DATA EXCHANGE FOR MACHINE SHOP SIMULATION

Y. Tina Lee
Yan Luo

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

ABSTRACT

Manufacturing simulation systems normally provide interfaces for data import and export using proprietary formats. Dependent data interfaces limit simulation applications in the manufacturing industry. This paper describes a mechanism for transferring data between a traditional database and XML files, which represent a machine shop data model developed at the National Institute of Standards and Technology (NIST). The data model is described by XML (eXtensible Markup Language) and UML (Unified Modeling Language). The data transfer mechanism is based on Document Object Model (DOM), XML Path Language (XPath), and Open Database Connectivity (ODBC) database engines. A prototype implementation is also described in this paper.

1 BACKGROUND

Today's industry is faced with global competition. Simulation technology can reduce cost, save time, and improve quality. A manufacturing system is composed of a large number of distinct processes that all influence product cost, product quality, and productivity. The interactions among these various facets of a manufacturing system are complex, and decisions made concerning one aspect have ramifications that extend to the others (Stoll 1991). The need for an integrated and automated information system has emerged as a result of the reorganization and expansion of companies when a factory wants to move towards computer integrated manufacturing (Ricardo 1999). Information models play a key role to integrate information and share data in industry. Information integration and sharing require a united and neutral information model (Luo 2000).

Neutral information models provide a foundation for data representation, transfer, and integration. Manufacturing enterprises use several tools to build such models including IDEF1x (Integrated Computer Aided Manufacturing Definition Language 1 Extended) (Appleton 1985),

EXPRESS (ISO 10303-11) (ISO 1994), UML (Unified Modeling Language) (URL 2005), and XML (eXtensible Markup Language) (W3C 2005). IDEF1x is a formal graphical language for building relational data models, which was developed by the U. S. Air Force and is more popular in the USA. EXPRESS was designed to meet the needs of ISO 10303, informally known as the STandard for the Exchange of Product model data (STEP) (SC4Oline 2005). It has also been used in a variety of other "large scale" modeling applications. UML is a graphic representation for software systems' artifacts, but it is also useful for database design (URL 2005). XML is a simple, flexible text format derived from the Standard Generalized Markup Language (SGML, ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, it is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere (W3C 2005). XML helps make information exchange among software applications in a globally distributed computing environment possible.

McLean 2005 describes an interface information model, developed at NIST, to represent machine shop data for manufacturing simulation.

Database management systems such as the Structured Query Language (SQL) Server, Oracle, DB2, and Access, have been used to manage enterprise data for many years. Nevertheless, custom translators are needed to transfer data between an existing database system and an XML representation of that data. This paper focuses on the machine shop data model, its representation in XML, the machine shop database, and the mechanism to data transfer between them.

1.1 Machine shop data model

NIST's machine shop data model was developed to support both NIST's System Integration of Manufacturing Application (SIMA) (Carlisle 2001) program and the Software Engineering Institute's Technology Insertion Demonstration

and Evaluation (TIDE) program (McLean 2002). The model is presented in both graphical form, in UML, and textual form, in XML. The model has been developed with two goals in mind: (1) to support the integration software applications at pilot facility, the Kurt L. Lesker Company's machine shop (Lesker 2005), and (2) as a standard data interface between manufacturing simulators and other manufacturing software applications. The model contains twenty major entities. They are: 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.

