Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

NIST XML SIMULATION INTERFACE SPECIFICATION AT BOEING: A CASE STUDY

Roberto F. Lu

Commercial Airplanes Manufacturing R&D
The Boeing Company
Seattle, WA 98124-2207, U.S.A.

Guixiu Qiao
Charles McLean

Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8260, U.S.A.

ABSTRACT

Efficient and consistent simulation data management is indispensable and a challenging problem to be solved in modeling manufacturing and business processes. An extensible markup language (XML) based simulation interface specification is being developed by the National Institute of Standards and Technology (NIST). The proposed NIST document contains a prototype generic simulation data specification, which is an endeavor to fill a void in exchanging reusable simulation data. A case study was performed at Boeing Commercial Airplanes (BCA), utilizing the NIST XML simulation interface specification. Entity classes in this case study simulation model contain asynchronous servers, multi-input-output buffers, bi-directional cranes, labors (manpower), processes, and machines on different shifts. This model can be executed from a batch control language document, that is derived from the proposed NIST XML-based simulation specification. This case study illustrates a feasible method for using the XML-based NIST specifications in a discrete event simulation model of a manufacturing process.

1 INTRODUCTION

Discrete event simulation has been one of the most valuable technologies for assisting in decision evaluations in manufacturing and business processes. The competitiveness of an enterprise may be determined by its competencies in executing the lean concept, which can be influenced by the efficiency of practicing simulation modeling technology.

There are many different types of simulation technologies being exercised at The Boeing Company: physical mock-ups, flight deck simulations, finite element analyses, dimensional verifications, numerical control machining verifications, computational fluid dynamics, digital assembly modeling, ergonomics, and manufacturing-process-related discrete event simulations.

Simulation modeling exercises have been providing a quick and low-cost method for assessing and minimizing risks of change, streamlining manufacturing processes, validating production capabilities, and visualizing new systems. However, a consistent data integration specification has been absent in executing simulation projects. Hence, it takes real effort when exchanging simulation models during scenario iterations among project participants.

It will be mentioned in one of the Winter Simulation Conference 2003 panel discussions about the future of simulation (McLean et al. 2003) that operational simulation modeling was recently defined as number one among the top ten technologies at BCA. These ten technologies were categorized in three segments. Simulation was categorized under "Breakthrough" technology, which was placed above the "Enabling" and the "Core" technology segments. The BCA Manufacturing Research and Development (MR&D) group has been conducting simulation projects in areas of factory layout optimization, resource forecasting, rate change, production consolidations, and new product processes.

To address some of the issues associated with exchanging manufacturing information between existing information sources and manufacturing applications, the NIST has been developing an information model in an XML-based exchange file format that facilitates the exchange of manufacturing information between simulation applications and other manufacturing applications and/or data sources (Mclean, Jones et al. 2002).

This case study utilized the NIST XML simulation interface specification in a potential BCA wing manufacturing process line simulation at The Boeing Company. All information needed by simulation is written in a shop data file with the XML-based exchange file format. Using a translator, this shop data file is translated to the BCL file which can be executed directly by the simulation tool. The case study model uses QUEST software from the DELMIA Corporation, which is part of the Dassault Systems Group. Simulation modeling intends to verify a sizable possible capital investment to integrate multiple operations into one pulsing wing assembly line. For different scenarios, the content of the shop data file (e.g., process data) can be modified and used to drive a new scenario simulation quickly. For complex simulation models such as this case study model, this approach method is rather effective.

A brief explanation of the proposed NIST simulation specifications will be shown in section 2, and the simulation modeling portion of the case study example will be discussed in section 3. A more detailed demonstration of the model integration approach and means of the XML specification application will be shown in sections 4 and 5, respectively. Portions of the modeling results will be included in section 6.

2 THE PROPOSED NIST SPECIFICATION

XML is a markup language for documents containing structured information (Walsh 1998). XML is a simple and flexible text format originally designed to meet the challenges of large-scale publishing (Quin 2003). XML is also playing an increasingly important role in the exchange of data on the Web (Quin 2003). Most simulation software packages are object oriented and can be accessed in a structured manner within or via an external interface. Thus, XML is a preferred protocol for establishing simulation modeling specifications.

