CMSD:
A Model Supporting
Manufacturing and Simulation Application Integration

Frank Riddick
Y. Tina Lee
Manufacturing Systems Integration Division
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899, USA
301-975-3892, 301-975-3550
frank.riddick@nist.gov, leet@nist.gov

Keywords:
CMSD, data exchange, information model, and manufacturing simulation

ABSTRACT: *Standard representations for key manufacturing entities could help reduce the costs associated with simulation model construction and data exchange between simulation and other software applications. This change would make the use of simulation technology more affordable and accessible to a wide range of potential industrial users. To help foster this change, the Core Manufacturing Simulation Data (CMSD) specification was created. CMSD is a neutral, computer-interpretable representation for manufacturing shop floor related information. CMSD defines an information model that describes the characteristics of and relationships between the core manufacturing entities that define shop floor operations. This paper provides an overview of CMSD and describes how it can be used to enable data exchange between simulations and other manufacturing applications.*

1. Introduction

Simulation is a valuable tool that is used to evaluate the behavior and performance of complex manufacturing systems over time. Using simulation, managers and engineers can analyze the expected performance of those systems and propose changes to improve their performance. This analysis can be done quickly and economically before potentially disruptive system modifications are implemented. Noting these benefits, the National Research Council has repeatedly identified simulation as a high priority research area [1, 2].

Simulation is greatly underutilized in industry, particularly in the area concerned with the analysis of shop floor production systems. One of the principal reasons is the difficulty of identifying, accessing, and collating the relevant production-related data so that it can be used in the creation and execution of simulation applications. Frequently, the data is spread across several applications. Existing shop floor data standards, such as those from the Instrument Society of America [3] and Open Applications Group [4], do not solve this integration problem [5]. A critical thing missing from these standards is the ability to capture the stochastic nature exhibited by many properties of common shop floor related manufacturing entities.

To combat data interoperability problems, the Core Manufacturing Simulation Data *(*CMSD) specification has been developed under the auspices of the Simulation Interoperability Standards Organization (SISO) [6]. The standard facilitates the integration of and data exchange between manufacturing applications of shop floor operations, with specific support for the needs of manufacturing simulation applications. It defines a neutral, high level, extensible framework for describing the important aspects of the manufacturing entities (equipment, materials, personnel, production processes, etc.) that are used in manufacturing operations. It specifically facilitates the integration of simulation applications by providing a means to define aspects of manufacturing entities that are governed by stochastic processes, in such a way that the information can be exchanged and shared.

The rest of this paper is organized as follows. In section 2, a discussion of the motivation for developing CMSD is presented. The conceptual model on which CMSD is based is presented in section 3. In section 4, an overview of the structure and contents of the information model defined by the CMSD specification is presented. Information about the use of the Unified Modeling Language (UML) and Extensible Markup Language (XML) in the creation of CMSD is also discussed. In Section 5, a discussion of a case study that used CMSD as the basis for integrating simulations with other manufacturing applications is presented. The paper concludes with section 6, where a summary of the information about CMSD in this paper is presented.

2. The Motivation for Developing CMSD

The CMSD specification defines an information model that provides neutral descriptions for the essential entities in the manufacturing domain and the relationships between those entities that are needed to create manufacturing-oriented simulations. This model facilitates the exchange of information between manufacturing-oriented simulations and other applications in manufacturing domains. The CMSD effort is an attempt to:

- foster the development and more widespread use of simulations in manufacturing operations,
- facilitate data exchange between simulation and other manufacturing software applications,
- enable and facilitate better testing and evaluation of manufacturing software, and
- increase general manufacturing application interoperability.

The key reason for undertaking this effort is simulation technology has continued to be underutilized in manufacturing, limiting opportunities to make productivity and competitiveness improvements. Recently the Gartner group [7] identified simulation as a key technology for businesses. A key driver for their suggestion is that simulation can be used to cost-effectively explore multiple "what-if"scenarios when making business decisions.

The initial work on CMSD grew out of a project whose intent was to assess the information storage, representation, and application integration needs of a small electronics manufacturer [8]. The goal was to 1) create neutral definitions of manufacturing entities that could be exchanged between their production applications, so that 2) their applications could be integrated with simulations of their shop floor. This would allow production to be analyzed and optimized based on real and up-to-date production data [9].

