INFORMATION MODELING: FROM DESIGN TO IMPLEMENTATION

Y. Tina Lee
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001, U.S.A.
Email address: ylee@nist.gov

Today's manufacturing industry greatly relies on computer technology to support activities throughout a product's life cycle. Effective and efficient information sharing and exchange among computer systems have been critical issues. Formal information modeling languages that describe information requirements unambiguously is an enabling technology that facilitates the development of a large scale, networked, computer environment that behaves consistently and correctly.

This paper describes an information modeling process. Emphasis is placed on the need for the implementation of "quality" information models to facilitate an integrated manufacturing environment. How information models are used to define data requirements and how, in a practical application, information models enable information sharing and exchange are described. Several information modeling methodologies, modeling languages, and implementation methods are reviewed. In addition, recommendations on building practical information models are presented.

Keywords: data exchange, information modeling

1. INTRODUCTION

In 1970s, the Standards Planning and Requirements Committee (SPARC) of the American National Standards Institute (ANSI) published a three-schema architecture for database management systems [1]. The three schemas include an external schema (the user view of the information), an internal schema (the computer view of the information), and a conceptual schema (a logical neutral view of the information.) The conceptual view is a single, integrated definition of the data within an enterprise that is unbiased toward any single application of data and independent of how the data is physically stored or accessed. It provides a consistent definition of the meanings and interrelationship of the data in order to share, integrate, and manage the data. The need to define conceptual schemas has led to the development of semantic modeling techniques. An important benefit of having a fully developed, semantic information model is that the model can be used to define various applications and build sharable databases. During the 1970s, the relational data model was introduced to represent the conceptual schema level [2]. As the relational database management system (DBMS) design techniques grew, the need to design shared databases was recognized. Information modeling techniques provide a way to develop specifications for shared databases. These modeling techniques are useful for improving the quality of a database design.

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 sharable, stable, and organized structure of information requirements for the domain context. An information modeling language is a formal syntax that allows users to capture data semantics and constraints. In 1976, an Entity Relationship graphic notation was introduced to develop relational data models [2]. Since then, languages for information models have continued to evolve: the Integrated Computer Aided Manufacturing (ICAM) Definition Language 1 Extended (IDEF1X) [3], the EXPRESS Language [4,5], and the Unified Modeling Language (UML) [6] are some examples.

The combination of emerging technologies, global competition, and market diversification is imposing a great need for transferring information timely and reliably. Information models that support an integrated manufacturing environment are either not available or just for limited and proprietary usages. An international effort for standardizing the representation of product information to support the life cycle of products in diverse industries, under the informal name of STandard for the Exchange of Product model data (STEP) [7], has been under way and led by the International Organization for Standardization (ISO). The National Institute of Standards and Technology (NIST) has also launched two large programs, System Integration of Manufacturing Applications (SIMA) [8] and National Advanced Manufacturing Testbed (NAMT) [9], to support the U.S. industry in the area of information-based manufacturing. SIMA addresses the information interface needs of the U.S. manufacturing community. NAMT is an effort to build a showcase for the future of manufacturing that will enable U.S. manufacturers and their suppliers to rapidly introduce affordable and quality products.

This paper documents the author's experience in information modeling gained by developing information models for manufacturing domains, such as plant layout [10], process planning [11], discrete-event simulation [12], and apparel pattern making [13] for the SIMA and NAMT Programs. It describes how to develop information models and how to make the information models useful for the application development environment. Section 2 describes modeling methodologies. Section 3 describes some formal modeling languages. Section 4 outlines procedures to develop information models. Section 5 describes implementation methods and implementation issues with recommendations. Section 6 presents the process of developing a real-life information model. The final section presents the conclusions of the paper.

2. MODELING METHODOLOGIES

Information modeling is a technique for specifying the data requirements that are needed within the application domain. There are different practices in developing an information model. The underlying methodologies for the recent modeling practices are based on three approaches: the entity-relationship (ER) approach, the functional modeling approach, and the object-oriented (O-O) approach. The ER approach focuses on how the concepts of entities and relationships might be applied to describing information requirements. The emphasis of the functional modeling approach is placed on specifying and decomposing system functionality. The O-O approach focuses on identifying objects from the application domain first and then operations and functions.

The ER approach is based on a graphical notation technique [2]. Various ER extensions have been introduced since then. The basic constructs in an ER model are the entity type, the relationship type, and the attribute type. The notation is easy to understand and the technique has been useful in modeling real problems. There are commercial tools available to map ER models into commercial DBMSs.