Commonly applied XML document validation methods include: Document Type Definition (DTD), which defines rules for a specific type of document; and the XML Schema, which is a W3C specification for validating the content of an XML document (Raab et al. 2002). XML Schemas add capabilities, such as defining data types for text elements as integer, string, date, etc., that were not present in DTDs. Here, both Unified Modeling Language (UML) and the schema were applied in this specification as part of efforts to support the development of a shop data information model.

An Extensible Stylesheet Language (XSL) Transformation (XSLT) is needed, in this particular case, to integrate the proposed NIST XML document to a BCL file for the simulation modeling practice in the QUEST software. XSLT is an XML-based language for transforming XML document content into other document formats (Means 2002).

Other XML related parsing technologies such as Document Object Model (DOM), Simple Object Access Protocol (SOAP), Simple API for XML (SAX), and so on are not explored in this case study.

The data format in the proposed NIST simulation specifications (McLean, Lee et al. 2002) consists of the following groups: general and miscellaneous, organizational structures, product and process specifications, product operations, resource definitions, and layout.

XML was the encoding mechanism for the exchange file, hereafter referred to as the Shop Data File. The Shop Data File provides a mechanism for configuring the shop model and sharing data between the simulation and the other shop software applications. The file contains not only executable or computable data to be processed by the simulation, but also descriptive text intended only for human interpretation. It also contains a network of cross-reference links between the various types of data required to plan and manage operations within the shop. It supports references to other external computer files and/or paper documents that provide more appropriate mechanisms or standards for encoding or representing data such as part drawings. Subsets of individual data types, i.e., substructures, may be created, stored, and/or exchanged using the file (Mclean, Jones et al. 2002).

The Shop Data Information Model contains four (4) major supporting data structures and fifteen (15) major manufacturing data structures.

Supporting data structures include:

1. time-sheets,
2. probability-distributions,
3. references, and
4. units-of-measurement.

Manufacturing data structures include:

1. organization-directory,
2. calendars,
3. skill-definitions,
4. operation-definitions,
5. resources (containing stations, machines, employees, cranes, tool-catalog, and fixture-catalog),
6. layout,
7. parts,
8. bill-of-materials group,
9. inventory,
10. process-plans (containing routing-sheets, operation-sheets, and machine-programs),
11. work (containing orders, order-items, jobs, and tasks),
12. purchase orders,
13. schedule,
14. maintenance-requirement-definitions, and
15. setups.

Data elements within the Shop Data Information Model are grouped into three categories: data element keys, commonly used data elements, and unique manufacturing shop data elements. Data elements keys and commonly used data elements are those which were found repeatedly at many levels of the data structures within the model. Data element keys are elements that serve as individual pointers or a collection of pointers to the data model. The commonly used data elements are presented according to data types of the element, including basic data elements, data elements with prefixes/suffixes, and complex data elements. A data element may be a basic data element, which can not be further subdivided into elements, or a complex data element, which contains more than one basic data element.

The overall structure of the shop-data XML document is shown below:

```
<shop-data type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <units-of-measurement />
  <organization-directory />
  <calendars />
  <resources />
  <skill-definitions />
  <setup-definitions />
  <operation-definitions />
  <maintenance-definitions />
  <layout />
  <parts />
  <bills-of-materials />
  <inventory />
  <procurements />
  <process-plans />
  <work />
  <schedules />
  <time-sheets />
  <references />
  <probability-distributions />
</shop-data>
```

3 CONTENTS OF THE CASE STUDY MODEL

The case study is in a discrete event simulation model about a possible future wing assembly line inside one of the Boeing manufacturing facilities.

The potential process flow of the case study wing manufacturing line is shown in the figure 1 below.

