightness, vertical straightness, yaw, pitch and roll. In addition to these slide errors, there are spindle errors, parallelism errors, orthogonality errors, and thermally induced errors. The relationships among these errors can be quite complicated. The magnitude of these errors will determine how closely the final machined part is to the desired part.

To address these problems, NIST has initiated the virtual machine tool project. The project will 1) develop information models which capture machine tool error characteristics, 2) build high fidelity machine tool simulations which incorporate those information models, and 3) show that these simulations accurately predict part geometry. The results are based on procedures defined in national and international standards such as ANSI/ASME B5.54 and the ISO 230 series (ANSI/ASME B5.54 1992, and Donmez et al 1987). However, there are no provisions in these standards to (a) store the information obtained from the series of prescribed measurements in any type of electronic format so that the results may be shared, and (b) predict the output of a given machine in terms of the accuracy of any desired machined part. Using the information models defined in this project, we can instantiate PART 21 [4] files to address (a). As for (b), preliminary simulation experiments have been conducted on a turning machine with parallelism and orthogonality errors. Statistical analysis of the simulated output compares favorably with the output from the real machine.

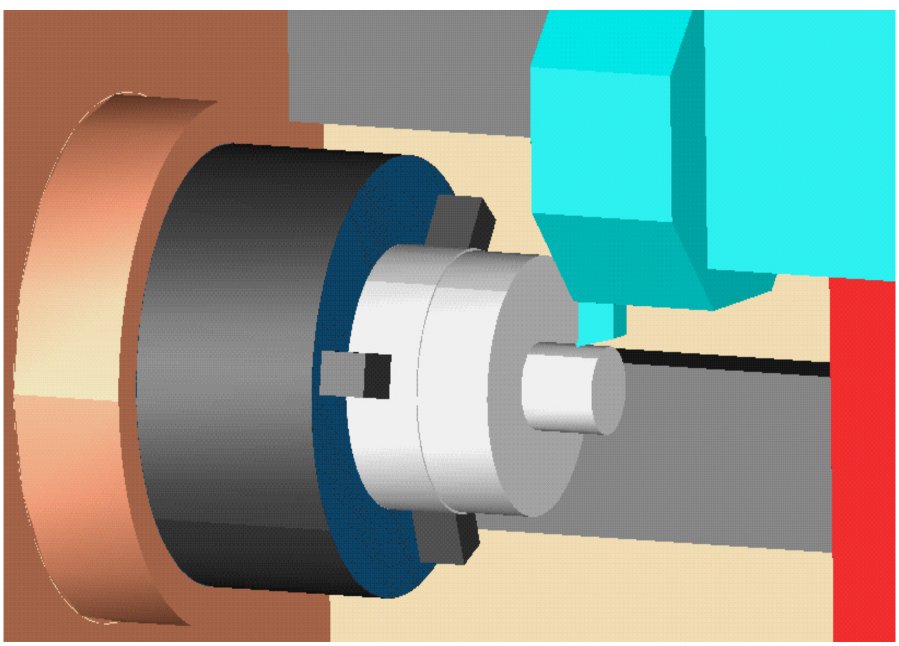

Figure 8. Combined PRO/ENGINEER®/ ADAMS®

Two commercial software applications were used in these experiments: PRO/ENGINEER ® and ADAMS®. The solid model of the turning center was developed in the PRO/ENGINEER ®. A corresponding kinematics model of the same machine was developed in ADAMS®. The two models were integrated via an interface provided in ADAMS ® to define motion characteristics of each element of the solid model. model of machine tool cutting a part. However, in order to simulate the actual machine behavior, a separate error generation module was developed and added to the original ADAMS ® model. This module incorporates the mathematical representations of the parallelism and orthogonality errors into the kinematics model of the machine. This augmented kinematics model allows the user to calculate the resultant error of the tool position with respect to the part being cut (see Figure 8).

Using such a system, a tool path can be generated by specifying position of each axis as a function of time. During the simulation, the tool point moves along the part profile in the machine workspace. At every calculation point, the dynamic machine model adds the nominal position of the tool point plus the calculated error to determine the actual position in machine space. When the graphical simulation is completed the individual nominal tool positions and corresponding error values are stored in a file which can be exported to PRO/ENGINEER®. This data, together with information about the nominal part geometry can be used to compute differences between the virtual part and the desired part. Eventually, we hope to develop a virtual Coordinated Measuring Machine (CMM) which can inspect our virtual part.

