ASSEMBLY SIMULATION IN THE DESIGN PROCESS IN A DISTRIBUTED ENVIRONMENT

Charles McLean[1], Deogratias Kibira[2], William Reiter[3], Paul Maropoulos[4]
[1]National Institute of Standards and Technology, Gaithersburg, MD, USA; [2]Makerere University, Kampala, Uganda; [3]Oregon State University, Corvallis, OR, USA; [4]University of Durham, Durham, UK

Abstract: Simulation technology is applicable across a distributed enterprise in the pursuit of bringing high quality products to the market in minimum time. The proliferation of computing technology and software applications now makes simulations available to many organizations in an enterprise. Significant focus has been devoted to manufacturing facility simulation and the automation of these simulations. The focus in this paper is assembly simulation early in a product design process in a distributed environment. The authors introduce a simulation framework, activity model, distributed simulation architecture, neutral data interfaces, and a case study of the use of assembly simulation early in the design process in a collaborative distributed environment.

Key words: manufacturing simulation, data standards, assembly simulation

1. INTRODUCTION

The primary reason for building manufacturing simulations is to provide support tools that aid the manufacturing design and engineering decision-making processes. Simulations are typically developed as a part of a case study commissioned by manufacturing management to address a particular set of problems, for example the design of an assembly line for producing a new power tool. The objectives of the case study determine the types of simulation models, input data, and output data that are required. Neutral model libraries and interface data standards could simplify the simulation analyst's job and significantly improve the simulation case study process.

The Manufacturing Simulation and Visualization Program at the National Institute of Standards and Technology (NIST) is focused on accelerating the development of standards for simulation model libraries and data. As a part of our program strategy, we are working with a number of industrial and academic collaborators to develop manufacturing simulation data standards. Simulation standards for models and data could help to accelerate the modeling process and reduce modeling costs.

In this paper, we present our research in the application of manufacturing simulation to the design and engineering of assembly lines for the production of small mechanical products. In this section, we briefly describe current simulation practices, suggest solutions for improvement, and introduce a framework for simulation standards. In subsequent sections, we describe activity and information models for concurrent engineering of product designs with production systems using simulation, an architecture for integrating distributed manufacturing simulations, and a simulation case study of a power tool assembly line.

1.1 Current Simulation Practice

Simulation textbooks typically recommend that a ten to twelve step process be followed in the development of simulation models. The recommended approach usually involves the following steps: (1) problem formulation, (2) setting of objectives and overall project plan, (3) model conceptualization, (4) data collection, (5) model translation into computerized format, (6) code verification, (7) model validation, (8) design of experiments to be run, (9) production runs and analysis, (10) documentation and reporting, and (11) implementation[1]. Unfortunately, this approach often leaves considerable work and possibly too much creative responsibility to the simulation analyst.

Using this approach, the process of modeling and simulation is perhaps as much an art as it is a science. Simulations are often developed from scratch, so the skill of the individual analyst may figure significantly in the quality of the results that are obtained. There is little opportunity for the analyst to build upon the work of others since each simulation is built as a custom solution to a uniquely-defined problem. Input data from other manufacturing software applications is not often in the format required for simulation, so data must often be abstracted, reformatted, and/or translated. Furthermore, pressure from manufacturing management to obtain quick results may have a negative impact on the performance of the simulation analyst and the quality of results obtained.

How could the manufacturing modeling and simulation process be improved? Today simulation analysts typically code their models from scratch using commercial commercial simulators. They often build custom data translators to reformat and import required data. A better solution would be to simplify the process through modularization, i.e., the creation of re-usable simulation model building blocks. Simulations would be constructed by assembling or configuring, modular building blocks. Similarly, neutral interface formats for transferring data between simulation and other manufacturing applications are also needed. Data would ultimately be imported directly into the simulators without translation using standard data input formats.

The development of neutral, vendor-independent data formats for storing simulation models could greatly improve the accessibility of simulation technology to industry by enabling the development of reusable models. Such neutral, simulation-model formats would enable the development of reusable models by individual companies, simulation vendors, equipment and resource manufacturers, consultants, and service providers. Model libraries could be marketed as stand-alone products or distributed as shareware. Neutral model formats would help enlarge the market for simulation models and make their development a more viable business enterprise. Standard formats for models would make it possible for simulation developers to sell model libraries much the same way clip art libraries are sold for graphics software packages today.

