INTEGRATION OF MANUFACTURING SIMULATIONS USING HLA

Charles McLean and Frank Riddick
Manufacturing Systems Integration Division
National Institute of Standards and Technology (NIST)
Gaithersburg, MD (USA)
email: charles.mclean@nist.gov, frank.riddick@nist.gov

KEYWORD

Manufacturing, simulation, architectures, HLA, standards

ABSTRACT

To fully implement simulation-based acquisition, it will be critical to be able integrate manufacturing simulations from the supply chain down to the shop floor process level. Currently no standard exists for integrating these different types of simulations. This paper presents an overview of a neutral reference architecture for integrating distributed manufacturing simulation systems with each other, with other manufacturing software applications, and with manufacturing data repositories. The architecture identifies the interfaces that need to be standardized to integrate manufacturing simulations. It uses the Department of Defense High Level Architecture (HLA) as an integrating infrastructure.

The architecture identifies the software building blocks and interfaces that will facilitate the integration of distributed simulation systems and enable the integration of those systems with other manufacturing software applications. Other manufacturing software applications that must be integrated with simulation include, but are not limited to systems used to: 1) design products, 2) specify processes, 3) engineer manufacturing systems, and 4) manage production. The architecture also addresses the decomposition of simulation systems into major component modules. It is being developed in cooperation with organizations in Europe and Japan as a part of the international Intelligent Manufacturing Systems (IMS) MISSION project.

1. INTRODUCTION

Scientists and engineers within the NIST Manufacturing Systems Integration Division of the Manufacturing Engineering Laboratory are developing an architecture for distributed manufacturing simulation in collaboration with representatives from a number of outside organizations. The organizations are principally participants in the IMS MISSION Project. MISSION is just one of many international collaborative projects that are currently underway as part of the IMS Program.

"The goal of MISSION is to integrate and utilise new, knowledge-aware technologies of distributed persistent data management, as well as conventional methods and tools, in various enterprise domains, to meet the needs of globally distributed enterprise modelling and simulation. This will make available methodologies and tools to support the definition of appropriate manufacturing strategies and the design of appropriate organizations and business processes. This goal will be achieved by establishing a modelling platform incorporating engineering knowledge and project information which supports space-wise and control-wise design, evaluation and implementation over the complete enterprise life cycle. This will be the foundation stone for an architecture to support engineering co-operation across the value chain of the entire extended enterprise."(MISSION 1998)

NIST is currently serving as the U.S. Regional Coordinator for the IMS MISSION project. For further information on the overall IMS Program, see the IMS Web page at [www.ims.org].

2. DISTRIBUTED MANUFACTURING SIMULATION

This document takes a broad view of distributed manufacturing simulation (DMS). Normally a DMS may be thought of as a manufacturing simulation that is comprised of multiple software processes that are independently executing and interacting with each other. Together, these simulation software processes may model something as large as a manufacturing supply chain down to something as small as an individual piece of industrial machinery. Different software vendors may have developed the basic underlying simulation software. The modules may run on different computer systems in geographically dispersed locations. The simulation may be distributed to take advantage of the

functionality of specific vendor's products, protect proprietary information associated with individual system models, and/or to improve the overall execution speed of the simulation through the use of parallel computer processors.

DMS may also refer to a distributed computing environment where other non-simulation manufacturing software applications are running and interacting with one or more simulation systems. Engineering systems may interact with simulation systems through service requests. That is, they submit data to a simulator for evaluation. For example, a computer-aided manufacturing application that has generated a control program for a machine tool may submit that program to a simulator to verify that it is correct.

Another view of DMS is a computer environment comprised of multiple, functional modules that together form what today is commonly a single simulation system. Such an environment may include model building tools, simulation engines, display systems, and output analysis software.

Why build distributed manufacturing simulation systems? A distributed approach increases the functionality of simulation. For example, it could be used to:

- 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,
- model multiple systems in a single factory with different simulation requirements such that an individual simulation-vendor's product does not provide the capabilities to model all areas of interest,
- allow a vendor to hide the internal workings of a system through the creation of run-time simulators with limited functionality,
- create an array of low-cost, run-time, simulation models that can be integrated into larger models,
- take advantage of additional computing power, specific operating systems, or peripheral devices (e.g., virtual reality interfaces) afforded by distributing across multiple computer processors,
- provide simultaneous access to executing simulation models for users in different locations (collaborative work environments), and
- offer different types and numbers of software licenses for different functions supporting simulation activities (model building, visualization, execution, analysis).

