A Simulation Architecture for Manufacturing Interoperability Testing

Charles McLean, Sanjay Jain, Frank Riddick, Y.Tina Lee
National Institute of Standards and Technology
Gaithersburg, MD 20899
charles.mclean@nist.gov

Keywords: virtual manufacturing, simulation, software architectures, information models, interface standards, interoperability testing

Abstract:
Manufacturing systems are often costly to develop and operate. Simulation technology has been demonstrated to be an effective tool for improving the efficiency of manufacturing system design, operation, and maintenance. But manufacturing simulations are usually developed to address a narrow set of industrial issues, e.g., the purchase of new equipment or the modification of a manufacturing process. Once the analysis is complete a particular simulation model may not be used again. If simulations could be made more modular and easily integrated, they could have tremendous value as tools for manufacturing interoperability testing. This paper presents a modular reference architecture to facilitate the integration of manufacturing simulation systems with other support and testing applications. Opportunities for testing are also discussed that will be enabled by the implementation of the architecture.

1. Introduction

Manufacturing systems tend to be large, complex, and expensive to construct and operate. Due to hardware-acquisition, maintenance, and space costs, academic and research institutions cannot afford to duplicate real manufacturing systems in their laboratories. Student and researcher hands-on experiences with manufacturing systems are often limited to individual or small groups of machine tools in laboratory shops, prototype work cells, or tabletop manufacturing systems. Manufacturing research and testing could be significantly enhanced if manufacturing systems could somehow be brought into the laboratories of academic and research institutions. Computer simulation technology now allows us to construct large, realistic virtual worlds in software. The military and the entertainment industry have made extensive use of this technology for a number of years. The industrial world is just beginning to recognize the potential of this technology. Virtual manufacturing enterprises could be used by a variety of organizations involved in manufacturing for research, testing, and training. This paper focuses on how this may be applied to manufacturing interoperability testing.

The paper presents an architecture for manufacturing testing based upon simulation systems as well as actual commercial manufacturing software and hardware systems. The architecture is being implemented at the U.S. National Institute of Standards and Technology (NIST) as part of the Virtual Manufacturing Enterprise Project within the Manufacturing Interoperability Program. The systems under development at NIST will be used primarily as research tools for testing and evaluating interface specifications and standards.

2. Simulation-based Testing

Simulation technology enables the construction of technically correct, dynamic models of organizations, systems, and processes. The models, once validated, can be used for supporting decisions for design and operation of the systems to achieve desired performance. Testing applications include the use of simulation primarily by operations personnel or operations support engineers to test new methods, processes, and equipment before integrating them into operations. Testing applications are also applicable at design and prototype stages. Integrated simulation technology can be used to support the following testing applications in the manufacturing domain.

- Perform interoperability testing with models of systems being integrated. For example a model of a robot controller may be integrated with a model of the robot for testing purposes to ensure interoperability.
- Perform interoperability testing with emulated physical equipment. For example, a physical programmable logic controller may be tested with an emulated conveyor system before the physical conveyor system is installed or even delivered.
- Evaluate the capability of the delivered process, system or design to meet interface specifications.
- Perform conformance and acceptance tests using simulations to create the specified range of inputs for a delivered system or process.
- Evaluate whether new systems, processes or designs meet performance specifications, for example, test program for robots and other machinery using simulations.
- Develop metrics to allow the comparison of predicted performance against "best in class" benchmarks to support continuous improvement of manufacturing operations.

Models will need to be carefully validated, however, the procedures used may be more focused on functional and deterministic validation rather than statistical validation used for system-level research applications that use stochastic factors. The validation procedures should be defined to ensure common practices. Supporting applications that exercise the models through the range of parameters defined in the specifications should be provided to facilitate the process.

Associated development of test cases and procedures would help by allowing a common scale on which alternate artifacts can be tested. Vendors of artifacts can use the results from standard test cases to highlight their products. Customers can use the results from the standard test cases for initial screening of vendors and then proceed with testing using company specific data. The test bed with associated test procedures and test cases will benefit both researchers and industrial personnel due to large reduction in effort for testing of new artifacts. Researchers and developers from manufacturing and vendor organizations would gain by unbiased testing of the developed and delivered artifacts. Finally, operations personnel would gain the ability to perform objective testing and savings of time involved.

