EMERGING STANDARDS FOR DESIGN MANAGEMENT SYSTEMS

George Konstantinow, Ph.D.

Hughes Aircraft Corporation
Santa Barbara Research Center

ABSTRACT

The complex processes involved in designing and manufacturing electronic products require computer based engineering support. Design management systems providing such support not only help manage the large volumes of interconnected engineering data typically produced in electronic design but also control the production of designs, revisions to the design data, and designer access to tools and data.

This paper discusses standards for engineering design management and guidelines for building design management systems. Several models for managing electronic product data, managing the design process, and setting up integrated design support environments are reviewed. These models are tied together in an overview of the efforts of the IEEE DASS Working Group on Design Management, tasked with producing Recommended Practices for all aspects of design management.

INTRODUCTION

Requirements for design management

The processes involved in designing and manufacturing electronic products are so complex that they are virtually impossible without the support of integrated computer based engineering systems. To address the obvious need for such systems, current research and development in design automation is focusing heavily on producing standards for hardware description languages and categorizing the types of information encountered in the electronic design process.

A central problem in the search for design automation standards, however, is to determine not only how to automate design activities in engineering but also how to manage the large volume of intricately interconnected data that result. As automated design tools proliferate and become more sophisticated, it becomes easier for designers to create and modify designs quickly. At the same time, it becomes all the more important to have engineering data management systems in place to control design revisions, coordinate the activities of design team members, and monitor the delivery of design products.

Sources for design management standards

A well-constructed design management system serves to identify engineering data, control how those data are produced, and give the designer reliable access to design tool sets. Just as standards in the disciplines of software configuration management and database management allowed far-reaching advances in software engineering and data processing methods, so will the establishment of standards for design management (especially for electronic product development) lead to progress towards the goal of truly integrated design and manufacturing systems.

This potential is fully recognized by various standards organizations. The IEEE is developing the VHSIC Hardware Description Language (VHDL) as a standard and is investigating support environments through the Design Automation Standards Subcommittee (DASS) of the Design Automation Technical Committee (DATC).[1] The Electronic Industries Association (EIA) has approved the Electronic Design Interchange Format (EDIF) Standard and is investigating data management requirements through its G33/G34 committees.[2] The National Bureau of Standards (NBS) is developing the Initial Graphics Exchange Specification / Product Data Exchange Specification (IGES/PDES) for electronic product information.[3] The Department of Defense (DoD) is sponsoring the Engineering Information Systems (EIS) initiative to ensure that design management systems for military electronics and embedded systems projects successfully integrate design tools from multiple vendors and make them interoperable.[4]

Other groups are investigating requirements for design automation and design management with equal interest. De facto standards are emerging from the academic and industrial communities as a result of carefully planned, long-term development efforts.[5,6] Also, the market presence of software for design data and process control indicates that vendors are looking for marketable solutions to the design management problem and will be attentive to formally developed standards in their product planning.[7,8]

Design management standards issues

The purpose of this paper is to highlight issues to be considered in formulating standards for engineering design management. In the next section, several models will be reviewed which address requirements for managing electronic product data, managing the design process itself, and incorporating methodologies and tools into an integrated environment. In the section following that, the investigation of design management standards will be discussed in terms of the efforts of the IEEE DASS Working Group on Design Management (WGDM). The charter of the WGDM is to produce Recommended Practices for Management of Computer System and Hardware Design (targeted for IEEE certification in 1989). This group coordinates its activities and shares information with the remaining DASS working groups on VHDL standardization, information modeling, and system design.[9]

MODELS FOR DESIGN MANAGEMENT SYSTEMS

Data management

Electronic product design, like other CAD disciplines, is a data-intensive endeavor. As tools for electronically producing designs became available, it was recognized that database management system (DBMS) technology is required to track the multitude of drawings and ancillary files comprising a full design. Simple file management techniques are not sufficient.

Seen from the point of view of information modeling, however, generic DBMS's are also not sufficient for managing the types of data and their interrelations which come up throughout all phases of product design.[10,11,12] A major issue in establishing design management standards is determining representations of electronic design entities and data types which could be shared among different database models. For example, system components and their attributes could be identified well in a relational model but a hierarchical model might be better suited to describe their composition and interconnections.

