ADVANCED GRAPHICS Magazine

& GENERAL COMPUTING

Vol.2
OCTOBER 1986

Advanced Graphics Magazine is produced quarterly by Apollo Computer (UK) Ltd.

The new range of Hitachi high resolution monitors represents a real breakthrough in this competitive market.

They have many new features designed to improve even Hitachi's high standards of clarity, performance and reliability.

These features include the Hitachi developed elliptical aperture CRT to give operators a crisper and brighter picture over the entire screen, not just in the centre.

At the top of the range, a 20″ 0.26mm dot pitch tube gives 1700 x 1250 resolution, and sophisticated techniques such as DDC help to bring a convergence error to less than 0.2mm even at all four corners.

These stylish and ergonomically designed monitors, which meet both UL and VDE standards, are available now from local stocks with full back-up facilities.

Right across the board, they've got the edge over the competition.

New Media Products

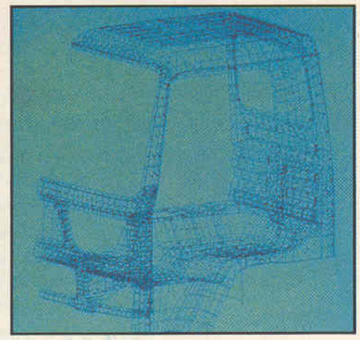
HITACHI SALES (UK) LIMITED, HITACHI HOUSE, STATION ROAD, HAYES, MIDDLESEX UB3 4DR. TEL: (01) 848 8787.

HITACHI

Vol.2
OCTOBER 1986

ADVANCED GRAPHICS Magazine

This Issue ... Technical Publishing — essential integration



cover...

Chernobyl, May 1st 1986, observed by the French satellite SPOT 1, showing the reactor site (middle) and cooling pond (right). Red areas, derived from the infra-red detector band, indicate areas of vegetation.

The image is effective 10 metre resolution, achieved by "fusing" a 20 metre multi-spectral image (3 bands) and a 10 metre panchromatic image, combining correlated edge information from the higher resolution, panchromatic scene, and spectral information from the lower resolution multi-spectral scene.

The adaptive, linear predictor algorithm developed at TASC (The Analytic Sciences Corporation) to achieve this was very compute intensive, but took only two minutes when implemented on an Alliant FX/8.

Advanced Graphics Magazine is published by Octagon Marketing Group, for Apollo Computer (UK) Ltd.

Editor: John Abbott

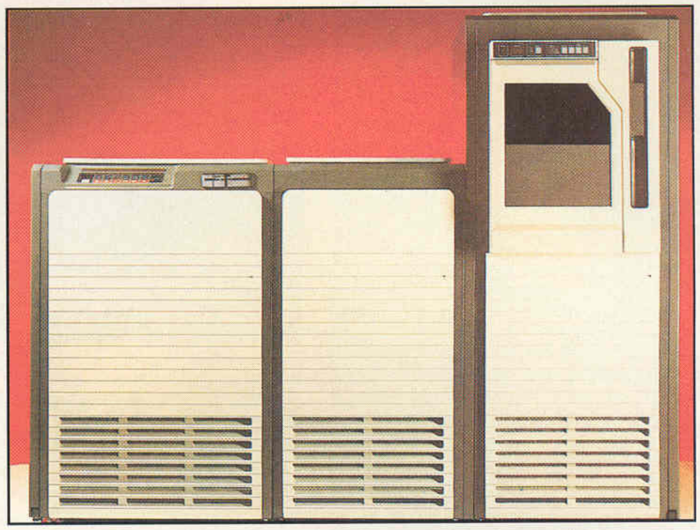
Production Director: M.G. Evans

Advertising Manager: Clive Farrand

Advanced Graphics Magazine, Octagon House, 32-36 Little Horton Lane, Bradford BD5 0AL Tel: (0274) 309930. London office: 7 Kensington High Street, London W8 5NP Tel: 01-938 2151.

Parallel Processing bringing the power to the people

Parallel processing has for long been confined to the groves of academia. Advanced Graphics looks at how one company is set to change all this.



The DOMAIN DSP9000 compute server

Parallel processing, the art of harnessing multiple processors to co-operate and simultaneously deal with different parts of a single problem, has become one of the computer industry's major talking points. As supercomputers begin to stretch the limits of single CPU performance, interest has been driven towards providing similar or greater power by combining smaller processors which can be quite cheaply built using existing VLSI technology. So far, though, parallel processing has had little commercial acceptance.