1.2 Database model

Using Microsoft Access, a database model has been developed to map onto the XML model's entities (Lee 2003). The database provides the possibility of data sharing in a distributed simulation environment. There were three major objectives of the database implementation:

- to demonstrate the feasibility of the information model
- to develop a pilot database system and then to migrate to a large database management system
- to support the integration of manufacturing applications and simulations used in machine shops

The database contains a set of relational tables that can be represented in a tree-shape structure. *Shop-data* is the database model's very top level, thus the model's first level relationship expresses the connections between entity *shop-data* and the other major entities. About 500 tables and relationships have been developed.
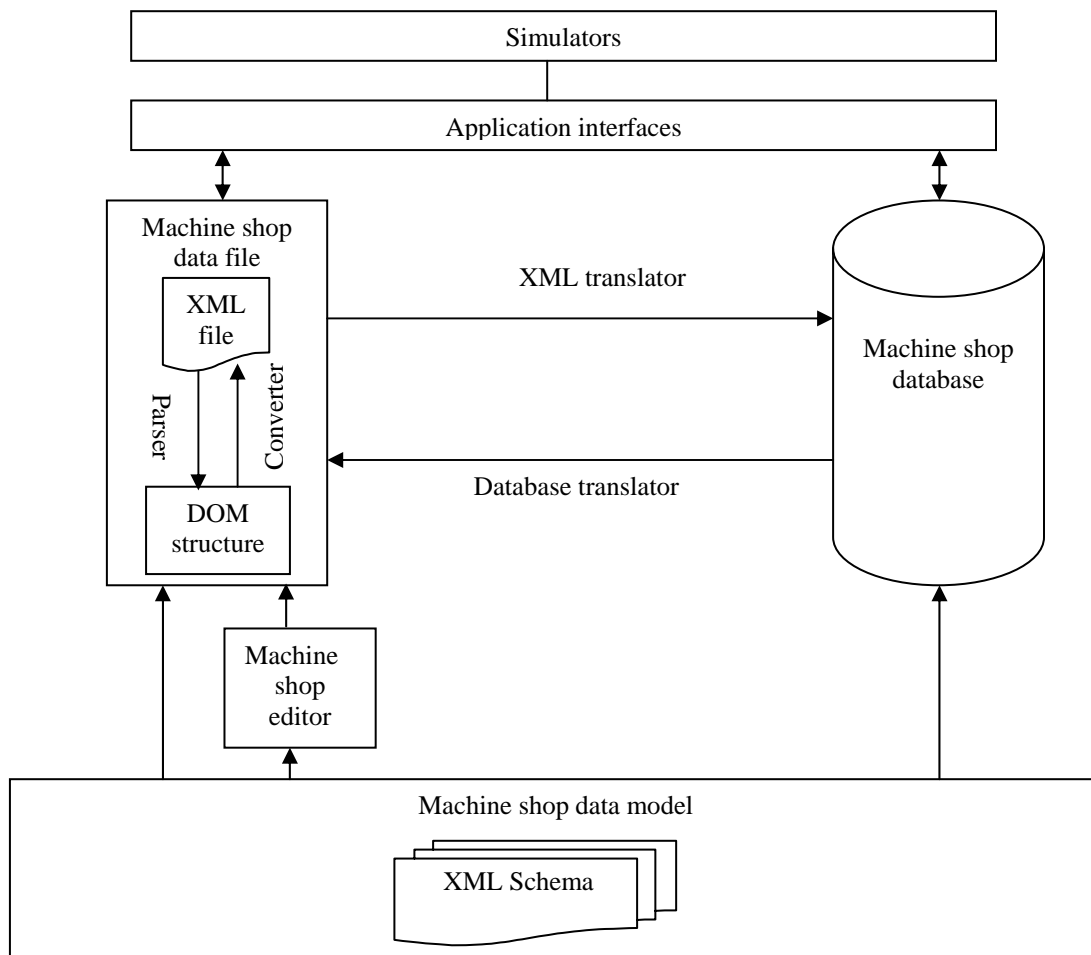

Figure 1: Data transfer based on the machine shop data model

2 DATA TRANSFER FRAMEWORK

Neutral data formats are often necessary when exchanging data between simulation models of different systems. This section describes the approach to develop data translators and data interfaces that support machine shop manufacturing simulation. The implementation is based on: the Open Database Connectivity (ODBC) (MSDN 2005) for database access, the Document Object Model (DOM) (W3C 2005) for XML file operation, and the XML Path Language (XPath) (W3C 2005) for querying nodes.