3. The CMSD Conceptual Model

The CMSD conceptual model lays out the key abstractions for the important entities and entity interrelationships for the domain of shop floor operations. This conceptual model defines the entities that represent the manufacturing concepts covered by CMSD, along with the key relationships between those entities. The intent of the model is to cover the concepts that are most closely related to matters of defining and managing shop floor operations. In this model, no information about the architecture of the software applications that are used to control and manage shop floor operations is represented. The focus is on representing the data entities, and the interrelationships between those data entities, that are stored/manipulated/exchanged by manufacturing shop floor applications.

The CMSD conceptual model is presented in the form of an entity relationship diagram [10]. Since this model is intended to represent high level concepts, only the entities and their direct interrelationships are represented. Figure 1 illustrates some of the major data entities of the conceptual information model and their relationship to each other. For relationships, direction indicators are provided to explicitly indicate each entity's role in the relationship. For example, the *BillOfMaterrials* data entity, located at the top left of the figure, is a reference to a bill of materials that defines the substructure of parts of a particular part type. Part or part type together is referred as the *Part/PartType* data entity in the figure.

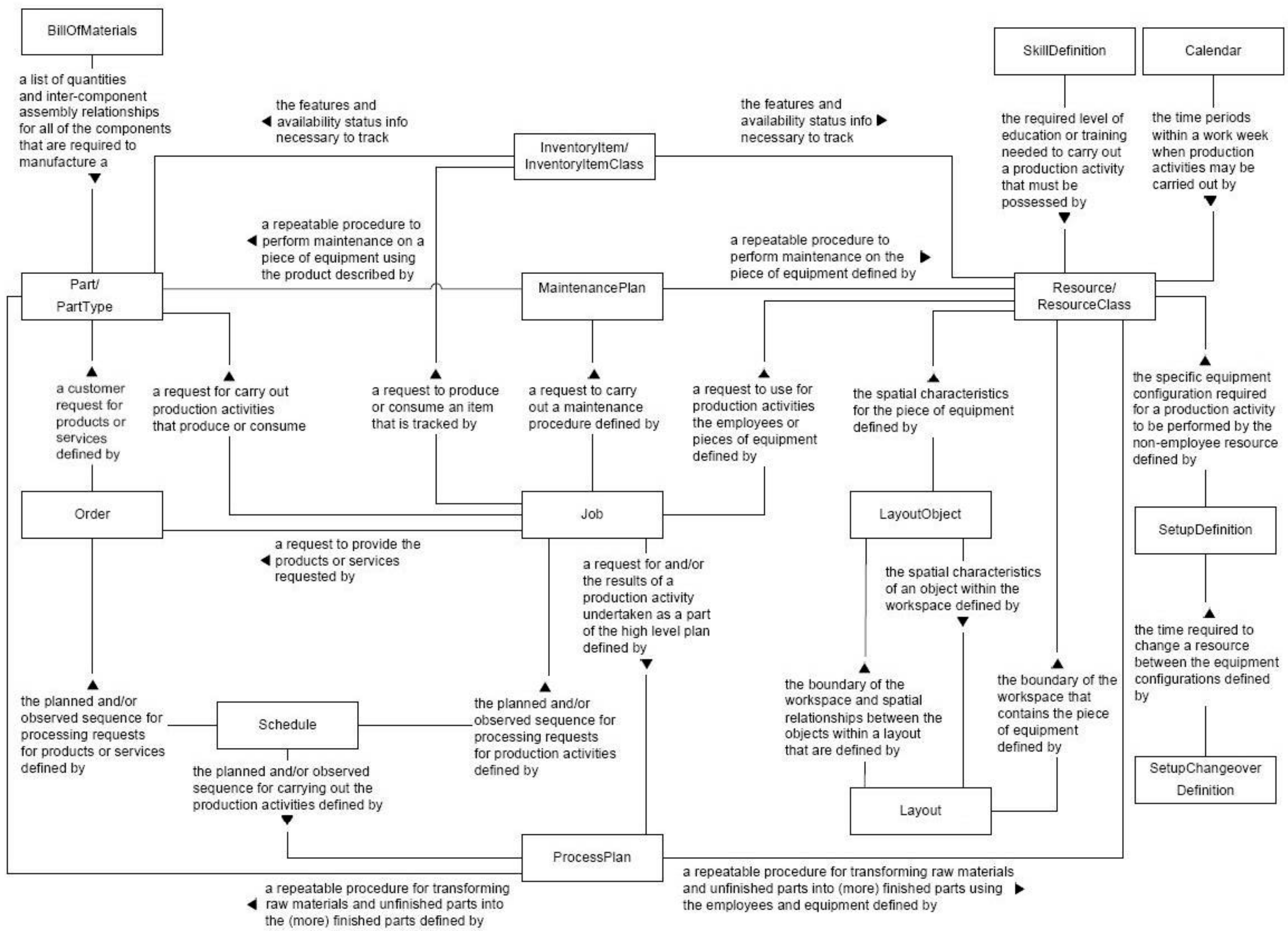
4. The CMSD Information Model

The CMSD conceptual model is critical for setting the universe of discourse for CMSD, but its definition is too high-level to facilitate unambiguous information definition and exchange. To meet this challenge the CMSD Information Model was defined [6].

4.1 The CMSD Information Model development process

Using the CMSD conceptual model as a starting point and keeping in mind the goals for CMSD, the CMSD Information model was developed. The approach taken was to refine and extend the manufacturing entities of the conceptual model with attribute information, in such a way that the refined entities could function as recognizable but neutral representations of their associated concepts. Also, the model was extended with additional entities that enabled aggregations of CMSD data to be constructed so that: 1) the uniqueness of entities in the aggregation can be unambiguously defined; 2) interrelationships between entities can be unambiguously defined; and, suitable realizations for aggregations of CMSD data exist so that the aggregations may be stored and/or exchanged between software applications.

Figure 1 - The CMSD Conceptual Model



In addition to the information in the CMSD conceptual model, the content of the CMSD information model is based on:

- reviews of published research about information representations to support manufacturing and shop floor operations,
- reviews of formal standards that cover some part of manufacturing operations,
- reviews of open specifications supported by consortia with broad membership in the manufacturing community,