A handful of manufacturers have succeeded in producing machines with, in some cases, hundreds of processors and a potential supercomputer performance. Their ability to exploit the new architectures, however, has been restricted by the need for applications to be re-written or at best heavily adapted. At present the knowledge and skills needed to do this are rare.

A relatively small Massachusetts company, Alliant Computer Systems, has recently come up with a different approach. Although on a smaller scale than some of the grand designs sitting in research departments, it looks set to bring the new technology to a far wider audience. The company has taken the pragmatic stance that the way to commercialise parallel processing is to allow existing software to run unchanged. In doing so, it has removed the biggest immediate obstacle to the technology's acceptance.

Fortran

With the Alliant system, widely used Fortran applications can be recompiled to take advantage of up to eight floating point vector processors. The machines offer a peak performance of 949 UMFLOPS, 35.6 MIPS, and are aimed squarely at the new minisupercomputer class. These lie below the Cray's but are much more powerful than superminis for computer-intensive engineering and scientific problems. The machines also come in at a supermini price of between £125,000 and £1m.

As with any start-up hardware manufacturer, Alliant faces the problems of trying to rapidly establish itself in international markets. However, a recent agreement with Apollo Computer promises to remedy the situation. Apollo will not only integrate and market the machines to its world-wide DOMAIN user base, under the name of the DSP9000 compute server, but will also market the Alliant systems in their own right to new customers seeking this level of price/performance.

Alliant has an interesting history. It was founded in 1982 by three engineers who had all worked for Data General. One of these was president Ron Gruner, who had led the Fountainhead project that lost the race (immortalised in Tracey Kidder's book Soul of a New Machine) to build DG's first 32-bit machine.

Opportunity

The founders took the frightening step of starting the company without any clear business or product strategy. They began by looking for academic work with the potential for commercial exploitation and realised that parallel processing, if it could be opened up to wider markets, held a major opportunity.

Within a few months, the company found what it was looking for in the work of David Kuck, a professor at the University of Illinois. His work was showing that there was no reason why existing Fortran programs should not be able to take advantage of parallel processing. By providing a Fortran 77 and VAX-compatible compiler that automatically converted software to take advantage of multiple processors, Alliant would be able to offer users a painless route to more power. The company could also rapidly make the essential mass of third party applications available on the new machine.

Kuck was hired as a consultant and three years later, with the help of multi-million dollar venture capital backing, the first shipments took place. In contrast to most start-ups, Alliant has been profitable since then and now has 32 systems installed, with a value of $15m.

The Alliant approach to concurrency is based on a compiler that takes a high-level view of optimising code. The compiler picks out simple and complex program loops, which on a single CPU system would have each iteration processed serially, and identifies them for allocation to available Computational Elements that can then process them in parallel. The compiler also exploits code that can be optimised for the vector processing facility of each CE.

The Computational Elements have a peak performance of 11.8 MFLOPS for either 32-bit or 64-bit precision. Any number of them can be assigned to a specific task, while others deal with the rest of the system workload. The machines also have one or more Interactive Processors running the Concentrix Berkeley 4.2-based operating system to free the CEs for compute-intensive tasks.

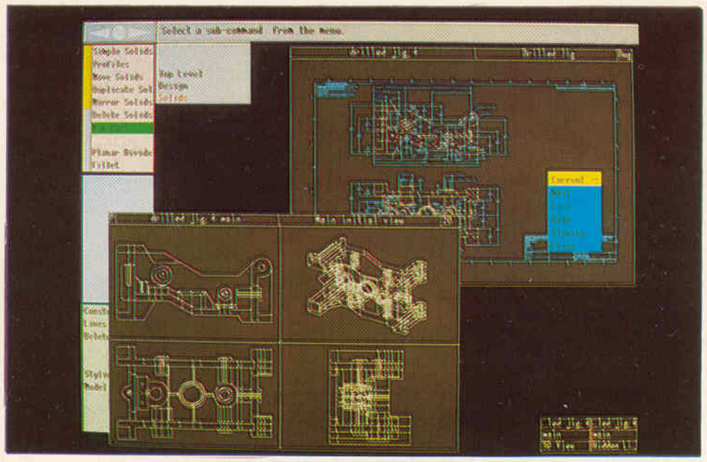
Synchronisation

An important feature is that control and synchronisation of concurrent processes is handled by Alliant-designed custom hardware. The efficiency in co-ordinating the different CEs is vital, since the overhead of using software to do so could nullify the benefits of the parallel approach. The Alliant approach also means that the company can offer a wide range of systems of increasing power, which can be field upgraded by adding CPUs.