A different view of design entity representation is given by an object-oriented approach.[13] Object orientation retains the benefits of "traditional" database models for identifying components and dependencies. This approach adds the ability to represent historical entities (design versions, revisions, and configurations) easily and to exploit abstraction and information hiding for representing the design entities themselves. The benefits of this approach for design management are that object-orientation is flexible in allowing the user to group components and define complex objects, it permits easy modifications to component designs with a minimum of impact on dependent designs (by virtue of information hiding), and it promotes reusability by the very way designs or components are represented (i.e. indexed, located, and documented).

The notion of reusability is important not just for object-orientation but for design management in general. Two types of data bases which emphasize shared data may be applied in a design management system. One is the repository, which is used dynamically during development. Of this type, a designer may use a public data base for accessing or submitting completed designs (where they are formally reviewed and controlled for release) and a private data base for modifying designs, checking alternatives, and controlling interim documents relating to phases of the design life cycle. Another data base type is the library, which is used during and after development and evolves more slowly than the first type. Of this type, a designer may rely on an archive to track design history, cell libraries to include components easily in designs, and libraries of standard (reusable) parts to tailor generic design objects to specific system requirements.

Process management

In building a design management system, it is important to focus not only on the data to be controlled but also on the multi-stage process by which designs are produced. As design tools proliferated and migrated to distributed workstations, it was recognized that automated control systems would be required to keep the interfaces among these tools straight and to track design information through well-defined design steps with formal review points.

The standard models applicable to the design process are data flow and structured process flow models. A design management system handles data at all stages of conceptual, logical, and physical design of the product and coordinates the transfer of those data between design steps. Data flow models may be used to identify input and output constraints for requirements analysis, schematic capture, placement and routing, simulation, and layout. It may even be appropriate to have the system electronically manage results or by-products of prototyping, system test, and fabrication stages, depending on how close the connections are between design and manufacturing in specific applications.

A model which extends the scope of data tracking through design stages is the configuration management (CM) system. Models for automated software CM are directly applicable to the engineering design process. The major functions served by a CM system are configuration identification (identifying and collecting related entities making up versions of a design), configuration control (controlling access and changes to design entities to assure that all design information is kept current), status accounting (tracking and reporting on design revisions and monitoring design dependencies among the process stages) and support for audits and reviews (preparing data and documentation for formal reviews and monitoring acceptance and sign-off procedures).

Configuration management addresses only a subset of requirements for product life cycle management. In an extended model for product management, three responsible areas exchange information: design engineering, configuration management, and quality assurance (QA). In this scenario, a design management system supports primarily the activities of QA to verify the consistency and completeness of the designs and all supporting information. The system automates delivery of design information to QA for such verification and automates maintenance of all relevant product control documentation provided by CM. The design management system may even incorporate expert systems to aid QA in the validation and verification process.[14]

Integrated environments

It is clear that a design management system addresses more than just data requirements for electronic product design or just control requirements for tracking the design process. Models for integrated design environments address the need (noted earlier) to identify design data, control data production, and maintain and allow access to design tool sets. For example, it is often important to know not only which version of which design is to be delivered for fabrication but also which version of which design tool (purchased or produced in-house) was used in the production of that design.

Of course, an integrated environment must be built upon a solid data base kernel addressing the particular design data requirements discussed earlier. That is, in building a design management system, one must analyze the data base schemata appropriate to user, logical, and physical views encountered in the engineering design process. An underlying data base kernel can be extended to account for design tools by incorporating a user interface, a procedural interface, a functional interface, a DBMS, and a design tool set. All user access and activities (including addition of new functions) are handled at the top level, the user interface. Each such activity is represented and implemented as a series of procedures and controlled at the second level, the procedural interface. A procedure in turn calls on a series of low-level functions at the third level, the functional interface, both to handle data base transactions and to invoke cataloged tools.

The benefits of a structured, integrated environment as described above are that it makes it easy to connect tools and procedures from various sources (exchangeable parts). The scenario presented is analogous to an Ada programming support environment in software engineering. In fact, VHDL, which itself is modeled after Ada in its typing and packaging constructs, calls for a support environment to manage both analysis tools and VHDL design libraries (coordinated with language functions, as is the case with Ada).[15] Such a model may be generalized as a three-level, layered "Design Support Environment" (DSE):

1) a kernel (KDSE) for controlling the design library and providing a logical interface to workstations, plotters, and other peripheral devices;

2) a minimal set (MDSE) for controlling and maintaining both design tools (schematic capture, placement and routing, layout, etc.) and management tools (configuration management, project management, requirements analysis, documentation, etc.); and

3) a full DSE for controlling top-level procedures (design creation, modification, maintenance, etc.), relying on functions from the levels below.