NIST proposed an architecture for a generic, data-driven, machine shop simulator (McLean 2002). The architecture includes a neutral shop data file, an XML data processor, a system supervisor and reporter, a machine shop emulator, a discrete event simulator, and a user interface system. The machine shop data model is a key factor in integrating the generic machine shop simulator effectively and efficiently. The data model is now being transferred into schemas using an XML schema language (van der Vlist 2002).

Figure 1 shows how the data transfer is accomplished. Simulators provide the support for the manufacturing activities such as design, process, manufacturing, assembly, and production management. Simulation systems can import/export simulation data from/to data files/databases using their data interfaces. A machine shop database and an XML machine shop data file are built based on the machine shop data model. The XML parser and DOM converter, XML translator, and database translator are custom-built software platforms. An XML parser converts from XML to a structural, in-memory representation. The DOM converter serializes and reformats the DOM structure into an XML file. The XML and database translators allow data to be converted between a database structure and an XML file format respectively. The machine shop editor can be used to generate an XML file via the user-friendly interface. The data model/XML schema serves as a neutral data format for representing and exchanging machine shop data.

3 TRANSLATORS

This section describes the XML translator and database translator that are identified in Figure 1. The XML translator is used to transfer data from an XML file to a machine shop database, while the database translator generates an XML instance file from a database.
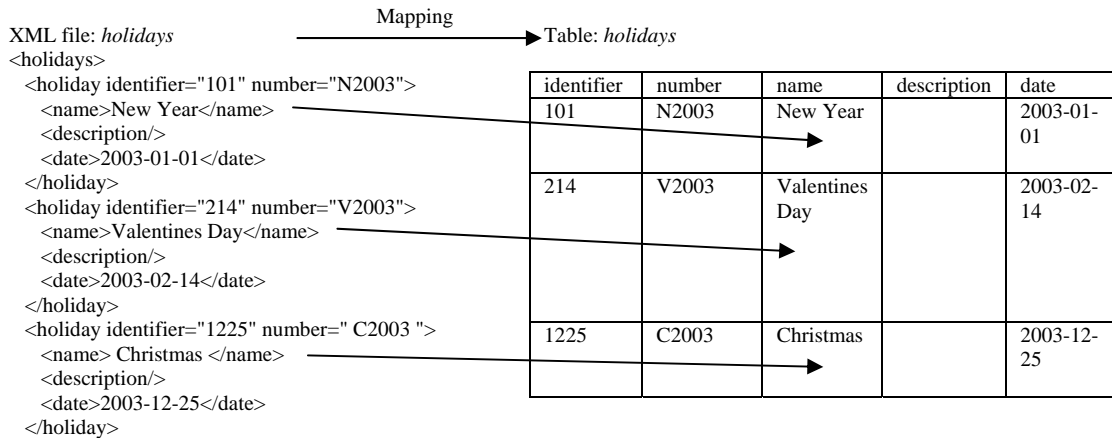
XML file: *holidays*   Mapping →   Table: *holidays*

```
<holidays>
  <holiday identifier="101" number="N2003">
    <name>New Year</name>
    <description/>
    <date>2003-01-01</date>
  </holiday>
  <holiday identifier="214" number="V2003">
    <name>Valentines Day</name>
    <description/>
    <date>2003-02-14</date>
  </holiday>
  <holiday identifier="1225" number=" C2003 ">
    <name> Christmas </name>
    <description/>
    <date>2003-12-25</date>
  </holiday>
```

identifier	number	name	description	date
101	N2003	New Year		2003-01-01
214	V2003	Valentines Day		2003-02-14
1225	C2003	Christmas		2003-12-25