The effectiveness of Alliant's parallel processing is limited by the structure of the programs it is dealing with — and by the intelligence of the compiler. Alliant points out that performance can be improved further by adding programmer specified directives. Meanwhile, the automatic parallel processing features will continue to improve with experience. To add to the available Fortran, other compilers for C and Lisp are being developed that take full advantage of the multiple CPUs. Currently, Alliant systems can be linked to Apollo DOMAIN systems via Ethernet, but the next DOMAIN release will fully integrate them as DOMAIN servers.

Apollo will have a demonstration centre with a large Alliant configuration in Frankfurt, although UK managing director John Parkinson points out that this country will soon have its own capability. "We are already in the right market for the machines," he says. "25% of current Apollo customers are potentially serious prospects." However, Parkinson is equally excited at the prospect of the systems taking Apollo into new markets. "The most promising UK prospects are not currently Apollo users," he points out.

Apollo adds the DSP9000 mini-supercomputer to the DOMAIN family — see page 23.



The Romulus-D CAE design system from Shape Data.

Technical Publishing — Essentially Integrated

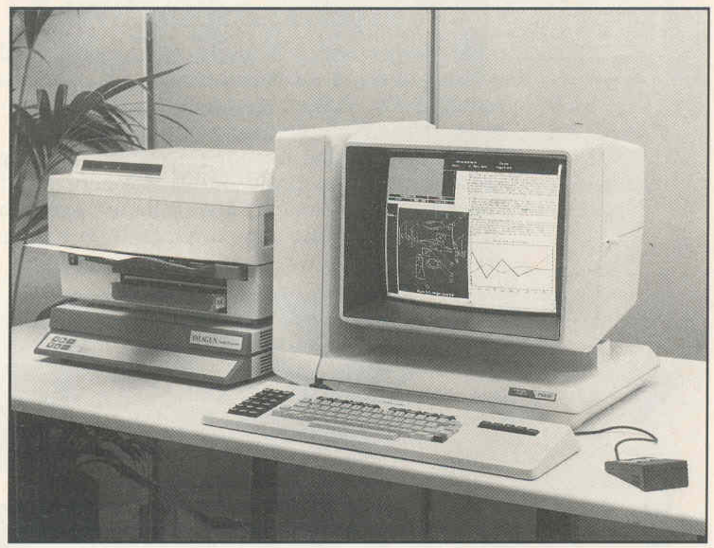
The demand for office and technical publishing systems is growing fast with the increasing perception by manufacturers and developers of the importance of quality documentation as a vital part of the finished product.

December of this year will see a team from Shape Data of Cambridge travelling to Detroit, Michegan, for this year's Autofact show. They will be there to announce the launch of a new design engineering software product called Romulus-D. The system, designed to run on Apollo networked DOMAIN workstations, is an interactive system based on solid modelling which integrates draughting and design management, and it is aimed at mechanical engineering and design companies, especially those involved in complex assembly work.

The lead up to such a launch is always a race against time, ensuring that all elements of the product have been fully completed so as to maximise the impact it makes. Bernard Williams, Product Services Manager at Shape, believes Romulus-D to be of a particularly high standard, and as the documentation must be regarded as an important aspect of the product, he was concerned that documentation quality should reflect that of the software itself.

This led to Shape Data approaching Amazon Computers Limited, with a view to buying technical publishing software. Amazon's systems division specialises in supplying Interleaf Workstation Publishing Software (WPS) on Apollo DOMAIN workstations running under AEGIS. Shape, already a major Apollo user with around 20 workstations for development projects, took Interleaf on three of its existing nodes for training, specification, and documentation, with an Imagen laser printer. This was installed in early Summer, and immediately put to work on the Romulus-D project, initially to prepare the specifications.

Interleaf has now integrated the specification and documentation processes at Shape, ensuring that information remains stable from the specification stage. An added advantage has been that some specifications, with minor amendments, have been turned into end-user documents. Another significant advantage to Shape has been that it can integrate graphics from Romulus-D for use in the documentation, eliminating lengthy "cut and paste" operations. Williams aims to have all written material produced by the system eventually, and is planning to double its size within the next six months.



A typical Interleaf WPS system as supplied by Amazon.

Amazon Computers, based in Milton Keynes, is part of the "technology transfer" company Cogen Ltd, which is wholly owned by the Legal and General Assurance Society. Amazon is involved in a number of specialist projects, developing products in areas such as fluid flow and thermal analysis. At the same time, the systems division has identified other products that need to be taken into the market-place.

