WHAT A SYSTEM OF THE FUTURE MIGHT LOOK LIKE...

Paper for the International Council for Building Research Studies (CIB), Working Commission W78 Colloquium June 5-7 1984 at the Building Research Establishment, Garston UK.

Paul Richens, Technical Director, Applied Research of Cambridge Limited
23 April 1984

HARDWARE

There are many developments possible or likely in the next five years or so. I have picked two which will certainly determine the appearance of future systems; and a third where technology appears to be ahead of any demand from our industry.

The CAD Workhorse

Most currently installed CAD systems rely on a central processor for computation and filestore; augmented by a variable amount of local processing in the workstation. The central processor is normally a time-sharing super-mini, such as a VAX, Prime or Eclipse. This architecture is being displaced rapidly by the "CAD workhorse" for technical applications requiring high interactivity and moderate amounts of computation.

The typical workhorse is a dedicated single-user computer with a bit-map display, multi-tasking virtual-memory operating system, local discs, connected to similar machines by a Local Area Network. Filestore is either distributed around the network, or concentrated at a "file-server" node. Specialised nodes deal with printing, plotting, and interconnection to other networks.

Raster Graphics

Raster technology is now firmly in the ascendancy, not only for CRT displays, but increasingly in the form of electrostatic and ink-jet plotters and laserprinters. New display technologies such as LCD and plasma panel are also raster oriented.

A modern 1000 line non-interlaced colour display is more pleasant and more productive to use than anything that has gone before. It is still difficult and expensive to achieve some of the interactive effects of the original vector-refresh displays (e.g. rubber-banding, tumbling, dragging). On the other hand raster displays are leading in many new directions.

They are capable, and more effective, when drawing solids rather than lines. This could eventually greatly change the appearance of a document such as an architects working-drawing, which currently (even when computer originated) displays its long association with penmanship.

Profoundly different styles of interaction such as pop-up menus, icons and multi-windowing are inherent in the technology, especially when it is closely integrated with a personal computer. Voice input and output may be a gimmick, but new pointing devices such as the mouse and touch-sensitive screens are not. In conjunction with the multi-window multi-task screen they allow a single user to pursue several activities at once. This will avoid the unnatural "linearisation" of design activities which is enforced by most present-day systems.

Public Data-bases

Massive public databases are already of importance in the financial and legal worlds. They can be accessed over telephone or television cable, or distributed on optical discs. Designers will increasingly expect to have on-line access to material usually found in the office library, such as cost information, building products, design standards, government regulations, client standards, contract procedures and standard specifications.

THE SYSTEM OF THE FUTURE

THERE WILL BE MANY

DESIGN WILL REMAIN FRAGMENTED

INTERCONNECTIONS VITAL

LOOSE-FIT LOW-CONSTRAINT BROAD-SPECTRUM

HARDWARE

BIT MAP RASTER GRAPHICS

LASER PRINTER/PLOTTER

DEDICATED PERSONAL COMPUTER

LAN INTERCONNECTIONS

SOFTWARE - LEVEL 2½ INTEGRATION

- DISINTEGRATED MODELS

MULTIPLE MODELLING TOOLS

INTERCONNECTED INPUTS

SUBDIVIDED BY DISCIPLINE

VISIBLE AS LAYERS

COMBINED BY DOCUMENT EDITOR

The most important use of computers in the design office at present is for the preparation of production information; drawings, specifications, schedules and bills of quantity. The packages used in these areas are largely separate, as are the secondary packages used for structural design, pipe-sizing, visualisation and so on. Software for architectural design (as opposed to documentation) is rare. Any system which manages to do two of these things tends to be billed as "integrated".

The concept of integration is worthy of some analysis; below I propose four levels of integration and discuss their merits. Let me say at the outset that I believe too much integration to be like a straight-jacket. It may control the unruly patient, but it hardly encourages flexibility, self-expression or a spirit of co-operation.

Level 0 - Tools

An unintegrated (but still capable) CAD system would consist of a selection of the following broad-spectrum software tools; all are aimed at the production of documentation:

a) Word processor - edits text
b) Drafting system - edits line drawing
c) Painting system - edits raster graphics
d) Spread sheet - edits numbers
e) Business graphics - edits diagrams
f) Database, enquiry and report generator - edits data and schedules.

An editor is a system for the origination, amendment and reformatting of documents. Each is aimed at a particular form of documentation, and is very effective in producing it. Each has its own method of storage and of output. This makes it difficult to present the same information in different ways; or to combine information from different sources.

Level 1 - Interconnected Tools

The basic documentation tools can be interconnected in two ways. One is to provide interfaces that convert the stored information from one editor into a form suitable for another. So office systems will usually allow spread-sheet tables to be transferred to the word-processor, or database enquiries to the business graphics package. Similarly a drafting system may allow some scheduling; or perhaps generate visualisations which would be edited by a painting program.

SOFTWARE

The most important use of computers in the design office at present is for the preparation of production information; drawings, specifications, schedules and bills of quantity. The packages used in these areas are largely separate, as are the secondary packages used for structural design, pipe-sizing, visualisation and so on. Software for architectural design (as opposed to documentation) is rare. Any system which manages to do two of these things tends to be billed as "integrated".

The concept of integration is worthy of some analysis; below I propose four levels of integration and discuss their merits. Let me say at the outset that I believe too much integration to be like a straight-jacket. It may control the unruly patient, but it hardly encourages flexibility, self-expression or a spirit of co-operation.