- reviews of manufacturing entity representations in simulation software widely used in manufacturing, and
- discussions with domain experts from different manufacturing arenas.

As a part of the development process, several drafts of the CMSD specification were produced. These drafts were used as the basis for several case studies that involved integrating simulation and manufacturing applications. These case studies were invaluable for assessing the feasibility of using CMSD for manufacturing application integration. In section 5 of this paper, a description of one of these case studies, where CMSD is used as a basis for data exchange with a simulation application, is presented.

4.2 The CMSD Multiple Modeling Language Approach

To better support the use of CMSD as an integrating mechanism for applications on different systems, the CMSD information model is being specified using different modeling languages.

The normative representation of the information model has been created using the Unified Modeling Language (UML) [11], a widely-accepted standard of the Object Management Group (OMG). It is a graphically-oriented language that provides a rich set of modeling mechanisms that for CMSD allows intricate entity structures, relationships, and instance constraints to be defined. By using UML to define the CMSD information model, a visual representation has been created that provides the best opportunity for the largest audience to access and understand the model's content.

In addition to the CMSD UML model, a version of the CMSD information model is being created using the XML Schema definition (XSD) language [12]. XML [13], a specification supported by the World Wide Web Consortium (W3C), is a format for structured documents, where the content of the document exists between named "tags" that describe the content. An XML schema defines a specific kind of document and specifies the allowable tags and the arrangement of allowable tags in that kind of document. The CMSD XML Schemas are a representation of the same model as the CMSD UML model. They facilitate the definition of XML instance documents containing information structured according to the CMSD information model and provide an easy to use mechanism for validating those documents.

4.3 CMSD UML Model Overview

The CMSD UML model is designed as a suite of interrelated collections of information modeled as UML classes contained within UML packages, presented visually as a series of UML class and package diagrams. The primary function of UML packages in the CMSD model is to partition and group, by major areas of manufacturing, the class definitions that realize related manufacturing concepts. When necessary to improve model clarity, some packages may be further partitioned into subpackages. The manufacturing concepts within each package are modeled as UML classes and the characteristics associated with each entity are modeled as UML class attributes. Attributes associated with a class may be defined directly in the class, or may be defined through an aggregation association with another class defined in the model. Within each package, a series of UML class diagrams is used to present details about the content of each class and its relationship to the other classes in the model.

