Application of machine shop data model in manufacturing simulation

Y. Luo and Y. T. Lee
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD, USA

Abstract - Simulation technology holds tremendous promise for the manufacturing industry. A machine shop data model has been developed at the National Institute of Standards and Technology (NIST) to represent neutral machine shop data for supporting machine shop manufacturing simulation [1]. This paper discusses the data model and its application in manufacturing simulation. The paper also proposes an application framework that integrates the simulation model, machine shop database, and machine shop XML (Extensible Markup Language) data document. A prototype system is developed to present the application of machine shop data model using the proposed framework.

Keywords: information model, manufacturing simulation, simulation model

Disclaimer: *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.*

1.0 Introduction

Manufacturing industry is characterized by increasing pressure on product development time and cost. Simulation technology holds tremendous promise for reducing costs, improving quality, and shortening the time-to-market for manufactured goods [2]. Standard data interfaces, among simulation modules or between simulation and other software applications, could make information sharing effectively and efficiently, and hence promote the utilization of simulators. An information model, that represents machine shop data and facilitates data sharing among machine shop's manufacturing execution system, scheduling system and simulation system, has been developing at the National Institute of Standards and Technology (NIST) to support both NIST's System Integration of Manufacturing Application (SIMA, [3]) program and the Software Engineering Institute's Technology Insertion Demonstration and Evaluation (TIDE, [4]) program. Information integration and sharing require the supporting of a united and neutral information model [5]. Some modeling languages such as IDEF1-Extended (IDEF1x, [6]), EXPRESS (ISO 10303-11, [7]), the Unified Modeling Language (UML, [8]) and the Extensible Markup Language (XML, [9]) are most frequently used by manufacturing enterprises for information modeling. IDEF1x is a formal graphical language for relational data modeling, the language was developed by the U. S. Air Force and is more popular in the USA. EXPRESS was designed to meet the needs of ISO10303, informally known as the STandard for the Exchange of Product model data (STEP, [10]). It has been used in a variety of other "large scale" modeling applications. UML is a graphic representation for software systems' artifacts; it is also useful for database design. XML is a format for structured documents; it helps make information exchange among a globally distributed computing environment possible. NIST's machine shop data model [1] is developed using UML and XML. The model has been implemented based on the XML model and relational database. This paper focuses on the data model, simulation application, and prototype development.

2.0 Machine shop data model and machine shop database

2.1 Machine shop data model

A Machine shop data model has been developing at NIST as a part of efforts that support the development of standard data interfaces. The model contains twenty major entities. The name and brief description of the major entities are described as follows.

- *Organizations* is used to maintain the organizational structure, contacts and address information for the manufacturing organization and its customers and suppliers.
- *Calendars* identifies the shift schedules that are in effect for a period of time, breaks, and holidays.
- *Resources* describes all the resources that may be assigned to tasks in the shop. The resource types available in the machine shop environment include: stations and machines, cranes, employees, and tool and fixture catalog items.
- *Skill-definitions* lists the skills that an employee may possess and the levels of proficiency associated with those skills.
- *Setup-definitions* typically specifies tool or fixture setups on a machine. Tool setups are typically the tools that are required in the tool magazine. Fixture setups are work holding devices mounted on the machine. Setups may also apply to cranes or stations.
- *Operation-definitions* defines the operations that may be performed at a particular station or group of stations in the shop.
- *Maintenance-definitions* defines preventive or corrective maintenance to be done on machines or other maintained resources.
- *Layout* defines the location of reference points within the shop, area boundaries, paths, resource, and part objects.
- *Parts* provides elements for part specifications, group technology codes, customers, suppliers, as well as links to bills of materials, process plans, drawings, part models and other references.
- *Bills-of-materials* cross-references the parts and quantities required in a hierarchical bill-of-materials structure.
- *Inventory* identifies the instances and locations for parts, materials, tools, and fixtures inventory.
- *Procurements* identifies the external purchases that have been created to satisfy the part inventory and manufacturing requirements.
- *Process-plans* specifies a set of process plans that are associated with production and support activities for a particular part or parts. A process plan has routing sheets and operation sheets that correspond to the job and task level in the work hierarchy.
- *Work* is used to specify a collection of a hierarchy of production orders, jobs, and tasks. It is also used to specify a collection of internal support orders for maintenance activities, inventory picking and tool preparation.
- *Schedules* lists planned assignment or mapping of work to resources and resources to work.
- *Revisions* specifies information about a set of revisions of the subjects. Information included in the element are each revision’s identification, description, date, and creators.
- *Time-sheets* provides a list of individual time sheet elements. A time-sheet is used to log the hours that an employee has worked, the time an employee has taken off from work, and accrual of leave hours.
- *Probability-distributions* specifies distributions that are used to vary processing times, breakdown and repair times, and availability of resources.
- *References describes* the information about reference materials that support or further define the data elements contained within the shop data structure.
- *Units-of-measurement* describes various measurement units used in the file, for example, the measurement may be for distance, speed, mass, time duration, or currency.