Level 0 - Tools

An unintegrated (but still capable) CAD system would consist of a selection of the following broad-spectrum software tools; all are aimed at the production of documentation:

a) Word processor - edits text
b) Drafting system - edits line drawing
c) Painting system - edits raster graphics
d) Spread sheet - edits numbers
e) Business graphics - edits diagrams
f) Database, enquiry and report generator - edits data and schedules.

An editor is a system for the origination, amendment and reformatting of documents. Each is aimed at a particular form of documentation, and is very effective in producing it. Each has its own method of storage and of output. This makes it difficult to present the same information in different ways; or to combine information from different sources.

Level 1 - Interconnected Tools

The basic documentation tools can be interconnected in two ways. One is to provide interfaces that convert the stored information from one editor into a form suitable for another. So office systems will usually allow spread-sheet tables to be transferred to the word-processor, or database enquiries to the business graphics package. Similarly a drafting system may allow some scheduling; or perhaps generate visualisations which would be edited by a painting program.

A second method of integration is to combine outputs from the various editors into a single document. Such an approach is being used in the printing industry; with the development of effective raster hard-copy devices it should become widespread. It will be possible to combine images from drafting, visualisation, painting and business-graphics systems with text from word-processing, spread-sheet and database sources, into a single document. (These techniques are also of great interest in the Computer Aided Instruction field; they hope to combine animation with the rest to produce a kind of interactive instruction manual).

At this level of integration information is still kept in multiple independent forms. It becomes attractive to load drawings with additional information to aid interfaces to other programs. The MOBIL system at Oxford is working at this level.

Level 2 - Sharing Data

Level 1 of integration is beginning to share information between programs. More important is the sharing of information between users. This is a difficult problem when we look at the needs of a design team, especially a multi-disciplinary one. What happens when a heating engineer wants to edit the same drawing as the architect? The level 2 approach allows them to integrate their activities by dis-integrating their data. Each has his own drawing which he can amend. At the same time he can see the other's drawing as a background to his own, in its last stable state. The last point is important; much design work is doodling, and you don't want your colleagues to act upon it until you've committed yourself to a solution.

Level 3 - Total Design Database

The existence of many different representations of the same design information in scattered files belonging to different editors is theoretically unattractive. Would it not be better to have a single total design database which stores each fact once, from which all the documentation producers could derive their input?

This third level of integration has never been achieved, despite several valiant attempts. One problem is that a design does not have a single state; alternatives are evaluated, and "doodling" must be accommodated without locking-out other users of the information. Another is that no-one has been able to design such a database, capable of representing all facets of a building design.

The most common approach to the database is the "parts" or "things" model. (It is a technique used by BDS, RUCAPS, ACROPOLIS, etc.) The building is represented by located instances of predefined "things" or components. Each thing is (in its most general case) a 3D solid. Each location describes its orientation and position in the building space. From this database it is possible to extract line drawings, visualisations and schedules, even clash reports. This looks powerful. The drawbacks are in fact numerous. If what you want are working drawings, it is a very laborious way to get them. Many building elements are hard to fragment into things (masonary, concrete). Costing and detailing tend to focus on the junctions between things rather than the things themselves. The things do not adequately describe the spaces they enclose.

Attempts to overcome these problems we generally (following Ed CAAD) refer to as "stuff" models. They usually incorporate a good deal of building knowledge (GABLE is a small-scale example), but are consequently limited to forms of construction that they understand. (A parts model is quite general purpose). In the future they will probably use techniques such as rule-based procedures, object-oriented or extensible languages. They will be designed to be modified and tailored to new forms of construction or analysis. They will be rather personal tools, for the dedicated enthusiast. To anyone else they are liable to appear intricate, unstable and unmanageable. The supply of suitable enthusiasts is severly limited, though some educationists are hopeful of changing that.

Occasionally an enthusiast finds himself in a very large public-sector body (or a Japanese construction company), and may be able to develop such a "building-intelligent" system for use in his own organisation. He will attempt to build-in the collective wisdom of the organisation, so that his user colleagues have no need to tailor it. Such systems usually turn out to be very large and complex, short-lived, organisation-dependent and poorly accepted by their users.

So although such attempts are of great interest, and new developments in software will make them more practicable, we do not see them as being important for the integration of design activity.

THE SYSTEM OF THE FUTURE

Of course, there won't just be one. They will be manifold, and at all levels of integration. Communications between these systems will become an industry in itself (and so far IGES has made the problem worse rather than better). Building design will remain fragmented, carried out on many separate sites, and by many technologies. Something altogether more "loose-fit" than the total integration of level 3 will be necessary.

I refer to it as the "Dis-integrated" system, and place it at level 2 1/2.

Its main characteristic is that design information is sub-divided according to discipline and perhaps geography; and that a user can see this information as a set of "overlays". Some overlays he can modify, others he cannot because others are working on them. What raises it above the second level of integration is that his overlays need not be simply drawings, they can also be 3D models of the sort envisaged at level 3. The difference is that the models are supplementary rather than fundamental, and will eventually be converted to drawings. We expect "things" models to be used in specialist areas where they are really beneficial - furniture and fittings for interior design, where visualisation is important; building services, where coordination and clash-checking in 3D is needed. We expect "stuff" models to be important in the early design stages where they can help with conceptualisation and early analysis. We expect them to be the personal tools of individual designers. The information will be transferred into a more straightforward form (e.g. drawings) before detailing starts. We expect the drawing editor to be the "front-end" through which any combination of models and drawings can be viewed and manipulated, as we see the "Document Editor" being the way drawings and images can be combined with any other machine resident information from free-hand sketches to the clauses of the NBS specification.