A package diagram that shows the packages of the CMSD UML model is presented in Figure 2. The CMSD package defines packages that contain definitions for all of the classes and relationships that make up the CMSD information model. Although it does not directly define any classes, several nested subpackages are defined within the scope of the CMSD package. It is within these nested packages that the classes and relationships that define CMSD information are directly defined. Each of the CMSD package's subpackages defines a focused, cohesive set of classes and relationships for a specific subset of CMSD information. The top level of CMSD packages are the *Layout*, *Part Information, Production Operations, Production Planning*, *Resource Information*, and *Support* packages.

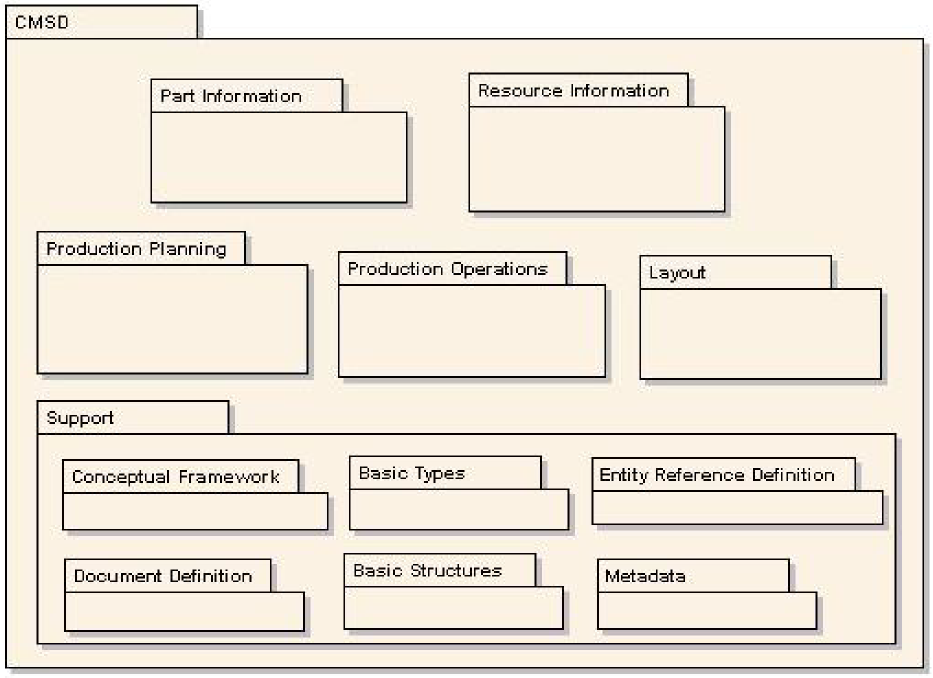


Figure 2 - CMSD Package Diagram

4.4 Example CMSD Class Diagram

Figure 3 presents the class diagram for two of the classes in the Resource Information package. Brief descriptions of these classes are provided below.

The *Resource* class defines information about a piece of equipment or an employee, or a collection of pieces of equipment and/or employees, that are used in the manufacturing process. An important thing to notice about *Resource* is that it is a subclass of the *UniqueEntity* class. *UniqueEntity* indicates that particular entity is unique within a given CMSD document (the main unit of storage or exchange for CMSD information). *Resource* instances can be uniquely identified and differentiated from each other by the value of their Identifier attributes. Subclasses of *UniqueEntity* inherit a required Identifier attribute along with several optional attributes. Also, *Resource* has a required attribute named *ResourceType* that describes, in the general, the kind of manufacturing asset or assets the resource represents. This attribute indicates its associated resource belongs to one of twelve general categories for manufacturing resources. Examples of the categories are machine, station, employee, and fixture. Other information that can be defined for a resource includes the current operational state of the resource, the gross dimensions of the resource, the "class" to which a resource belongs, and information about the group members for resources that define a specific identifiable collection of other resources.

Attached to *Resource* is a UML Constraint definition. It indicates that only *Resource* instances whose *ResourceType* is "employee" are allowed to contain information about employee skills, and that Resource instances that have setup information (information about the current configuration of a piece of equipment) may not have a *ResourceType* of employee.The *ResourceClass* class provides a means to create a classification scheme for resources based on descriptions of the characteristics that those resources possess. Like *Resource*, *ResourceClass* is a subclass of *UniqueEntity*, indicating that a *ResourceClass* instance can be uniquely identified and differentiated from other *ResourceClass* instances by the value of its inherited Identifier attribute. It also has a required *ResourceType* attribute and several optional attributes.