1.2 Simulation Standards Framework

How can we determine what simulation standards need to be developed? It is the authors' contention that the same basic analytical and model development processes are being repeated over and over again by simulation analysts around the world. Although a simulation analyst may think that each modeling problem is unique, considerable commonality can be found in each problem's component elements. If the different types of modeling problems addressed by simulation analysts could be classified according to a uniform scheme, commonalities could be exploited.

What factors were considered in creating a uniform framework for classifying the various aspects of manufacturing simulation problems? The goal of the framework is to identify the boundaries of manufacturing simulation and offer an initial skeleton that can be used to organize requirements for simulation model and data standards. A major aspects of the framework is a system for manufacturing simulation classification. The classification attributes are: 1) the industrial market sector, 2) the hierarchical level of the manufacturing organization, system, or process, 3) the simulation case study, and 4) model elements, input, and output data. For a more detailed discussion on the framework for classifying manufacturing simulations, see reference[2].

2. PRODUCTION SYSTEM ENGINEERING MODELS

Production system engineering may be used to specify designs for transfer lines, job shop environments, group technology cells, automated or manually-operated workstations, customised multi-purpose equipment, or entire plants. Some examples of the functions performed as part of this include: 1) identification of product specifications and production system requirements, 2) modeling and specification of manufacturing processes, 3) plant layout and facilities planning, 4) simulation and analysis of system performance, 5) specification of interfaces and the integration of information systems, 6) task and work place design, and 7) management, scheduling and tracking of projects.

An activity model has been developed to identify some of the functions and data required to support concurrent engineering of production systems. The model defines inputs, outputs, controls, and mechanisms for carrying out the functions using the Integrated Definition Method (IDEF0) modeling techniques[3]. The top level of the model identifies the production system engineering function, its inputs, and its outputs. The next level of the model decomposes the engineering process into five major functions or activities listed below and defines data flows between those activities:

- Define the production system-engineering problem
- Specify production processes required to produce the product
- Design the production system
- Model the system using simulation and evaluate its performance under expected operating conditions
- Prepare project plans and budgets

Figure 1 for an illustration of the top levels of the IDEF model. The model is discussed in further detail in references[4,5].

Detailed information models are under development to specify each data input and output identified in the process model. The information models can being used to implement shared-databases, exchange files, messages, and program calls for passing information between the commercial software tools. The data required to support the production system engineering activity model is briefly described below.

Production system requirements describes a production system engineering problem. It defines the constraints that characterize the production system that is to be designed. A standard model for production system requirements data would ultimately enable application developers to access common or shared databases that describe the intended system. This data includes: product identification data and key product attributes, production system and engineering project type, manufacturing constraints and issues, critical milestone dates and schedules, expected or estimated costs, and identification of manufacturing data for related products.

Product design describes the product to be manufactured, components, production resources, and other physical objects within the production system itself. Product designs contain various types of data including: product structure or bill of materials, geometry, topology, tolerances, and configuration management data. The specifications must provide neutral, common structures and information models for describing and accessing product data, e.g., ISO 10303 Standard for the Exchange of Product Model Data, commonly referred to as STEP.

Assembly process specification defines the product assembly process. Information contained within the process specification includes: process identification, process resources, process time and cost, and process relationships. Other process data that may be part of this specification includes:

- activity relationship matrices are defined that describe how different processes relate to each other, e.g., required proximity or location;
- specification of requirements for processes, tooling, job skills, timing and line balancing, quality control, process audits, production yield;
- development of process and inspection plans, process description sheets,
- development of time standards for operations;
- ergonomic analyses of manual tasks;
- value engineering analysis (idetermination of activities or steps that can be eliminated).

Plant layout includes: site or facility identification data, drawing annotations, dimensions and tolerances, reference points, location and orientation of process machinery (both two dimensional footprints and solid models), material handling systems, tooling, fixture devices, safety systems, human ergonomic models, work areas, storage areas, aisles, building structural elements, utilities and services, and layer naming conventions within drawings and models.