The functional approach addresses the system's processes and the flow of information from one process to another. It uses objects and functions over objects as the basis. The approach often uses data-flow diagrams. A data-flow diagram shows the transformation of data as it flows through a system. The diagram consists of processes, data flows, actors, and data stores. This approach has been in wide use.

In the objected-oriented approach, the fundamental construct is the object, which incorporates both data structures and functions. The building blocks in the O-O model are object classes, attributes, operations, and associations (relationships.) The objected-oriented approach has the following advantages: easier modeling of complex objects, better extensibility, and easier integration with O-O DBMS and O-O programming code.

Choosing an appropriate modeling methodology is a judgment that must be made at the beginning of the modeling work. In general, an information model, developed in any methodology, is a representation of entities, attributes, and relationships among entities. However, each information model has a different emphasis. The emphasis often depends on the viewpoint associated with the model. Occasionally there are multiple viewpoints for the model. The viewpoints of the model help to decide the type of information modeling methodology to be used. For example, the ER approach is a better selection if data requirements are at the higher levels of detail. In the case where functions are more important and more complex than data, the functional approach is recommended. The O-O approach, however, may provide better extensibility and may be more compatible with the intended implementation environment. The disadvantage of the ER model is its lack of preciseness in supporting the detailed levels. Very often the data requirements of the application may need to be changed and most changes are function related; if the information model was developed using the functional approach, these changes may lead to a major modification to the model. Finally, the major obstacle for using the O-O approach is that the approach requires a critical paradigm shift in thinking compared with other data modeling approaches -- from considering only the data to considering both the data and the functions.

3. MODELING LANGUAGES

Quite a few information modeling languages, for different methodologies, have been developed or are under development. These information modeling languages provide various ways of formally representing an information model. In general, the languages are presented in two forms: graphical form and textual form. The former is represented by diagrams that are formed by graphic symbols. The structure of the latter is specified by a context-free grammar that includes formal language syntax and semantics. The graphical form is designed primarily for humans, while the textual form is for both humans and machines. This report concentrates on three modeling Languages: IDEF1X, EXPRESS, and UML. The reason these languages are chosen is three-fold: they are formal languages, they are either standardized or in the public domain, and they are most frequently used in the manufacturing areas.

The ICAM Definition (IDEF) Language was developed from the U.S. Air Force ICAM Program during the 1976 to 1982 timeframe [3]. The objective of the ICAM Program was to increase manufacturing productivity through the systematic application of computer technology. IDEF includes three different modeling methods: IDEF0, IDEF1, and IDEF2 for producing a functional model, an information model, and a dynamic model respectively. IDEF1X is an extended version of IDEF1. The language is in the public domain. It is a graphical representation and is designed using the ER approach and the relational theory. It is used to represent the “real world” in terms of entities, attributes, and relationships between entities. Normalization is enforced by KEY Structures and KEY Migration. The language identifies property groupings (Aggregation) to form complete entity definitions.

EXPRESS was created as ISO 10303-11 for formally specifying information requirements of product data model. It is part of a suite of standards informally known as the STandard for the Exchange of Product model data (STEP). It was first introduced in the early 1990s [4,5]. The language is a textual representation. In addition, a graphical subset of EXPRESS called EXPRESS-G is available. EXPRESS is based on programming languages and the O-O paradigm. A number of languages have contributed to EXPRESS. In particular, Ada, Algol, C, C++, Euler, Modula-2, Pascal, PL/1, and SQL. EXPRESS consists of language elements that allow an unambiguous object definition and specification of constraints on the objects defined. It uses SCHEMA declaration to provide partitioning and it supports specification of data properties, constraints, and operations.

UML is a modeling language for specifying, visualizing, constructing, and documenting the artifacts, rather than processes, of software systems [6]. It was conceived originally by Grady Booch, James Rumbaugh, and Ivar Jacobson. UML was approved by the Object Management Group (OMG) as a standard in 1997. The language is non-proprietary and is available to the public. It is a graphical representation. The language is based on the objected-oriented paradigm. UML contains notations and rules and is designed to represent data requirements in terms of O-O diagrams. UML organizes a model in a number of views that present different aspects of a system. The contents of a view are described in diagrams that are graphs with model elements. A diagram contains model elements that represent common O-O concepts such as classes, objects, messages, and relationships among these concepts.