The reason that both *ResourceClass* and *Resource* are needed in CMSD is that sometimes the manufacturing entity being defined is a description of a "kind" of resource (e.g., a variable speed milling machine model 123 from company XYZ) and sometimes the manufacturing entity being defined is a specific instance of a resource (e.g., the variable speed milling machine model 123 from company XYZ with serial number 11223 bought on 12/31/04).

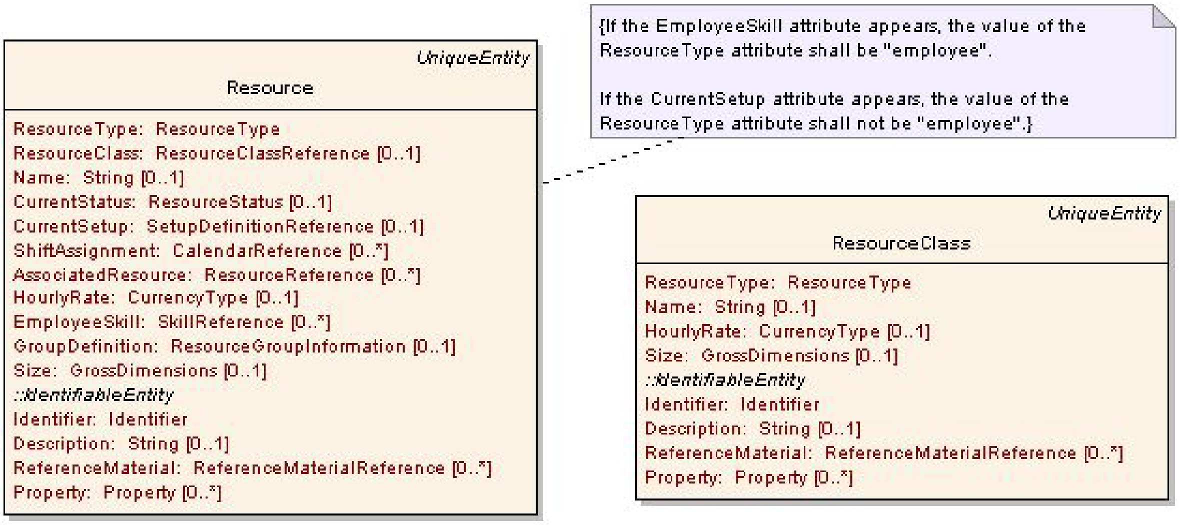


Figure 3 - The *Resource* and *ResourceClass* UML Diagram

4.5 Example Resource Information in XML

In Figure 4, a fragment of an XML document is shown that contains resource data organized according to the CMSD specification. In general, the class name and attributes of the Resource class are represented as XML elements in the fragment. XML elements are present to hold information from the required Identifier and ResourceType attributes of the Resource class, as well as for the optional Size and Description attributes. In addition several Property XML elements are present. These elements reflect the fact that Resource class instances may have information for 0 to many Property attributes. Property attributes are an extension mechanism in CMSD that provides a way for many classes in CMSD to add information about their associated manufacturing entities for which attributes are not predefined.

The fragment shown is only intended to illustrate how resource data should look in a CMSD document instance. Because of the amount of material involved, the complete format of a CMSD document and the XML Schema that would be used to validate the document are not presented here.

5. A Case Study that used CMSD to accomplish simulation integration

To validate its ability to be used as the basis for integrating simulation applications with other manufacturing applications, CMSD has been used in several case studies. One such project was undertaken by Volvo Trucks, Chalmers University of Technology, and the National Institute of Standards and Technology (NIST) [14]. Figure 5 shows the engine line assembly process that includes two parallel lines and nine workstations for each line.