Figure 2: Extracting data from an XML file document

3.1 XML translator

In this section we describe the algorithm to transfer data from an XML file to a database. The XML file has been validated with the machine shop schema before input into the XML translator. The database contains a collection of relational tables, which contain records that are generated from the XML data file. The XML file contains the machine shop data. Each XML element is extracted and mapped to a machine shop database table. The extracted information will be used to generate records of table mapping to element. Figure 2 presents a sample XML file containing data *name* in the element *holiday*, e.g., "New Year," "Valentines Day," and "Christmas," these data will be imported into the table *holidays* (*name*). The flowchart about importing data to machine shop database from XML file is presented in Figure 3.
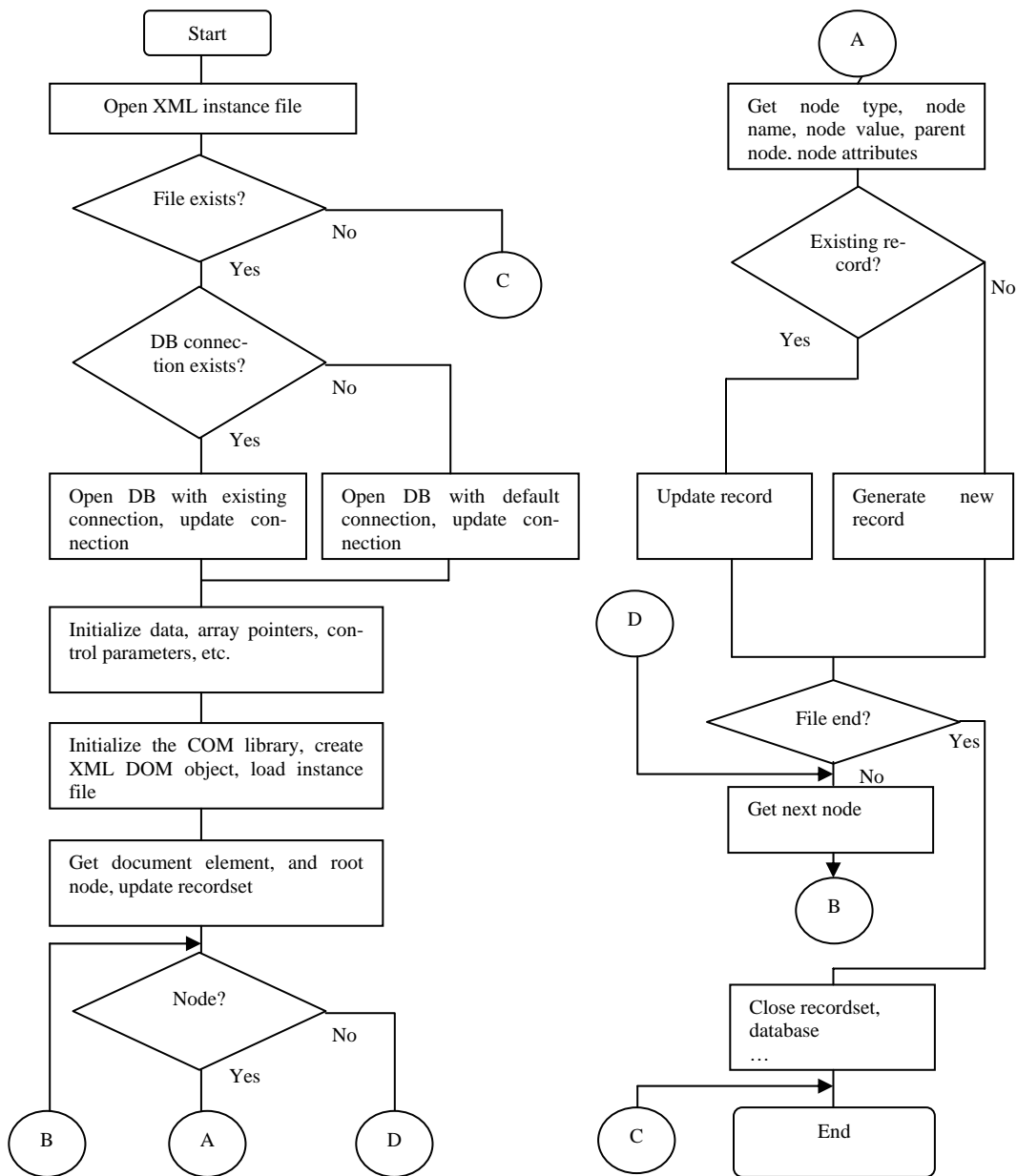
Figure 3: Flowchart for data import