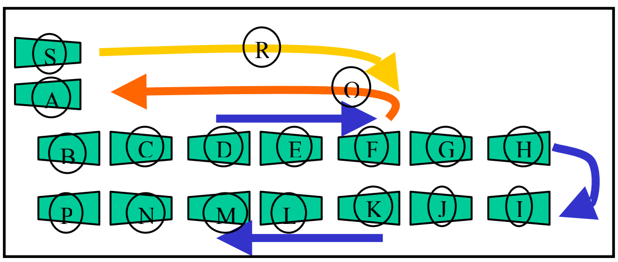

Figure 1: The Wing Process

Incoming wings arrive at a holding buffer, the position S, via the only overhead crane in the system. The wing will then be moved by crane to position B, which is the starting position of the pulsing line. Different processes take place along the whole line from the position B to position P as wings progress down the line. An additional process position outside this line takes place between positions E and F, where the wing will be moved by crane (route O) to holding position A. Two different ground transportation devices are designed in the system. The "high speed" dolly transports wings to and from the position A, one at a time. The "low speed" dolly carries one wing at a time throughout the wing line from position B to position P. Processes call for additional wing components at positions H, K, L, and P. At the end of the pulsing line, wings are moved by crane from the position P to position A. Then wings will be ready to be transported by high speed dollies to the airplane final assembly facility from position A.

Information collected for each position to establish this model is categorized in the list below:

1. Crane is required to move the part in,
2. Part movement time,
3. Incoming part schedule,
4. Incoming part arrival location,
5. Incoming part unloading time,
6. Setup time at the position,
7. Processing time at the position,
8. Unloading time at the position,
9. Crane required to move out,
10. Cost of each event,
11. Labor requirements,
12. Labor shift schedule, and
13. Labor team arrangements.

The following QUEST entity classes are utilized in this exercise:

1. Automated Guided Vehicle (AGV) controller,
2. AGV path system (crane),
3. AGV (crane),
4. AGV sub-resources (hoist and cross travel),
5. Crane AGV decision points,
6. Labor controllers,
7. Labors,
8. Buffers,
9. Machines,
10. Parts,
11. Processes,
12. Sinks, and
13. Sources.

The logic in this model was designed to reflect both potential benefits and "jig-locks," which is an undesirable situation when no entity can logically move forward. Routing conditions and logics are thoroughly assessed for all entities in the model. For example, the logic of the position A in this model was carefully examined to handle multiple possible wing movements: completed wings stop by this location en route to the final assembly; partially completed wings after position E are moved by crane to this position for an outside process to take place; wings that went through the outside process, treated as different part classes, will be carried by the "high speed" dolly to this position and followed by crane moving (route R) to position F or a holding buffer (not shown in figure 2) between positions E and F.

4 MODELING INTEGRATION APPROACH

A high level illustration of the modeling approach is shown in figure 2. Incoming right-hand and left-hand wings are

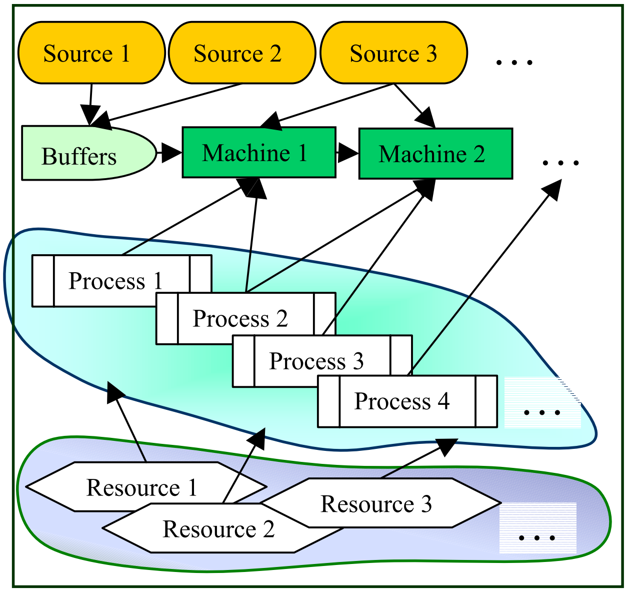

Figure 2: Modeling Approach

generated by different sources per a predetermined production schedule. Other wing components, such as ailerons, struts, flaps, slats, and wing tips are generated by their respective sources and delivered directly to their point of use machines by processes assigned at their relevant machines.