The simulated operation of the production line in each simulation was to be based on cycle time and breakdown information derived from historical data stored in several databases. This requirement indicated that to study the line's operation during different time periods, some mechanism to integrate the simulations with the historical data would be necessary. Often, to meet such requirements two point-to-point integration efforts are necessary. By using CMSD, information from the different historical data sources could be collected and stored in a CMSD formatted XML file, and each simulation could then import the data. With this approach, the integrated system could be used to evaluate the production lines operation over many more time periods than would otherwise be possible. Also, evaluations of the performance of the two different simulation applications were ensured of being based on the simulations' execution using the same input data. A diagram of the integrated system is shown in Figure 6. For a detailed description of the project see [15].

```xml
<Resource>
    <Identifier>123abc</Identifier>
    <Description>Variable speed lathe
        in bldg 101</Description>
    <ResourceType>machine</ResourceType>
    <Size>
        <Unit>foot</Unit>
        <Width>6</Width>
        <Depth>3</Depth>
        <Height>4</Height>
    </Size>
    <Property>
        <Name>SerialNumber</Name>
        <Value>111222333</Value>
    </Property>
    <Property>
        <Name>ModelNumber></Name>
        <Value>XYZcorp-lathe-77</Value>
    </Property>
    <Property>
        <Name>MinimumSpindleSpeed</Name>
        <Unit>RPM</Unit>
        <Value>100</Value>
    </Property>
    <Property>
        <Name>ObservedMTBF</Name>
        <Description>The distribution that describes the
            likelihood of this machine failing</Description>
        <Unit>hours</Unit>
        <Distribution>
            <Name>weibul</Name>
            <DistributionParameter>
                <Name>shape</Name>
                <Value>10</Value>
            </DistributionParameter>
            <DistributionParameter>
                <Name>scale</Name>
                <Value>1000</Value>
            </DistributionParameter>
        </Distribution>
    </Property>
</Resource>
```


Figure 4 - An XML Fragment with Resource Data

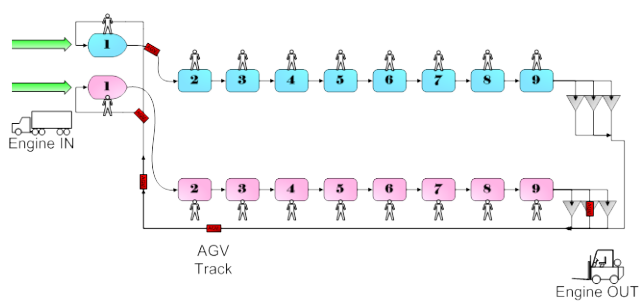


Figure 5 - An Engine Line Assembly Process Simulation

(Source: Johansson et al. [14])

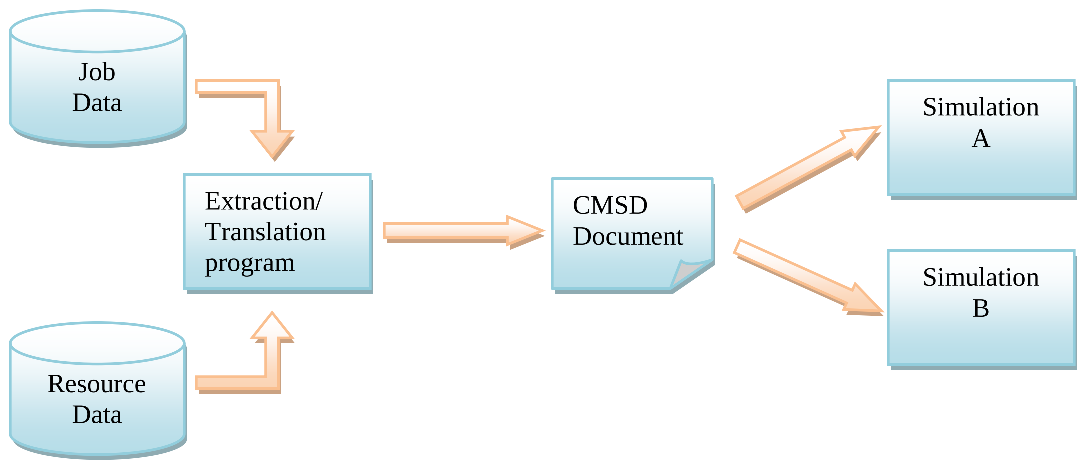


Figure 6 – Integrated Engine Line Evaluation System

6. Conclusions

Currently, small machine shops do not typically use simulation technology. A key limiting factor in the use of simulation technology is access to the data that defines the important aspects of manufacturing operations. Much of the data exists in other manufacturing applications, but the data is often incomplete or inconsistent between different applications. This makes the process of identifying, defining, extracting, and exchanging the data needed for simulation applications to be costly and time consuming. In turn, data incompatibility issues make simulation creation, simulation reuse, and integration between simulation and other manufacturing applications too costly for many manufacturers to undertake. The CMSD specification helps to resolve some of the long-standing interoperability issues that make integrating simulation and other manufacturing applications difficult and expensive. Standards like CMSD help to reduce the systems integration costs incurred by manufacturers. This paper presents information about the CMSD specification, from its conceptual underpinnings, to the CMSD information model that is being standardized, to an example of how it has been used to integrate simulations with other manufacturing applications.