2.2 Machine shop database

The database provides the possibility of data sharing in a distributed simulation environment. A database model has been developed to map onto the XML model's entities [15]. The database contains a set of relational tables that can be presented in a tree shape structure. About 500 tables and tables' relationships have been developed. *Shop-data* is the database model's very top level, thus the model's first level relationship expresses the connections between the entity *shop-data* and the major entities.

3.0 Simulation applications

Figure 1 presents a simulation application framework that is based on the machine shop data model. A simulation model can be generated in a simulation system. Simulation data can be imported/exported into/from a simulation system by internal or external interfaces. Internal interface imports/exports data using an internal data structure provided by the system. The external interface uses widely accepted or standard data formats. The paper focuses on external interfaces. Different manufacturing simulation systems use different modeling constructs and distribution methods, system-related simulation data are specifically defined for the system use purpose. Thus in this paper, simulation data are divided into model-related data and system-related data. The converters reformat data exchange between simulation model data format and machine shop data format. Machine shop database and XML instance document are external data sources. The translators are developed to exchange data between the machine shop database and the XML document. The machine shop data model is a key for information integration.

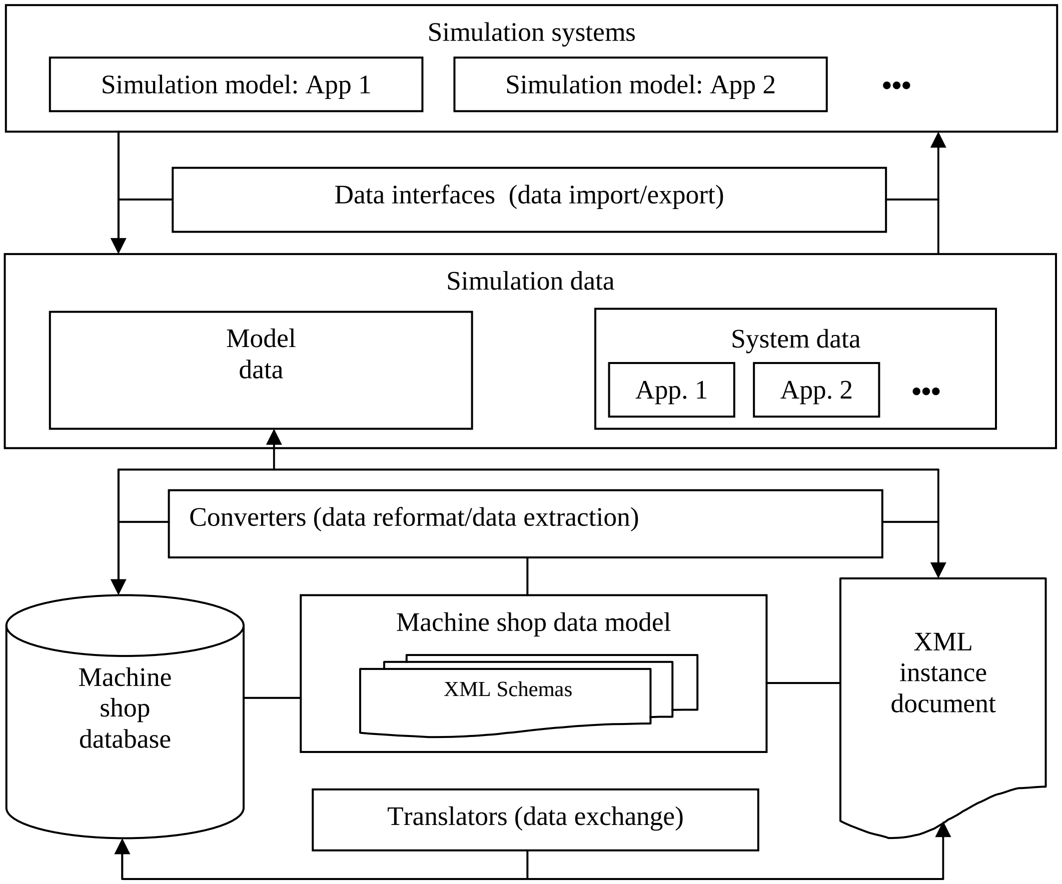


Figure 1: Application framework based on machine shop data model

A prototype system that contains two simulation software environments, Arena (the Rockwell Software Inc., v5.00, [11]) and ProModel (the ProModel Corporation, v6.0.20, [12]), has been implemented. Simulation model data of Arena can be exported/imported from/into an Arena model database. ProModel uses its internal function XSUB() that calls Dynamic Link Library (DLL) to perform external tasks.

