AN INTERFACE DATA MODEL SUPPORTING MANUFACTURING SIMULATION

Y. Tina Lee, Yan Luo, and Chuck McLean
National Institute of Standards and Technology
100 Bureau Drive, MS 8260
Gaithersburg, MD 20899-8260, USA
E-mail: leet@cme.nist.gov, luoyan@cme.nist.gov and mclean@cme.nist.gov

KEYWORDS

data exchange, information model, machine shop operations, simulation

ABSTRACT

Simulation technology is underutilized by the manufacturing industry. Standard data interfaces, among simulation modules or between simulation and other software applications, could make information sharing effective and efficient, and hence promote utilization of simulators. An information model that represents machine shop data and facilitates data sharing among the manufacturing execution system, the scheduling system and the simulation system's modules in a machine shop has been under development at the National Institute of Standards and Technology (NIST). A set of relational database tables representing all elements in the information model is being developed. The model will be used to support the integration of software applications at a pilot machine shop located in Pittsburgh, PA. The information modeling project is part of efforts that support the development of standard data interfaces. This paper presents an overview of the information model, the mapping from the information model to relational database tables, and the future project plan.

1 INTRODUCTION

Today's industry is characterized by increasing pressure on development time, manufacturing cost, and product quality. In the manufacturing industry, the 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 between these various facets of a manufacturing system are complex, and decisions made concerning one aspect have ramifications that extend to the others (Stoll 1991). A major cause that an enterprise cannot respond to a sudden change is lack of fundamental thinking about the relationship between management and information in the manufacturing enterprise (Inoue et al. 2000). The information technology is critical to enterprise integration. Formal information modeling languages that describe information requirements unambiguously are an enabling technology that facilitates the development of a large scale, networked, computing environment that behaves consistently and correctly (Lee 1999).

A machine shop data model, as an interface information model, has been under development at NIST to support both NIST's System Integration of Manufacturing Application (SIMA) program (Carlisle and Fowler 2001) and the Software Engineering Institute's Technology Insertion Demonstration and Evaluation (TIDE) program (McLean et al. 2002). SIMA supports NIST projects applying information technologies and standards-based approaches to manufacturing software integration problems. The goal of the TIDE program is to demonstrate the cost-saving and efficiency benefits that small manufacturers can gain by using advanced commercial off-the-shelf software and the information technology. TIDE focuses on helping firms that supply goods and services important to national defense; however, much of the work of TIDE is broadly applicable to small businesses.

Information integration and sharing require the support of a united and neutral information model (Luo 2000). Some languages such as the Integrated Computer Aided Manufacturing (ICAM) Definition Language 1 Extended (IDEF1X) (D. Appleton Company 1985), EXPRESS (ISO 10303-11 1994), Unified Modeling Language (UML) (Object Management Group 2003) and Extensible Markup Language (XML) (The World Wide Web Consortium 2000) are most frequently used by the manufacturing enterprise for information modeling. IDEF1X is a formal graphical language for relational data modeling, developed by the U. S. Air Force, and is more popular in the USA. EXPRESS was designed to meet the needs of the STandard for the Exchange of Product model data (STEP) (ISO 10303-1 1994), it has been used in a variety of other "large-scale" modeling applications.

UML is a graphic representation for artifacts in software systems, and is also useful for database design. XML is a format for structured documents, and it helps make possible information exchange in a globally distributed computing environment. NIST's machine shop data model is developed using UML and XML. The model is currently being implemented using XML documents and relational database tables. XML is chosen for supporting web users while relational databases are commonly used by today's small manufacturers.

2 OVERVIEW OF THE MACHINE SHOP DATA MODEL

NIST's machine shop data model is presented in both the graphical form, in UML, and the textual form, in XML. The model has been developed with two goals in mind: a) the model will be used to support the integration of software applications at a pilot facility, the Kurt L. Lesker Company's machine shop, and b) the model will be promoted as a standard data interface for manufacturing simulators and possibly other software applications. The model contains twenty major elements. The name and brief description of the major elements are described as follows.