The next section outlines the role that software architectures will play in enabling the development of distributed manufacturing simulations.

3. SOFTWARE ARCHITECTURE

In their book, Software Architecture: Perspectives on an Emerging Discipline, Mary Shaw and David Garlan, explain the significance of software architectures:

"As the size and complexity of software systems increase, the design and specification of overall system structure become more significant issues than the choice of algorithms and data structures of computation. Structural issues include the organization of a system as a composition of components; global control structures; the protocols for communication, synchronization, and data access; the assignment of functionality to design elements; the composition of design elements; physical distribution; scaling and performance; dimensions of evolution; and selection among design alternatives. This is the software architecture level of design."(Shaw and Garlan 1996)

A distributed manufacturing simulation architecture is needed to address the integration problems that are currently faced by software vendors and industrial users of simulation technology. Neutral simulation interfaces would help reduce the cost of data importation and model sharing -- and thus would make simulation technology more affordable to users. The definition of a neutral architecture for distributed manufacturing simulation is the first step towards identifying the information models, interfaces, and protocols for integrating these systems.

This step can be achieved by decomposing the distributed manufacturing simulation architecture into three major functional views: *Distributed Computing Systems, Simulation Systems, and Manufacturing Systems.* Each architectural view defines a set of system elements, data models, and interface specifications for integrating distributed manufacturing simulations. Aspects of each view are interrelated to and interconnected with aspects of the other views. The views can be thought of as three sides of a cube.

3.1 Distributed Computing Systems View

This architectural view is primarily concerned with simulation as a set of computers and software processes that are simultaneous executing and communicating with each other across a computer network. This view also addresses issues pertaining to the general management and integration

of the software applications that are used to generate models and data for the simulations. The fact that the software processes are simulations or simulation-related is not particularly critical in this view. This view is not concerned with simulation or manufacturing data content.

This view provides the infrastructure that allows us to implement simulation development and execution environments as distributed systems. Elements of this view includes: hardware computing platforms; operating systems, distributed computer processes, integration infrastructures, process initialization and synchronization, software development environments (including but not limited to: editors, compilers, system build utilities, debuggers, source code, general subroutine and header libraries, run-time modules, and system test data), communications systems, information models and data dictionaries, work flow management systems, database management systems and databases, product data management (PDM) systems, version control and configuration management, computer file systems and files, system installation and maintenance utilities, computer security and data protection services, license verification systems, and World Wide Web access mechanisms. It also includes various input and output peripheral devices, such as digital cameras, scanners, monitors, projection displays, printers, and virtual reality (VR) interfaces.

There are five major clusters of information systems that are relevant to the distributed manufacturing simulation problem: 1) software development systems; 2) design, engineering, production planning, and simulation model development systems; 3) distributed manufacturing simulation execution systems; 4) manufacturing management, control, production, support systems, and 5) distributed manufacturing data repository systems. Each implementation of this architecture will undoubtedly be based on different information systems and physical configurations.

The Software Development Environment is used to develop simulation engines, visualization systems, integrating infrastructures, and other software applications. It is not the central focus of the architecture and will not be addressed in this paper. The Design, Engineering, Production Planning, and Simulation Model Development Environment contains the systems that generate models and data used by simulation and manufacturing itself. The Distributed Manufacturing Simulation Execution Environment contains simulation engines executing models, visualization systems, and infrastructure systems to manage and integrate those simulations. The Manufacturing Management, Control, Production and Support Systems Environment contains the “real” systems that are used to run and perform the manufacturing operations.

A communications network connects environments with each other and the Manufacturing Data Repository. The Repository is a consolidation of the various data stores and management systems that are used by the various information systems environments. It logically integrates the file systems, Web pages, data bases, and libraries used for the storage of data by design, engineering, production planning, real manufacturing systems, simulation model development, and executing distributed manufacturing simulations. In different implementations of the architecture, the repository may reside on a single computer system, a file server, or be geographically distributed across a network.