3.1 ProModel environment

ProModel is a simulation and animation tool designed to model manufacturing systems of all types quickly and accurately, it is capable of modeling complex systems [14]. A data interface is being developed to import data from an XML document based on the machine shop data model. The purpose of this task is to initialize data of a shipyard simulation model from the XML document. The ProModel's data interface uses an internal function XSUB() to call an external subroutine inside a DLL file. XSUB() [12] is a powerful statement in ProModel. Using XSUB(), we can access the entire functionality of any thirty-two bit Windows programming language such as C, C++, or Pascal. We can use XSUB() for sophisticated file input/output and to make simulations interactive. XSUB() will copy the parameters following the function name to a block of memory, then pass the function a pointer to that block of memory. The DLL executes the external program to load XML document, query data, and return value to XSUB(). XSUB() will send the value to a related cell for simulation model to use. Figure 2 presents the system architecture. All initial data will be imported from the XML document in order to run the shipyard simulation model.

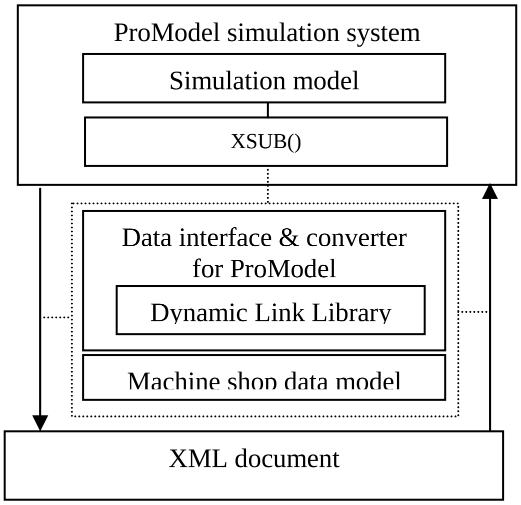


Figure 2: Machine shop data model for ProModel

3.2 Arena environment

Arena is used for various environments from customer service to manufacturing, it can create a basic model, refine model, simulate model, analyze simulation results, and so on [11]. As the Arena data format of simulation model (.DOE) is proprietary, two internal ActiveX dynamic link libraries (smExportToDatabase.dll and smImportFromDatabase.dll) are provided to import/export to/from an Arena model database. A set of data process utilities has been developed to support data transfer between the Arena model database and XML data document. These utilities contain four main 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 an Arena model database.

