directions on where we may go

By Jon H. Pittman and John C. Dill

Design is an information-intensive process. The architect is constantly engaged in collecting, refining, organizing, and presenting information. In fact, architecture can be viewed as information management.

Architects collect information from clients in the form of programmatic data and generate information in the form of design and production documents. In addition, they must provide information to other parties, collect information from other parties, and attempt to organize and filter information from other parties (see Figure A).

Information is exchanged between many different parties in many different forms

The individuals and groups who must exchange information during the design and construction process include (as shown in Figure A) the architect, client, building user(s), consultants, engineers, planning agencies, construction managers, contractors, subcontractors, suppliers, community organizations and regulatory agencies. In addition, the client may request additional information such as "as-built" drawings or data to be used for facility management after the project has been completed. It is also common for architects to enter into joint ventures with other architects for specific projects. The joint firms must, obviously, exchange information throughout the design process.

Traditionally, architectural information has been transferred in a variety of forms, including sketches, working drawings, written specifications, telephone conversations, change orders, shop drawings, correspondence, photographs, physical models, renderings, computer printouts, building codes, product catalogs and supplier quotations. Using these forms and others, individuals and groups involved in the design process compiled information in the form most convenient for their own needs. Others who had to use information prepared by one group had to extract the information they needed and perhaps organize it into a different format.

Although the process of differentiating, extracting, integrating and organizing information is slow by traditional method, it is a task for which humans are very well suited. People are very adept at picking a meaningful pattern out of an extremely ambiguous field of information.

With computers, the need becomes finding a way to exchange information among various systems

Now, with the increased use of computers in the design and construction process, the opportunity to exchange data through computers has presented itself. However, new problems have emerged with this opportunity. Although computers can handle raw data at much faster rates than humans, they are not as adept at extracting useful information from varied sources and integrating it into a meaningful form.

Each computer manufacturer and computer software developer, as well as each group of users, has his own conceptual model of the way data is organized and related in the computer. The type of data that is stored and the ways in which data is stored in one computer or software system is invariably different from other systems. This poses a "Tower of Babel" problem. Many individuals are trying to use computers to work toward a common goal without having common communication.

As the architecture profession begins to use computers more and more in architectural practice (and as our clients and colleagues increase their use of and understanding of computers),the need to exchange information in digitial form will increase. This will require that some standard be established to ensure that information produced by one group and computer system is meaningful to other groups and systems.

To address this issue, several attempts have been made to develop a standard for information exchange between computer systems. It is important that architects be aware of these efforts so they may provide input to the formation of these standards and so they will be aware of the need for the products that they purchase to support them.

To help make architects aware of these efforts and some of the issues surrounding them, let's explore information exchange in more detail, the ways in which computers exchange information, the concept of an information standard, and finally some examples of information exchange specifications.

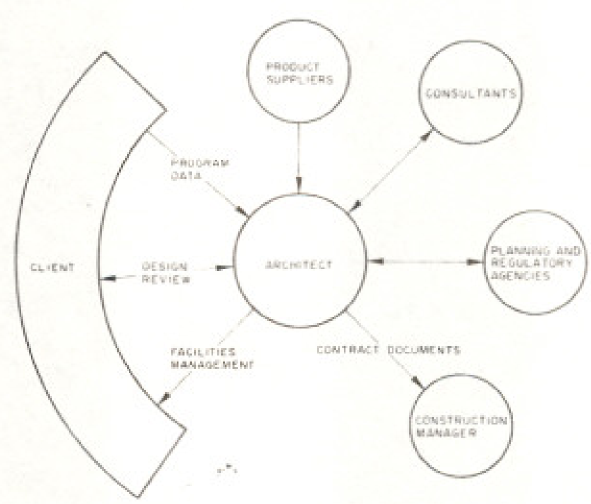


Figure A—The architect exchanges information with many parties during the course of practice.

How information is exchanged between computer systems affects their usefulness

For purposes of this discussion, our goal is somehow to transmit a drawing and associated information from one system to another. Since this information will be in a computer file of some kind, the task becomes one of:

- transmitting a file from one system to another, and
- ensuring that the contents of the file can be "understood" by the application program in the receiving system.

