A neutral data interface specification for simulating machine shop operations

Y. Tina Lee
Charles R. McLean

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

Email: leet@cme.nist.gov Phone: (301) 975-3550 Fax: (301) 258-9749
Email: mclean@cme.nist.gov Phone: (301) 975-3511 Fax: (301) 258-9749

Keywords: simulation, standards, machine shop operations, XML

Abstract

In most cases, the effort required to develop a meaningful simulation for a small machine shop exceeds the resources available. Machine shops typically do not have staff with appropriate technical qualifications required to develop custom simulations of their operations.

Furthermore, simulators are not designed to use traditional shop data in its native format, so models and data import routines usually must be developed from scratch. If simulation software vendors were to try to develop generic job simulation models, they would still be faced with the problem that there are no standard formats for much of the data required to run the models. Thus, if someone wanted to input a specific shop's data into one of these hypothetical simulators, custom data translators would still need to be developed at possibly considerable expense.

This paper provides an overview of work currently underway at the National Institute of Standards and Technology (NIST) to develop neutral, standard, data interfaces for machine shop simulation that are being developed to support the simulation industry and manufacturing users.

1 Introduction

Standard interfaces could help reduce the costs associated with simulation model construction and data exchange between simulation and other software applications -- and thus make simulation technology more affordable and accessible to a wide range of potential industrial users. Currently, machine shops do not typically use simulation technology because of various difficulties and obstacles associated with model development and data translation. Machine shops typically do not have staff with the appropriate technical qualifications required to develop custom simulations of their operations or custom translators to import their data from other software applications. If in-house staff or external consultants are available, shop management is often unwilling to invest the time, effort, and funding required for simulation modeling activities.

NIST is working with a number of industrial partners and researchers to develop neutral formats for machine shop data to facilitate simulation and modeling activities. A machine shop information model, as a neutral interface format, has been under development to support both NIST's System Integration of Manufacturing Application (SIMA) program and the Software Engineering Institute's (SEI) Technology Insertion Demonstration and Evaluation (TIDE) Program. SIMA supports NIST projects in applying information technologies and standards-based approaches to manufacturing software integration problems (Carlisle and Fowler 2001). The TIDE Program is sponsored by the Department of Defense and SEI; it is currently engaged in a number of other projects with various small manufacturers in the Pittsburgh, Pennsylvania area. The technical work is being carried out as a collaboration between NIST, SEI, Carnegie Mellon University, Duquesne University, the iTAC Software AG, and the Kurt J. Lesker Company (KJLC).

KJLC is an international manufacturer and distributor of vacuum products and systems to the research and industrial vacuum markets. KJLC manufactures complete, automatically controlled vacuum systems with special emphasis on custom-designed, thin-film-deposition systems for research in alloys, semiconductors, superconductors, optical and opto-electronics. A machine shop is contained within the KJLC manufacturing facility. KJLC's machine shop operation has been used to help define the requirements for simulation modeling and data interface specification activities described in this paper. Their facility will also be used as a pilot site for testing and evaluation of the simulation models, neutral data interfaces, and other software developed under this TIDE project. For more information on KJLC, see `<www.lesker.com>`.

The machine shop information model was developed with two goals in mind: (a) support for the integration of software applications at a pilot facility -- KJLC's machine shop, and (b) promotion as a standard data interface for manufacturing simulators and possibly for other software applications, such as manufacturing execution system and production scheduling system. The information model is continuing to evolve based on experience and feedback from KJLC's implementations and others involved in this effort.

The objective of the information modeling effort is a standardized, computer-interpretable representation that allows for exchange of information in a machine shop environment. The information model, when completed, must satisfy the following needs: support data requirements for the entire manufacturing life cycle, enable data exchange between simulation and other manufacturing software for machine shops, provide for the construction of machine shop simulators, and support testing and evaluation of machine shops' manufacturing software. Data structures contained within the information model include organizations, calendars, resources, parts, process plans, schedules, and work orders for machine shops.

The Integrated Computer Aided Manufacturing (ICAM) Definition Language 1 Extended (IDEF1X), EXPRESS, Unified Modeling Language (UML), and Extensible Markup Language (XML) are most often used by the manufacturing enterprises for information modeling. IDEF1X is a formal graphical language for relational data modeling, developed by the U. S. Air Force (D. Appleton 1985). EXPRESS (ISO 1994b) was designed to meet the needs of ISO (International Organization for Standardization) 10303, commonly called the STandard for the Exchange of Product model data, STEP (ISO 1994a), and 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 (OMG 2003). XML is a format for structured documents and it enables information exchange in a globally distributed computing environment (W3C 2000).

This paper describes the concept, methodology, specification, and applications of the machine shop information model. This is an extended paper that is based on the conference paper presented at the IFIP WG5.7 ("Integration in Production Management" of the International Federation of Information Processing) Working Conference on Human Aspects in Production Management (McLean and Lee 2003).

2 Standard interfaces

This section describes our approach to developing standard data interfaces that support the machine shop manufacturing simulation. We have proposed an architecture for a generic, data-driven, machine shop simulator, and have been constructing a prototype simulator based on the architecture using commercial off-the-shelf software (McLean et al. 2002). The architecture for the generic machine shop simulator is divided into the following component elements: a neutral shop data file, an XML data processor, a system supervisor and reporting module, a machine shop emulator, a discrete event simulator, and a user interface system. The machine shop information model is a key factor in integrating the generic machine shop simulator effectively and efficiently.