- Generating simulation model

This module reads an 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 3 presents an architecture for an Arena application. Data transfer between the simulation model and Arena model database is through the internal ActiveX dynamic link libraries. As for data exchange between Arena model database and XML data document is executed using Document Object Model (DOM, [17]), XML Path Language (XPath, [17]), and Open Database Connectivity (ODBC, [16]) database engine.

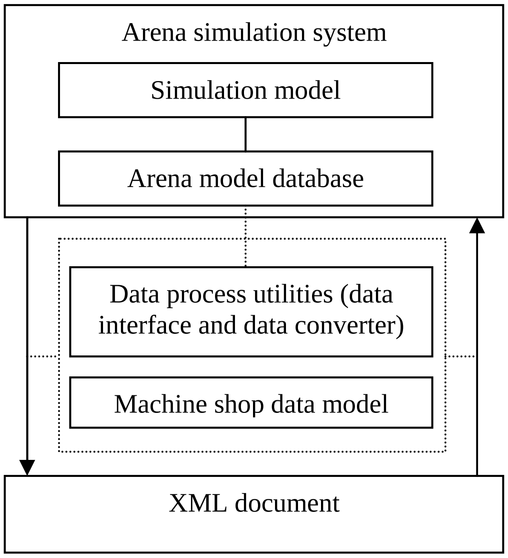


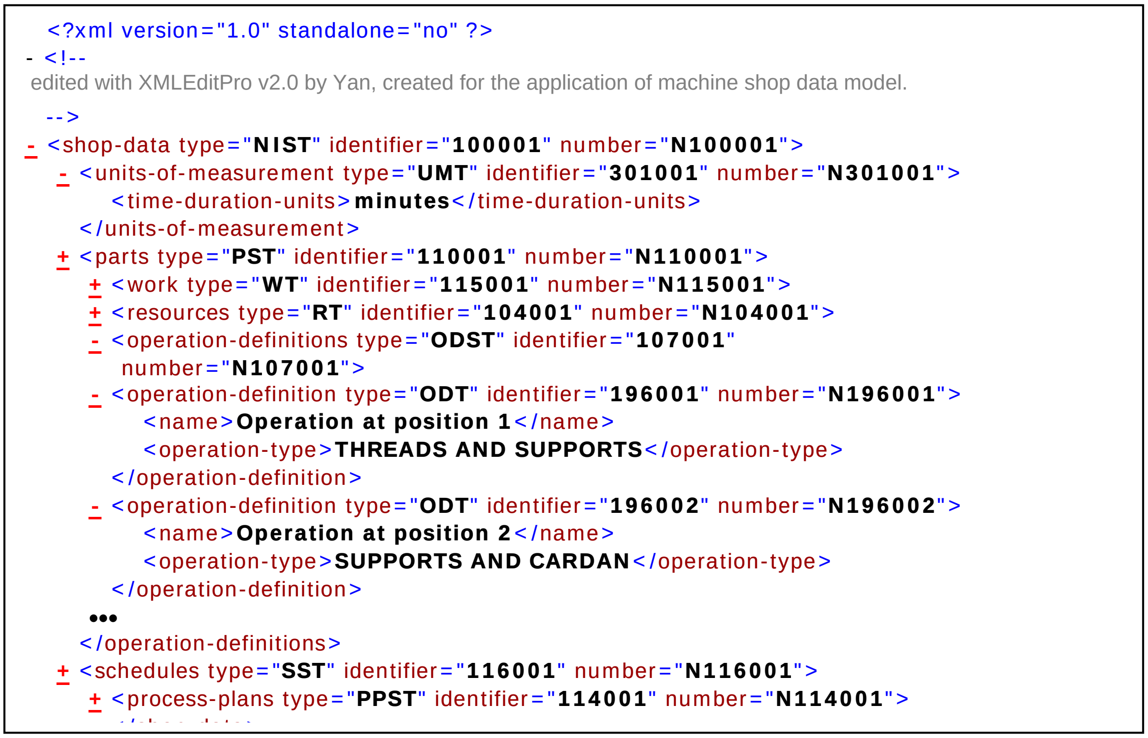
Figure 3: Machine shop data model for Arena

3.3 Arena implementation demonstration

A simulation model is presented here to demonstrate an Arena implementation. A sample model has been selected from the Arena sample models library. The model demonstrates the operations of a truck assembly line. A new truck chassis enters the line every 9.5 minutes. It is then conveyed down the line from work position to work position, whereby each work position adds a part or performs an operation [13].