3. Reference Architecture

The purpose of a reference architecture is to identify the major modules, module functions, and interfaces for a software system. The proposed manufacturing simulation architecture defines a distributed system that may be comprised of simulation as well as real manufacturing systems. Simulation provides technically correct models of human and manufacturing organization behavior, systems, and processes. The proposed testing approach envisages a capability for connecting and using real manufacturing systems alongside the virtual gaming and simulation systems. The architecture allows the integration of real manufacturing systems with simulation software, but a discussion of those integration issues is beyond the scope of this paper.

3.1 A Distributed System Approach

Techniques for integrating distributed simulation systems have evolved over the years. Within the simulation world, the High Level Architecture (HLA) has been used to integrate distributed simulation systems. While HLA has become a standard [1] for distributed military simulations, the manufacturing world has not adopted the standard.

The proposed architecture is composed of a distributed collection of simulation applications and HLA integration mechanisms that allow the applications to work together. Why is the architecture based upon a distributed set of manufacturing simulations rather than a single monolithic one? A distributed approach increases the functionality of simulation enabling users to do things that they could not do with a monolithic system. For example, a distributed approach would allow users to:

- utilize simulation tools developed by different software developers that are specialized to model specific aspects of manufacturing. Individual simulation-vendor's products may not provide the capabilities to model all areas of interest
- allow a vendor to hide the internal workings of a simulation system through the creation of run-time simulators with limited functionality
- provide simultaneous access to executing simulation models for users in different locations (collaborative work environments)
- take advantage of additional computing power, specific operating systems, or peripheral devices (e.g., virtual reality interfaces) afforded by distributing across multiple computer processors
- offer different types and numbers of software licenses for different functions supporting simulation activities (model building, visualization, execution, analysis).
- create an array of low-cost, run-time, simulation models that can be integrated into larger models
- model supply chains across multiple businesses where some of the information about the inner workings of each organization may be hidden from other supply chain members
- simulate multiple levels of manufacturing systems at different degrees of resolution such that lower level simulations generate information that feeds into higher levels.

Figure 1 shows the major elements of the manufacturing simulation reference architecture. The elements of the architecture include clusters of simulation applications, data servers, an integrated infrastructure, and a simulation management module.

3.2 Simulation Applications

- Simulation applications will be used to model the behavior of real manufacturing systems. Several clusters of manufacturing simulators are envisioned. Each cluster and possible simulation applications are briefly introduced below.*Supply chain simulators* can be used to model the organization and management of supply chains. Organizations that may be simulated include supply chain headquarters, manufacturing primes, suppliers, transportation networks, warehouses, distribution centers, retailers, and customers. Some of the issues that may be addressed include lead times, inventory levels, production capacity, operations under surge conditions, and information flows.
- *Enterprise organization simulators* can be used to model the internal business processes of various

departments within the manufacturing organization, such as customer order servicing, design, engineering, production, and inventory management. Business process modeling techniques may be used to analyze order flow and processing times in order to streamline operations and minimize non value-added functions.

- *Manufacturing system and equipment simulators* can be used to model the normal operations, failure modes, and maintenance of various manufacturing equipment, such as fabrication, assembly, material handling, quality, and packaging systems. Examples of some of the equipment making up these systems includes machine tools, coordinate measuring machines, robots, storage and retrieval systems, and conveyors. Discrete event simulation techniques may be used to analyze operation times, capacity, queue lengths, bottlenecks, buffer storage requirements, inventory levels, etc.
- *Physical process simulators* can be used to create accurate models of the physical transformations that products and tooling undergo in various manufacturing industries. Industries that will have unique process simulations include metalworking, electronics, food, textiles, plastics, and chemicals/refining. For example, a physical process simulator for metalworking may model processes associated with a machine tool's operation. Information obtained from the simulation may include changes to work piece geometry, chip formation, tool wear, chatter, thermal and mechanical variations to the machine.