- The element *organizations* is used to maintain the organizational structure, contacts, and address information for the manufacturing organization, and its customers and suppliers.
- The element *calendars* identifies the shift schedules that are in effect for a period of time, breaks, and holidays.
- The element *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.
- The element *skill-definitions* lists the skills that an employee may possess and the levels of proficiency associated with those skills.
- The element *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.
- The element *operation-definitions* defines the operations that may be performed at a particular station or group of stations in the shop.
- The element *maintenance-definitions* defines preventive or corrective maintenance to be done on machines or other maintained resources.
- The element *layout* defines the locations of reference points within the shop, area boundaries, paths, resource, and part objects.
- The element *parts* provides attributes for part specifications, group technology codes, customers, suppliers, as well as links to bill of materials, process plans, drawings,
part models, and other references.
- The element *bills-of-materials* cross-references the parts and quantities required in a hierarchical bill-of-material structure.
- The element *inventory* identifies the instances and locations for the part, material, tool and fixture inventory.
- The element *procurements* identifies the external purchases that have been created to satisfy the part inventory and manufacturing requirements.
- The element *process-plans* specifies a set of process plans that are associated with production and support activities for a particular part or parts.
- The element *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.
- The element *schedules* lists the planned assignment or mapping of work to resources and resources to work.
- The element *revisions* specifies information about a set of revisions of the subjects. Information included in *revision* are each revision's identification, a description, the date, the creator, etc.
- The element *time-sheets* provides a list of individual *time-sheet* elements. A *time-sheet* is used to log the hours that an employee works, the time the employee takes off from work, and the accrual of leave hours.
- The element *probability-distributions* specifies the distributions that are used to vary processing time, breakdown and repair time, availability of resources, etc.
- The element *references* is used to express information about reference materials that support or further define the data elements contained within the shop data structure.
- The element *units-of-measurement* describes various measurement units used in the file such as for distance, speed, mass, time duration, currency, etc.

3 THE DATABASE MODEL

A database provides the possibility of data sharing in a distributed simulation environment. A set of relational tables that represent the elements within the machine shop data model and the relationships among elements are being developed.

3.1 Database structure design

A database model is designed to map onto the XML documents. The database contains a set of relational tables presented in a tree shape structure. Table 1 is a design view of the data structure of *shop-data.* The design view contains three types of information presented in three columns: Field, Data Type, and Description. The

"Field" column identifies attributes of the element and special elements that are used by the database system, or the referenced or referencing tables. The "Data Type" column defines the data type format for the field. The "Description" column is reserved for information about the attribute usage, domain, definition (that defines the internal relationship between tables), reference, cardinality relationship, etc. The cardinality relationship specifies how many specific instances of the child element could be related to the parent element. The cardinality relationship may be one to zero or one, one to zero or more, one to one or more, or exactly "n" occurrences, and is presented in the design view as [0: 1], [0: *], [1:*], [n], respectively.

Table 1: Data Structure for *shop-data*

Field	Data Type	Description
INDEX	Number	[1],system use
type	Text	[1],type.type-category = "shop-data"
identifier	Number	[1]
number	Text	[1]
name	Text	[0..1]
description	Text	[0..1]
REFERENCE-KEYS	Text	[0..1],reference-keys.SOURCE
REVISIONS	Text	[0..1],revisions.SOURCE
ORGANIZATIONS	Number	[0..1],organizations. SHOP-DATA
CALENDARS	Number	[0..1],calendars.SHOP-DATA
RESOURCES	Number	[0..1],resources.SHOP-DATA
SKILL-DEFINITIONS	Number	[0..1],skill-definitions. SHOP-DATA
OPERATION-DEFINITIONS	Number	[0..1],setup-definitions. SHOP-DATA
MAINTENANCE-DEFINITIONS	Number	[0..1],maintenance-definitions. SHOP-DATA
LAYOUT	Number	[0..1],layout.SHOP-DATA
PARTS	Number	[0..1],parts.SHOP-DATA
BILL-OF-MATERIALS	Number	[0..1],bill-of-materials. SHOP-DATA
INVENTORY	Number	[0..1],inventory.SHOP-DATA
PROCUREMENTS	Number	[0..1],procurements. SHOP-DATA
PROCESS-PLANS	Number	[0..1],process-plans. SHOP-DATA
WORK	Number	[0..1],work.SHOP-DATA
SCHEDULES	Number	[0..1], schedules.SHOP-DATA
TIME-SHEETS	Number	[0..1],time-sheets. SHOP-DATA
REFERENCES	Number	[0..1],references.SHOP-DATA
PROBABILITY-DISTRIBUTIONS	Number	[0..1],probability-distributions. SHOP-DATA