The Distributed Manufacturing Data Repository may include the following types of data stores and management systems: computer file systems, Web pages and files, object-oriented database management systems, relational database management systems, special purpose library management systems, and source-code control systems for software. A consistent, common data access interface mechanism will be used to simplify access to the data repository by all software environments and applications within those environments. References to documents in the data repository may be specified as Uniform Resource Locators (URLs), see (URL Web Site 2000). This will allow the identification of documents, both remotely and locally stored using the well-established standard World Wide Web access mechanism.

In the future, additional data management schemes and data stores may be added to the repository structure. From this point forward in this document, the Distributed Manufacturing Data Repository and Common Data Access Mechanism will be treated and represented as a single module.

3.2 Simulation Systems View

This architectural view is concerned with the specifics of building, initializing, running, observing, interacting with, and analyzing simulations. In this view, simulation systems, tools, and supporting applications should be viewed generically; i.e., independent of the manufacturing domain. The same system elements could be used for simulating other problem domains. Major elements of this view include: simulation coordination and management, visualization systems, manufacturing data preparation and model development tools, simulation models, discrete event and

process simulation engines, component module and template libraries, mathematical and analytical models, input distributions, timing and event calendars, and output analysis tools.

The component elements of this environment are: 1) product design applications and tool kits, 2) manufacturing engineering applications and tool kits, 3) production management applications and tool kits, 4) simulation model development applications and tool kits, 5) work flow management system, and 6) distributed manufacturing data repository.

In this environment, the work flow management system provides the integrating infrastructure. It manages and sequences activities within the applications and tool kits that generate manufacturing models and data. Tool kits are tightly coupled suites of applications that work together to perform a related set of functions. Tool kits may be manually driven or more automated expert systems.

Product design applications may include conceptual and detailed design, solid modeling, bill of materials generation, design handbooks, parts catalogs, and various analysis tools. Some manufacturing engineering applications may include process planning and process specification, plant layout, machine tool programming, time standards development, line balancing, and tool and fixture design. Production management applications may include manufacturing resource planning, batch and lot sizing, and scheduling applications. Simulation model development tools include functions such as flowcharting, diagramming, model definition, and user level programming.

Figure 1 illustrates the relationship of the various elements of the distributed manufacturing simulation execution environment. The integration infrastructure for this environment, the Run-Time Infrastructure (RTI), is based on the U.S. Department of Defense High Level Architecture (HLA) developed by the Defense Modeling and Simulation Office (DMSO), see (Kuhl et al 2000). The HLA was developed by DMSO to provide a consistent approach for integrating distributed, defense simulations. Several implementations of the HLA RTI software are currently available from different sources. Currently, there is no interoperability across 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 simulation is called a federation. Each simulator, visualization system, real production system, or output analysis system that is integrated by the HLA RTI is called a federate. 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.

A DMS Adapter Module is incorporated into each DMS federate, see Figure 2. The DMS Adapter will handle the transmission, receipt, and internal updates to all FOM objects used by a federate. The DMS Adapter Module will contain a subroutine interface and data definition file that will facilitate its use as an integration mechanism by software developers. The goal of the DMS adapter is to ease 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.

Several functions may be needed for the proper operation of a distributed simulation that are logically outside of any one simulation federate. In the distributed manufacturing simulation environment, the Manufacturing Simulation Federation Manager is the architectural element that provides these functions. Some of its functions may be to: execute initialization scripts to launch federates, provide initialization data to federates, assist in federation time management, and provide a user interface so that users can monitor and manipulate the federation and invoke federation services.

3.3 Manufacturing Systems View

This architectural view is concerned with modeling the behavior and data of specific manufacturing organizations and systems, from the supply chain down to individual machines on the factory floor. Major elements of this view include, but are not limited to:

- manufacturing organizational templates and structures - business process and organizational models
- supply chain systems - refineries, mills, factories, warehouses, distributors, transportation systems, wholesalers, retailers, customers, ...
- manufacturing facility departments, areas, and subsystems - design, engineering, procurement, finance, production shops, work cells, production lines,

workstations, inventory storage areas, shipping and receiving, ...
- production resources and support equipment - machine tools, inspection equipment, material handling systems, storage buffers, robots, workers,
- tools and materials - cutting tools, hand tools, jigs and fixtures, consumables, components, part blanks, sheet and bar stock, work-in-process (WIP), …
- manufacturing information systems - design, engineering, production planning and scheduling, tool management, shop floor data collection systems, and
- manufacturing documents and data - work flow patterns, orders, jobs, product data, part designs, process plans, production calendars, schedules, layouts, and other reference data (machinability data, statistical distributions).