The assembly line contains the following processes: “entrance,” “assembly line one,” “turns,” “assembly line two,” and “exit.” “entrance” contains two activities: “assign the type of truck to assemble” and “access a conveyor and convey to next position on line one.” “assembly line one” contains ten processes: “threads and supports,” “supports and cardan,” “suspension,” “cardan,” “support and tanks,” “axes,” “air tanks,” “grease,” etc. “turns” contains: “turn chassis I” and “turn chassis II.” “assembly line two” contains: “exhaust and front bumper,” “engine and exchange,” “cage and radiator,” “bumper,” “direction bar,” “electrical connectors,” “torpedo,” “lights and battery,” “tanks and fuel,” “supply and control,” “tires,” “tests,” and “tires fixturing.” “exit” contains: “exit from conveyor,” “count the production,” “collects production time,” and “go to the truck store.”

An XML data document is generated manually as Figure 4. The document is validated using the machine shop schemas. This document consists of *parts*, *work*, *resources*, *operation-definitions*, *schedules*, *units-of-measurement*, and *process-plans* elements. The XML document is converted to an Arena model database, which contains data tables, e.g., "ModuleTables" Table, a Module Table, "RepeatGroupTables" Table, a Repeat Group Table, "ModelLevels" Table, "Submodels" Table, "Connections" Table, "NamedViews" Table, "ProjectParameters" Table, "ReplicationParameters" Table, and "Reports" Table. For each type of module in the model, a table exists in the database that records all instances of that module. These tables are referred to as module tables. If a module's structure includes a repeat group (i.e., a repeatable set of operands), then a separate table for that repeat group exists in the database. These tables are referred to as repeat group tables. The simulation model generated dynamically in Arena from an XML instance document as Figure 5.



```
  <?xml version="1.0" standalone="no" ?>
- <!--
 edited with XMLEditPro v2.0 by Yan, created for the application of machine shop data model.
  -->
- <shop-data type="NIST" identifier="100001" number="N100001">
  - <units-of-measurement type="UMT" identifier="301001" number="N301001">
      <time-duration-units>minutes</time-duration-units>
    </units-of-measurement>
  + <parts type="PST" identifier="110001" number="N110001">
    + <work type="WT" identifier="115001" number="N115001">
    + <resources type="RT" identifier="104001" number="N104001">
    - <operation-definitions type="ODST" identifier="107001"
       number="N107001">
    - <operation-definition type="ODT" identifier="196001" number="N196001">
        <name>Operation at position 1</name>
        <operation-type>THREADS AND SUPPORTS</operation-type>
      </operation-definition>
    - <operation-definition type="ODT" identifier="196002" number="N196002">
        <name>Operation at position 2</name>
        <operation-type>SUPPORTS AND CARDAN</operation-type>
      </operation-definition>
    •••
    </operation-definitions>
  + <schedules type="SST" identifier="116001" number="N116001">
    + <process-plans type="PPST" identifier="114001" number="N114001">
```

