STANDARDS FOR ENGINEERING INFORMATION MODELS

Curtis H. Parks
General Dynamics/Pomona
1675 W. Mission Blvd.
Pomona, CA 91769-2507
(714)868-4923

ABSTRACT

There is no longer an issue that information modeling may be important to data definitions about products and as expressed in data transfer languages. Information models have been required as deliverables in several large military contracts focused on automation of product design and manufacture. The problems to be addressed by design standards committees is that of managing the wide range of information modeling languages, the products of several public domain efforts to produce such models, the criteria for evaluation, review, consensus and change, and the ownership of and credit for these models. Other issues to be addressed are the changes which result when models are integrated into a larger scope (e.g., electrical products as a part of electro-mechanical products), and the impact of semantics of information entities on data dictionary standards.

INTRODUCTION

An information model is a logical view of data used by an enterprise (also referred to as a domain of discourse). When the enterprise is a company or corporation there is little impact on the standards community. Recently the technique of information modeling has been applied to product definition-data[1]. The intended benefit is associated with the migration of data from design to computer integrated manufacturing (CIM). For the CIM application the "enterprise" includes CAE/CAD system, software, and database (DBMS) vendors, and the public domain transfer format. An information model held in common becomes a strategic part of the control architecture for integrating the data-driven applications.

The information model, in turn, has the potential to establish a syntactical and semantic base on which knowledge-based applications can be built and shared. Both information models and data dictionaries require a single meaning for each data element. In turn, the relationships in which the data element participates influence the element's definition. An example of this would be the use of "signal". It may be a difference in potential between two electrical ports, or alternatively, the name of a network. A model incorporating the entity "signal" is very different for each definition. If an entity is needed for each relation structure suggested by the definition, then each entity must have a unique name. When these meanings and data relations are held in common within the using community, higher-level issues such as CIM and knowledge-based systems can be addressed.

The information model of electrical product data can be considered a subject for standardization. The prior paragraphs suggest the benefits as a foundation for shared knowledge-based systems. The standardization of an information model has quite different implications from that of other levels of electrical information such as a digital bus configuration. These implications are based on the nature of the information model and the role it serves.

The model, when mature, is the most stable when compared to the using applications and the physical data structures which relate to the model. But an information model may be dynamically developed in a piecemeal fashion. The standardization process must allow for the evaluation of a part of the total enterprise, and a "level of stability" assigned based on the review and consensus that part has received.

In turn, as the scope of a model is expanded, either through growth or integration, the "non-core" elements may change to insure a data fit. An example is seen in the Electrical Functional Model submitted to the IEEE[2] in which an acknowledgement is made that when the scope is extended to consider versions of a functional design, new attributes would be introduced into many of the model entities.

There is a type of modeling language that is recognized as an intermediate form. These languages are more closely aligned to machine readability (rather than human) and usually include data constraints (e.g., 8 numeric digits) needed by a proposed physical usage. Intermediate languages suggest to a reviewer a familiar feeling based on the way they relate to an identifiable implementation. They are also easier to publish with their text-list syntax. Like an electrical design, changing and reviewing the schematic is much more effective than when working only with a netlist. The analogy between an intermediate language for data, and a netlist for an electrical design[3] suggests that the more cognitive data model should always be published. The intermediate form may be a recommendation for a common physical representation. In turn a DBMS itself may have an information model[4] and one or more intermediate forms.

The problem of which organization should define an information model consensus and management infrastructures must be addressed. At present this is a recognized need within the Design Automation Standards Subcommittee (DASS) chaired by Mr. Ronald Waxman, University of Virginia. The cooperative work on electrical product information modeling is being reported to him, and to his Information Modeling Working Group chaired by Mr. Stephen Piatz, Unisys Corp. In like fashion, the logical integration of the electrical product model with mechanical products and product geometry models is an objective of the Initial Graphics Exchange Specification/Product Data Exchange Specification (IGES/PDES) organization. Mr. Bradford Smith, National Bureau of Standards is chairman of this organization. His staff coordinates the meetings and work of the industry and vendor members of the open voluntary organization for IGES enhancements and testing. The PDES development is utilizing logical data modeling and is focused on shareable product data for the automated manufacturing environment.