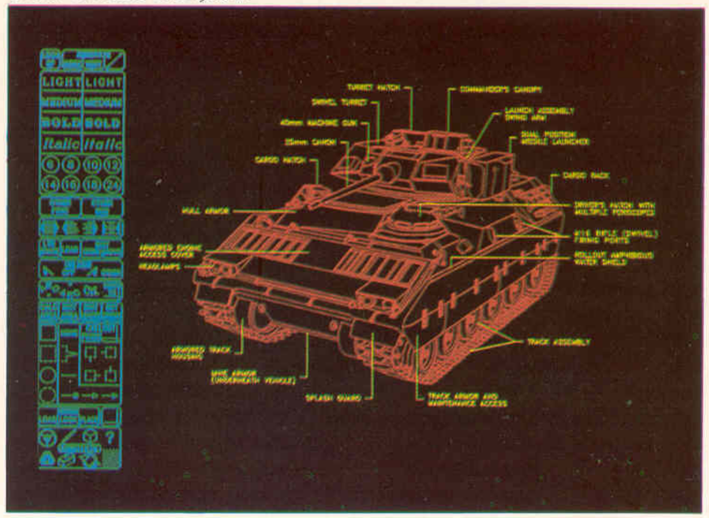
Interleaf is one such product. Interleaf Inc. of Cambridge, Mass., introduced its WPS and TPS (Technical Publishing Software) products in America in May 1984, and has since become a market leader in the office and technical publishing, and graphics arts markets. Now the product is at last becoming fully available in the UK.

David Boucher, one of the founders of Interleaf in 1981, has told the story* of how initial market research revealed highly paid consultants sitting at their desks drawing little boxes with rulers. He also realised that in the majority of companies the job of document production was distributed around an organisation, not entrusted to one person. Thus the Interleaf package concentrated on graphics, and ease of use. Stuart Holt, Business Development Manager with Amazon, believes the market area covered is one of the fastest growing areas in computing. And, of course, functionality has been enhanced by the development of fast and reliable laser printers, which can produce high quality, camera ready artwork at a fraction of the traditional cost.

Amazon distributes the Interleaf WPS system, TPS being limited to U.S. sales at present (except as part of large and expensive systems such as Kodak's KEEPS). TPS is generally aimed at customers using workstations exclusively for publishing work, and is designed to allow integration with specialist peripherals such as image scanners and typesetters.

Another Amazon customer is Selenia-Autotrol which has developed and marketed its own technical illustrator set of drawing tools for use by dedicated illustrators. On screen menus group drawing functions together for ease of use, and these can be used for anything from simple schematics to complex exploded view drawings, or orthographic and other projections. Built into the package is the ability to transfer complete graphics and text files to the Interleaf package for final preparation, and output to the laser printers. The company is targeting its marketing approach both to small manufacturing companies, and to larger sectors, such as the aerospace industry. A large opportunity exists, and Selenia-Autotrol expects 1987 to be the year when it makes major inroads into these markets.

The company expects to install an Interleaf package for its own internal use this month (October), but will also be acting as an OEM with Amazon, incorporating Interleaf with the Autotrol software in its Electronic Technical Publishing System, based on Apollo hardware. A complete package, which includes hardware, Selenia-Autotrol software, and Interleaf, costs around £35,000.



Selenia-Autotrol's Tech Illustrator - illustrations can be annotated with a high degree of accuracy and clarity.

The options for office and technical publishing for workstation users are expanding rapidly through the activities of companies such as Amazon and Selenia-Autotrol. And these efforts are likely to have a much wider impact as the quality of documentation improves throughout those industries which take advantage of the new technology. Visitors to Detroit's Autofact exhibition will no doubt be impressed by the quality documentation of Romulus-D even before they take a detailed look at the software itself.

**see The Yates Perspective, vol 4 no 1, January 17th, 1986 — published by Yates Ventures Inc.*

Documentation and design database management

by Michael Bosworth, Context Corporation

Documentation of complex and technical products is rapidly becoming a costly bottleneck in the design process. Before, during and after actual design, the product itself is reflected in a database of documentation which includes such documents as product proposals, specifications, and manuals. Documentation costs normally considered are those associated with product manuals. The hidden costs of inadequate documentation tools are lost time-to-market, quality compromise and poor revision support. A new class of product support documentation tools can substantially reduce such costs throughout the design cycle.