Though the thrust of this article is the latter and might be called "information transfer," the former, which we could call "data transfer," is certainly a necessary component. How do we do data transfer?

One way, certainly, is to copy the data onto paper, as a printout, or digitizing in a drawing. Although highly inefficient and error-prone, this scheme has actually been used many times in the past, even in production systems in very large companies.

A much more effective means is to copy the data onto some electronic storage medium that can be carried between systems, such as a floppy disk or magnetic tape. Here, of course, we must ensure that a compatible format is used. In other words, if the sending system uses a 1600 bpi, 9-track unlabled tape with 80 character records, the receiving system must be able to read such records. Further, both systems must use the same encoding scheme, either ASCII (e.g. the number 142 represented as the character string '142') or binary (142 represented as '10001110').

A still more effective scheme is a data link, transmitting the data over a communications link, such as a telephone line or a direct wire. Similar incompatibility issues occur here, too. Economic issues are important. For example, using a dial telephone network to transmit a 4-million-byte file from Chicago to Los Angeles at 1200 bits per second would take well over 10 hours, even assuming no errors, and be considerably more expensive than mailing a tape, even using an overnight service. Using a high-speed data link (e.g. ARPANET) would reduce the time but increase the cost. If file transfers occur often, on the other hand, the network may be economical.

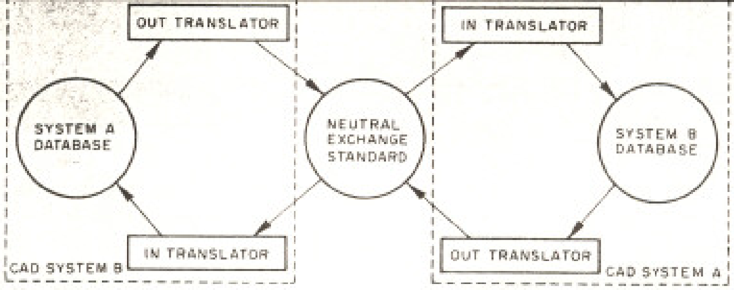
The subject of networking and data transmission is too large and complex to be more than mentioned here. For further reading see any standard introductory text.

What is an information exchange standard, and why do we need one?

In architectural practice, each firm has its own standards and methods. The ways in which architects describe building details, lay out a drawing, and relate various drawings to each other and to contract documents may be quite similar from firm to firm, but with subtle differences. The standards for a given firm have evolved over time and have been shaped by a variety of factors to fit the needs of architects and their clients.

In a similar way, the developers of computer-aided design systems have evolved unique ways of organizing information influenced by the type of hardware, the software, the type of data used by the system, the needs of the users of the system, the methods used for generating data used by the

Figure B—A neutral information exchange standard allows two CAD systems to transfer information even though they may have different internal representations of that information.



system, and the developers' concept of what information is necessary for a particular task and how it is collected, generated, analyzed, and displayed.

As with architects, there may be considerable variation between one standard and another. However, since the factors affecting the formation of a standard in the computer-aided design community are more diverse, it is likely that the standards vary significantly from one system to another.

To communicate information stored in one system to another system, some format for the exchange of this information must be agreed upon by the developers of the two systems. This agreed-upon means of data transfer is an information exchange standard. Such a standard defines the form of information exchange from one system to another. If one has some information in system "A" and wishes to transfer it to system "B," one must first translate the information on "A" into a neutral form.

The neutral form is, in effect, the form described by the information exchange specification. One must then translate the information to system "B." With each translation some information may be lost. If one wishes to then transfer the information back to system "A" even more information may be lost.

Let's look at some problems involved in defining an information exchange standard

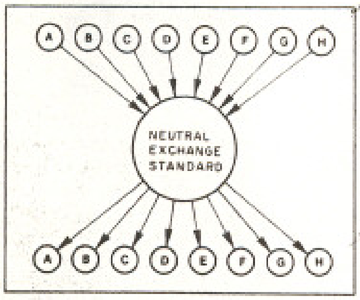
Let's use the analogy of a Russian scientist who wishes to send a document to a Japanese colleague. Suppose that the Russian did not have a Russian-Japanese translator available, but that a Russian-English and an English-Japanese translator were available. The Russian would have to have the document translated from Russian to English and then from English to Japanese. With each translation, some information would be lost. Each language has a particular set of concepts that can be expressed, but there is not necessarily a one-to-one correspondence in translation. The general meaning of the document can probably remain intact through the translation but subtle nuances may be lost.