Different manufacturers will create different supply chain organizations and arrangements of systems within each organization. The DMS architecture must be flexible enough to allow these different system configurations, but still enable increased integration. As such, the architecture does not mandate a particular manufacturing organization. It does require the development and specification of one DMS FOM.

Many objects in the FOM may reference documents containing more detailed information that are stored in a file system, PDM system, or database. An example of such a document might be a part design file or a process plan. The Extensible Markup Language (XML) can be used to define new document types, see (Goldfarb and Prescott 2000). XML allows for the definition data that has semantic information in addition to the data values. XML data-type-definitions (DTD) may be used to define new document formats. Advantages of this approach include: the set of supported document types can be easily extended, each individual document format can be easily modified, COTS tools are available to implement creation, parsing, interpreting, and displaying the documents, XML documents from other sources can easily be supported, different instances of file structures may be created to convey the same semantic information, XML-enabled browsers can intelligently display the data, and semantic validation of the files is possible. Even without the DTD, XML files are often both human and machine readable because of the semantic information that is included.

There are potentially many document types that will be stored as distributed manufacturing simulation data. Some of these document types have widely-accepted or standardized formats. Examples of these include the many kinds of CAD files (DXF, IGES, etc.), image files (GIF, TIFF, BMP, etc), and executables (EXE, com, bat, dll, etc.). However, many manufacturing documents do not have standardized format. Schedules, BOMs, and process plans are examples of such documents. While it is easy to come up with acceptable representations for such data that are appropriate for short-term use, it is highly likely that these representations will need modifications, possibly major modifications, over time. A mechanism is needed to allow the definition of extensible formats for new document types without adversely affecting the rest of the DMS architecture or interfaces. XML can be that mechanism. XML DTD's must be stored in and uniquely accessible from the DMS data repository. An initial set of document formats should be developed and allowed to expand over time as the need arises.

4. CONCLUSION

This document has provided a brief overview of the distributed manufacturing simulation architecture that is being developed as a part of the IMS MISSION Project. The approach taken in the architecture is to facilitate integration of existing commercial systems with minimal new development work. The architecture also should enable experimentation with research systems that are based on evolving technology. The architecture describes the major system modules, data elements or objects, and interfaces between those modules. It uses the DOD High Level Architecture and Run-Time Infrastructure as an integrating infrastructure. Prototypes of each of these systems will be developed, tested, and integrated with commercial simulation systems, modeling tools, and other related manufacturing software applications as part of the IMS MISSION Project.

REFERENCES

MISSION Consortium: Intelligent Manufacturing System (IMS) Project Proposal. March 1998. *Modelling and Simulation Environments for Design, Planning and Operation of Globally Distributed Enterprises (MISSION*), Version 3.3. Shimuzu Corporation, Tokyo, Japan.

Shaw, M.; Garlan, D. 1996. *Software Architecture: Perspectives on an Emerging Discipline*. Prentice-Hall: Saddle River, NJ.

A beginners guide to URLs, 2000. http://www.ncsa.uiuc.edu/demoweb/url-primer.html.

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

Goldfarb, C.; Prescod, P. 2000 *The XML Handbook*, Prentice Hall, Upper Saddle River, NJ, 2000.

Work described in this paper was sponsored by the NIST Systems Integration for Manufacturing Applications (SIMA) Program. No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. The work described was funded by the United States Government and is not subject to copyright.