3.3 Manufacturing Simulation Data Model

If a number of software applications including simulators are going to share data, they should have a common understanding of its meaning and structure. In this section, the concept of a common, shop information model is introduced. The primary objective of this model is to develop a structure for exchanging shop data between various manufacturing software applications, including simulation. The idea is to use the same data structures for managing actual production operations and simulating the machine shop. The rationale is that if one structure can serve both purposes, the need for translation and abstraction of the real data would be minimized when simulations are constructed. The mapping of real world data into simulation abstractions is not, for the most part, addressed in the current data model. Figure 2 illustrates some of the major elements of the conceptual data model and their relationships to each other. For a more detailed discussion of the data model, see [2] or [3].

Maintaining data integrity and minimizing the duplication of data is an important requirement. For this reason, each unique piece of information appears in only one place in the model. Cross-reference links are used to avoid the creation of redundant copies of data.

The current version of the data model is focused on machine shops and contains twenty major elements. Each of the major data elements are italicized in the discussion that follows. The data elements are called: *Organizations, Calendars, Resources, Skill-definitions, Setup-definitions, Operation-definitions, Maintenance-definitions, Layout, Parts, Bills-of-materials, Inventory, Procurements, Process-Plans, Work, Schedules, Revisions, Time-Sheets, Probability-distributions, References,* and *Units-of-measurement*. Due to space limitations, the entire model is not shown or discussed in detail. The remainder of this section discusses the data elements and their significance.

Perhaps a good place to start the discussion of the data model is with the customer. Machine shops are businesses. They typically produce machined parts for either internal or external customers. Data elements are needed to maintain information on customers. The types of organizational information that is needed about customers are very similar to the data needed about suppliers that provide materials to the shop. The same types of organizational data are also needed about the machine shop itself. For this reason, an *Organizations* element was created to maintain organizational and contact information on the shop, its customers, and its suppliers.

Organizations can be thought of as both a phone book and an organization chart. The element provides sub-elements for identifying departments, their relationships to each other, individuals within departments, and their contact information. Various other types of information needs to be cross-referenced to organizations and contacts within structure, e.g., customer orders, parts, and procurements to suppliers.

The operation of the machine shop revolves around the production of parts, i.e., the fabrication of parts from raw materials such as metal or plastic. The raw materials typically come in the form of blocks, bars, sheets, forgings, or castings. These materials are themselves parts that are procured from suppliers. The *Parts* data element was created to maintain the broad range of information that is needed about each part that is handled by the machine shop. Part data includes an identifying part number, name, description, size, weight, material composition, unit-of-issue, cost, group technology classification codes, and revision (change) data. Cross-reference links are needed to the customers that buy the parts from the shop and/or the suppliers that provide them as raw materials. Links are also needed to other data elements, documents, and files that are related to the production of parts including part specification documents, geometric models, drawings, bills-of-materials, and process plans.

The *Bills-of-materials* element is basically a collection of hierarchically structured parts lists. It is used to define the parts and subassemblies that make up higher-level part assemblies. A bill-of-materials identifies the component or subassembly required at each level of assembly by a part-number reference link. The quantity required for each part is also indicated. Cross-reference links are needed between parts that are assemblies and their associated bill-of-materials.

The *Parts* and *Bills-of-materials* elements establish the basic definition of parts produced or used by the shop. Another element, Inventory, is used to identify quantity of part instances at each location within the facility. Inventory data elements are provided for parts, tools, fixtures, and materials. Materials are defined as various types of stock that may be consumed partially in production, e.g., sheets, bars, and rolls. Structures are provided within inventory to keep track of various stock levels (e.g., reorder point level) and the specific instances of parts that are used in assemblies.

The *Procurements* element identifies the internal and external purchase orders that have been created to satisfy order or part inventory requirements. Cross-reference links are defined to *Parts* to identify the specific parts that are being procured and to Work to indicate which work items they will be used to satisfy.

The *Work* data element is used to specify a hierarchical collection of work items that define orders, production, and support activities within the shop. Support activities include maintenance, inventory picking, and fixture/tool preparation. *Work* is broken down hierarchically into *orders, jobs*, and *tasks*.