In trying to exchange information between two scientist. However, the problem is likely to be worse since human languages are very rich and complex and can express one concept in a variety of ways whereas the computer may not have such a wide range of alternatives available to represent information.

Architectural design is an iterative process. An architect generates a design solution, evaluates that solution, and successively refines it until it "fits" the design criteria. Many people, including the client, consultants, engineers, other architects, planning boards, etc., may be a part of this process. If the parties involved in the design process have computer systems, it is probable that they will wish to use them. Thus design information may go through many translations, potentially losing information through each.

It is clear that, if care is not taken, the original meaning of the design information could be lost in much the same way that information is lost or distorted beyond recognition in the old party game in which a sentence is passed around a circle of people by having each person

Figure C—A neutral exchange standard only requires 2xN translators (n "in translators" and n "out translators").



whisper to the next in the circle.

One may ask why a single such standard is necessary. Why not write a translator between each pair of CAD systems? There are two reasons why this is impractical. First, there are a large number of CAD systems on the market. To allow each system to exchange data with any other system would require a large number of translators. If we know that there are "n" CAD systems on the market, nx(n-1) translators would be needed to ensure that data could be exchanged between the systems.

Each time a new system came on the market, a new set of CAD system developer upgraded his products, the translators would have to be upgraded as well. It is clear that this would be a monumental task! With one information exchange standard, each manufacturer would be responsible for maintaining a translator to and from the standard format, thus resulting in only 2xn translators (see Figure C).

The second reason for a single, neutral standard is that manufacturers of various CAD systems are competitors. They, in all probability, view their internal information structures as proprietary. To write a translator between any two CAD systems, it is necessary to have a detailed knowledge of each system's data storage structures and mechanisms. CAD system developers might understandably be hesitant to divulge such information to their competitors. With a single standard in the public domain, CAD-system developers could develop their own translators, thus providing data transfer capability while ensuring that knowledge of the internal workings of their system remains confidential. Thus, it is clear that a single information exchange standard is the most reasonable approach to exchange of architectural CAD data.

An information exchange standard should allow one to transfer as much information as possible between computers. Obviously, the more information that can be described by a standard, the more that can be transferred. Care must be taken to ensure that information stored implicitly in the structure or arrangement of information as well as the information that is explicitly defined is transferred.

To summarize, an information exchange standard is a format upon which architects have agreed to transfer information from one CAD system to another. There is currently no information exchange standard for architectural CAD data, but several potential standards are evolving. It will be to the architect's long-term benefit to participate in their formation.

What efforts are under way to develop an information exchange standard?

Several attempts have been made to develop information exchange standards. Some are important to the architect. Let's look at IGES, perhaps the most

transfer capabilities of two commonly used standards for computer graphics, and finally a data transfer specification for microcomputers.

- *IGES—Initial Graphics Exchange Specification.* IGES is specifically tailored to the exchange of CAD data. Development of IGES began in January 1980 by a technical committee composed of CAD/CAM industry representatives and coordinated by the National Bureau of Standards. IGES Version 1.0 was adopted by ANSI (the American National Standards Institute) in September 1981. The original goal of IGES was to provide "a data format for product design and manufacturing information created and stored in a CAD/CAM system in computer-readable form.

IGES allows CAD data to be stored in neutral form (a generic format not specific to any given CAD system) and to be translated from one CAD system to another. Initial users of IGES were large companies that had developed special-purpose software in-house and who wished to create integrated CAD systems with that software and/or to use that software in conjunction with a turnkey system. In addition, CAD system developers began to implement IGES translators for their systems. Currently, 32 CAD-system suppliers have committed themselves to supplying IGES translators for their products. These suppliers include several major CAD system developers for the architectural market.