Production resource models identifies the types of production resources and attributes required to support assembly. Production resources nominally include machinery, test systems, material handling devices and carriers, tooling, work holding devices, materials, and staff. These models must not only characterize the attributes of production resources, but also enable encapsulation of their geometry and behaviors to support simulation.

Production system evaluation represents the results of the evaluation of a production system design. Examples of evaluation data include cost, performance measures, system reliability, safety, ergonomics, and timing data that is either associated with a particular production system or industry benchmark data used for reference and comparison purposes. Cost data may include: system design, engineering and construction costs, training costs, material costs, consumable costs, utility costs, unit production cost, touch labor cost per unit, estimated system maintenance costs. Performance measures include process yield, throughput, machine utilization, cycle time, work-in-process, and inventory turnover rates. Reliability data include machine downtime, mean-time-between-failures, and mean-time-to-repair systems. Safety data may include identification of processes and chemicals that are hazardous to employees, systems, or products. Ergonomic data identifies the level of physical effort and/or special physical abilities are needed to perform the production tasks.

Engineering management includes the project management and cost data required to support the engineering of production systems. Project planning data includes: project phases, tasks, resources, and timing data (early/late start and end dates, estimated task durations, slack and float, and lead times.) Budget and cost data may include: project phase, planning, labor, tooling, capital equipment, projected maintenance, information and control system, operational, training, licensing and inspection, construction, installation, material (components, consumables), overhead (utilities, labor multipliers, area usage), and rental costs.

The information models for this data are being defined using the Unified Modeling Language (UML)[6]. The Extensible Markup Language (XML)[7,8] is being used to define the exchange formats for the information models. XML is a standard supported by the World Wide Web (W3C) organization[9]. XML supports the development of structured, hierarchical data entities that contain a high level of semantic content, that is both human and machine interpretable. Inexpensive tools are available to support creating and manipulating XML data. Being a textual format facilitates XML's use with the integration mechanisms commonly available in existing commercial off-the-shelf (COTS) software applications (file import/export, sockets, pipes, etc.). A significant portion of the UML models, XML structures, and supporting text describing the data specifications has been completed[10].

3. DISTRIBUTED MANUFACTURING SIMULATION ARCHITECTURE

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 simulation 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);
- offer different types and numbers of software licenses for different functions supporting simulation activities (model building, visualization, execution, analysis).

A distributed manufacturing simulation (DMS) architecture was developed to address the integration problems that are currently faced by software vendors and industrial users of simulation technology[11]. A neutral reference architecture enables the integration of distributed manufacturing simulation systems with each other, with other manufacturing software applications, and with manufacturing data repositories. Other applications include, but are not limited to: 1) product design, 2) process specification, 3) manufacturing systems engineering, and 4) production management and scheduling. The DMS architecture has been developed based upon extensions to the U.S. Department of Defense - Defense Modeling and Simulation Office's (DMSO) High Level Architecture (HLA). The HLA was developed to provide a consistent approach for integrating distributed, defense department simulations[12].

The architecture is comprised of three major functional views: *Distributed Computing Systems*, *Simulation Systems*, and *Manufacturing Systems*. Each view defines a set of system elements, data models, and interfaces 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. Due to space limitations, we will only describe the simulation view in this paper.

The simulation systems view is concerned with the specifics of building, initializing, running, observing, interacting with, and analyzing simulations. 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.

Figure 2 illustrates the relationship between the various elements of the distributed manufacturing simulation execution environment. The integration infrastructure for this environment is the HLA Run-Time Infrastructure (RTI). Several implementations of the HLA RTI software are currently available from different sources, but distributed simulation modules running across a network must use the same RTI software on each computer platform.

An HLA-based distributed simulation is called a federation. Each simulator, visualization system, real production system, or output analysis system 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.

In the NIST DMS architecture, a DMS Adapter Module is incorporated into each DMS federate. The adapter simplifies development by reusing implementations of some housekeeping and administrative subroutines. The Adapter handles the transmission, receipt, and internal updates to all FOM objects used by a federate. The Adapter contains a subroutine interface and data definition file that facilitates its use as an integration mechanism by software developers. It also 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.