7. References

[1] National Research Council: "Committee to Study Information Technology and Manufacturing. Information Technology for Manufacturing: A Research Agenda," National Academy Press, Washington, DC, 1995.

[2] National Research Council: "Committee on Visionary Manufacturing Challenges. Visionary Manufacturing Challenges for 2020," National Academy Press, Washington, DC, 1998.

[3] ISA-95.com: "ISA-95: The International Standard for the Integration of Enterprise and Control Systems," http://www.isa-95.com/, accessed on May 12, 2010.

[4] Open Applications Group (OAGi): "The Open Applications Group Integration Specification (OAGIS)," http://www.openapplications.org/, accessed on May 12, 2010.

[5] Lee, Y. T., Riddick, F., McLean, C., and Leong, S.: "Current Activities Related to the Core Manufacturing Simulation Data Standards Development Effort," In Proceedings of the Fall 2008 Simulation Interoperability Workshop, Orlando, FL, September 14-19, 2008.

[6] Simulation Interoperability Standards Organization (SISO): "Standard for: Core Manufacturing Simulation Data – UML Model, SISO-STD-008-2010-DRAFT-V1.0," http://www.sisostds.org/, accessed on April 30, 2010.

[7] Gartner Group: "Gartner Identifies the Top 10 Strategic Technologies for 2010", http://www.gartner.com/, accessed on May 12, 2010.

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

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

[10] Martin, J. and McClure, C.: "Diagramming Techniques for Analysis and Programmers," Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1985.

[11] Object Management Group: "Introduction to OMG's Unified Modeling Language," http://www.omg.org/gettingstarted/what_is_uml.htm, accessed on May 12, 2010.

[12] Van der List, E.: “XML Schema,” O’Reilly& Associates, Sebastopol, California, 2002.
[13] World Wide Web Consortium: “Extensible Markup Language (XML),” http://www.w3c.org/XML, accessed on May 12, 2010.
[14] Johansson, M., Johansson, B., Leong, S., Riddick, F., and Lee, Y. T.: “A Real World Pilot Implementation of the Core Manufacturing Simulation Data Model,” In Proceedings of the 2008 EURO Simulation Interoperability Workshop, Edinburgh, Scotland, UK, June 16-19, 2008.
[15] Johansson, M., Leong, S., Lee, Y. T., Riddick, F., Shao, G., Johansson, B., Skoogh, A., and Klingstam, P.: “A Test Implementation of the Core Manufacturing Simulation Data Specification,” In Proceedings of the 2007 Winter Simulation Conference, 1673-1681, Institute of Electrical and Electronics Engineers, Inc, Piscataway, New Jersey: 2007.

ACKNOWLEDGMENT: Work described in this report was sponsored by the National Institute of Standards and Technology (NIST), Gaithersburg, Maryland. The work supports NIST projects applying information technologies and standards-based approaches to manufacturing software integration problems. The work described was funded by the United States Government and is not subject to copyright.

DISCLAIMER: Company names and products may have been identified in the context of this paper. This does not imply a recommendation or endorsement of the software products by the authors or NIST, nor does it imply that such software products are necessarily the best available for the purpose.

AUTHOR BIOGRAPHIES

FRANK RIDDICK is a computer scientist in the Manufacturing Simulation and Modeling Group in The National Institute of Standards and Technology (NIST) Manufacturing Systems Integration Division. He has participated in research and authored several papers relating to manufacturing simulation integration and product data modeling. He holds a Master's Degree in Mathematics from Purdue. His email address is <frank.riddick@nist.gov>.

Y. TINA LEE is a computer scientist in the Manufacturing Simulation and Modeling Group at NIST. She joined NIST in 1986. Her major responsibility in recent years has been to develop information models to support various manufacturing application areas. 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@nist.gov>.