Version 1.0 of IGES was primarily aimed at the general CAD/CAM community and allowed for the representation of geometric data such as size, shape, and position. Version 2.0 of IGES included extensions to accommodate printed circuit board technology and description of finite elements.

IGES tries to provide a very general format for the storage and transmission of computer-aided-design data. Each piece of data in an IGES file is represented by an entity. An entity may be one of three types: geometric, annotation, or structure. A geometric entity describes the physical shape and size of the object being represented. Geometric entities include points, lines, curves, surfaces, and planes. Annotation entities allow notations to be on a drawing and dimensioning entities. Structure entities allow relationships between other entities to be expressed. Thus entities may be placed into groups or notations may be placed into groups or notations may be tied to geometric entities. In addition, a general purpose entity called a property may be attached to other entities to provide other information.

As IGES currently stands, it could be used to transfer architectural drawings from one CAD system to another. It is not, however, adequate to describe a building in detail. It is difficult to use and has grown up as a definition standard for industrial products. Thus, it may not yet be suited for architectural use.

There is currently interest, however, in extending IGES so that it meets the needs of the architecture, engineering, and construction communities. Although IGES may need some work, it provides a good foundation for architecture, engineering, and construction data exchange. It would benefit the architectural profession to become involved in the development of this specification.

- *SIGGRAPH CORE* and *GKS*. These systems have arisen through efforts at defining graphics standards—i.e. standards for computer graphics. Work in this area has been under way since 1974 at both the national and international levels. Among the better known of these efforts are those of the ACM (Association for Computing Machinery), SIGGRAPH (Special Interest Group on Graphics), Graphics Planning Committee's "Core" proposal and the German standards group (DIN) GKS (Graphical Kernel System).

The thrust of these efforts is specification of a device-independent computer graphics system. A major benefit is portability of graphics applications programs, i.e. an application such as a CAD system need not be rewritten for each new graphics device in which it is installed.

An additional component of these efforts is the specification of a so-called "metafile," a file format for device independent graphical information. The basic purposes of the metafile are to transport graphics information between systems, to transport it between applications, and archival storage.

An important aspect is the ability also to store nongraphic information. While the 1979 CORE specification did not allow this, later efforts by the ANSI X3H3, the formal standards body for graphics in the United States do support this. Version 7.0 of the GKS also specifically provides a means for application programs to read that information back. Though this provides a basic mechanism for therefore of less interest than the IGES specification for this purpose.

- *DTS*. Readers with microcomputer systems may already have run into the problem of interchanging data between systems or programs. Suppose one wishes to look at the results of running a spreadsheet analysis (i.e. VisiCalc) with a plotting package. Unless the plotting program can read the file output by the spreadsheet program, one must print the spreadsheet and re-enter all the relevant data by hand into the plotting program. This may be a time-consuming and error-prone task. To solve this kind of problem, the DIF (Data Interchange Format) file specification was defined. The DIF file is simply an application-independent format for storing certain types of information. Thus programs able to read DIF files can be used to process the output of *any* program that creates a DIF file.

The need to transfer data between computers is clear in many applications

As architects increasingly rely on computers for drafting, engineering analysis, inventory, and other computer-aided design functions, the need to transfer architectural information between computers will be felt. Technological decisions being made now will have far-reaching implications. An awareness of such issues as information exchange standards will make today's architects more intelligent consumers in the CAD marketplace and will increase the probability that future CAD systems will meet the needs of architectural practice in years to come.

Mr. Dill is on the Cornell University staff as manager of the Computer-Aided Design Instructional Facility. He has been involved in interactive computer graphics since 1964, and has an extensive industrial background, having been involved with computer-graphics research and development at General Motors Research Laboratories since 1969. At General Motors he was active in the design of graphic systems for computer-aided design applications. Among his continuing research interests are device-independent graphics, man-machine communications, color graphics in computer-aided design, and business graphics systems.

Mr. Pittman is a member of the computer group at Hellmuth, Obata, and Kassabaum. Prior to that he was an assistant professor in the Department of Architecture at Cornell University, working with the Program of Computer Graphics on the development of computer-aided design applications for architects. In the past, he has worked on the development of computer-aided design and architectural computer applications for Skidmore, Owings, & Merrill; Structural Dynamics Research