3.2 Database translator

We now discuss the database translator that exports data from a machine shop database to an XML file. The translator can generate the entire database, a selected root element, and responses to a query.

- *Entire Database*: As discussed above, the machine shop database is based on the machine shop data model. The database is a tree-shaped structure, the top level is the element *shop-data.* In this module, *shop-data* is considered as the root element. The algorithm will query all the records in every table of the database and hence generate an XML document that represents the entire database information.

- *Selected root element*: This module provides an algorithm to generate an XML file based on a root element selected. The algorithm queries the related records of tables that are referenced by the root element directly or indirectly. Direct reference means that the root element references other elements directly, e.g., the existent relationship between tables by fields. Indirect reference means that the reference relationship exists by the branch's elements. If the root element selected contains more than one record, one record will be selected as the root element record.

- *Data Query*: SQL provides functions to support database query. This module provides an algorithm to generate an XML file using SQL statements.

4 DATA MODEL IMPLEMENTATION

Discrete-event simulation tools offer considerable flexibility when used to model the behavior of manufacturing systems. These models can be built using provided Graphical User Interfaces (GUI) and some common programming languages. Simulation data can sometimes be imported from another simulation systems through exchange files or databases. These import activities are often performed through internal and/or external interfaces. Internal interfaces allow data be imported/exported with functions or data formats provided by the simulation system. External interfaces rely on standards, neutral data formats, or widely accepted data formats.

To demonstrate the data exchange role of the machine shop information model, prototype implementations are being performed at NIST. Different simulation systems use different modeling constructs and distribution methods. While demonstrating data exchange, we used only data related to machine shop operations; we ignored those data elements defined for a particular simulation system.

A prototype based on the Rockwell Software's Arena, Version 5.0 (Rockwell 2005) is introduced here. Since the format of the Arena database is proprietary, we developed two internal ActiveX dynamic link libraries (smExportToDatabase.dll and smImportFromDatabase.dll). We also developed a set of data process utilities to support data transfer between the Arena model database and XML instance document. These utilities include those translators described in section 3 and the following four modules:

- *Generating Arena model database*: This module opens an XML data document, which is previously validated by the machine shop schemas, and loads XML document data into the Arena model database.

- *Generating simulation model*: This module reads the Arena model database, and dynamically generates a simulation model in Arena.

- *Generating Arena model database*: This module exports simulation model data to an Arena model database.

- *Generating XML document*: This module generates an XML data document from an Arena model database.
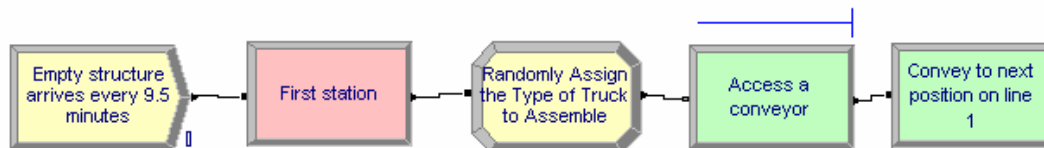