IDEF1X, EXPRESS, and UML all can be used to create a conceptual model, and each has its own characteristics. Although some may lead to a natural usage (e.g., implementation), one is not necessarily better than another. In practice, it may require more than one language to develop all information models when an application is complex. In fact, the modeling practice is often more important than the language chosen.

4. INFORMATION MODEL DEVELOPMENT PROCESS

This section describes the process for developing a "quality" information model. By a "quality" information model we mean that the model is complete, sharable, stable, extensible, well-structured, precise, and unambiguous. In general, the contents of an information model include a scope, information requirements, and a specification. The detailed explanations on each area are given in the following paragraphs.

The initial phase for developing an information model starts with the definition of the scope of the model's applicability. The scope specifies the domain of discourse and the processes that are to be supported by the information model. It is a bounded collection of processes, information, and constraints that satisfy some industry need. The scope statements include the purpose as well as viewpoints of the model, the type of product, the type of data requirements, the supporting manufacturing scenario, the supporting manufacturing activities, and the supporting stage in the product life cycle. The scope definition may be supported by an activity model and/or a data planning model. An activity model is a representation of the application context, data flows, and the processes of the application. It is a mechanism for gathering high level information requirements. A data planning model provides a high level description of the data requirements for the information model, as well as the relationships among the basic data components. It is used as a roadmap to establish interfaces across a wide range of data. A well-defined scope should be accurate, unambiguous, viable, and meet the industrial need. During the course of the modeling, the scope should be revisited and may be refined. Since the scope provides the boundaries of the application domain, it also serves as a guildine for evaluating the "completeness" of the information model.

After the scope is defined, the next phase is to conduct a requirements analysis. There is no standard method for collecting information requirements. However, requirements analysis may be accomplished by: literature surveys, standards surveys, domain experts' interviews, industrial data reviews, and state-of-the-art assessments. Depending on the scope, the analysis may include today's manufacturing practices, traditional practices, and near-future needs. It is important to capture data requirements accurately for the application scope while performing the requirements analysis. Industry reviews of the result of the analysis will help to ensure the completeness and correctness of the information requirements. As the result of the requirements analysis, information requirements should be documented. The definition of each identified information item should be included in the document. This document will be a strawman for developing the information model.

After the detailed scope and information requirements are defined, the next phase is to develop the model. This phase transforms information requirements into a conceptual model. The information model is independent of any physical implementation, and it should be developed using a formal modeling language. Each information requirement should be expressed in the model. The model should be sufficiently detailed to describe the data needs of the application fully.

To actually develop the information model, three types of design approaches can be taken: a top-down design, a bottom-up design, and a mixed or inside-out design. While the most effective way is to take the top-down design approach for modeling, it may not be possible or appropriate in all cases. An optimal design approach may depend on the individual application environment. Conceptualizing information requirements starts with grouping concepts, that is, to identify the model's units of functionality. After that, an abstraction process will be performed to establish the model's structure for each functionality. This abstraction process, which structures information requirements into entities, objects, or classes, may include generalization, specialization, aggregation, classification, and association. Classification is the grouping of objects with the same data structure and operation. Generalization, specialization, aggregation, and association are for establishing relationships among the model's elements. Generalization and specialization identify the "inheritance-from" and "inheritance-to" relationships. Aggregation identifies "subset-of" relationships. Association identifies "dependency" relationships. Once the structure of the model is established, it must then be laid out according to the syntax of the selected modeling language.

5. IMPLEMENTATION METHODS AND ISSUES

An information model provides a sharable, stable, and organized structure of information requirements. It is developed to preserve independence from both usage and implementation. Implementation independence allows users to select their implementation methods. Three types of implementation methods are currently used by the manufacturing community:

1) Data transfer via a working form, which is a structured, in-memory representation of data,
2) Data transfer via an exchange file, which is a file with a predefined structure or format, and
3) Data transfer using a database management system.

These implementation methods can be accomplished through programming languages and DBMSs. Method 1) uses the mechanism that accesses and changes data sequentially without actually moving the data around. All shared data are stored in memory. Method 2) requires a neutral file format for storing the data. The application systems read and write from files. Method 3) uses a DBMS where information is mapped onto and retrieved from databases. DBMSs' access methods are generally through either libraries of routines or embedded data access/manipulation languages. The types of DBMSs used include O-O DBMSs and relational DBMSs.