Orders may be either customer orders for products or internally-generated orders to satisfy part requirements within the company, e.g., maintenance of inventory levels of stock items sold through a catalog. The *Orders* element contains both definition and status information. Definition information specifies who the order is for (i.e., customer cross-references), its relative priority, critical due dates, what output products are required (a list of order items, i.e., part references and quantities required), special resource requirements, precedence relationships on the processing of order items, and a summary of estimated and actual costs. Order items are also cross-referenced to jobs and tasks that decompose the orders into individual process steps performed at workstations within the shop. Status information includes data about scheduled and actual progress towards completing the order

Jobs typically define complex production work items that involve activities at multiple stations and ultimately produce parts. *Tasks* are lower-level work items that are typically performed at a single workstation or area within the shop.

The *Process-plans* element contains the process specifications that describe how production and support work is to be performed in the shop. Major elements contained within *Process-plans* include routing sheets, operation sheets, and equipment programs. Routing and operation sheets are the plans used to define job and task level work items, respectively, in the work hierarchy. These process plans define the steps, precedence constraints between steps, and resources required to produce parts and perform support activities. Precedence constraints defined in process plan are used to establish precedence relationships between jobs and tasks. Equipment program elements establish cross-reference links to files that contain computer programs that are used to run machine tools and other programmable equipment that process specific parts. Each part in the *Parts* element contains cross-reference links to the process plans that define how to make that part. Jobs and tasks contain links back to the process plans that defined them.

The *Resources* element is used to define production and support resources that may be assigned to jobs or tasks in the shop, their status, and scheduled assignments to specific work items. The resource types available in the machine shop environment include: stations and machines, cranes, employees, tool and tool sets, fixtures and fixture sets.

The *Skill-definitions*, *Setup-definitions*, *Operations-definitions*, *Maintenance-definitions*, and *Time-Sheets* elements provide additional supporting information associated with resources. *Skill-definitions* lists the skills that an employee may possess and the levels of proficiency associated with these skills. Skills are referenced in employee resource requirements contained in process plans. *Setup-definitions* typically specifies tool or fixture setups on a machine. Tool setups are typically the tools that are required in the tool magazine. Fixture setups are work-holding devices mounted on the machine. Setups may also apply to cranes or stations. *Operation-definitions* specifies the types of operations that may be performed at a particular station or group of stations within the shop. *Maintenance-definitions* specifies preventive or corrective maintenance to be done on machines or other maintained resources. *Time-sheets* are used to log individual employee's work hours, leave hours, overtime hours, etc.

The *Layout* element defines the physical locations of resource objects and part instances within the shop. It also defines reference points, area boundaries, paths, etc. It contains references to external files that are used to further define resource and part objects using appropriate graphics standards. Cross-reference links are also provided between layout objects and the actual resources that they represent.

Schedules and *Calendars* data elements are used to deal with time. *Schedules* provides two views of the planned assignment of work and resources. Work items (orders, jobs, and tasks) are mapped to resources, and conversely, resources are mapped to work items. The planned time events associated with those mappings are also identified, e.g., scheduled start times and end times. *Calendars* identifies scheduled work days for the shop, the shift schedules that are in effect for periods of time, planned breaks, and holiday periods.

The four remaining major data elements are *Revisions, References, Probability-distributions*, and *Units-of-measurement.* The *Revisions* element is used repeatedly throughout many levels of the data model. It provides a mechanism for identifying versions of subsets of the data, revision dates, and the creator of the data. The *References* element identifies external digital files and paper documents that support and further define the data elements contained within the shop data structure. It provides a mechanism for linking outside files that conform to various other format specifications or standards, e.g., part design files. The *Probability-distributions* element defines probability distributions that are used to vary processing times, breakdown and repair times, availability of resources, etc. *Distributions* may be cross-referenced from elsewhere in the model, e.g., equipment resources maintenance data. *Units-of-measurement* specifies the units used in the file for various quantities such as length, weight, currency, speed, etc.

This information model and associated data formats are undergoing standardization under the Simulation Interoperability Standards Organization [4].