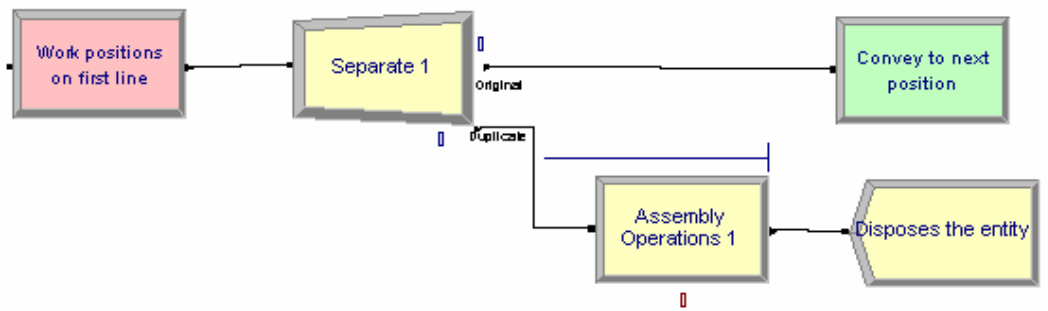Figure 5: The *arrival* process of a truck assembly

Figure 6: The *assembly-line-1* process of a truck assembly

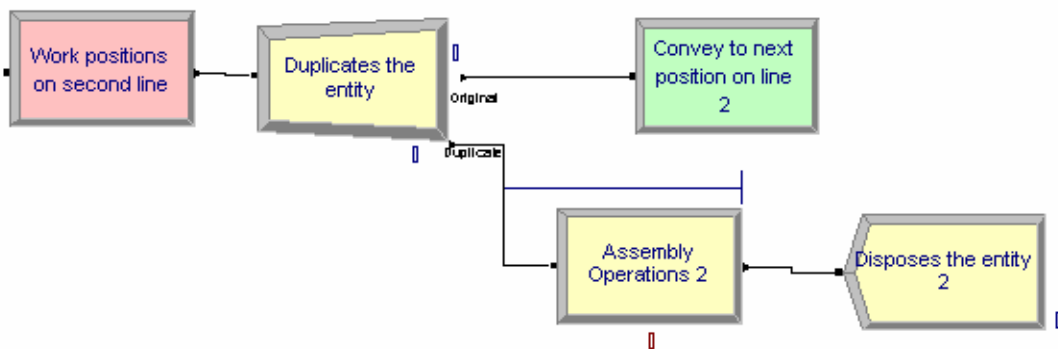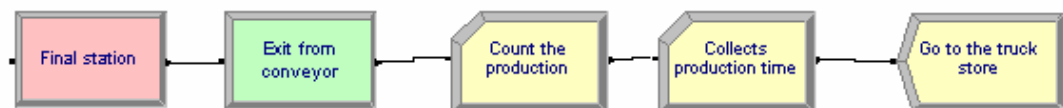
Figure 7: The *turn* process of a truck assembly

Figure 8: The *assembly-line-2* process of a truck assembly

Figure 9: The *exit* process of a truck assembly

To demonstrate these capabilities, we selected the truck assembly line model from the Arena sample models library. New truck chassis enters the line at a constant rate. They are conveyed down the line from one work position where five processes are performed: *arrival, assembly-line-1, turn, assembly-line-2*, and *exit*. Each process has two to fourteen activities. The *arrival* process executes every 9.5 minutes. *The assembly-line-1* process performs operations related to axes, air tanks, etc. The *turn* process performs turning operations at two turning stations. *The assembly-line-2* performs operations related to bumper, radiator, engine, etc. The *exit* process completes assembly activities by exiting from conveyor, delivery to truck store, and collecting statistics data. An XML instance document, which contains *parts, work, resources, operation-definitions, schedules, units-of-measurement,* and *process-plans* is manually generated to support this truck assembly line operation (see Figure 4). The XML document is then loaded into an Arena model database using the customer-built Arena utilities mentioned previously. As a result, an Arena simulation model can be generated dynamically from the model database using internal ActiveX dynamic link libraries. Figures 5-9 present screen captures of each process from the described Arena simulation model.