The information model will eventually be formulated into a schema using the XML Schema language (van der Vlist 2002). The information model/XML schema serves as a neutral data format for representing and exchanging machine shop data. With the neutral data format, machine shop data can be represented in working forms, in database tables, or in XML instance documents. The working form is a structured way of storing data in main memory. The database is designed to map the information model/XML schema to tables in the database. Figure 1 depicts the role of the standard interfaces. The XML parsers, "to/from Database Management System (DBMS) translators," and "to/from XML translators" are custom-built software programs. XML parsers convert XML schema's data elements to structural in-memory presentations, such as C++ data structures. Through the use of the neutral data format, "to/from DBMS translators" and "to/from XML translators" allow data to be converted among a user's data formats, database structures, and XML document formats.

To facilitate an implementation of the machine shop information model, two translators are being developed at NIST. One converts an XML instance document to an Access database (Microsoft 2004a); the other converts a database back to XML. XML data structures, which are parsed from the XML Schemas, are used as intermediate representation. A graphical user interface (GUI) system will also be generated to execute various functions, such as import, export, and translator execution.

3 Concept and methodology for the information model

The concept of the shop information model has been introduced briefly at the 2003 IFIP WG5.7 conference in Karlsruhe, Germany (McLean and Lee 2003). In this section, we further describe the concept of the shop information model from the user perspective. Our primary objective was to develop a structure for exchanging shop data between various manufacturing software applications, including simulation. The idea was to use the same data structures for managing actual production operations and simulating the machine shop. The rationale was that if one structure can serve both purposes, the need for translation and abstraction of the real data would be minimized when simulations are constructed. The mapping of real world data into simulation abstractions is not, for the most part, addressed in the current information model.

We also recognized that maintaining data integrity and minimizing the duplication of data were important requirements. For this reason, each unique piece of information appears in only one place in the model. Cross-reference links are used to avoid the creation of redundant copies of data.

An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse. The advantage of using an information model is that it can provide a shareable, stable, and organized structure of information requirements for the domain context. An information model serves as a medium for transferring data among computer systems that have some degree of compliance with this information model. For proprietary data, implementation-specific arrangements can be made when transferring those data (Lee 1999).

In general, the contents of an information model include a scope, a set of information requirements, and a specification. Information requirements serve as the foundation of the specification of the information model. A thorough requirements analysis is a necessity. The initial goal for the machine shop information model is to support data transfer needed for KJLC's machine shop operations. This information model, ultimately, will be promoted as a standard data interface to be used by other machine shops. Thus, the completeness and correctness of the information requirements and a consensus on the data requirements from the industry are also important issues.

A requirements analysis of machine shop operations have been performed. This analysis focused on the data that needed to be exchanged between the applications supporting the machine shop operations. The analysis included the machine shop visits, domain-experts interviews, industrial data reviews, and state-of-the-art assessments. Several workshops were held to gather requirements and reach a preliminary consensus on requirements. Participants at the workshops were the TIDE and SIMA project members from NIST, SEI, Carnegie Mellon, KJLC, etc. As a result, twenty major data requirements and a set of commonly-used data elements have been identified. Thus the current version of the machine shop information model contains twenty major elements. Each of the major data elements are italicized in the discussion that follows. The data elements are called: *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.* Examples of commonly-used data elements are *addresses, customers, due-dates, events,* and *suppliers*. Figure 2 illustrates some of the major elements of the conceptual information model and their relationships to each other. Due to space limitations, the entire model is not shown or discussed in detail. For more detailed information on the model, see (McLean et al. 2004). The remainder of this section discusses the data elements and their significance.

Perhaps a good place to start the discussion of the information model is with the customer. Machine shops are businesses. They typically produce machined parts for either internal or external customers. Data elements are needed to maintain information on customers. The types of organizational information that is needed about customers is very similar to the data needed about suppliers that provide materials to the shop. The same types of organizational data are also needed about the machine shop itself. For this reason, an *Organizations* element was created to maintain organizational and contact information on the shop, its customers, and its suppliers.

Organizations can be thought of as both a phone book and an organization chart. The element provides sub-elements for identifying departments, their relationships to each other, individuals within departments, and their contact information. Various other types of information needs to be cross-referenced to organizations and contacts within structure, e.g., customer orders, parts, and procurements to suppliers.

The operation of the machine shop revolves around the production of parts, i.e., the fabrication of parts from raw materials such as metal or plastic. The raw materials typically come in the form of blocks, bars, sheets, forgings, or castings. These materials are themselves parts that are procured from suppliers. The *Parts* data element was created to maintain the broad range of information that is needed about each part that is handled by the machine shop. Part data includes an identifying part number, name, description, size, weight, material composition, unit-of-issue, cost, group technology classification codes, and revision (change) data. Cross-reference links are needed to the customers that buy the parts from the shop and/or the suppliers that provide them as raw materials. Links are also needed to other data elements, documents, and files that are related to the production of parts including: part specification documents, geometric models, drawings, bills-of-materials, and process plans.

The *Bills-of-materials* element is basically a collection of hierarchically-structured parts lists. It is used to define the parts and subassemblies that make up higher-level part assemblies. A bill-of-materials identifies, by a part number reference link, the component or subassembly required at each level of assembly. The quantity required for each part is also indicated. Cross-references links are needed between parts that are assemblies and their associated bill-of-materials.