3.4 Simulation Integration Infrastructure

Although simulations are often implemented as individually executable computer processes, sometimes there is a need to divide simulations into multiple processes. These processes may need to run as a distributed simulation system on a single computer or over a network of computers. A distributed simulation system may be used to:

- divide a large simulation into smaller functional modules that can be used by multiple training packages
- provide a simulation service to other client applications
- enable coordinated simulations over a local Intranet or the Internet.

The High Level Architecture (HLA) is a standard, originally initiated by the Department of Defense (DoD), for implementing distributed simulation. It was developed by the Defense Modeling and Simulation Office (DMSO) to provide a consistent approach for integrating distributed, defense simulations. In HLA terms, the individual simulations are called federates and the distributed simulation is referred to as a federation. The HLA defines a framework by which individually executing federates can be combined into a distributed simulation federation.

The HLA framework has three major parts. The first part is a set of rules that federates and federations must adhere to ensure that a federation operates properly. The second part is the integration infrastructure called the Run Time Infrastructure (RTI). The RTI defines an interface that provides a number of services that federates can use to communicate (i.e., exchange simulation data), and coordinate their execution (i.e., synchronize simulation clocks) with other federates in a federation. The third part of the HLA is called the Object Model Template (OMT). The OMT provides a means for describing the format of the data that will be exchanged between federates. See [5] for more information on distributed simulation using HLA.

Several implementations of the HLA RTI software are currently available from different sources. There is, however, no interoperability across different vendor's RTI implementations. A distributed simulation running on different computer systems across a network must use the same RTI software as an integration infrastructure

An HLA-based distributed manufacturing simulation may include simulators, visualization system, real production system, and output analysis system as federates. One common data definition is created for domain data that is shared across the entire federation. It is called the federation object model (FOM). Each federate has a simulation object model that defines the elements of the FOM that it implements.

Integration of distributed simulations using HLA requires significant expertise and effort. NIST researchers have developed a Distributed Manufacturing Simulation (DMS) adapter that reduces the integration effort and provides access to basic HLA capabilities. The DMS adapter is in particular useful for integrating legacy simulations.

DMS Adapter Module is incorporated into each DMS federate. The DMS Adapter handles the transmission, receipt, and internal updates to all FOM objects used by a federate. The DMS Adapter Module will contain a subroutine interface that will facilitate its use as an integration mechanism by software developers. The DMS adapter eases the development of distributed manufacturing simulations by reusing implementations for some of the necessary housekeeping and administrative work. The DMS adapter provides a simplified time management interface, automatic storage for local object instances, management of lists of remote object instances of interest, management and logging for interactions of interest, and simplified object and interaction filtering. For a more detailed discussion of the NIST distributed manufacturing simulation architecture and the adapter module, see [6].

4. Conclusions

This paper presented an architecture for integrating simulation systems within the manufacturing domain. Such an environment will allow testing of practices or technologies for communicating information across the hierarchy as well as the decisions at each level.

It is proposed that the architecture be implemented as a common infrastructure that can be used to integrate independently developed simulation modules. The availability of such an infrastructure will strongly encourage development of simulation modules covering the breadth and depth of the manufacturing domain. Manufacturing personnel can select the modules applicable to their environment to create a capability to serve their testing needs.

An implementation of the architecture will provide a test bed for the Manufacturing Interoperability Program at NIST's Manufacturing Engineering Laboratory and other standards organizations. It can be used to test the interoperability of manufacturing applications including enterprise resource planning, scheduling, manufacturing execution systems, machine and material handling equipment control programs, and machine and robot programs. It can also be used to test the interfaces for such applications.

The proposed test bed will be highly effective if supported with repositories for templates and test case data. Academic and commercial researchers can use the templates and test case data to quickly test out new developments. The test case data can also serve as a benchmark for comparison of alternate approaches for similar applications and thus further spur development and help manufacturing personnel by providing a common scale to rank vendor offerings.

Implementation of the architecture as a common infrastructure will require development of standards at several fronts including the data models, interfaces, distribution and synchronization mechanisms and user interaction devices. NIST researchers have prepared draft standards for shop floor data and are working with the Simulation Interoperability Standards Organization for their formal acceptance. Current work in progress on integration of manufacturing simulations is expected to lead to more such activity in the future.