The selection of an implementation method is heavily dependent on the target environment where the application system resides. While the relational DBMS is generally desirable for data transfer, the traditional file-oriented systems are being used continually by many manufacturing applications. An O-O model is more easily implemented using an O-O language or an O-O DBMS; however, it can also be implemented using a conventional programming language or a relational DBMS [14].

A few lessons learned are described here.

a) Information requirements serve as the foundation of the model. A thorough requirements analysis is a necessity. Literature surveys, standard surveys, domain experts interviews, industrial data reviews, and state- of-the-art assessments are a source of capturing knowledge. Workshops are a good way to gather requirements and sometimes even to reach a consensus on the requirements.

b) Modeling is an iterative process, as refinements are often necessary. As iteration continues, the information model obtained at the end of each iteration is presented to the user community to obtain further feedback. Based on the feedback, either another iteration starts or the information model is cast in concrete.

c) It is useful to establish a set of naming conventions for a big and complex model in the beginning of the modeling effort. The naming conventions should be descriptive in nature. Advantages for using naming conventions are: consistency, ease of identifying entities, and ease of collaboration.

d) Developing a glossary of terms that are used by the applications is also useful. The purpose of the glossary is to provide a unique definition for each term to eliminate improper use due to conflicting definitions.

e) There are several common problems during the implementation process. The most fundamental effort is that if a particular information model serves as the medium for transferring the data, the application system should be brought in to some degree of compliance with this information model. Occasionally, there is no complete data mapping between the model and the system. This may be due to the fact that data requirements are not a complete set, or some private data from certain application systems are not intended to be shared. If the data requirements are not complete, further requirements analysis should be conducted. For proprietary data, implementation-specific arrangements should be made.

f) Using different measurement units is another common error in an implementation. Under this situation, the attributes in different units should be included in the information model.

g) Conflicts in precision is another issue. The information model should declare the specified precision for numeric data. If the application system carries a lower precision, the accuracy may be lost.

h) Sometimes the same terms may have different meanings or different terms may have the same meaning. The glossary mentioned in item d) that precisely defines all terms presented with the information model is an effective solution to this problem.

i) Finally, having industry reviews of the information model is critical. It helps to ensure the model's necessity, correctness, and completeness.

6. EFFORT TO DEVELOP A REAL-LIFE MODEL

This section describes how a real-life information model, Pattern Information Model [13], was developed. The information model was developed to support the Defense Logistics Agency's apparel research program in the area of electronic commerce. The model is for the exchange of two-dimensional apparel patterns. The development of the model was an integrated effort from several tasks:

a) File format evaluation: In the late 1980s, the American Apparel Manufacturers Association (AAMA) took the position that the apparel industry urgently needed a mechanism for automatic transfer of pattern data and hence asked NIST and the Apparel CIM Center of the University of Southwestern Louisiana to develop a neutral data format for the representation of 2-D pattern pieces for the apparel industry. As a result, it was recommended that the AutoDesk DXF format [16] be used as the framework to develop a near-term, neutral format for pattern data. In addition, it was recommended that a STEP application model of apparel patterns be developed for the apparel industry as a long-term strategy [17].

b) Glossary development: A study on apparel manufacturing terms, especially those used in the pattern-making process, was performed at NIST. As a result, a working set of terms and definitions from published literature was created to act as a catalyst in the development of a consensus glossary [18].

c) Requirements analysis: A task to identify information requirements for apparel pattern making was performed at NIST. Efforts included visiting and consulting with apparel manufacturers, Defense Support Center Philadelphia[1], apparel institutes, traditional dressmakers, and tailors; reviewing existing standards; studying industrial data; and actively participating in activities held by the Apparel Research Committee of AAMA and the DLA Apparel Research Network. As a result, an activity model of pattern making was developed [15], and a preliminary set of data requirements was identified [13]. In addition, "A Survey of Standards for the U.S. Fiber/Textile/Apparel Industry" [19], "A Bibliography on Apparel Sizing and Related Issues" [20], and "Body Dimensions for Apparel" [21] were published.

d) Model layout: Mapping the data requirements to an EXPRESS model was the next step. The schema presented in [13] was developed through three major iterations. The experience gained through the implementation of the prototype information model and recommendations received from apparel researchers provided useful inputs for improving the early versions of the model. A prototype of the current model has been demonstrated using two military garments. The model now can be used as the initial proposal for developing an official specification. It can also be extended to include all the information necessary for an apparel product throughout its development life cycle.

