MODELS FOR ENGINEERING INFORMATION SYSTEMS

Gio Wiederhold, John Smith, and Phillip Kaufman

1. Introduction

A meeting, just preceeding this VHSIC meeting will be held in Chicago (November 18,19 1985) to discuss the feasibility and approach to interface standards for Design programs to access Design databases, and deposit the design results into the database for use by others. We define database systems which can support such tasks as Engineering Information Systems (EIS).

During the oral presentation I will report on progress made at the meeting. The remainder of this note presents the basis for discussion of the interfaces which might become candidates for standardization,

2. Interfaces for Engineering Information Systems

The objective of this note is to provide a basis for the identification of interfaces in EIS. It is based in part on discussion held at an Air Force sponsored planning meeting, held at the Dayton Marriott Inn, 26 August 1985.

For the purpose of this note, we define an EIS more precisely as a system which encompasses software and hardware to aid management and engineering personnel in the specification, design, or acquisition, evaluation, validation, manufacturing, testing, and maintenance of engineered products. These products utilize digital, analog, and mechanical components and subsystems. An EIS to support these activities will include a large variety of software packages which have to be able to share information. An overview of the issues is provided by Randy Katz [Katz83]; the references section at the end lists a number of papers which define and address the issues raised by EISs.

The users of an EIS are mainly envisaged to be engineers involved in the design process, but can include sales engineers working on specifications and maintanance engineers obtaining and updating status records. Management access becomes important in the larger projects, where coordination of activities is a prerequisite to effective design. An EIS should enable all these professionals to focus on informed engineering decision-making, rather than on the capture, maintenance and distribution of the information itself. It should simplify the design of engineering tools, unify their application, and enhance their interoperability.

No complete integrated EIS as we visualize exists as of now; and we believe that the rapid advances in design technology make it unlikely that such a system can be effectively produced by any single source. To bring a choice of ever-improving EIS modules to the users standard interfaces should be defined. Prior to definition their position in hypothetical, but prototypical EISes has to be identified. The modules sketched in this note are intended to aid in the identification process.

3. Issues

An underlying relational model is typically used to represent the Information Resource (IR) of the EIS. Use of this database model makes the general literature on database theory accessible. However, EIS databases differ for commercial databases in that typical transactions take a substantial amount of time, leading to concepts of "long transactions" and "nested transactions". Also, the objects to be retrieved are not easily specified using traditional relational algebra, leading to "object-oriented data organizations". Thirdly, a comprehensive history has to be kept, leading to "version management" in addition to adequate "audit trails". We will expand slighly on the problems raised.

3.0.1 Transactions.

A Transaction is the unit of work performed from the point-of-view of the support system. In an reservations database such a unit may be a ticket sale; in an EIS it is likely to be a layout, a timing-simulation, a design change, etc. This means that typical transaction tend to be long. To permit sharing of design data it is now not possible to lock out other accesses during transaction which may change the database. Interface specifications may have to consider transfer of lock information.

Nested transactions are created when subtasks of major transactions are scheduled independently. A hierarchy of locks and status information is now required.

3.0.2 Objects.

An EIS typically deals in large units of information (e.g., a chip layout, or a set of simulation results) and must be concerned with maintaining the correctness of relationships between such information units. The objects can be composed both of distinct and of overlapping data element sets. A chip may be partitioned into distinct units for design. A power requirement estimation will deal with objects which use some data from all units. Accurate defintion of objects must be accomplished if an EIS is to keep track of design status and help assure consistency.

3.0.3 Versions.

An improved version of an object must be combined with matching objects to achieve a consistent design. Older versions must be retained, first untill a design is upgraded, but also to reflect past experience and units in the field. After parallel design transactions it may be neccessary to perform version merging operations to achieve a new design. Version identification and tracking is critical in EIS interfaces.

Audit trails assure that any problem can be traced back to its source. Without such feedback designers will be doomed to repeat their mistakes. Audit trails can also identify when a revision is made, the extent of rechecking and testing that is now required, considerably reducing the cost of improving the product.

4. Models

The primary goal of an EIS model is to focus on the relationships between various parts of the system in order to gain insight into the needed interfaces between parts of the system and to help guide the definition of needed interface and common service specifications and standards. With this goal in mind, the best model is one that highlights these interfaces and recognizes that, unlike most systems, a key aspect of an EIS is its composition of many different tools, each with possibly unique requirements for interfaces and services.

It is a goal to reach a reasonable balance point between overly constraining the model and its interfaces to the detriment of the functionality and performance of individual tools and making the model so generic that it serves no useful purpose.