5 INTEGRATION OF SCHEDULING AND SHOP FLOOR DATA COLLECTION PROJECT

During the last 10 years, many manufacturing companies have invested heavily in computer hardware and software which make it possible to collect data about the events on the shop floor literally as they are happening. This real-time data collection makes it possible to "close the loop" for operations management just as concurrent engineering is "closing the loop" for product design. Integration of the computer software, which collects this data with the computer software, which uses that data, continues to be major problem. While there is a general understanding about which tools need that information, there is no agreement on the content or format of that information. A project is underway to develop generic interface specifications for and demonstrate the integration of scheduling and shop floor data collection tools. This project is a collaborative effort between vendors, users, academia, and the NIST. It has three major benefits. First, the interface specifications will reduce the cost of integration for both manufacturers and vendors. Second, the integrated tool set will give academic researchers an opportunity to test their scheduling algorithms on different simulated shops. Third, it has the potential to improve shop performance and throughput.

The initial focus of the project was the development of an information model in EXPRESS (ISO 10303-11 1994) to specify formally the concept of shop floor status and a series of message that could be used to update that status. The model is comprised of four major entities - Buffer, Load, Material, and Resource (Jones, Riddick, and Rabelo 1996):

- Buffer: defines the state of an area used to hold loads. This entity provides the list of loads contained in the buffer, and could provide the resource, which is associated with. In our approach we do not consider the queue that sometimes is used in simulations software as a specific type of buffer. Furthermore, specifying one resource as a resource associated with a buffer, handles this situation.
- Load: defines the state of the collection of products. This entity provides information on the changes in the amount of parts in the load. It also provides time characteristics on the process, the states of the load and the resources that are currently associated with it.
- Material: defines the consumable item that is used in manufacturing processes.
- Resource defines the state for things that will be used to manufacture products; that may be operators, machines, tools or fixtures. This entity provides the description of the resource, maintenance time characteristics and information about the load currently associated with this resource.

These entities have been used in the development of three types of messages: create, delete, and change [8]:

- Create messages allow the system to create new elements (load, buffer, material, resource)
- Delete messages allow the system to delete elements when they become obsolete.
- Change messages allow the system to modify any dynamic characteristics of an element.

The change messages are constructed using keyword/value pairs. This provides:

- a variant number of attributes to be updated in the same message,
- the type of the value in the pair must correspond to the type of the attribute of the corresponding element of the status database,

- no two pairs in the same change message are allowed to have the same keyword.

The information model and change messages form the basis for one of the integrated tool sets in the virtual production facility. That tool set has four major functional components (see Figure 9): scheduler, dispatcher, shop floor simulator and shop floor data collector. The scheduler generates a schedule and passes it to the dispatcher as a file. The dispatcher interprets the schedule and generates lists of jobs with associated start and finish times for each resource within the simulated shop.

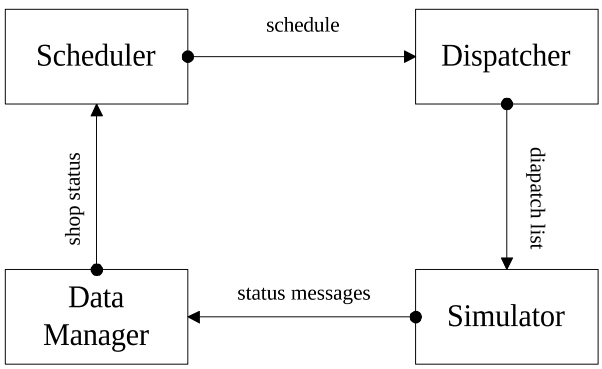

Figure 9. Systems Architecture for Scheduling Project

The shop floor simulator (see Figure 10) has been modified to accept orders at specified times, to execute jobs according to the sequences determined by the dispatcher, and to follow the routings for each job. As the simulation evolves over time, a series of messages are collected in files and sent to the shop floor data collector. The shop floor data collection system interprets these messages and keeps an updated view of the shop floor status. The schedulers can continually update their own internal model of the shop based on this status. They can also perform a "rescheduling" operation whenever major disruptions to the current schedule occur.

Currently, as noted above, the facility contains two commercial schedulers, which output a common file format for the schedule. This file is passed to the dispatcher that has been developed jointly by NIST and Ohio University. The data collector has been implemented in two ways. The first consists of a message handler and an ACCESS® database. In this integration scenario, shop status database is sent to AUTOSCHED®, which updates its internal shop model (currently implemented as a collection of files). The second consists of a message handler and an INGRES® database developed on top of JOBPACK®. In this integration scenario, shop status is sent to FACTOR®, which updates its internal shop model (currently implemented in INGRES®). In the near future, we will incorporate other schedulers, simulators, and shop floor data collectors into this integrated tool set.