The Manufacturing Simulation Federation Manager provides functions that are logically outside of any one simulation federate. It implements functions to execute initialization scripts that launch federates, provides initialization data to federates, assists in federation time management, and provides a user interface so that the federation and federation services can be interactively invoked, manipulated, and monitored.

4. ASSEMBLY LINE CASE STUDY

A simulation case study of the development of a simulation a small assembly line was conducted to validate activity model, data requirements, and integration issues[13]. The simulation was also designed to be integrated into a higher level supply chain simulation at a later time using the distributed manufacturing simulation architecture that was previously described. The parts to be assembled were characteristic of those manufactured by Black and Decker, a project partner. The research focused on developing assembly line designs and simulations for three manufactured products: a handheld power drill, a palm-grip finishing sander, and a jigsaw. Figure 3 shows an assembly station on the drill line simulation.

A simulation case study such as this would typically be used by industry to: 1) support the definition, design, and actualization of the overall system, 2) evaluate different decision options for the design or reconfiguration of the production line, e.g., loading options, part mix, number of stations, layout, allocation of operations to stations, setting of cycle times for operations, tooling requirements, equipment selection, process and support equipment, staffing options, workstation space requirements, buffer storage requirements, materials handling requirements, 3) generate cost and performance data for the operation that cannot be calculated in a straightforward manner, e.g., throughput, resource utilization, idle times, bottlenecks, effects of breakdowns, rejects, and rework, service and material replenishment requirements, and effect of shift scheduling on throughput and cost, and 4) demonstrate and/or visualize the operation of the line.

Our approach was to assume that we had to develop the manufacturing process design for a prototype product for which the assembly line had yet to be realized. The product might be either a digital or physical prototype. As such, following analytical procedure was used:

- analyze the product model to determine the components parts list and quantities;
- construct an assembly tree showing the different stages of assembly of the product and the sequence of the various assembly processes;
- aggregate assembly operations that can be performed at a single workstation using precedence relations between task steps and time associated with each step;
- determine the layout arrangement of workstations on the production line
- identify material handling replenishment requirements based on product demand and cycle times.

Delmia's QUEST and IGRIP simulators were used to develop the models. IGRIP provided the graphical ergonomic modeling of workstation operations. QUEST provided the discrete-event modeling of the overall production line. These tools complemented each other in the determination of process times, assembly line balancing, and discrete event analysis of the process. Other issues that were addressed included allocation of buffer storage spaces, production scheduling issues, and material handling requirements.

The data that was required to conduct the simulation study included: product design, bill of materials, assembly constraints, assembly process specification including operation sequences, process times, and testing requirements, expected product demand, failure and repair data for production line systems, system configuration and layout options, unit cost of labor, tooling, and materials.

5. CONCLUSIONS

Ultimately manufacturing simulation will have a major impact on the way products are designed and manufactured. A key factor in increasing the utility of the simulation is the establishment of standard data formats. This paper has presented a brief introduction to our work on the development of a framework for simulation standards, activity and information models for concurrent engineering of products and production systems, distributed manufacturing simulation architecture, and a case study for the simulation of a production line for small mechanical products.

ACKNOWLEDGEMENTS

Mention of commercial products in this paper does not imply approval or endorsement of any commercial product by NIST. This project is funded [in part] by NIST's Systems Integration for Manufacturing Applications (SIMA) Program and the Software Engineering Institute's Technology, Insertion, Demonstration, and Evaluation (TIDE) Program. SIMA 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.

REFERENCES

1. McLean, C., Leong, S., "A Framework for Standard Modular Simulation," *Proceedings of the 2002 Winter Simulation Conference,* ed. E. Yücesan,

C.Chen, J. Snowdon, and J. Charnes, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