Labor and AGV types of resource entities are managed by their group controllers, where fundamentally common logics are assigned for all entities in the same class. Process entities are stand-alone classes. Each machine is assigned two to six different processes depending on complexity of its activities. Processes of each machine are running constantly throughout the whole model simulation time. In the event that conditions of a process are satisfied, fulfilled with the required part, labor, and AGV, then the process will consume the designed process time and then produce the defined product, which will then be transferred by the designed labor or AGV to the next machine or buffer as an incoming part for further processes. At the end of the last process, the final product in the system is transferred to the sink where the entity is removed logically from the system. The whole system operates repeatedly as an endless loop against time.

Simulation Control Language (SCL) and Batch Control Language (BCL) are the procedural and the command languages, respectively, that are available to users of the QUEST software. The enormous capability of the SCL provides users with literally unlimited precise simulation modeling options within the QUEST modeling environment. The powerful integration ability of the BCL enables users to manipulate QUEST externally in many different ways (DELMIA 2002a, 2002b).

This case study employs the BCL part of the QUEST software, since the XML data source originated from an external document that was not part of the DELMIA QUEST software. Elements in BCL codes of the example simulation model are derived from the sub-data file content. Iterations of this set of BCL codes are driven from the proposed NIST XML specifications. There are SCL and BCL macros that can be assigned in the QUEST modeling environment. This study utilizes the BCL macro function to execute BCL codes in QUEST.

Upon calling up the BCL macro, the BCL codes will be read and executed line by line into the QUEST modeling work space. It is beyond the scope of this paper to list the entire contents of the case study BCL document, which consists of almost 1500 lines of codes. The main building blocks of the BCL document are listed below:

1. Define environment configurations,
2. Create and define part classes,
3. Create and define schedule classes,
4. Create and define source classes,
5. Create and define buffer classes, including orientations of stacking and labor points,
6. Create a sink class,
7. Create and define cycle processes,
8. Create and define setup processes,
9. Create and define machine classes, including machine locations, orientations, number of stacking points, stacking point locations and orientations, associated processes, labor controllers, etc.,
10. Create and define an accessory class,
11. Create and define an AGV crane class and its subclasses,
12. Create and define AGV decision point classes,
13. Create and define labor classes,
14. Create and define class connections,
15. Assign process classes to their respective machine classes,
16. Assign machine stacking points to specific part class families, e.g., right-hand or left-hand parts,
17. Assign AGV decision point class destinations,
18. Assign schedule classes to machine classes, and
19. Assign routing logics and constraints.

This BCL file establishes the simulation model as shown in figure 3. Here is a portion from the BCL code document:

```
…
CREATE CYCLE PROCESS 'ProcRHwingPosC'
SET PROCESS 'ProcRHwingPosC' TIME TO 28800
SET PROCESS 'ProcRHwingPosC' PART REQUIREMENT
TO ANY PART CLASS 0
SET PROCESS 'ProcRHwingPosC' PART REQUIREMENT
TO PART CLASS 'Part737RHwing' 1
…
SET PROCESS 'ProcRHwingPosC' LABOR
REQUIREMENT TO 'lbrMechanics' 4
…
CREATE BUFFER CLASS 'bfrPosA'
SET 'bfrPosA' COLOR TO $YELLOW
LOCATE ELEMENT 'bfrPosA_1' AT -44,287,0
ROTATE ELEMENT 'bfrPosA_1' to 0,0,180
SET 'bfrPosA' LABOR CONTROLLER TO ELEMENT
'LbrCtrlHiSpDolly_1'
```

```
SET 'bfrPosA' NUMBER OF STACK POINTS TO 4
LOCATE STACK POINT 1 OF 'bfrPosA' AT COORD -
8,1,5.8,-4.5,0,30
LOCATE STACK POINT 2 OF 'bfrPosA' AT COORD -
8,65,5.8,4.5,0,-30
LOCATE STACK POINT 3 OF 'bfrPosA' AT COORD -
8,1,0,-4.5,0,30
LOCATE STACK POINT 4 OF 'bfrPosA' AT COORD -
8,65,0,4.5,0,-30
SET 'bfrPosA' NUMBER OF LABOR POINT TO 2
LOCATE LABOR POINT 1 OF 'bfrPosA' AT COORD -
8,1,0,-4.5,0,30
LOCATE LABOR POINT 2 OF 'bfrPosA' AT COORD -
8,65,0,4.5,0,-30
…
```

The model integration approach in this study explores simulation software capability aspects more than possible methods to collaborate amongst users by using web technologies (Kehris 2002).