References

[1] IEEE Standards Association, IEEE 1516-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. (Accessed at http://standards.ieee.org/)

[2] Lee, Y. Tina, Charles McLean, and Guodong Shao, "A Neutral Information Model for Simulating Machine Shop Operation." Proceedings of the 2003 Winter Simulation Conference, eds: S. Chick, P.J. Sanchez, D. Ferrin and D.J. Morrice, Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2003, pp.1296-1304.

[3] McLean, Charles, Y. Tina Lee, Guodong Shao, and Frank Riddick, "Shop Data Model Interface Specification," NISTIR 7198, National Institute of Standards and Technology, Gaithersburg, MD, January 2005.

[4] Simulation Interoperability Standards Organization, Core Manufacturing Simulation Data Product Development Group, (accessed at http://www.sisostds.org/index.php?tg=articles&idx=More&article=42&topics=20 on Feb. 26, 2007)

[5] Kuhl, F., R. Weatherly and J. Dahmann, Creating Computer Simulations: An Introduction to the High Level Architecture, Prentice Hall, Upper Saddle River, NJ, 1999.

[6] McLean, Charles, Frank Riddick, and Y. Tina Lee, "An Architecture and Interfaces for Distributed Manufacturing Simulation," Simulation: Transactions of the Society for Modeling and Simulation International, Volume 81, No. 1, Sage Publications, San Diego, CA, January 2005, pp. 15-32.

Biographies

CHARLES R. MCLEAN is a computer scientist and Group Leader of the Manufacturing Simulation and Modeling Group at the National Institute of Standards and Technology. He has managed research programs in manufacturing simulation, engineering tool integration, product data standards, and manufacturing automation at NIST since 1982. He has authored more than 50 technical papers on topics in these areas. He is on the Executive Board of the Winter Simulation Conference and the Editorial Board of the International Journal of Production, Planning, and Control. He is formerly the Vice Chairman of the International Federation of Information Processing

(IFIP) Working Group on Production Management Systems (WG 5.7). He holds an M.S. in Information Engineering from University of Illinois at Chicago and a B.A. from Cornell University. His e-mail address is <mclean@cme.nist.gov> and his web address is <http://www.mel.nist.gov/msidstaff/mclean.chuck.html>.

SANJAY JAIN is an Assistant Professor in the Department of Decision Sciences, School of Business at the George Washington University (GWU), and works part-time at NIST under a research arrangement. Sanjay serves as an associate editor of the *International Journal of Simulation and Process Modeling* and also as a member of the editorial board of *International Journal of Industrial Engineering*. He is a senior member of the Institute of Industrial Engineers and a member of APICS - The Association for Operations Management. He received a Bachelors of Engineering from Indian Institute of Technology (IIT)-Roorkee, India, a Post Graduate Diploma from National Institute of Industrial Engineering, Mumbai, India, and a Ph.D. in Engineering Science from Rensselaer Polytechnic Institute, Troy, New York. His email address is <jain@gwu.edu> and his Web address is <http://home.gwu.edu/~jain>.

FRANK RIDDICK is a staff member in the Manufacturing Simulation and Modeling Group in The National Institute of Standards and Technology (NIST) Manufacturing Systems Integration Division. He has participated in research and authored several papers relating to manufacturing simulation integration and product data modeling. He holds a Master's Degree in Mathematics from Purdue. His email address is <riddick@cme.nist.gov>.

Y. TINA LEE is a computer scientist in the Manufacturing Simulation and Modeling Group at NIST. She joined NIST in 1986. Most recently, she has been working on the design and development of interface information models to support the Software Engineering Institute (SEI) Technology Insertion Demonstration and Evaluation (TIDE) project. Previously she worked at the Contel Federal Systems and at the Sperry Corporation. She received her BS in Mathematics from Providence College and MS in Applied Science from the College of William and Mary. Her e-mail address is <leet@cme.nist.gov>.

Acknowledgement
Work described in this report was sponsored by the National Institute of Standards and Technology (NIST) Manufacturing Interoperability Program, Gaithersburg, MD. The Manufacturing Interoperability Program supports NIST projects applying information technologies and standards-based approaches to manufacturing software integration problems. The work described was funded by the United States Government and is not subject to copyright.