2. Banks, J.(ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, New York: John Wiley and Sons, 1998.
3. http://www.idef.com
4. McLean, C., "Tools and techniques for modelling production systems," *Strategic Management of the Manufacturing Value Chain, Proceedings of the International Conference of the Manufacturing Value-Chain, Troon, Scotland August 1998*, ed. U. Bititci and A. Carrie, Boston: Kluwer Academic Publishers, 1998.
5. McLean, C., Leong, S., "Industrial Need: Production System Engineering Standards," NISTIR 6019, National Institute of Standards and Technology, Gaithersburg, MD, 1997.
6. Alhir, S. *UML in a Nutshell*, Cambridge, MA: O'Reilly & Associates, 1998.
7. DuCharme, B., *XML: The Annotated Specification*, Upper Saddle River, New Jersey: Prentice Hall, 1999.
8. Goldfarb, C., *XML Handbook*, Upper Saddle River, New Jersey: Prentice Hall, 2002.
9. http://www.w3.org
10. McLean, C., Lee, T., Shao, G., Riddick, "Shop Data Model And Interface Specification," Draft NISTIR, National Institute of Standards and Technology, Gaithersburg, MD, 2003. (in review)
11. McLean, C., and Riddick, F., "The IMS MISSION Architecture for Distributed Manufacturing Simulation," *Proceedings of the Winter Simulation Conference 2000*, Orlando, FL, December 2000.
12. Kuhl, F., Weatherly, R., and Dahmann, J., *Creating Computer Simulations: An Introduction to the High Level Architecture*, Prentice Hall: Upper Saddle River, NJ, 1999.
13. Kibira, D., McLean, C., "Virtual Reality Simulation of a Mechanical Production Assembly Line," *Proceedings of the 2002 Winter Simulation Conference*. ed. E. Yücesan, C.Chen, J. Snowdon, and J. Charnes. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 2002.

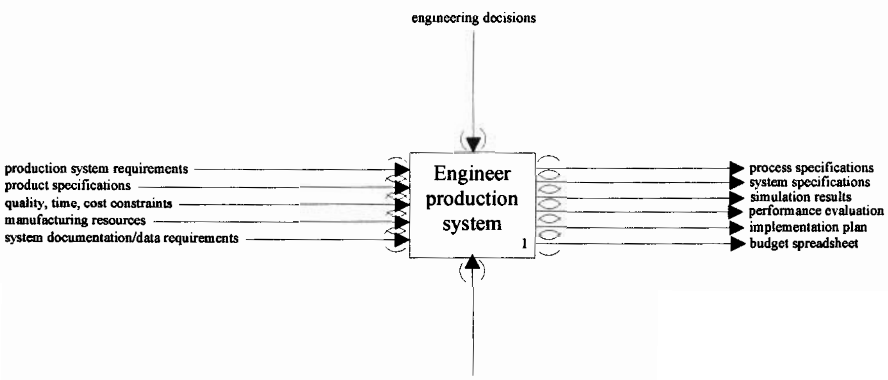

Figure 1 IDEF0 Activity model for production system engineering

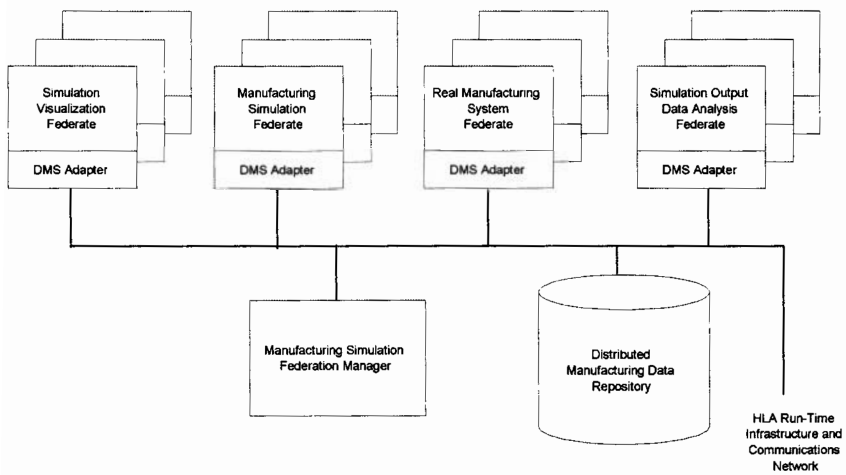

Figure 2 Simulation system view of distributed manufacturing systems architecture

Figure 3 Final assembly workstation in the drill line simulation