The *Parts* and *Bills-of-materials* elements establish the basic definition of parts produced or used by the shop. Another element, *Inventory,* is used to identify quantity of part instances at each location within the facility.

Inventory data elements are provided for parts, tools, fixtures, and materials. Materials are defined as various types of stock that may be partially consumed in production, e.g., sheets, bars, and rolls. Structures are provided within inventory to keep track of various stock levels (e.g., reorder point level) and the specific instances of parts that are used in assemblies.

The *Procurements* element identifies the internal and external purchase orders that have been created to satisfy order or part-inventory requirements. Cross-reference links are defined to *Parts* to identify the specific parts that are being procured and to *Work* to indicate which work items they will be used to satisfy.

The *Work* data element is used to specify a hierarchical collection of work items that define orders, production and support activities within the shop. Support activities include maintenance, inventory picking, and fixture/tool preparation. *Work* is broken down hierarchically into orders, jobs, and tasks.

Orders may be either customer orders for products or internally-generated orders to satisfy part requirements within the company, e.g., maintenance of inventory levels of stock items sold through a catalog. Orders contain both definition and status information. Definition information specifies whom the order is for (i.e., customer cross-references), its relative priority, critical due dates, what output products are required (a list of order items, i.e., part references and quantities required), special resource requirements, precedence relationships on the processing of order items, and a summary of estimated and actual costs. Order items are also cross-referenced to jobs and tasks that decompose the orders into individual process steps performed at workstations within the shop. Status information includes data about scheduled and actual progress towards completing the order.

Jobs typically define complex production work items that involve activities at multiple stations and ultimately produce parts. Tasks are lower-level work items that are typically performed at a single workstation or area within the shop.

The *Process-plans* element contains the process specifications that describe how production and support work is to be performed in the shop. Major elements contained within *Process-plans* include routing sheets, operation sheets, and equipment programs. Routing and operation sheets are the plans used to define job- and task-level work items, respectively, in the work hierarchy. These process plans define the steps, precedence constraints between steps, and resources required to produce parts and perform support activities. Precedence constraints defined in a process plan are used to establish precedence relationships between jobs and tasks. Equipment program elements establish cross reference links to files that contain computer programs that are used to run machine tools and other programmable equipment that process specific parts. Each part in the *Parts* element contains cross-reference links to the process plans that define how to make that part. Jobs and tasks contain links back to the process plans that defined them.

The *Resources* element is used to define production and support resources that may be assigned to jobs or tasks in the shop, their status, and scheduled assignments to specific work items. The resource types available in the machine shop environment include: stations and machines, cranes, employees, tool and tool sets, fixtures and fixture sets.

The *Skill-definitions, Setup-definitions, Operations-definitions, Maintenance-definitions,* and *Time-sheets* elements provide additional supporting information associated with resources. *Skill-definitions* lists the skills that an employee may possess and the levels of proficiency associated with these skills. An example of a skill might be the ability to operate a particular type of machine. Skills are referenced in employee resource requirements contained in process plans. *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. Setup is referenced in the setup configuration that is associated with a particular machine, tool, fixture, crane, or station. *Operation-definitions* specifies the types of operations that may be performed at a particular station or group of stations within the shop. Operations are referenced in routing sheets and operation sheets contained in process plans. *Maintenance-definitions* specifies preventive or corrective maintenance to be done on machines or other maintained resources. A maintenance order is a work item to be done on a particular resource within the shop and *maintenance-definitions* is referenced in a maintenance order. *Time-sheets* is used to log individual employee's work hours, leave hours, overtime hours, and so on.

The *Layout* element defines the physical locations of resource objects and part instances within the shop. It also defines reference points, area boundaries, and paths. It contains references to external files that are used to further define resource and part objects using appropriate graphics standards. Cross-references links are also provided between layout objects and the actual resources that they represent.

Schedules and *Calendars* data elements deal with time. *Schedules* provides two views of the planned assignment of work and resources. Work items (orders, jobs, and tasks) are mapped to resources, and conversely, resources are mapped to work items. The planned time events associated with those mappings are also identified, e.g., scheduled start times and end times. *Calendars* identifies scheduled work days for the shop, the shift schedules that are in effect for periods of time, planned breaks, and holiday periods.

The four remaining major data elements are *Revisions, References, Probability-distributions,* and *Units-of-measurement.* The *Revisions* element is used repeatedly throughout many levels of the information model. It provides a mechanism for identifying versions of subsets of the data, revision dates, and the creator of the data. The *References* element identifies external digital files and paper documents that support and further define the data elements contained within the shop data structure. It provides a mechanism for linking to outside files that conform to various other format specifications or standards, e.g., STEP part design files. The *Probability-distributions* element defines probability distributions that are used to vary processing times, breakdown and repair times, availability of resources (that determines which resources are able to be used or assigned during a simulation run), among others. Distributions may be cross-referenced from elsewhere in the model, e.g., equipment resources maintenance data. *Units-of-measurement* specifies the units used in the file for various quantities such as length, weight, currency, and speed.

The next section provides a detailed illustration of a small portion of the overall information model, and UML and XML file structures.