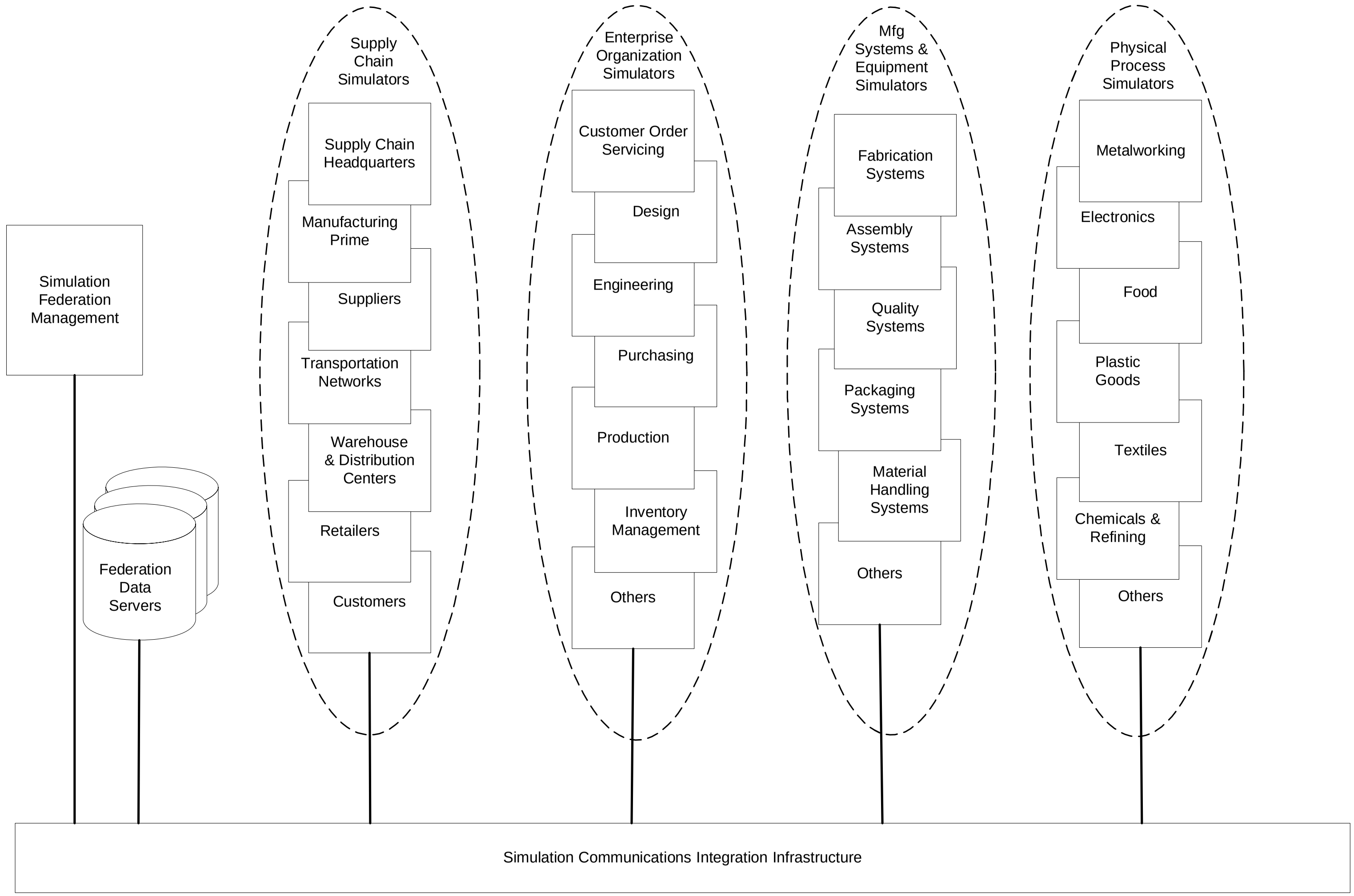

Figure 1. Manufacturing simulation system clusters, simulation federation management, data servers, and the integration infrastructure.