7. CONCLUSION

We have described a flow for designing and implementing a quality information model. This flow starts with choosing the information modeling approach: ER, functional, or O-O. It proceeds to selecting the right combination of modeling languages. Once these tools for setting up the environment are chosen, we then define the scope of applications, determine the information

[1] Defense Support Center Philadelphia, formerly Defense Personnel Support Center, is a DLA organization. The organization is responsible for supplying patterns to government contractors.

requirements, and write down the conceptual data model using a formal data definition language.

In implementing the information model itself, which is to be shared by different components of a manufacturing process or exchanged among CAD/CAM systems, we have to determine if the data transfer is to be based on an in-memory storage structure, disk files, or a database management system. To keep the information model design and implementation process streamlined and out of chaos, naming conventions and a glossary should be established. Different requirements of numerical precisions and measurement units should be recorded and coordinated to maintain system flexibility. Finally, industrial review to obtain user feedback and expert opinions help to enhance the system performance and user satisfaction.

REFERENCES:

[1] Tsichritzis, D., and Klug, A., eds., “The ANSI/SPARC DBMS Framework Report of the Study Group on Database management Systems,” Infosystems, Vol. 3, 1978.

[2] Chen, P. P., “The Entity-Relationship Model - Towards a Unified View of Data,” ACM Transactions on database Systems, Vol. 1, No.1, March, 1976.

[3] D. Appleton Company, Inc., “Integrated Information Support System: Information Modeling Manual, IDEF1 - Extended (IDEF1X),” ICAM Project Priority 6201, Subcontract #013-078846, USAF Prime Contract #F33615-80-C-5155, Wright-Patterson Air Force Base, Ohio, December, 1985.

[4] ISO 10303-11:1994(E), Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: The EXPRESS Language Reference Manual.

[5] Schenck, D., and Wilson, P. “Information Modeling the EXPRESS Way,” Oxford University Press, New York, NY, 1994.

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

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

[8] http://www.mel.nist.gov/msid/sima.

[9] http://www.mel.nist.gov/namt.

[10] Lee, Y. T., “Initial Manufacturing Exchange Specification (IMES): Requirements Analysis for the Plant Layout Application,” NISTIR 6139, National Institute of Standards and Technology, Gaithersburg, MD, July, 1998.

[11] Ellis, K., Jones, A., and Lee, T., “Requirements Analysis: Process Plan Specification - Workstation Level,” NISTIR 6172, National Institute of Standards and Technology, Gaithersburg, MD, June, 1998.

[12] Bartolotta, A., McLean, C., Lee, Y. T., and Jones, A., “Production Systems Engineering: Requirements Analysis for Discrete-Event Simulation,” NISTIR 6154, National Institute of Standards and Technology, Gaithersburg, MD, April, 1998.

[13] Lee, Y. T., “Data Sharing Implementation Based on the Information Model for Apparel Pattern Making,” NISTIR 5969, National Institute of Standards and Technology, Gaithersburg, MD, January, 1997.

[14] Rumbaugh, J., Blaha, M., Premerlani, Eddy, F., and Lorensen, W., “Object-Oriented Modeling and Design,” Prentice-Hall, Inc., Englewood Cliffs, NJ, 1991.

[15] Lee, Y. T., “Extensions of the Prototype Application Protocol of Ready-to-Wear Apparel Pattern Making,” NISTIR 5727, National Institute of Standards and Technology, Gaithersburg, MD, October, 1995.

[16] AutoCAD Release 11 Reference Manual, AutoDesk, Inc., August, 1990.

[17] Efe, K., Delcambre, L., Steward, A., and Remedios, I., “Evaluation of Neutral Data Formats for the Representation of 2-D pattern Pieces,” USL A-CIM Technical Report #2, University of Southwestern Louisiana, LA, February, 1990.

[18] Read, M. E., “Apparel Manufacturing Glossary for Application Protocol Development,” NISTIR 5572, National Institute of Standards and Technology, Gaithersburg, MD, February, 1995.

[19] Pawlak, C. G., “A Survey of Standards for the U.S. Fiber/Textile/Apparel Industry,” NISTIR 5823, National Institute of Standards and Technology, Gaithersburg, MD, April, 1996.

[20] Lee, Y. T., " A Bibliography on Apparel Sizing and Related Issues," NISTIR 5365, National Institute of Standards and Technology, Gaithersburg, MD, February, 1994.

[21] Lee, Y. T., "Body Dimensions for Apparel," NISTIR 5411, National Institute of Standards and Technology, Gaithersburg, MD, April, 1994.