4 Specification of the information model

The specification of the information model defines elements, attributes, constraints, and relationships between elements for the domain context. The specification should be laid out using some formal information modeling language. An information modeling language provides a formal syntax that allows users to capture data semantics and constraints unambiguously. Three types of methods that implement information models are currently used by the manufacturing community:

- Data transfer via a working form, which is a structured, in-memory representation of data. The method uses a mechanism that accesses and changes data sequentially without actually moving the data around. All shared data are stored in memory.
- Data transfer via an exchange file, which is a file with a predefined structure or format. This method requires a neutral file format for storing the data. The application systems read and write from files.
- Data transfer using a database management system. This method uses a database management system where information is mapped onto and retrieved from databases.

These implementation methods can be accomplished through translators that are developed using programming languages and database management systems. The selection of an implementation method is heavily dependent on the target environment where the application system resides. While the relational database is generally desirable for data transfer, the traditional file-oriented systems are being used continually by many manufacturing applications.

A specification for the machine shop information model has been developed based on the information-model concept described in section 3. Figure 3 shows the top level of the model in UML representation. The *shop-data* element is represented by a *type*, an *identifier*, and a *number*. Optional elements include: *name*, *description*, *reference-keys*, *revisions*, *units-of-measurement*, *organizations*, *calendars*, *resources*, *skill-definitions*, *setup-definitions*, *operation-definitions*, *maintenance-definitions*, *layout*, *parts*, *bills-of-materials*, *inventory*, *procurements*, *process-plans*, *work*, *schedules*, *time-sheets*, *references*, and *probability-distributions. Type* is an attribute of *shop-data* and is an enumeration to describe types about *shop-data*. *Identifier* is a key to uniquely identify the object internally within the system, and it is generated automatically by the system when the object is created. *Number* is also a unique key for identifying the object either when taken alone or possibly together with the object type, and the uniqueness is to the user or the user's organization. *Type*, *identifier* and *number* are required attributes. *Name* is used to identify the object by the user or user's organization. It is provided for readability. *Description* is used to describe the nature of the subject. *Reference-keys* refers to reference documents or files that are stored external to the model. When a data element's name suffixes with "*-key*" or "*-keys*", these data elements serve as pointers to the model to avoid redefining the same set of information. All other attributes, such as *organizations*, *calendars*, *resources*, etc., are major elements of the model that were introduced in section 3.

The machine shop information model specification is documented using both UML and XML structures. XML is chosen to support Web users while UML's standard graphical notations provide visual communications. UML is a graphical representation; the language is for specifying, visualizing, constructing, and documenting, rather than processing. XML is a format for structured documents, thus XML documents are decodable.

XML supports the development of structured, hierarchical data entities that contain a high level of semantic content, that is both human and machine interpretable. There are several supporting standards from W3C that make working with XML easier. These include Document Object Model (DOM) for manipulating XML documents, XML Schema for defining the format of XML documents, and Extensible Style-sheet Language (XSL) for translating XML documents to other formats, see `<www.w3.org>`. There also exist commercial off-the-shelf software applications to implement creation, parsing, interpreting, and displaying of XML documents. The current version of the XML specification of the information model has been developed using Microsoft XML Notepad (Microsoft 2004b).

The current version of the specification includes XML documents that are well-formed, but may or may not be validated. Data should be validated before being imported to a legacy system. An XML Schema is a specification of the elements, attributes, and structures; it is not only useful for documentation, but also for validation or processing automation. Validation is the most common use for schema in the XML environment.

UML provides several modeling types, from functional requirements and activity analysis to class structures and component description. The modeling type used to map to the XML documents is the UML class diagram. A UML class diagram can be constructed to represent the structural and behavioural features visually. Since the behavioural feature is not relevant to the XML specification, that feature is omitted here (Carlson 2001).

The complete specification is not given here due to its size. Instead, a sample data element specification is described. The data element of *orders* is chosen for illustration in this section. *Orders* is a subgroup of work and consists of a set of individual *order* data elements. It specifies a collection of production work orders to be processed within the shop. Each *order* contains the order definition and/or order status section. The order definition contains attributes of the order including a list of order items, i.e., a listing of individual parts that make up a particular order. The order status describes information about scheduled and actual progress toward completing the order. The same part may be listed in the order multiple times in different order items if each instance has unique attributes, e.g., different due dates.

The UML information model for the *orders* element is shown in figure 4. The *orders* element has the attributes of *type* (which is a string), an *identifier* (which is an "int" or integer value), a *number* (which is a string), an optional *name* (which is a string), an optional *description* (which is a string), and an optional *reference-keys* and *revisions* (they are user-defined data types). Figure 4 illustrates the cardinality relationships among *orders*, *order*, *order-definition*, and *order-status*. An *orders* element contains some *order* elements. Each *order* is defined by *order-definition* and has an *order-status*. *Orders* and *order* have a "one" to "one or more" relationship, i.e., there may exist one or many *order* instances for an *orders* instance. Similarly, there may exist zero or one *order-definition* instance and zero or one *order-status* instance for an *order* instance. Each *order-definition* instance is defined by one *customers* instance, one *due-dates* instance, and zero or one of *priority-rating, order-items, precedent-constraints, resources-required,* and *cost-summary* instances.