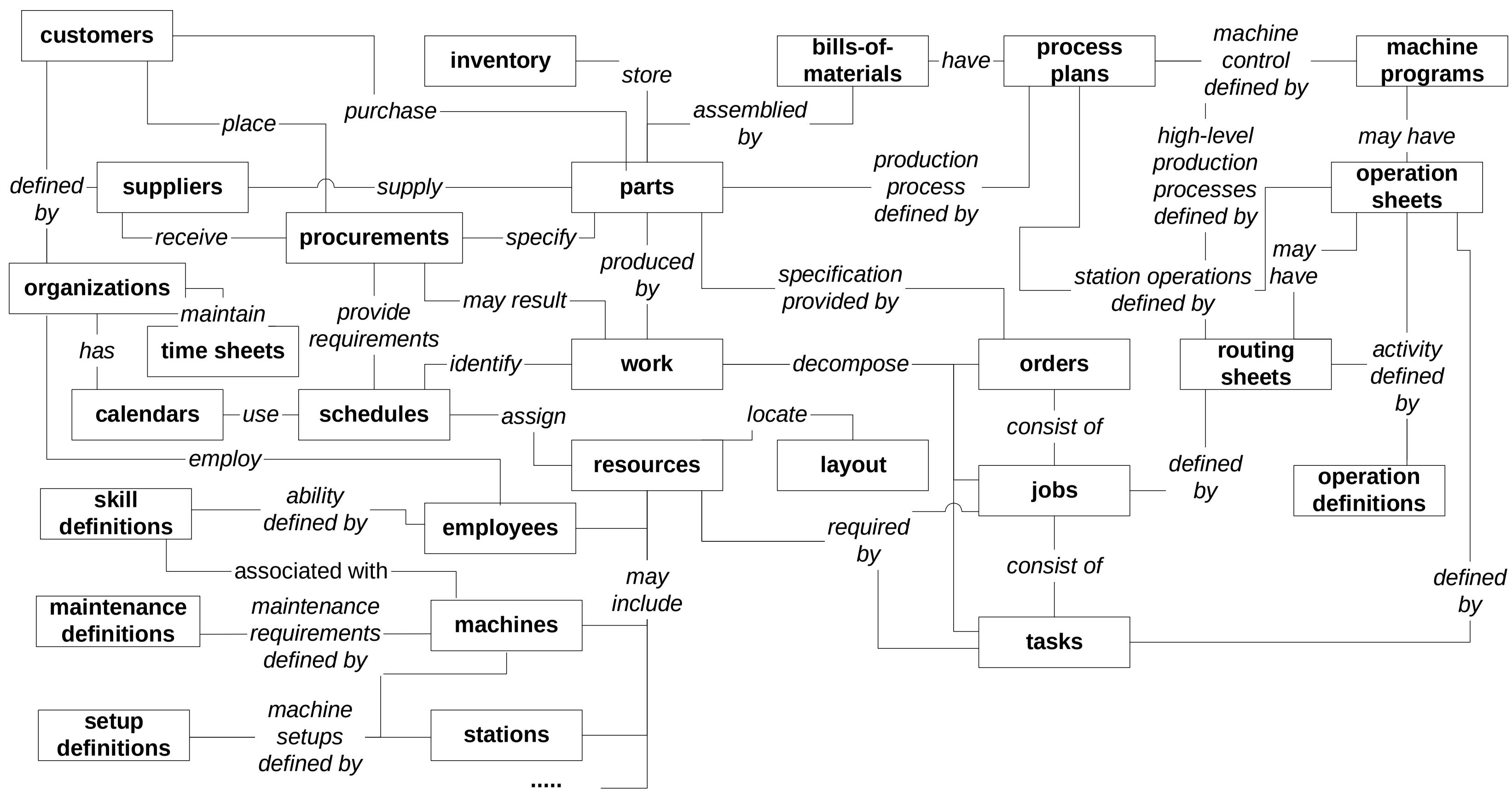

Figure 2. Major elements of the manufacturing conceptual data model and their relationships to each other