Figure 4: The general XML document structure for truck assembly

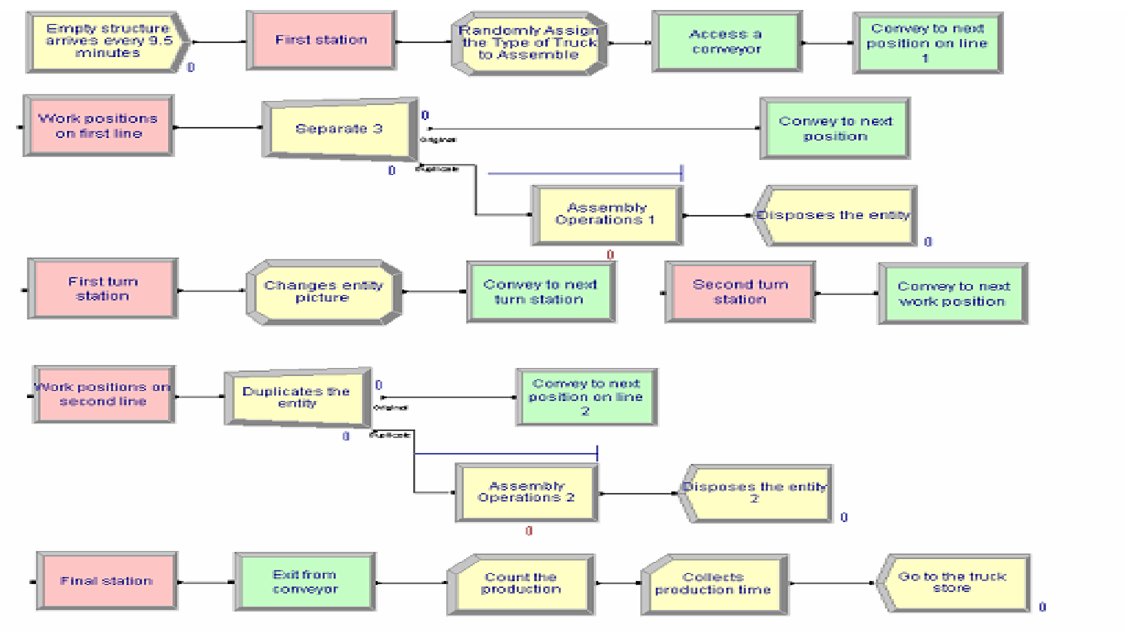


Figure 5: An Arena simulation model generated from an XML instance document

4.0 Conclusions and future work

The paper presents an application framework to integrate simulation model, machine shop data model, machine shop database, and XML instance document. A prototype system is developed to present the application of a machine shop data model using the proposed framework. The simulation model can import/export model data into/from an XML data document and/or a machine shop database. The model has recently been uploaded onto the Simulation Interoperability Standards Organization (SISO, [18]) document library [19] for the standardization process. It will be the strawman of the first standard of the SISO's Core Manufacturing Simulation Data Product Development Group.

5.0 References

[1] C. McLean, Y. T. Lee, G. Shao, and F. Riddick. "Shop data model and interface specification," NISTIR 7198, National Institute of Standards and Technology, Gaithersburg, MD, 2005.

[2] S. Leong and C. McLean. "The Role of Simulation in Strategic Manufacturing," Working Group on Integrated Production, 2001.

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

[4] C. McLean, A. Jones, Y. T. Lee, and F. Riddick, "An Architecture for a Generic Data-Driven Machine Shop Simulator,"Proceedings of the Winter Simulation Conference, 2002.

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

[6] D. Appleton Company, Inc. "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, Wright-Patterson, OH, 1985.

[7] ISO 10303-11. "Industrial Automation Systems and Integration-Product Data Representation and Exchange - Part 11: The EXPRESS language Reference Manual," 1994.

[8] http://www.rational.com/uml/

[9] The World Wide Web Consortium. "Extensible Markup Language (XML) 1.0," http://www.w3.org/TR/REC-xml/, 1998.

[10] ISO 10303-1. "Industrial Automation Systems and Integration-Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles," 1994.

[11] http://www.software.rockwell.com/index.cfm

[12] http://www.promodel.com/support/

[13] Rockwell Software Inc. The Arena Example Models Library, Version 5.00, 2000.

[14] C. R. Harrell and R. N. Price. "Simulation modeling using ProModel technology," Proceedings of the 2003 Winter Simulation Conference, 2003.

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

[16] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcintroduction_to_odbc.asp

[17] http://www.w3c.org/

[18] http://www.sisostds.org/

[19] http://www.sisostds.org/doclib/doclib.cfm?SISO_RID_1005843