5 APPLICATION OF THE XML SPECIFICATION

The need and benefit of simulating these potential wing line processes is partially controlled by some of the limitations of this facility. As shown in the simulation model screen copy (figure 3), this facility has space and resource constraints. Resources can be skilled mechanics, specialized fixtures, tools, stands, and/or material-handling equipment.

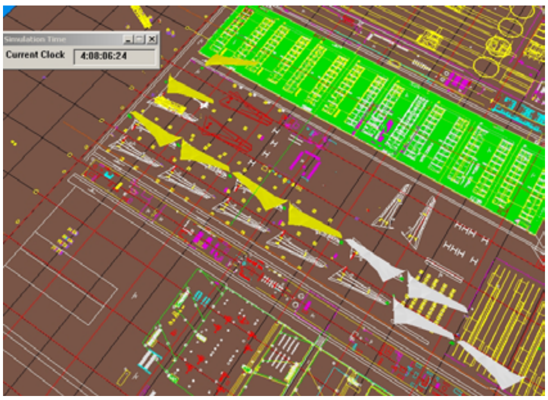

Figure 3: The Case Study Model

Furthermore, some of the proposed processes already exist in the subject facility. Transition logistics and integration methods that exist between the existing wing line and possible future processes are some of the other reasons for this modeling exercise.

BCA MR&D has been conducting discrete event simulation modeling projects without a standard specification that is equivalent to the depth and breadth of the XML-based document proposed by the NIST. This case study is the first attempt to integrate the NIST's generic specifications to a specific MR&D simulation project. This new method of integration has great potential for improving efficiencies in BCA MR&D upon conducting discrete event simulation projects in the future.

The entity types and their attributes in the QUEST simulation model are listed in section 3. Data structures of the NIST standard are listed in section 2. The XML-based Shop Data File is a generalized specification which encompasses more data types and groups than the scope of this case study simulation model. The BCL codes in this modeling approach are derived from the Shop Data File. "Code generation techniques using technologies such as XSLT are playing an increasingly important part in software projects …" (Ashley 2003). A partial example of the Shop Data File is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
    <shop-data identifier="1" type="One"
    number="shop data" mlns:xsi=
    http://www.w3.org/2001/XMLSchema-instance
    xsi:noNamespaceSchemaLocation =
    "D:\1Rf10664D\MAARS2003\WSC2003\SimulationPra
    cticeCaseStudy\process.xsd">
    …
    <process-plans type="process" identi-
    fier="1000" number="Barnhart">
    <process-plan type="process" identi-
    fier="1001" number="timesheet">
    <name>ProcRHwingPosC</name>
    <operation-sheets>
    <operation-sheet type="1" identifier="2001"
    number="labor requirement">
    …
    <plan-definition>
    …
    <resources-required>
    …
    <employees-required>
    <employee-required>
    <minimum-employees>4</minimum-employees>
    <maximum-employees>6</maximum-employees>
    …
    </employee-required>
    </employees-required>
    </resources-required>
    </plan-definition>
    </operation-sheet>
    </operation-sheets>
    …
    <machine-programs>
    <machine-program type="DriveMatic" identi-
    fier="4001" number="mach">
    <estimated-durations>28800</estimated-
    durations>
    </machine-program>
    </machine-programs>
    </process-plan>
    </process-plans>
    …
</shop-data>
```

"Reuse is the most compelling feature of the XML assembly line because it saves so much time." (Nicholson 1999). The application of the XML specification, in this case, successfully incubated a BCL file for the QUEST model. The same specification in the Shop Data File can be reused with a different set of attribute data to generate another BCL file for scenario iterations.

6 MODELING RESULTS

Fundamental findings in this discrete event simulation modeling study are machine utilizations, labor team utilizations, part resident time, crane utilizations, dolly utilizations, and system takt time-driven process performances.

Entity class utilization related modeling results are normally not difficult to collect. QUEST, as with many other discrete event simulation softwares on the market, has built-in functions that provide individual entity and group utilization summaries. Contrarily, it is a rather lengthy task to derive multiple specific process parameters based on different system takt time targets.

One of the desirable modeling results is the ability to change important attributes of certain elements in the model, and to observe the impact upon the system performance. Some of the important attributes in each of the thirty plus processes are:

- Process time,
- Labor requirements, labor class type and labor quantity,
- Part requirements, and
- Transportation requirements.

Here is an example XSLT document that addresses some of the mentioned attributes:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl =
"http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsl:template match="/">
  <html>
   …
  <head />
   …
  <body>
     <xsl:apply-templates />
      …
  </body>
  </html>
</xsl:template>
 …
<xsl:template match=
"estimated-durations">28800</xsl:template>
 …
<xsl:template match=
"maximum-employees">6</xsl:template>
 …
<xsl:template match=
"minimum-employees">4</xsl:template>
 …
<xsl:template match=
"name">ProcRHwingPosC</xsl:template>
 …
</xsl:stylesheet>
```

