Context: Lisp Machine in the Database Environment

DATA MANAGEMENT CONCEPTS AND GOALS IN CAD

by: John M. Switlik
Senior Technical Professional
Sperry Corp
Bloomington MN

CAD DBM NEEDS

CAD design and information needs

Computer-aided design (CAD) systems create and use a variety of valuable data for such purposes as part design, manufacturing, and management. Although the integration of such information is desirable, the complexity that arises from the sharing of CAD data by both applications and users requires the use of data base management (DBM) tools. To provide the best possible computer assistance for users in design, engineering, and manufacturing, DBM for CAD systems must be specifically oriented to handling design objects, assisting in moving objects between application phases, and supporting ad hoc information needs throughout a project.

Business-oriented DBM tools

Fortunately, DBM tools used in business-oriented environments, such as a data base management system (DBMS), can play a part in some aspects of CAD. However, the complexity and types of data structures needed to support CAD applications often prevent the use of business DBM tools. As a result, DBM tools specifically for CAD have been and will continue to be sought. In this article, the author discusses business-oriented DBM and CAD DBM capabilities and requirements.

[Type here]

Auerbach Publishers Inc (1987):
Computer-Aided Design, Engineering, and Drafting

dumped to files on disk for quick I/O with file pathname maintenance for object control. The control information consists of data for administrative and handling purposes (e.g., name, description, parts list). Another model uses a series of direct access files that provide a buffering scheme, allow multiple access to the data, and provide a DML for programmers. The DML in this case takes away the need to be aware of the files.

Customized DBMSs

Custom-developed DBMSs can be of this variety and usually are optimized for a particular CAD approach, perhaps even affording object-level views. In addition to supporting a DML, the customized DBMS may support data definition for mapping complicated application data structures to the DBMS. In many cases, application concurrency is read-only, and one application is responsible for updating and for serving all data update requests. The need to view a conglomeration of CAD data as an integral object requires that changes be implemented and facilitated at the object level and controlled at lower levels.

Relational Model

Advantages and disadvantages of the model

The relational model is associated with a tabular view of data. A relation is a table of rows (tuples) and columns (attributes). Tables can have key variables that are used to build relationships. Most DBMSs based on this model provide an easy-to-use syntax for query formulation. The actual access of tuples is handled by the DBMS and is beyond the worry of the user. The main drawback of the model is the difficulty in representing complex data structures, such as those needed for graphics or a hierarchy of objects, without some major enhancements to the relational model. There has been much effort to provide hardware support for this model.

Network Model

Requirements of the model

The network model consists of records and sets; each set has an owner record and zero or more member records. Typically, a DBMS using this model tracks set relationships through embedded pointers. Query formulation in this model requires knowledge of record definitions and set relationships. Also, knowledge of navigational currency in the DBMS is required to determine at what point in the data base the next action will occur. This need for tracking position can be quite cumbersome for the programmer with a very large data base.

Disadvantages of the model

A data base built on this model can be fairly complicated. Programmers specializing in writing data base routines for others to use may be needed. In addition, because of the model's record orientation, a design object that has tens of thousands of records would create storage overhead in set pointers, pathing

AUERBACH ®

[Type here]

Auerbach Publishers Inc (1987):
Computer-Aided Design, Engineering, and Drafting

had been stored using a network-model-based DBMS and enters the data base into computer memory, where the data base is put into data structures that can support the CAD application. The structures that were in memory are changed through the terminal connected to the computer. Permanent changes would be made through the applications software writing out the revision from the structures that were in memory to the DBMS data base.

Applications management

In both approaches a group of specialists buffered applications software management from data base concerns. This group performed the data base design and developed data-handling routines. The use of a DBMS provided tools for the DBM group.

File-access-based model

Another design data base approach used files and wrote large data structures related to a design to storage in an efficient manner for I/O. This storage would be put into files with meaningful file names. Data for an object was then accessed through pathname manipulations. This approach could be augmented with a DBMS that handled project data, part relationships, and the file name handling requirements. Attributes of a part, or composite, could include pathname data.

FUTURE DEVELOPMENTS IN CAD DBM

Application management and ad hoc information improvements

DBM should support CAD with proper commands, storage techniques, and control mechanisms. Currently, project applications software provides this capability with some additional capability acquired from DBMSs and other DBM tools. Most ad hoc requirements are met through low-level access routines written by applications software management. Several areas of CAD DBM should improve in the future. For application data structures that can be put into a relational form, a relational data base machine might provide ways to efficiently handle large amounts of shared data and support most data requirements for project and management data. For CAD applications requiring complex data structures, new data-handling techniques supported by enhancements to the relational model are needed.

Object-Oriented DBMSs

Capabilities

Research in an object-oriented approach to data bases looks promising, but several unresolved technical issues suggest that the approach may be inadequate. Nonetheless, one expected benefit of an object-oriented DBMS is the ability to use data in any current DBMS or data system and still derive all the benefits of object management.

Potential benefits

An object-oriented DBMS is expected to provide object-level DML commands, ad hoc reporting capability for project

AUERBACH ®

[Type here]

management, interactive access to an object, and application program views of and actions on objects. Because the object-level DML will interface with any physical storage mechanism, the storage of data could be optimized for handling large data structures. Even with object views, data transfer between machines that may be incompatible requires data transformations. However, data dictionary techniques can be used to establish a universal view of data. An object would then be defined not only in the external form needed to support applications but also in the particular format needed for a specific physical representation. Methods to transform from one representation to another could be attached to an object. This universal view of data also has implications of a conceptual nature.

Another benefit of the object approach may be in distributed processing. In sending a message to the object to which the activity is to be done, it would be possible to send an object to a processor carrying its activity information. The object would then be able to invoke the appropriate activity.

Symbolic processing

In current systems, applications give data meaning. Symbolic processing provides the opportunity to go beyond this approach with more semantic views of data. The benefits to be derived from the augmentation of DBM by symbolic processing will accrue to system users, designers, and DBM personnel.

[Type here]

Auerbach Publishers Inc (1987):
Computer-Aided Design, Engineering, and Drafting