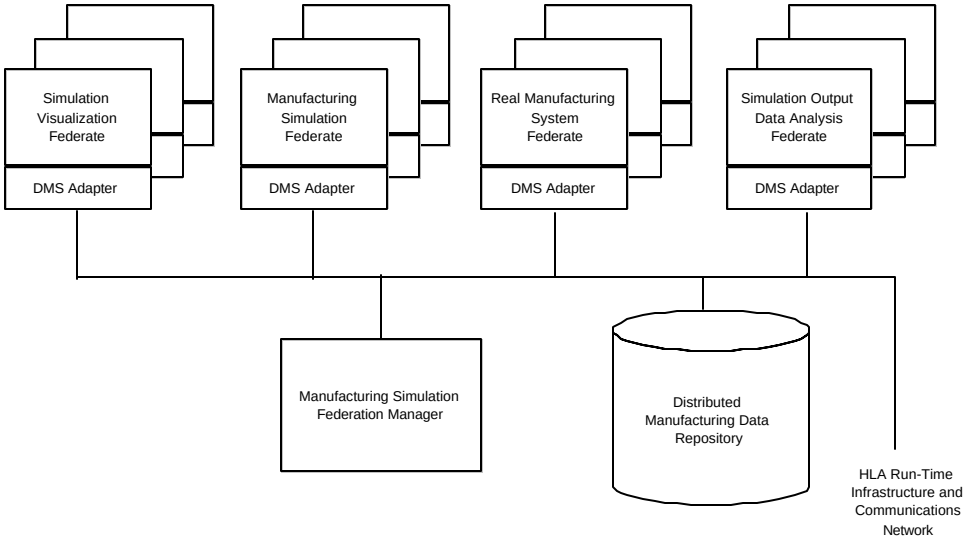

Figure 1: Distributed Manufacturing Simulation Environment elements integrated by the HLA Run-Time Infrastructure.

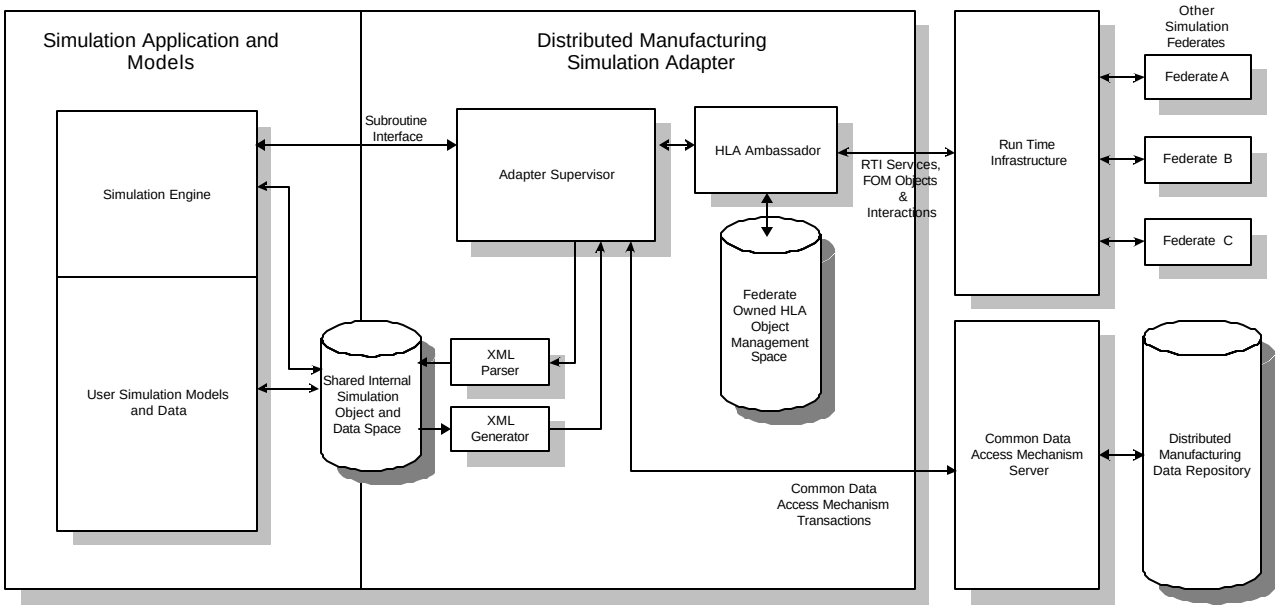

Figure 2: A decomposition of the DMS Adapter into its component modules.