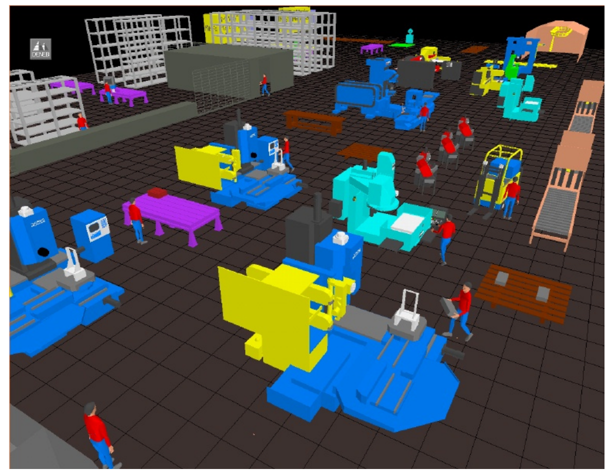

Figure 10. Shop Floor Simulation

6 SUMMARY

The simulation-based production facility will continue to evolve over the next several years. The next major project will involve the development of a virtual supply chain. The focus will be on production and operations management policies within the supply chain. The chain will include suppliers that operate on both push and pull strategies. We intend to build and simulate business models, information flow models, and material flow models, which describe the operational aspects of a supply chain.

ACKNOWLEDGEMENTS

Work described in this paper was sponsored by the US Navy Manufacturing Technology program and the NIST Systems Integration of Manufacturing Applications program. Certain commercial software and hardware products are identified in this paper. This does not imply approval or endorsement by NIST, nor does it imply that the identified products are necessarily the best available for the purpose.

REFERENCES

Simpson, J., Hocken, R., and Albus, J., The Automated Manufacturing Research Facility of the National

Bureau of Standards, *Journal of Manufacturing Systems*, Vol. 1, No. 1, 17-32, 1982.

McLean, C., “Computer-Aided Manufacturing Systems Engineering”, *IFIP Transactions: Advances in Production Management Systems*, North Holland, Amsterdam, Netherlands, 341-348, 1993.

McLean, C., Interface Concepts for Plug-Compatible Production Management Systems, *IFIP WG 5.7: Information Flow in Automated Manufacturing Systems*, North Holland, 307-318, 1987.

ISO/IS 10303-21, “Industrial automation systems and integration - Product Data Representation and Exchange - Part 21: Clear text encoding of exchange structure,” ISO, 1 rue de Varambe, Case Postale 56, CH-1211 Geneva, Switzerland, 1994.

ANSI/ASME B5.54, *Methods for Performance Evaluation of Computer Numerically Controlled Machining Centers*, ASME, New York, 1992.

Donmez A. et al., “A Generalized Mathematical Model for Machine Tool Errors,” *Modeling Sensing and Control of Manufacturing Processes*, PED- vol. 23/DSC-vol. 4, Book No. H00370, K. Srinivasan, D.L.E. Hardt, and R. Komanduri, eds., ASME, New York, 1987.

ISO 10303-11:1994, "“Industrial automation systems and integration - Product Data Representation and Exchange - Part 11: Express Language Reference Manual," ISO, 1 rue de Varambe, Case Postale 56, CH-1211 Geneva, Switzerland, 1994.

Jones, A., Riddick, F., and Rabelo, L., Development of a Predictive/Reactive Scheduler using Genetic Algorithms and Simulation-based Scheduling Software, *Proceedings of AMPST’96*, 589-598, 1996.

M. Iuliano and A. Jones, “Controlling Activities in a Virtual Manufacturing Cell”, *Proceedings of the 1996 Winter Simulation Conference*, 1062-1067, Coronado, CA, Dec. 8-11, 1996

DR. ALBERT JONES is currently heading up projects at the National Institute of Standards and Technology (NIST) to investigate the functional and integration requirements for the next generation simulation tools. Prior to this assignment, Dr. Jones spent several years as Deputy Director of the Automated Manufacturing Research Facility at NIST. During that time, Dr. Jones worked with various NIST and academic researchers on system architectures for shop floor control, cell control, and distributed scheduling. He received his MS in Mathematics and Ph.D. in Operations Research from Purdue University. Dr. Jones is currently on the Executive Boards for the Winter Simulation Conference and the Engineering School at Loyola of Baltimore. He is Manufacturing Editor for several leading journals. He has Chaired or Co-chaired several international conferences, and has served on several proposal evaluation panels for NSF, NIST, and ARPA.

MICHAEL IULIANO is a Computer Scientist in the Manufacturing Engineering Laboratory at the National Institute of Standards and Technology (NIST). Previously, he was a Software Engineer at Vitro Corporation, which is an U.S. Naval Defense contractor. He received a BS in Electrical Engineering from Drexel University in 1987. Over the past several years he has participated in research projects that include manufacturing engineering, scheduling, shop floor data collection, plant layout, assembly planning, production data management, computer aided design, operations planning, characterization of machine tool errors, simulation, and virtual reality.