Shop-data is the model's very top level, and it is defined by a type, an identifier, a number, and optionally a name, a description, references, revisions, units of measurement, organizations, calendars, resources, skill definitions, setup definitions, operation definitions, maintenance definitions, a layout, parts, bills of materials, an inventory, procurements, process plans, work, schedules, time sheets, and probability distributions. Field *INDEX* in Table 1 is a special element used by the system, its instance should be an integer, and there will be exactly one instance for a *shop-data* instance. *Type* is an attribute of *shop-data* and is an enumeration to describe types about *shop-data*. *Name* is another attribute, and there may be zero or one instance for *a shop-data* instance. *ORGANIZATIONS* is an attribute served as a pointer pointing to the table of *organizations*. (The *organizations* table view is omitted due to the page limit.) A *shop-data* instance may have zero or one instance of *ORGANIZATIONS*. *ORGANIZATIONS* in the *shop-data* table points to the field of *shop-data* in the *organizations* table.

3.2 Elements and relational tables

Elements in the XML model are used to describe fundamental features or common features of machine shop data. They are represented by tables in the database model. For illustration, two elements are selected and described here: the element *stock-level-quantities* and element *task*.

Figure 1 is another presentation view about the table/element of *stock-level-quantities. Stock-level-quantities* specifies the quantities of tools, fixtures, materials, or parts for various purposes, and is represented by the following attributes*: on-hand-quantity, allocated-quantity*, *safety-stock-quantity, required-quantity*, *on-order-quantity*, *back-order-quantity*, and *work-in-progress-quantity*.

Figure 1: Element of *stock-level-quantities*

The XML source of *stock-level-quantities is* presented as follows.

```
<stock-level-quantities>
    <on-hand-quantity />
    <allocated-quantity />
    <safety-stock-quantity />
    <required-quantity />
    <on-order-quantity />
    <back-order-quantity />
    <work-in-process-quantity />
</ stock-level-quantities >
```

The element *task*, shown in Figure 2, is a child element of *tasks* and is represented *by task-definition* and *task-status. Task-definition* specifies who the task is for (*customers*), the relative priority of this task (*priority-rating*), critical due dates for the task (*due-dates*), the process plan the task uses *(process-plan-multi-key*), the number of times the task activities need to be repeated (*repetition-count*), parameters used by the task (*parameters*), what products are required (*part-quantities*), resource requirements (*resources-required*), child work items (*child-work-items*), precedence relationships between child work items (*precedent-constraints*), and summary of estimated and actual cost data (*cost summary).* Child work items for a task may be pick orders or tool orders. *Task-status* specifies information about scheduled and actual progress towards completing the task.