The same *orders* element presented in XML is shown below:

```
<orders type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <order type="" identifier="" number="">
        <name />
        <description />
        <reference-keys />
        <revisions />
        <order-definition>
            <customers />
            <priority-rating />
            <due-dates />
            <order-items/>
            <precedent-constraints />
            <resources-required />
            <cost-summary />
        </order-definition>
        <order-status>
            <work-scheduled-progress />
            <work-actual-progress />
        </order-status>
    </order>
</orders>
```

5 Related Work

To facilitate the implementation of the machine shop information model and to demonstrate the model's feasibility and capability, tasks/projects have being performed in the following four major areas:

- Specification Extension

  Schema Development–

  The XML documents specification is now being extended to schemas using the World Wide Web Consortium (W3C) XML Schema, an XML schema language. The schema conversion is being done by using Examplotron (van der Vlist 2001) and Trang (Thai 2002) converters. Examplotron generates Regular Language Description for XML Next Generation (RELAX NG) (relaxng.org 2003) schemas where RELAX NG is another schema language. Trang, based on RELAX NG, converts between schema languages, including RELAX NG, W3C XML Schema, etc.

  Supply Chain Specification–

  Under the new project in the Doyle Center for Manufacturing Technology Program (a new TIDE outgrowth and extension program), the information model is currently extending the capability to include the supply-chain interface data. A set of preliminary interface data requirements for a proposed, generic supply-chain-simulation system has been identified (Umeda and Lee 2004). The requirement list will be added into the information model when the requirement list is reviewed by the project members. Project partners for this Doyle Center project include NIST, Carnegie Mellon University, Duquesne University, and Catalyst Connections.

- Case Study

  Boeing Case Study –

A case study in which the machine shop information model was applied at the Boeing Company has been completed (Lu and Qiao 2003). The case study represents a discrete-event-simulation model about a possible future wing assembly line inside one of the Boeing manufacturing facilities. Entity classes in the case-study simulation model contain asynchronous servers, multi-input-output buffers, bi-directional cranes, labors, processes, and machines on different shifts. The case study model was developed using DELMIA's QUEST. Manufacturing data needed by simulation are provided with XML instance documents based on the machine shop data information model. These XML instance documents are converted into Batch Control Language (BCL), a QUEST's command language, and then executed by QUEST (DELMIA 2002).

KJLC prototype –

Under the TIDE project, Carnegie Mellon has successfully developed a prototype external scheduling system for KJLC. The prototype scheduling system used data from the iTAC's Manufacturing Execution System (MES) (iTAC 2004) to create a schedule for each resource (machine, person) that optimizes the execution of jobs. This was done frequently as realities of the shop floor change. The MES provided the required real-time information to support the scheduling program. Interfaces between the MES and the prototype scheduling system were through XML documents based on the machine shop information model. The initial test interface data set used in the system included *calendars*, *resources*, *setup-definitions*, *skill-definitions*, and *units-of-measurements*.

Shipyard Case Study –

A generic shipbuilding simulation system has been designed using the discrete event simulation tool, ProModel™ (ProModel 2004). The simulation system can be used to analyze the schedule impact of new workload, evaluate production scenarios, and identify resource problems (McLean and Shao 2001). A case study that implements the information model with shipyard data is under development for the generic shipbuilding simulation system. The purpose of this study is to identify a neutral data interface for the generic shipbuilding simulation system, hence to promote the utilization of simulations in the shipbuilding industry. Shipyard data used in this case are those necessary for construction and execution of the simulation system.

- Standardization

The machine shop information model will be proposed as a candidate standard to be considered by a formal standards body. A preliminary plan is in process for standardization through the Simulation Interoperability Standards Organization (SISO) (SISO 2004). The Standards Activities Committee of SISO is a sponsor of the IEEE (Institute of Electrical and Electronics Engineers, Inc.) standards.

- Tool Development

Machine Shop Database Model –

A database model containing a set of tables that are mapped onto the machine shop information model has been generated (Lee and Luo 2003). The objectives of the database development are to demonstrate the feasibility of the information model, to develop a pilot database system and then to migrate to a large database management system, and to support the integration of manufacturing applications and simulations used in machine shops.

Shop Data Editor –

A graphical user interface was developed using various controls within Visual Basic™ (Microsoft 2004d) to simplify data entry and population of the internal object structure and XML-based data exchange file. Development of the data editor is continuing to support the input of data for the entire machine shop information model.

XML Data Processor –

A prototype was developed to import and export scheduling data using Microsoft's XML Document Object Model (Microsoft 2004c), a standard library of application programming interfaces for accessing on XML document. It also transferred data into internal Visual Basic™ object structures. The prototype verified the structure of software that processes the XML files.

Translators –

A set of software programs has been developed at NIST to facilitate the implementation of the machine shop information model. These programs in Windows are capable of validating XML documents, editing XML documents, and translating data between XML documents and database tables. The XML documents and database tables are all structured using the machine shop information model.

6 Conclusions and Future Work

This paper described the work being carried out by NIST in developing a neutral model and data exchange format for machine shop data. The objective of the information modeling effort was a standardized, computer-interpretable representation that allows for the efficient storage and exchange of manufacturing life cycle data in a machine shop environment.

The information model will continue to evolve based on the experience and feedback from others involved in this effort. A long data file is a general comment from implementers and readers. Eliminating duplicate information from the model is a current effort. The model is now being transformed into a schema using an XML schema language. There are also plans to expand the model to include assembly line, supply chain, and other domain areas. A database implementation using Microsoft Access has been developed. The information model will be proposed as a candidate standard to be considered by a formal standards development organization.

The current model addresses the exchange of real-world data between simulation and other manufacturing software applications. An extension of the information model and exchange file format is needed to support the simulation abstraction process. This model would be used to maintain data regarding the mapping of real-world objects into their simulated representation. For example, as part of the abstraction process data values may be approximated, different colors may be substituted for real objects, shapes and sizes may be changed, and probabilistic distributions may be substituted for actual arrivals and other time-dependent events.