The latter problems of consensus and management are discussed and reported in the mentioned organizations. The nature of the information model itself and the languages implications will be discussed in the remainder of this paper.

MODELING LANGUAGE PERSPECTIVE

An extensive study of many modeling languages was made in an ISO report[5]. Of those described, two have been investigated for use in an open organization for product definition data[6]. These two are the most likely graphics-oriented languages to be encountered in public-domain product data models. Predicate logic may also find a use in product data modeling work.

NIAM

The Nijssen Information Analysis Method is a binary information language. Entities are represented by a circle around the entity name. The circle may be dashed indicating the enclosed word is a lexical entity. The entity may be connected to another entity by a line indicating a relationship exists. The relation line has a divided box inserted to contain the role labels. The labels allow the relation to be read (to form an English sentence) in either direction. Cardinality of the relation entities is not restricted and, further, groups of relations may have membership algebra notations.

$IDEF_1X$

The ICAM Definition, as published in the ISO reference, has been extended to include additional relational algebra and revised graphics which are easier to produce on a line-printer[7].

The $IDEF_1X$ information modeling is an entity, attribute, relationship language. An entity is represented by a rectangle. The data name assigned the entity and an optional identification number is written above the rectangle. The rectangle may have rounded corners indicating an existence dependency on another entity. The attributes of the entity are written inside the rectangle. Those attributes which are necessary to uniquely identify an instance of the entity are called keys and are written above a line drawn across the rectangle. When two entities are related, a line is drawn from the bottom of the independent (parent) entity rectangle toward the lower positioned dependent (child) entity. This line ends in a "big dot" above the dependent entity. The big dot may have cardinality imposed by the relation to the dependent entity. For the independent entity exactly one instance will participate in a relation. This language is that of a relational structure. An entity's keys migrate to its dependent entity, and the model is subjected to the relational normalization rules. A special relation connection is provided to declare a generalization. In this structure, each dependent entity is a "type-of" its related independent entity.

PREDICATE LOGIC

A linguistic/mathematical method is found in some disciplines[8], and a number of variations can be found. These fall roughly into categories of "interpretive" or "first order", and some of these languages also utilize limited graphic constructs.

COMPARISON

All three language types follow the three-schema[9] paradigm of separating data for verification apart from using applications and the physical format. All three also provide mechanisms for integration into larger scope information models. The opportunity also exists for converting a model into a different language without loss of the vital semantics and data integrity constraints. The primary differences are in the way humans perceive and work with each language. In turn, NIAM and IDEF are supported by computer tools which assist model development and with the database definition in several database structures. The binary construct of NIAM is easy to learn, and easy to use as to construction of diagrams. It tends to be larger in terms of graphics structures than the other two. The $IDEF_1X$ diagrams require a bit more graphics discipline to establish a "level of abstraction" ordering to the diagrams. The inclusion of non-graphic attributes within entities is compact structurally, but that advantage is somewhat offset by the key attribute migration requirement. The predicate logic types are rich in terms of expressive power, which in turns tends to decrease the probability of a good review by subject area experts who find data schemas remote to their subject interests. In verification by "structured walk-through" of NIAM OR $IDEF_1X$, however, the models are read into sentences which are predicate logic in nature.

KEY ISSUES

Information models belong to, and function to provide a key part of the control architecture of an enterprise. When the enterprise of those who wish to share data is industry wide, that definitive material becomes a common, valuable resource wherein the IEEE standards committee functions as the resource manager. That management task will require review, publish and consensus assessment functions for contributed models. The usual method of one-time review and publishing will likely require modification due to the piecemeal-evolutionary nature of the subject. The cognizant organization may also be required to effect model integration and arbitration activity. The most challenging task of such an organization will be that of identification of models, or model parts which are conceptual. Data model conceptuality is not presently well understood or easily qualified, but relates to the degree of independence from the using applications of the data (including their entry languages), and associated physical databases.