Three prototypical models were suggested at the preparatory meeting in Dayton, August 26, 1985, and they are sketched in this note. The purpose of these sketches is not to delimit the type of EIS to which the future definitional work can apply; but rather to provide a basis for a common vocabulary in this arena.

The three models were termed the:

- CenEIS: Centralized Engineering Information System
- DisEIS: Distributed Engineering Information System
- ComEIS: Communicating Engineering Information System

We will now describe each of these models succinctly and also provide some references to typical proposals.

To clarify the distinction between the models we will resort to simple sketches. We will use mnemonics for the various aspects of the systems:

HWThe Hardware. Hardware is isolated from the software as much as possible by the enclosing Operating System. Hardware includes the computer itself and all peripherals including the user's terminals and/or graphics displays.

WSWorkstation Hardware. The architecture is similar, but typically not shared, and less controllable.

OSThe Operating System. Operating System is used generically to mean the complete environment of basic hardware independent services used by applications. It includes interfaces to Input/Output devices. Also included is working file storage that is accessed by an application separate from the data managed by the Data Management System.

LOSThe Local Operating System on a Workstation. Supports Communication with other OSs.

DMSThe Data Management System. The database management system handles all data that is not private to an individual application. The DMS maintains information integrity via activity control, version control, configuration control and release control. Only sharable objects are kept in the DMS. Other, local, data may be managed directly, still via the OS, to avoid DMS overhead. Such data is then available only to the owning tool or application.

LDMLocal Data Management on a workstation. Follows protocols of the DMS to assure global integrity maintenance.

DDSA Distributed Database System is comprised of a DMS and a number of LDMs.

LDSLocal Data Storage facilities on a workstation. May not have the integrity maintenance of a DMS. Has often high performance for the specialized workstation tasks.

T_XTool or application X. The individual applications tasks. For example, physical layout generator, functional simulator, silicon compiler.

ExImExport Import. A special tool for bringing data in and out of the EIS.

XInterface nodes for an integrated EIS. Nodes represent the joining of arcs between multiple applications that must communicate information. Each node is a potential for the standardization of an interface, or for standardization of part of an interface. Thus, if it is anticipated that many applications will need to exchange a particular type of data than that data and any associated service requests are candidates for standardization.

Standardized representation could then be used directly between tasks or standardized formatted data could be managed by the DMS and accessed by multiple applications.

In the sketches the various arcs represent communications paths between different parts of the EIS. Each arc always includes the commands and requests from one part of the EIS to another. In general, arcs also include the data flow between parts of the EIS.

Whether data is actually passed directly between tasks or is stored in the file system or managed by the DBMS, the flow is illustrated as a process to process arc. In general, arcs are bidirectional. A double line is used to indicate that a single system component coordinates both of the connected nodes.

CenEIS

A Centralized Engineering Information System represents the traditional database view applied to EIS. The database is viewed as the central resource. Tasks which utilize the database initiate a transaction to read, or read and update a fragment of data of interest to them. The data accessed are locked against simultanuous update. The actual engineering tasks may be carried out at remote workstations so that the acquired data is removed from the direct control of the central system.

Insufficient attention has been given to issues of representation change between the central database and the workstation fragment. While the archival functions of the central system is often well served by a relational implementation, this is not true for the workstation where engineering is a performance consideration require more binding.

```
------------------------------------------------------------------
----------------------
|         OS         |
|   ---------------  |
------            |   |             |  |         ------
| T1 |----------------X->         <-X-----------| Tx |
------            |   |     DMS     |  |         ------
------            |   |             |  |       -----------
| T2 |----------------X->           |  |       |   LOS   |
------            |   |             |  |       | ------- |
------            |   |          <-X---------X-| Tw  | |
| T3 |----------------X->           |  |       | ------- |
------            |   |             |  |       | | LDS | |
|--------------------|       |---------|
.  .  .                        |         HW         |      |   WS     |
----------------------       -----------
CenEIS concept
------------------------------------------------------------------
```

DisEIS

The distributed database EIS model is illustrated below. It consists of a distributed database system (DDS), an information repository (IR), and four interfaces. The DDS is a vision of some future information system, and is not constrained by the features of existing commercial database system products. The general concept of the model will be described first, and then each of the interfaces will be discussed in turn.

The general concept is that the DDS will be responsible for the capture, maintenance, and dissemination of information generated in the distributed engineering environment. Its functions are identical to that of a centralized DMS. The concept of distribution means that a DDS is managing and dispensing large objects, perhaps entire OS files. For this task the DDS is acting as a directory/controller (using a small volume of record data) of a much larger volume of data files.

The DDS and the IR may be physically distributed across several hardware systems (including workstations and mainframes) to gain advantages of performance, reliability and autonomy.