There are also experimental development activities underway to test the viability of the model with real-world applications. A generic manufacturing simulator is being developed at NIST for the TIDE Program (McLean et al. 2002). This model is also being used in the TIDE Program to integrate a manufacturing execution system with a real-time adaptive scheduler, and the manufacturing simulator. An aerospace manufacturer is also working on a prototype simulation based on the specification.

Acknowledgments and disclaimer

This project is funded by NIST's SIMA Program and the SEI TIDE Program. SIMA supports NIST projects applying information technologies and standards-based approaches to manufacturing software integration problems. 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.

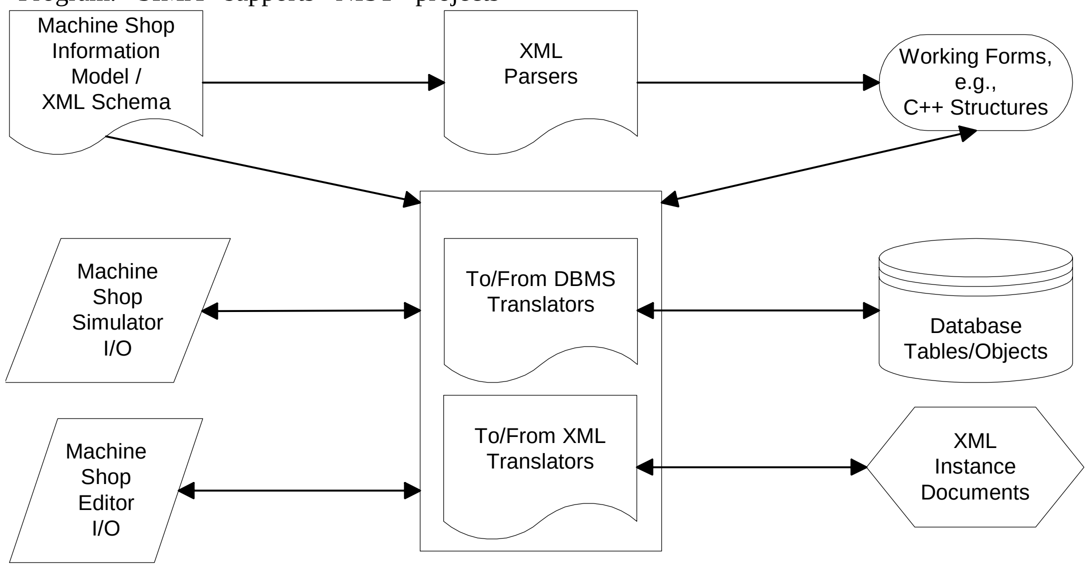

Figure 1. Concept for the standard data interfaces.

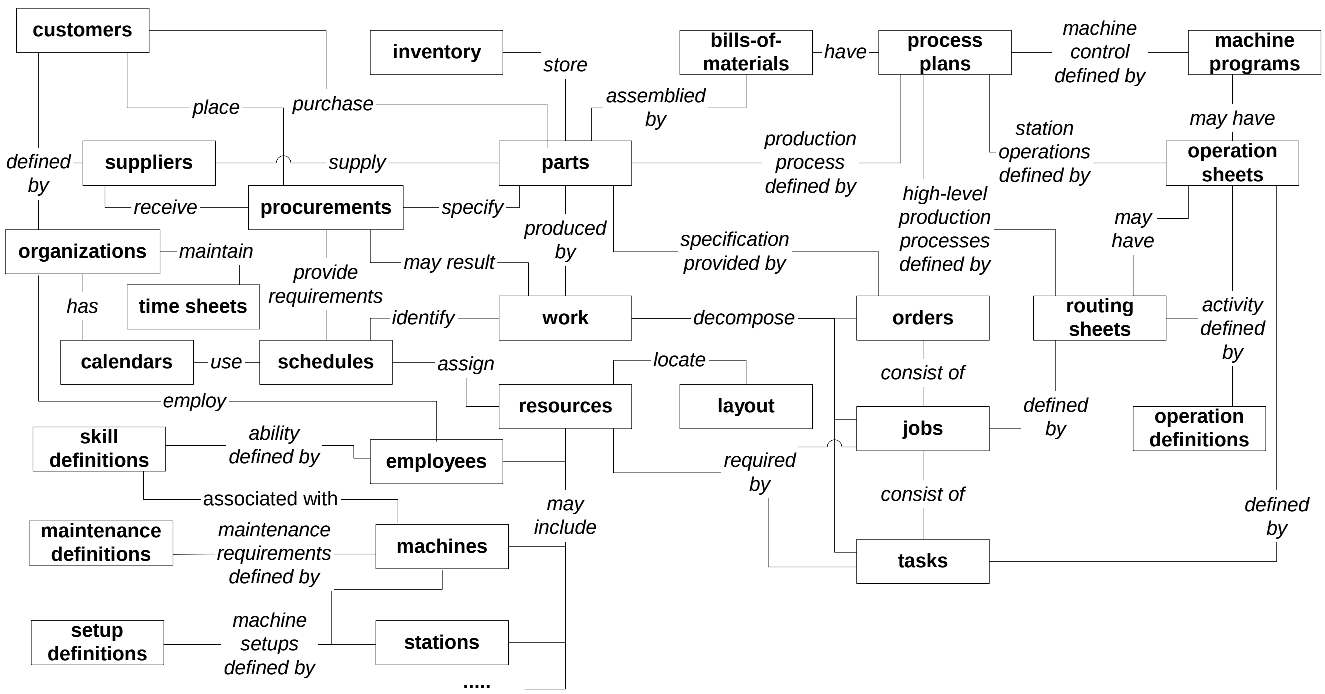

Figure 2. Concept for the machine shop information model.