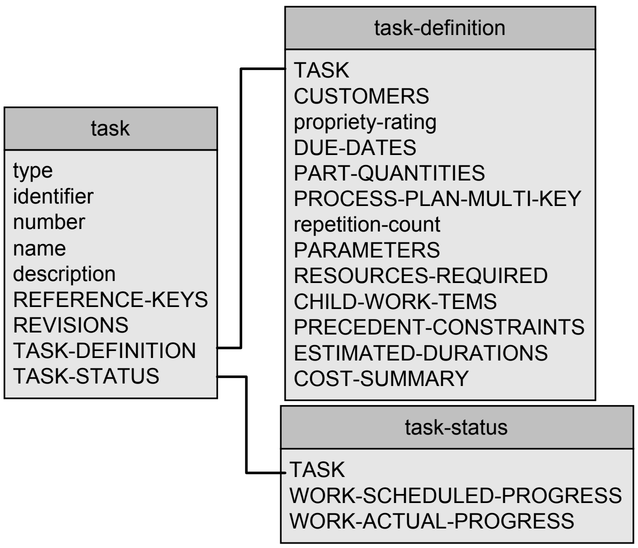

Figure 2: Element of *task*

The XML source of *task* is presented as follows:

```
<task type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <task-definition>
        <customers />
        <priority-rating />
        <due-dates />
        <part-quantities />
        <process-plan-multi-key />
        <repetition-count />
        <parameters />
        <resources-required />
        <child-work-items />
        <precedent-constraints />
        <estimated-durations />
        <cost-summary />
    </task-definition>
    <task-status>
        < work-scheduled-progress />
        <work-actual-progress />
    </task-status>
</task>
```

3.3 Element relationships

The machine shop data model can be represented in a tree shape expression. There are different levels of relationships among elements. The element *resources* has been selected and presented in this subsection to demonstrate the element relationships. *Resources* is a collection of shop resource records and is represented by *stations*, *machines*, *cranes*, *employees*, *tool-catalog*, and *fixture-catalog* items. *Stations* specifies a collection of station-type or station-group-type resources; *machines* specifies a collection of machine-type or machine-group-type resources; *cranes* specifies a collection of crane-type resources; likewise for *employees, tool-catalog,* and *fixture-catalog*. Figure 3 shows the elements' parent-child relationships. These relationships are built by adding referenced or referencing element names to the database tables as table fields.

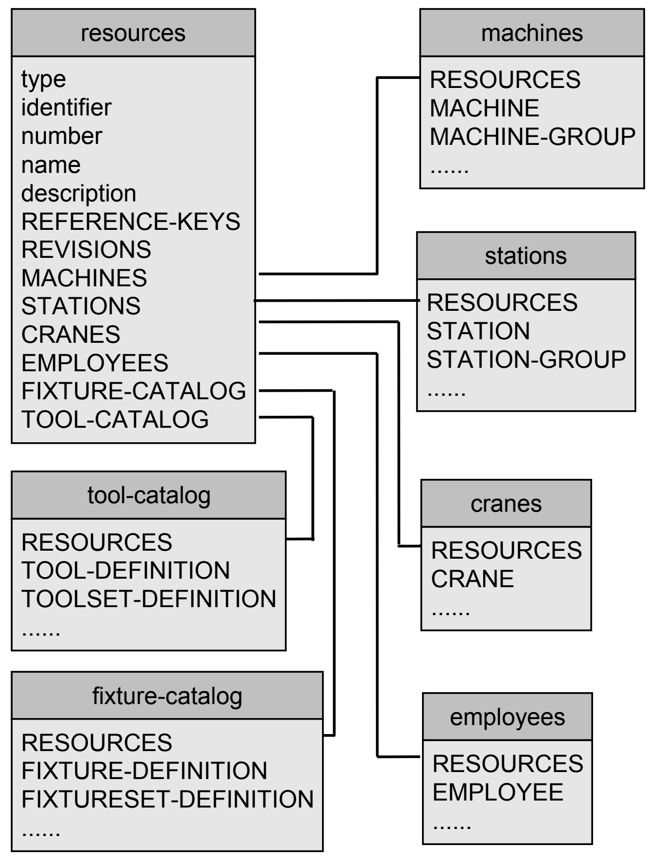

Figure 3: Element Relationships

4 SIMULATION AS A DATABASE APPLICATION

Normally a distributed manufacturing simulation (DMS) may be thought of as a manufacturing simulation that is composed of multiple software processes that are independent executing and interacting with each other (McLean, et al. 2000). DMS needs the support from scattered data sources. A distributed database management system is used to collect, analyze, and manage data. Figure 4 shows a prototype application based on the machine shop data model. Simulation tools provide the support for design, process, manufacturing, assembly, and production management. Interfaces can be built through internal or external network.