The matching XML statements that this XSLT pulled from the NIST xml specifications are shown in the previous section, 5. The main matching statements of the BCL document are listed at the end of section 4.

Influences from an XML specification change, to the capture of changes in a stylesheet, followed by the initiations of a modified BCL file that changed the simulation model, demonstrated an innovative methodology in simulation modeling practices. This exercise shares some common concepts with an overview of simulation-based shop floor control (Son et al. 2002) where process plans and external production orders are read into a common system. Simulation models are then generated automatically via external data.

The proposed NIST XML-based simulation specifications raised the possibility for improving simulation modeling methods. Some additional modeling scenarios that can be exercised by this same method are:

- varying the number of positions bases on the system takt time,
- dynamic allocation of labor classes among different positions,
- further process break-downs within each position,
- finite dynamic system scheduling links with more active part, crane, and labor movements, and
- optimizing among processes and resources by using the same set of XML-based data (Lu and Qiao 2003).

7 CONCLUSIONS

This case study successfully applied the proposed NIST simulation specifications in XML to a typical BCA MR&D simulation project using the QUEST software. Positive results from this new simulation modeling technique were increased efficiency in modeling iterations, creating a form of standard simulation data exchange, and used an innovative method for executing simulation model constructions.

Additional related information can be found in many of the previous Winter Simulation Conference proceedings. Areas of future work exist with: a graphical user interface for collecting simulation modeling data with automatic match up to the standard specification, expanding the XML specifications for business processes in additional to manufacturing processes, strengthening the parsing capabilities for extracting meaningful information from the specification into individualized formats for each given simulation software package, and expanding the same methodology to mass customization of manufacturing-related simulation practices with the application of the Petri Net (Qiao et al. 2002).

ACKNOWLEDGMENTS

Authors of this paper would like to express their sincere acknowledgments to Tina Lee, Gordon Shao, and Frank Riddick of the NIST for their technical contribution to the XML Shop Data File; to Swee Leong of the NIST for his leadership in the Simulation Standard Consortium; and to Wayne Hixson of Boeing for his tremendous support in this joint endeavor. The NIST portion of the project is funded by NIST's System Integration of Manufacturing

Application Program and the Software Engineering Institute's Technology Insertion Demonstration and Evaluation Program. No approval or endorsement of any commercial product by both NIST and The Boeing Company is intended or implied.

REFERENCES

Ashley, P. 2003. XSLT Simplify Development and Maintenance of Microsoft .NET Projects with Code Generation Techniques. *MSDN Magazine*. Vol. 18, No. 8, 60 – 71.

DELMIA Corporation. 2002a. *QUEST® Batch Control Language Reference Manual*. Troy, MI: DELMIA Corp.

DELMIA Corporation. 2002b. *QUEST® Simulation Control Language Reference Manual*. Troy, MI: DELMIA Corp.