The workstation interface provides the coupling between the DDS and a workstation used by an engineer. The workstation must provide a human interface through which the engineer can access and manipulate information from the IR, set up engineering activities, and invoke tools. The DDS must provide services to the workstation to support this human interface. Due to differences between workstations, the workstation interface will need to be individually tailored for each type of workstation. A set of software components is required to make this job as easy as possible. The workstation interface may be resident on the workstation itself, or on a mainframe.

The tool interface provides the coupling between engineering tools and the DDS. This interface provides a framework for integrating tools into the overall facilities of the EIS. In particular, the interface should support uniform methods for controlling tools, moving data to/from tools, archiving the results of tool activities, and monitoring tool operational status. When format conversion is required for data moving to/from a tool, the workstation interface should invoke the necessary conversion routine. Data transfer between tools must be coordinated through the DDS so that the necessary activity information can be captured in the IR. The tool interface may be resident on a workstation or on a mainframe.

The exchange interface allows information to be exchanged with the external environment, that is, to systems and tools that are not integrated with the EIS. The interface will need to support various standards for the exchange of engineering information. Otherwise, the function of the exchange interface is similar to the tools interface.

The existing DBMS/file system interface must provide facilities for easily coupling a foreign DBMS or file system to the EIS. This is a practical necessity in many installations which are already using traditional DBMS/file systems for partial information management. This interface must support the exchange of queries and data across different query languages and data models. The interface will need to be tailored individually for each distinct DBMS and file system. A set of software components is required to make this task as easy as possible.

```
------------------------------------------------------------------
--------------------
|   -------------  |
------            |   |           |  |        ------
| T1 |----------------X->       <-X-----------| Tx |
------            |   |           |  |        ------
--------          |   |           |  |      --------------
| ExIm |<-------------X->         |  |      |   LOS   |   |
--------          |   |           |  |      | ------ | W |
|   |---XX---------|       | | Tw | | S |
| O |            |  |       | -------|   |
-------------     | S |         <=============>| LDM |   |
|   |  LOS  |     |   |           |  |      --------------
| W |------ |     |   |           |  |
| S |  LDM <===========>           |  |      --------------
|   |     | |     |   |       DMS |  |      |   LOS   |   |
-------------     |----------     |  |      | ------ | W |
storage only      |  HW     |     |  |      | | Tv | | S |
node          |         |     |  |       | -------|   |
```

```
|          |   <=============>| LDM |   |
|          |     |  |      --------------
--------------------
Distributed Database EIS Model
------------------------------------------------------------------
```

ComEIS

An excellent analogy for modeling an EIS, based upon the structure of communications and services between various non-homogeneous tools and their environment is the ISO Open Systems Interface model for layered communications systems.

The Communications Model thus highlights the communication of data, not the data itself or where it is physically stored. It is independent of system configuration, not caring if data storage is centralized or distributed.

```
-------------------------------------------------------------------
----------------X----------------
|               |               |
|               |               |
------          ------          ------
| T1 |----------| T2 |          | T3 |
------          ------          ------
|  |            |  |            |  |
|  |     --------  |   ----------  |
|  |     |         |   |           |
|  |     |         -------------   |
|  -------------------------   |   |
|        |             |   |   |   |
-----    |    ----------   |   |   |
\
   |   /             |   |   |
\
  |  /               \  |  /
\
 | /                 \ | /
\
|/                   \|/
X                     X
|                     |
|                     |
-------------               |
|    OS     |               |
|           |         -------
|  -------  |         |     |
|  |     |  |---------| DMS |
|  | HW  |  |         |     |
|  |     |  |         -------
|  -------  |
|           |
-------------
A COMMUNICATIONS MODEL
-----------------------------------------------------------------
```

5. Goals.

Within the models which we have described interactions must take place between the tools and the datasystems. We expect both areas to continue to evolve at a rapid pace. To take advantage of progress it is neccessary to avoid having to redesign all complementary components. Standards can address this issue.

Any standard can be seen as restrictive and in advance of its time. However, if we wait until we have a stable world we will have wited too long. We do expect that some trade-off maximal interface bandwidth versus sharability will be tolerable.

We need standards here that are nor restrictive, that can grow as the needs are better recognized and as new tasks and tools are defined. That implies a having a general structure, into which specific definitions can be inserted. We do not now have a proposal to discuss — that is the goal of this workshop. of

Acknowledgement

This work is in part supported by the Air Force Wright Aeronautical Laboratories at WPAFB, Dayton Ohio, by the Coauthors' institutions (Computer Corporation of America, Cambridge MA and Silicon Compilers, San Jose CA) and by DARPA under contract N39-84-C-211 to Stanford University.