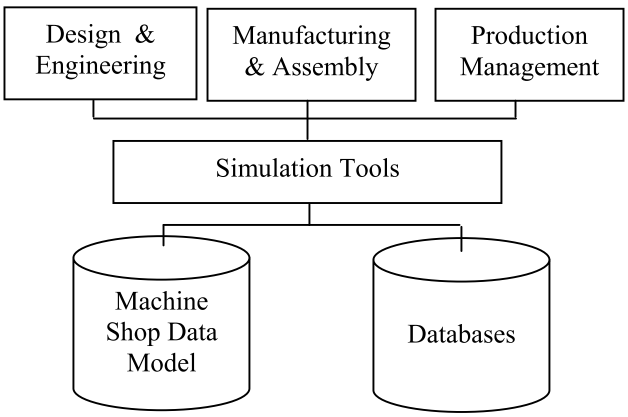

Figure 4: Implementation

5 CONCLUSIONS

This paper provides an overview of work currently undertaken at NIST to develop an information model as a neutral data format for machine shop simulations. The information model and a set of relational tables that mapped onto the information model are described. An effort to build a prototype machine shop simulator used by a pilot facility is undergoing, and the simulator can be reconfigured for use by other small machine shops. The simulator will implement the machine shop information model for the exchange or sharing of data. A simulation standards consortium, initiated by NIST, is being organized to develop pre-competitive neutral specifications for manufacturing simulation. The information model can be a candidate of a standard to be considered by the consortium or by other formal standards bodies.

DISCLAIMER

Certain commercial software and hardware products are identified in this paper. This does not imply approval or endorsement by NIST, nor does it imply that the identified products are necessarily the best available for the purpose.

ACKNOWLEDGEMENT

This project is funded in part by NIST's SIMA Program and the Software Engineering Institute's TIDE program. The authors wish to acknowledge technical contributions to this report by Fred Schenker, Project Manager of the TIDE Program, of Software Engineering Institute.

REFERENCES

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.

D. Appleton Company, Inc.. 1985. "Integrated Information Support System: Information Modeling Manual, IDEF1-Extended (IDEF1X)." Wright-Patterson Air Force Base, OH.

Inoue, H.; Y. Funyu; and S. Masami. 2000. "A New Strategic Enterprise Information Model Based on Super Parallel Evolution Computing," *Proceedings of Seventh International Conference on Parallel and Distributed Systems: Workshops*, 364 –369.

ISO 10303-1. 1994. "Part 1: Overview and Fundamental Principles." *Industrial Automation Systems and Integration-Product Data Representation and Exchange*. International Organization for Standardization, Geneva, Switzerland.

ISO 10303-11. 1994. "Part 11: The EXPRESS Language Reference Manual." *Industrial Automation Systems and Integration-Product Data Representation and Exchange*. International Organization for Standardization, Geneva, Switzerland.

Lee, Y.T. 1999. "Information Modeling: From Design To Implementation." *Proceedings of the Second World Manufacturing Congress* (Durham, U.K., Sept. 27-30). ICSC, Canada/Switzerland, 315-321.

Luo, Y. 2000. "Injection Molding Product Application Activities Models." *International Journal of Advance Manufacturing Technology*. 16(4), 285-288.

McLean ,C.; F. Riddick; and S. Leong. 2000. "Architecture for Modeling and Simulation of Global Distributed Enterprises." *Proceedings of the ASIM 2000 Conference* (Berlin, Mar. 8-9). ASIM-Mitteilung Nr. AMB68. 365-374.

McLean, C.; A. Jones; Y.T. Lee; and F. Riddick. 2002. "An Architecture for a Generic Data-Driven Machine Shop Simulator." *Proceedings of the 2002 Winter Simulation Conference*, (San Diego, CA. Dec. 8-11). IEEE Catalog Number 02CH37393C (CD).

Object Management Group. 2003. "Unified Modeling Language." http://www.omg.org/uml/. UML Resource Page.

Stoll, H.W. 1991. "Design for Manufacture: An Overview." In *Design for Manufacture: Strategies,* *Principles, and Techniques*. J. Corbett.; M.Doner; J. Meleka; and C. Pym (Eds.). Addison – Wesley Publishing Company, New York, NY, 107-129.

The World Wide Web Consortium. 2000. "Extensible Markup Language (XML) 1.0 (second edition)." http://www.w3.org/TR/REC-xml.html.