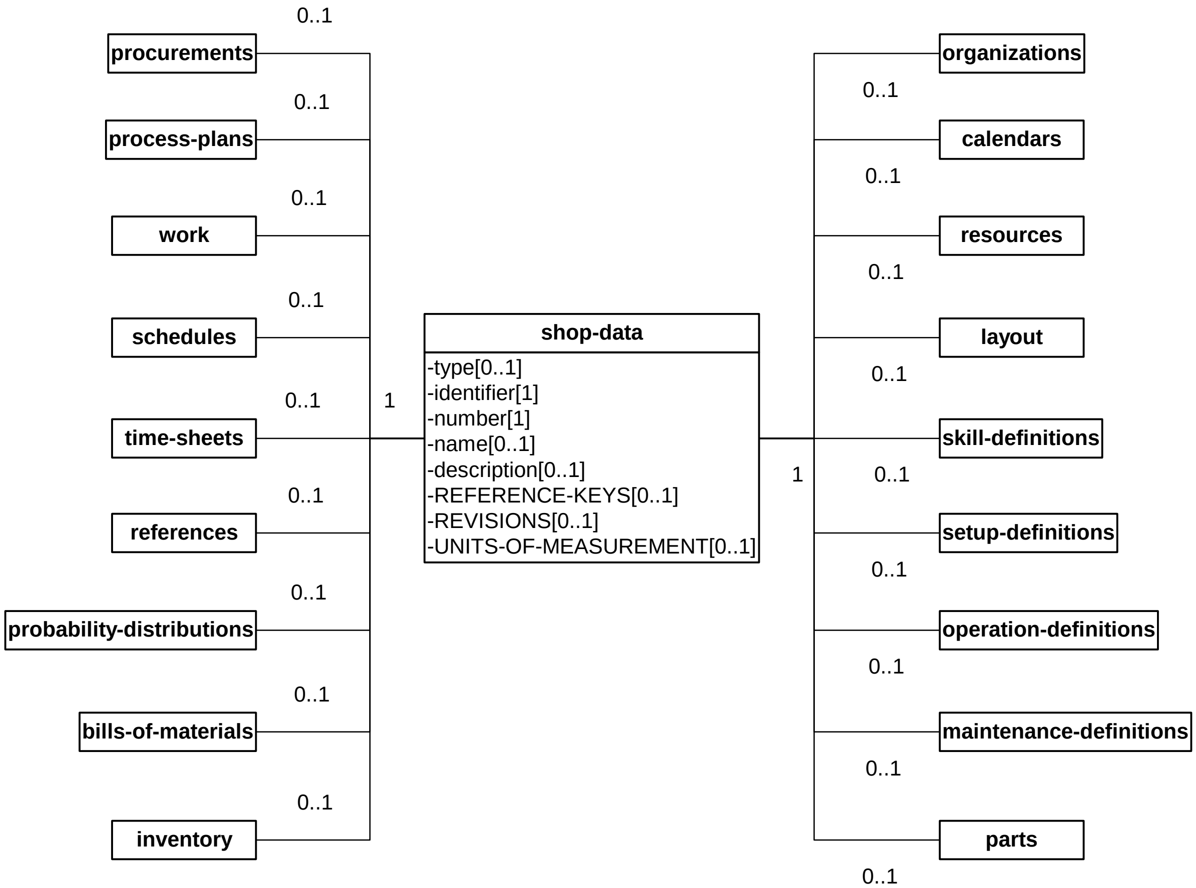

Figure 3. Top level of the machine shop information model.