A non-conceptual information model will unnecessarily restrict the evolution of applications which depend on the data syntactical constraints of the model. In turn, the model validity is at risk.

The standardization of the language used in information models is held by many to be less important or even not required. The reasons given are that all languages seem to be capable of defining the required data semantics and integrity constraints. They differ in such areas as human-cognitive, physical structure mapping requirements, and their differing ability to identify the degree of conceptuality. More experience with information modeling language translation and verification may be required to address the language issue.

Much of the database interest has turned to object-oriented data structures[10]. The process of creating data objects begins with the set of data elements (an information model) and aggregation is applied as would optimize the physical structure for a target application. It follows that any number of object-oriented structures can be assembled from a conceptual information model. As with transfer formats or other data views, the model will function as the common control element.

Two important models have been contributed to the IEEE/DASS. The first has the scope of electrical functionality (functional hierarchy, connectivity and input/output voltage values). The functional model was completed by an industry cooperative task team during the first quarter of 1987. The "Engineering Information Systems" Air Force program will likely add behavioral data elements. The second model addresses the features and topology of layered physical electrical products. It has been developed by an industry team during the second half of 1987.

Contributed and integrated models are a resource of data semantics for use in a database dictionary standard. The dictionary is vital to the sharing of data among using applications. The definition of each data item involved in a model will have been reviewed and determined correct for use in application data structures.

CONCLUSION

The introduction of data models as stardards discipline represents a new and vital challenge for the standards community. The rewards in terms of data definition stability for building higher-level data applications cannot be over-emphasized. The data modeling methodology presents a means for the capture of complex and abstract data structures. When consensus is reached through the standardization process for the subject data models, a foundation is achieved for a common understanding of the discipline of concern. This is a step toward a shared-data future in electrical product design, analysis and fabrication technology. Other product types are also beeing addressed for integration into a complete span of product logical definition. The key to successful data model standards relates not to languages as much as to what they codify.

REFERENCES

1. MacDonnell Douglas Co., Product Definition Data Interface Specification, Air Force Materials Laboratory AFWAL/MLTC, Document SS-560130100, Contract F33615-82-C-5036, 31 July 1984.

2. Gale, R., Parks, C., "IEEE/PDES Cal poly Task Team Final Report" Unpublished, 30 April 1987.

3. Parks, C., "Tutorial: Reading and Reviewing The Common Schema for Electrical Design Analysis" 24^{th} ACM/IEEE Design Automation Conference Proceedings, IEEE No 87CH2422-4, July 1987.

4. Jefferson, D., Fong, E. (eds) "Reference Model for DBMS Standardization" National Bureau of Standards, NBSIR-85/3173, May 1985.

5. Van Griethuysen, J., (ed) "Concepts and Terminology for the Conceptual Schema and the Information Base", ISO/TC97/SC5/WG3 publication ISO/TC97- SC5-N695, 15 March 1982.

6. Kelly, J., "Final Report of the Logical Layer Initiation Task" Unpublished, IGES/PDES office, National Bureau of Standards, Gaithersburg MD 20899, May 1986.

7. DACOM, Inc., "Information Modeling Manual IDEF1 Extended, ICAM Architecture" Part II Volume 5 revised Dec 1986, ICAM-CIM Library Wright Aeronautical Laboratories AFWAL/MLTC, Wright- Patterson AFB, OH, 45433- 6533.

8. Nilsson, N., Principles of Artificial Intellegence, Chapter 4, Tioga Publishing Co, Palo Alto CA, 1980.

9. Tsichritzis, D., and Klug, A. (eds.), "The ANSI/ X3/ SPARC DBMS Framework. Report of Study Group on Data Base Management Systems" AFIPS Press, Montrale, NJ, 1977.

10. Heiler, S., Dayal, U., Oresstein, J., Radke-Sproull, S., "An Object Oriented Approach to Data Management: Why Design Databases Need It", 24^{th} ACM/ IEEE Design Automation Conference, IEEE Doc No 87CH2422-4, p335, July 1987.