5   CONCLUSION

The paper presents a mechanism to transfer data between a machine shop database and XML files that are based on the machine shop data model. The data model will continue to evolve based on the experience and feedback from others involved in this effort. The machine shop information model was recently transferred to CMSD PDG of the SISO for standardization. The standard is for the modeling and simulation community in the manufacturing industry.

REFERENCES

Appleton Company, Inc. 1985. "Integrated information support system: Information Modeling Manual: IDEF1-Extended (IDEF1X)," ICAM Project Priority 6201, Subcontract #013-078846, USAF Prime Contract #F33615-80-C-5155, Wright-Patterson Air Force Base, OH.

Carlisle, M., and J. E. Fowler. 2001. "Systems Integration for Manufacturing Applications Biennial Report," Fiscal Years, NISTIR 6721. National Institute of Standards and Technology, Gaithersburg, MD.

ISO 10303-11. 1994(E). "Industrial Automation Systems and Integration-Product Data Representation and Exchange - Part 11: The EXPRESS language Reference Manual."

Lee, Y. T., and Y. Luo. 2003. "A Database Design for the Machine Shop Information Model," NISTIR 7077, National Institute of Standards and Technology, Gaithersburg, MD.

Lesker Company. 2005. Kurt J. Lesker Company [online]. Available via <www.lesker.com/newweb/index.cfm> [accessed June 3, 2005].

Luo, Y. 2000. "Injection molding product application activities models," International Journal of Advance Manufacturing Technology, 16(4), 285-288.

McLean, C.; A. Jones; Y. T. Lee and F. Riddick. 2002. "An Architecture for a Generic Data-Driven Machine Shop Simulation," Proceedings of the Winter Simulation Conference, San Diego, CA.

McLean, C.; Y. T. Lee; G. Shao and F. Riddick. 2005. "Shop Data Model and Interface Specification," NISTIR 7198. National Institute of Standards and Technology, Gaithersburg, MD.

MSDN. 2005. ODBC Overview [online] <msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcintroduction_to_odbc.asp> [accessed June 3, 2005].

Ricardo, J. G.; C.S.Pedro; B.P.Joao and S.G.Adolfo. 1999. "Integrating manufacturing systems using ISO 10303(STEP): an overview of UNINOVA projects," International Journal of Computer Applications in Technologies, 12(1), 39-45.

Rockwell Automation. 2005. Rrockwell Software [online]. <www.software.rockwell.com/index.cfm> [accessed June 3, 2005].

SC4Online. 2005. STEP Overview[online]. Available via <www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/STEP_(10303)> [accessed June 3, 2005].

SISO (Simulation Interoperability Standards Organization). 2005. Welcome to SISO [online]. <www.sisostds.org> [accessed June 3, 2005].

Stoll, H. W. 1991. "Design for manufacture: An overview," in: J.Corbett, M.Doner, J.Meleka and C.Pym, eds., Design for Manufacture: Strategies, Principles, and Techniques, Addison – Wesley Publishing Company, New York, NY, 107-129.

URL Resource Center. 2005. Unified Modeling Language [online]. Available via <www.rational.com/uml> [accessed June 3, 2005].

van der Vlist, E., 2002. XML Schema, Sebastopol, CA, O'Reilly & Associates, Inc.

W3C World Wide Web. 2005. XML [online]. Available via <www.w3c.org> [accessed June 3, 2005].

AUTHOR BIOGRAPHIES

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>.

Yan Luo is a guest researcher in the Manufacturing Simulation and Modeling Group at NIST. Prior to joining NIST, he was a research fellow of AvH at TU Darmstadt, Germany. He received his PH. D. from Shanghai Jiaotong University, China. His areas of interest include CAD/CAM, manufacturing automation, software technology, mechanical engineering, and manufacturing simulation and modeling. His e-mail address is <luoyan@cme.nist.gov>.