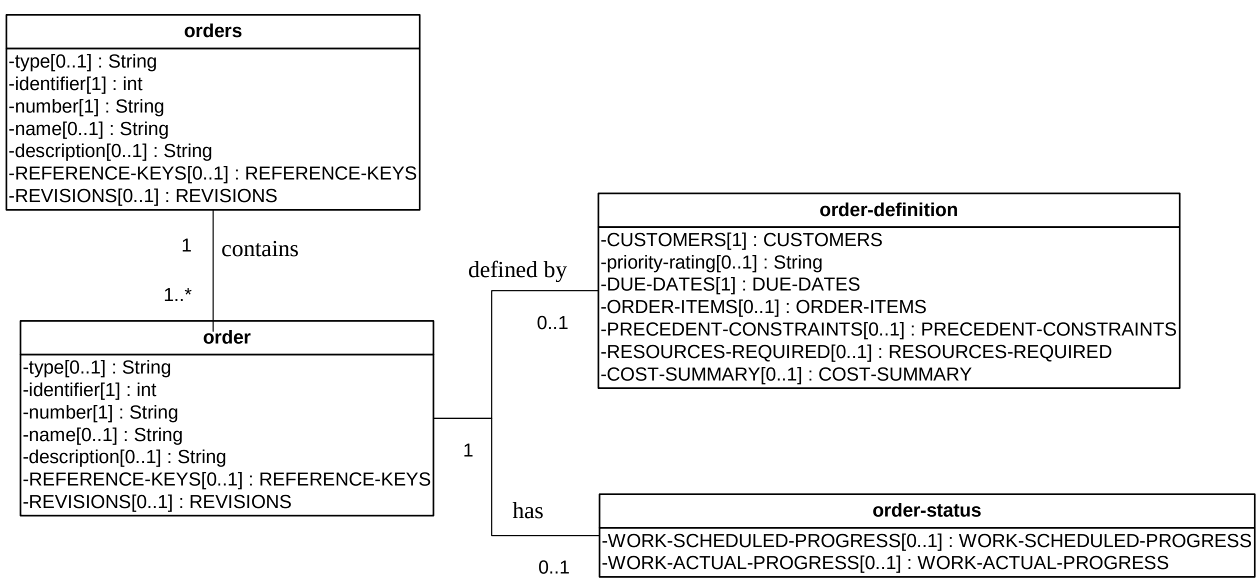

Figure 4. UML representation of the *orders* element.

References

CARLISLE, M., and J. FOWLER. 2001. Systems Integration for Manufacturing Applications Biennial Report. Fiscal Years, NISTIR 6721. Gaithersburg, Maryland: National Institute of Standards and Technology.

CARLSON, D. 2001. Modeling XML Applications with UML: Practical e-Business Applications (Addison-Wesley).

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

DELMIA Corporation. 2002. QUEST Batch Control Language Reference Manual. Troy, MI.

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

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

iTAC Software. 2004. The Traceability Experts [online]. Available online via http://www.itac.de/ger/index.html [accessed August 27, 2004]

LEE, Y. T. 1999. Information Modeling: From Design To Implementation. Proceedings of the Second World Manufacturing Congress, ed. S. Nahavandi and M. Saadat, pp. 315-321. Canada/Switzerland : International Computer Science Conventions.

LEE, Y. T., and Y. LUO. 2003. A Database Design for the Machine Shop Information Model. NISTIR 7077. Gaithersburg, Maryland: National Institute of Standards and Technology.

LU, R. F., and G. Qiao. 2003. NIST XML Simulation Interface Specification at Boeing: A Case Study. Proceedings of the 2003 Winter Simulation Conference, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, pp. 1230-1237. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

MCLEAN, C., A. JONES, T. LEE, and F. RIDDICK. 2002. An Architecture for a Generic Data-Driven Machine Shop Simulator. Proceedings of the 2002 Winter Simulation Conference, ed. E. Yucesan, C. Chen, J. L. Snowdon, and J. M. Charnes, pp. 1108-1116. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

MCLEAN, C., and G. SHAO. 2001. Simulation of Shipbuilding Operations. Proceedings of the 2001 Winter Simulation Conference, ed. B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, pp. 870-876. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

MCLEAN, C., and T. LEE. 2003. A Neutral Data Interface Specification for Simulating Machine Shop Operations. European Series in Industrial Management: Proceedings of the IFIP WG 5.7 Working Conference on Human Aspects in Production Management – Volume 2 (Karlsruhe, Germany), ed. G. Zulch, S. Stowasser, and H. S. Jagdev, V6, pp. 110-116. (Shaker Verlag).

MCLEAN, C., Y. T. LEE, G. SHAO, and F. RIDDICK. 2004. Shop Data Model And Interface Specification. Draft NISTIR. Gaithersburg, Maryland: National Institute of Standards and Technology.

Microsoft Corporation. 2004a. Access [online]. Available online via http://msdn.microsoft.com/office/understanding/access/ [accessed August 30, 2004]

Microsoft Corporation. 2004b. How to Use XML Notepad to create an XML Document. Access [online]. Available online via http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q296/5/60.ASP&NoWebContent=1 [accessed August 30, 2004]

Microsoft Corporation. 2004c. Microsoft XML Core Services (MSXML) 4.0 - DOM Developer's Guide [online]. Available online via http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk/htm/dom_devguide_overview_2g1j.asp [accessed August 27, 2004]

Microsoft Corporation. 2004d. Visual Basic Developer Center [online]. Available online via http://msdn.microsoft.com/vbasic/ [accessed August 30, 2004]

Object Management Group (OMG). 2003. Unified Modeling Language [online]. Available online via `http://www.omg.org/uml/` [accessed August 27, 2004]

ProModel Corporation. 2004. ProModel Home [online]. Available online via http://www.promodel.com/products/promodel/ [accessed August 27, 2004]

relaxng.org. 2003. RELAX NG Home Page [online]. Available online via http://relaxng.org/ [accessed August 27, 2004]

Thai Open Source Software Center Ltd. 2002. Trang [online]. Available online via http://www.thaiopensource.com/relaxng/trang.html [accessed August 27, 2004]

UMEDA, S., and Y. T. Lee. 2004. Integrated Supply Chain Simulation System – A Design Specification for a Generic Supply Chain Simulator. NISTIR 7146. Gaithersburg, Maryland: National Institute of Standards and Technology.

VAN DER VLIST, E. 2001. Examplotron [online]. Available online via http://examplotron.org/0/4/ [accessed August 27, 2004]

VAN DER VLIST, E. 2002. XML Schema (O'Reilly& Associates).

World Wide Web Consortium (W3C). 2000. Extensible Markup Language (XML) 1.0 (second edition) [online]. Available online via http://www.w3.org/TR/REC-xml.html [accessed August 27, 2004]