References

[BaKi86] Banerjee,J. and Kim,W.: "Supporting VLSI Geometry Operations in a Database System"; accepted for IEEE CS Data Engineering 2 conference, Feb.1986.

[BaKi85] Batory,D.S. and Kim,W.: "Modeling Concepts for VLSI CAD Objects"; ACM TODS, Vol.10 no.3, Sep.1985, pp.322–346.

[BBDM84] Brodie,M.L., Blaustein,B., Dayal,U., Manola,F., and Rosenthal,A.: "CAD/CAM Database Management"; IEEE Database Engineering, Vol.3, 1984.

[Donc79] deDoncker et al: "Interactive Database management for Integrated Circuit Design on the Operating System"; ACM-DA 16, 1979.

[Fisc83] Fischer,W.E.: "A Database Management system for CAD and CAM"; Informatik-Fachberichte 70, Springer Verlag, 1983, (in German, English abstract in Informatik-Fachberichte 94, Springer Verlag, 1985.).

[HNCL84] Hollar,L., Nelson,B., Carter,T., and Lorie,R.A.: "The Structure and Operation of a RDBS in a Cell-oriented Integrated Circuit Design System"; ACM-DA 21, 1984.

[John84] Johnson,H.R.: "Engineering Data Management Activities Within the IPAD Project"; IEEE Database Engineering, Vol.3, 1984.

[Juli85] Julien,Christian: "La Gestion des Donnees dans un Systeme de CAO de VLSI (in French)"; Bull.de Liason de la Recherche en Informatique et Automatique, INRIA Bull. no.102, Jun.1985, pp.11–16.

[Katz83] Randy Katz: "Managing the Chip Design Database"; IEEE Computer Magazine, Dec.1983, pp.16–36.

[KeBe86] Ketabchi, Mohammad and Berzins, Valdis: "Component Aggregation: A Mechanism for Organizing Efficient Engineering Databases"; To appear in Proc. IEEE CS Data Engineering 2, Feb.1986.

[KoLo83] 0! Koenig,W. and Lorie,R.A.: "Storage of VLSI Physical Designs in a Relational Database";

[MeCa84] Melkanoff,S.J. and Cammarata M.A.: "An Interactive Data Dictionary Facility for CAD/CAM Data Bases"; in Proc. Expert DataBase Systems, Univ.of south Carolina, vol.2, Oct.1984.

[MiWi83] 0! Milton,Jack and Wiederhold,Gio: "Network and Relational Systems in VLSI Design: Contradicting Performance?"; IEEE CS Database Engineering, Vol.1, 1983, pp.167–170.

[PKLM84] Plouffe,W., Kim,W., Lorie,R., and McNabb,D.: "A Database System for Engineering Design"; IEEE Database Engineering, Vol.3, 1984.

[RoYe84] Roussopoulos,N. and Yeh,Raymond T.: "An Adapatable Methodology for Database Design"; IEEE Computer, May 1984, pp.64–80.

[SpMF86] Spooner, D.L.; Milicia, M.A.; Faatz, D.B. : "Modeling Mechanical CAD Data With Data Abstraction And Object-Oriented Techniques"; accepted for IEEE CS Data Engineering 2 Conference, Feb.1986.

[StGu84] Stonebraker,M., and Guttman,A.: "Using a Relational Database Management System for Computer Aided Design Data–An Update"; IEEE Database Engineering, Vol.3, 1984.

[UdMi84] Udagawa,Y., and Mizoguchi,T.: "An Advanced Database System: ADAM–Towards Integrated Management of Engineering Data"; Proc. IEEE Data Engineering 1 Conf., Los Angeles, Apr.1984.

[Vern84] Vernadat,F.: "A Selected Bibliography with Keywords on Engineering Databases"; IEEE Database Engineering, Vol.3, 1984.

[WiBS] 0! Wiederhold,G., Beetem,A., and Short,G.: "A Database Approach to Communication in VLSI Design"; IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol.CAD-1 No.2, Apr.1982, pp.57–63.

[WWWi] Winslett-Wilkins,M. and Wiederhold,G.: "Relational and Entity-Relationship Model Databases and VLSI Design"; IEEE Database Engineering, Vol.3, 1984, pp.113-118. > 20, 22

[WBPW85] Wilkins,M.W., Berlin,R., Payne,T., and Wiederhold,G: "Relational and Entity-Relationship Model Databases and Specialized Design Files in VLSI Design"; ACM-DA, 1985, pp.410-416.

[Wied85] Gio Wiederhold: "An Annotated Bibliography on Databases in CAD"; Stanford CS department Sep.1985.

[ZaHe85] Zara,R.V., Henke,D.R.: "Building A Layered Database For Design Automation"; ACM-DA 22, 1985, pp. 645-651.