Kehris, E., and Doulgeri, Z. 2002. Improving Simulation Project Efficiency Using Web Technology. *Simulation*, Vol. 78, Issue 9, 568-579.

Lu, R. F., and Qiao, G. 2003. Simulation Modeling to Optimize Stochastic Manufacturing Processes and Resources by a Dynamic Monte Carlo Method. In *Proceedings of the 2003 Industrial Engineering Research Conference*. Available online via `<http://www.crystalball.com/articles/lu-qiao.html>` [accessed June 30, 2003].

McLean, C., Jones, A., Lee, T., and Riddick, F. 2002. An Architecture For A Generic Data-Driven Machine Shop Simulator. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1108–1116, vol. 2.

McLean, C., Lee, T., Shao, G., and Riddick, F. 2002. Shop Data Model and Interface Specification. Distributed in *the 2002 Winter Simulation Conference*.

McLean, C., and Leong, S. 2002. A Framework For Standard Modular Simulation. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1613–1620, vol. 2.

McLean, C., Leong, S., Harrell, C., Zimmerman, P., and Lu, R. 2003. Simulation Standards: Current Status, Needs, and Future Directions. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice.

Means, W. S. 2002. *Advanced XML*. Englewood, CO: ITCourseware, LLC.

Nicholson, S. 1999. The XML Assembly Line: Better Living Through Reuse. *GCA XML Europe '99*. 397 – 403.

Qiao, G., McLean, C., and Riddick, F. 2002. Simulation System Modeling for Mass Customization Manufacturing. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 2031 – 2036, vol. 2.

Quin, L. 2003. Extensible Markup Language (XML). Available online via `<www.w3.org/XML/>` [accessed June 30, 2003].

Raab, R., and Wright, T. 2002. *Fundamentals of XML*. Englewood, CO: ITCourseware, LLC.

Son, Y. J., Joshi, S. B., Wysk, R. A., and Smith, J. S. 2002. Simulation-Based Shop Floor Control. *SME Journal of Manufacturing Systems*. Vol. 21, No. 5, 380 – 394.

Walsh, Norman. 1998. What is XML? Available online via `<www.xml.com/pub/a/98/10/guide1.html#AEN58>` [accessed June 30, 2003].

AUTHOR BIOGRAPHIES

ROBERTO F. LU is an Associate Technical Fellow with the Boeing Commercial Airplane Manufacturing Research and Development group. He has been conducting job-shop, continuous flow, and large-scale system integration related simulation modeling projects as well as researching process optimization. He has presented his work at professional conferences, focusing upon discrete event simulation modeling and analytical process optimization. He received his bachelor's degree in materials science from Feng Chia University in Taiwan, a master's degree in mechanical engineering from Marquette University in Wisconsin, and another master's degree in industrial and systems engineering from Virginia Polytechnic Institute and State University in Virginia. He is a licensed and registered professional engineer. His e-mail address is: `<roberto.f.lu@boeing.com>`.

GUIXIU QIAO is a researcher in the Manufacturing Simulation and Modeling Group in the U.S. National Institute of Standard and Technology (NIST) Manufacturing System Integration Division. Before NIST, she worked as a postdoctoral researcher in the automation department of Tsinghua University in Beijing China. She received her Ph.D. degree from the Robot Research Institute of the Harbin Institute of Technology in 1999. Her research interests are mainly in virtual manufacturing, virtual reality, computer graphics, simulation integration, system control, and optimization. Her email addresses are `<qiao@cme.nist.gov>` and `<qiaogx@hotmail.com>`.

CHARLES MCLEAN is a computer scientist and Program Manager of the Manufacturing Simulation and Visualization Program at NIST. He also leads the Manufacturing Simulation and Modeling Group. 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 on Information Processing (IFIP) Working Group on Production Management Systems (WG 5.7). He is also the NIST representative on the Department of Defense Advanced Manufacturing Enterprise Subpanel. He holds an MS in Information Engineering from the University of Illinois at Chicago and a BA from Cornell University. His e-mail address is <mclean@cme.nist.gov>.