The Department of Defense (DoD) is particularly interested in the integrated environment model, as evidenced by its Engineering Information Systems (EIS) initiative.[16] The cornerstone of the EIS is an Engineering Information Model (EIM) satisfying, among other things, a requirement for a "neutral file" to which data formats from disparate tools may be translated. The availability of a central core of design data representations avoids the need to write direct interfaces pairwise between tools in the underlying tool set. Instead, each tool is accompanied by a translator to the controlled data base, ensuring that design tools from multiple sources may be integrated smoothly and new tools may be added without requiring a huge effort to write interfaces to existing tools. According to the EIS requirements, an EIM must be specified along with analysis of data flow and tool interoperability as a prerequisite for building a design management system.

Recognizing the need to establish integrated environments for design management, especially as concerns support environments for VHDL, the IEEE began investigation of design management issues and standards under the auspices of the DASS Working Group on Design Management (WGDM). The types of issues under consideration by that group are described next.

IEEE RECOMMENDED PRACTICE

The purpose of the DASS WGDM is to create a Recommended Practice for the management of electronic product development and related manufacturing information. The Recommended Practice covers all elements of design and manufacturing information accompanying design data which are required to allow control of the product over its life cycle. The Working Group will define applicable terminology, refer to related standards, and specify the types of design management information which should be transferred with the product.

The Design Management Recommended Practice is intended to offer guidelines for constructing design management systems without promoting any specific implementation methods. The basic requirement of a design management system as understood here is to manage electronic product development throughout the product life cycle. In that regard, to categorize practices and standards relating to design automation, the WGDM is investigating the following areas:

- Requirements
- Entities and Abstractions
- Hierarchies and Relations
- Transformations between Design Steps
- Rules
- Methodologies
- Tools
- Environment
- Design Management Planning
- Product Life Cycle Management
- Maintenance

In the task of building a design management system, each of these areas deserves special consideration in terms of information required, information produced, and procedures relating to product management:

Requirements
The system must provide for traceability to external customer requirements and be able to capture system engineering requirements relating to development. It is further desirable to have the system track changes to requirements automatically.

Entities and Abstractions
The system must accommodate different views of the design data (structural, functional, behavioral). It is further desirable to have a common data dictionary established and tailored to generic electronic design entities.

Hierarchies and Relations
The system must be able to reflect the hierarchical composition of electronic products (system, board, block, chip, circuit, gate, pin, etc.), relations among design components (interfaces-with, is-part-of, connects-to, etc.), and component attributes (part number, signal characteristics, timing, etc.).

Transformations between Design Steps
The system must provide control of the promotion of design data between design steps, recognizing procedures and constraints at each of those steps. The system must also be able to manage inputs and outputs, to control access to those data, and to allow status accounting for design changes at each of the steps.

Rules
The system must account for protocols followed during design, both for high-level functions (for example, collection of all pertinent design information required to pass from one design step to the next) and for low-level functions (for example, design rule checking in layout). It is further desirable to have non-procedural methods (for example, expert systems) implemented for various types of rule checking, particularly as applies to compliance with quality checks throughout the stages of design.

Methodologies
The system must allow the designer to identify a sequence of design steps appropriate to a particular design, to locate a corresponding tool set, and to determine all corresponding design products resulting from the application of those tools (and captured as part of the design data base).

Tools
The system must be able to manage tool sets for all phases of design both by identifying tools and controlling revisions or updates to tools within the design data base. It is further desirable to have a tool server as part of the system not only to control user access to tools but also to ensure that all designers are working with accepted versions of the tools.

Environment
The system must be able to meet usage requirements as constrained by the available engineering environment. These requirements relate to computing resources (computer network installed, development system used, and target system defined), project characteristics (life cycle assumed, project size, number of people, logistics employed), and availability of support (administrative organization, maintenance facilities, libraries).

Design Management Planning
System plans must indicate support for practices and procedures during design and development, quality assurance functions overseeing product design, and configuration management functions throughout the design life cycle. It is further desirable to have these plans included in the design data base as administrative information and to make them traceable throughout all design stages.

Product Life Cycle Management
The system must accommodate standards and procedures relating to the product life cycle and must be able to identify and control (for configuration management) all documentation relating to product development. It is further desirable to include such documentation as part of the design data base and to automate the process by which it is generated and reviewed.

Maintenance
The system must be structured for ease of maintenance internally to ensure that it remains available and effective across projects. It is further desirable to have mechanisms built-in to effect a smooth interface with systems from other contractors or agencies when applicable (for example, according to government data management standards).

SUMMARY

This paper has stressed the importance of considering the functions of a design management system when investigating design automation standards. Requirements for design management systems were reviewed in terms of models for engineering data management, design process management, and integrated environments incorporating methodologies and tools for design support. Issues currently under consideration by the IEEE in an effort to formulate engineering design management standards were then highlighted.

There is much activity currently under way to establish design management standards, especially in the critical and highly complex field of electronic product development. This interest is encouraging in that such standardization can only lead the industry closer to the ideal of reliable and effective computer-integrated manufacturing systems.

ACKNOWLEDGMENTS

I would like to thank John Graham, Brenda Ellis, Dave Rummler, and Joe Youmans, fellow members of the DASS WGDM, for their contributions to initial discussions on design management standards. I would also like to thank Darrel Lamb, a colleague at Hughes Aircraft, for his insights on the structure of design support environments.

For further information on the IEEE DASS Working Group on Design Management, please contact either co-chairperson:

John Graham
Attn: R154
Department of Defense
Ft. George Meade, MD 20755
(301) 688-6877

George Konstantinow, Ph.D.
Hughes Aircraft Corporation
Santa Barbara Research Center
75 Coromar Drive, B28 / MS 112
Goleta, CA 93117
(805) 562-7458

REFERENCES

1. R. Waxman, chair, "Design Data Representation and the DA Standards Environment," Proceedings 24th ACM/IEEE Design Automation Conference, Miami Beach, FL, June, 1987.

2. Electronic Design Interchange Format, Version 2.0.0, EIA Interim Standard No. 44, Electronic Industries Association, Washington, DC, 1987.

3. "PDES Development Forges Ahead," CAD/CIM Alert, June, 1987.

4. L. Linn and R. Winner, eds., The Department of Defense Requirements for Engineering Information Systems, IDA Paper P-1953, 2 vols., Institute for Defense Analysis, Alexandria, VA, July, 1986.

5. R.H. Katz, M. Anwarrudin, and E. Chang, "A Version Server for Computer-Aided Design Data," Proceedings 23rd ACM/IEEE Design Automation Conference, Las Vegas, NV, June, 1986, p. 27.

6. L.F. Saunders, "The IBM/VHDL Design System," Proceedings 24th ACM/IEEE Design Automation Conference, Miami Beach, FL, June, 1987, p. 484.

7. S.C. Baunach, "Project Management in the Technical Office," Computer-Aided Engineering, June, 1987.

8. J. Brouwers and M. Gray, "Integrating the Electronic Design Process," VLSI Systems Design, June, 1987, p. 38.

9. R. Waxman, "Design Automation Standards: The Promise and the Dilemma," IEEE Computer, June, 1987, p. 78.

10. R.H. Katz, "Managing the Chip Design Database," IEEE Computer, December, 1983, p. 26.

11. M.A. Ketabchi and V. Berzins, "Modeling and Managing CAD Databases," IEEE Computer, February, 1987, p. 93.

12. H.F. Korth and A. Silberschatz, Database System Concepts, McGraw-Hill, New York, 1986.

13. S. Heiler et al., "An Object-Oriented Approach to Data Management: Why Design Databases Need It," Proceedings 24th ACM/IEEE Design Automation Conference, Miami Beach, FL, June, 1987, p. 335.

14. R. Bhateja and R.H. Katz, "VALKYRIE: A Validation Subsystem of a Version Server for Computer-Aided Design Data," Proceedings 24th ACM/IEEE Design Automation Conference, Miami Beach, FL, June, 1987, p. 321.

15. M. Shahdad, "An Overview of VHDL Language and Technology," Proceedings 23rd ACM/IEEE Design Automation Conference, Las Vegas, NV, June, 1986, p. 320.

16. A.J. Gadient and J.L. Ebel, "Rationale for and Organization of the Engineering Information System Program," Proceedings 24th ACM/IEEE Design Automation Conference, Miami Beach, FL, June, 1987, p. 764.