A number of trends are affecting the documentation of complex technical products across the disciplines of electronic, mechanical and software design. One set of trends affects the product development process itself. Another affects the underlying product support documentation process.

Though the products of such industries as aerospace, telecommunications and computers differ widely, they share similar financial and process contexts. A simple product life cycle model for a technical product shows a period of substantial investment followed by a period of profit. However, two forces acting on that life cycle have driven the search for productivity tools in the last two decades. First is the increasing complexity of the products themselves.

One example is the rising transistor count of microprocessors, from less than 10,000 in Intel's original microprocessor to nearly one half million in the microprocessors of 1987. This increasing complexity has stretched out the investment period, thus postponing the beginning of profitability. A second profit-threatening trend has been increasing world competition in these industries, resulting in shortened product life. Countering that trend is the increasingly widespread use of design automation tools to speed the design and revision process as well as to increase functionality of resulting products. Successive waves of ATE in the early 1970s, CAD in the late 1970s, and CAE in the early 1980s have each altered or eliminated bottlenecks in their attempt to provide the ideal design process throughput.

"a major aircraft manufacturer estimates the cost of an average page of documentation as between $500 and $2,000."

In the same time period, a number of trends have affected the area of product support documentation. First, the use of design automation tools has made documentation itself more visible as a process bottleneck. Second, today's design automation tools have made the use of graphics increasingly pervasive in product design. As the graphics intensity of documentation necessarily increases, development and support page costs increase. For example, a major aircraft manufacturer estimates the product life cycle support costs of an average page of documentation as between $500 and $2,000. Finally, quality expectations have risen, due generally to the increased role of quality in product marketability and due specifically to the improved output devices available for today's product support documentation.

The failure to apply automation tools to the process of product support documentation has had several negative results. Most visible is the often low quality or non-existence of support documentation at a time of initial product shipment. Next most visible is the cost of supporting up to one million pages of documentation for a major product at life cycle costs over $500 per page. Least visible but perhaps most expensive of all is lost time-to-market resulting from poor communications in a team design environment early in the product development process.

If we look at that product development process in terms of product support documentation, we see a front-end stage consisting of product proposal and specification. Through this point the product exists only on paper. Subsequently, the product enters its development and production phase, tracked by the supporting product literature. Finally, they come together at time of shipment. However, it is typical that before, at and after initial product introduction, the entire process is subject to continual change. Documentation solutions to date have tended to focus on development and production of product literature itself. Documentation tools which address the entire process including revision can have a far greater impact on overall product profitability.

Improved Product Development

Consider now how documentation can affect time-to-market. The first effect is observed at the front end of the process, the creation of proposals and specifications for the product. As an example, consider the Safeguard ABM Program during the 1970s. The top level proposal for the program consisted of a document inches thick. After that, the system level specification comprised a bookshelf full of similar documents. To verify internal consistency of that documentation set, the program would rent a high school auditorium, layout the documentation hierarchically and have the design teams perform the validation. The time and opportunity cost of having those people act like computers was extremely high.

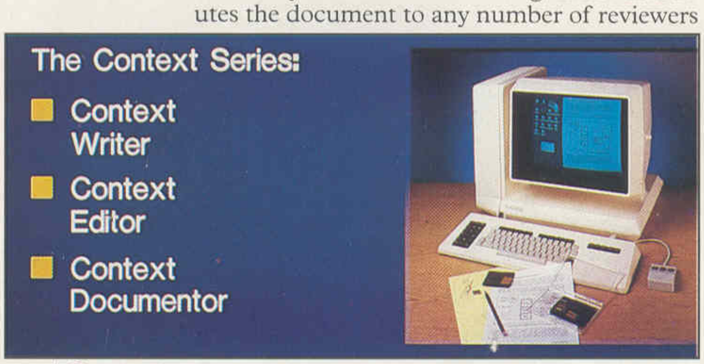
Documentation tools can have similar time-to-market effect at the production end of the documentation process. For example, with each 747 it ships, Boeing Aircraft Company is reputed to ship nearly a cargo hold full of product support literature. Any aircraft company must ensure completeness and quality of its support documentation or risk extreme product liability potential.

The problem is made more complex by the continuous application of revisions during and after the product development cycle. For hardware products in particular, new documentation must be generated to reflect change in the underlying product and be on time to permit product shipment. But the manufacturer must also maintain versions of the documentation for each revision shipped to the customer. Thus rises the need for configuration management of the documentation data base.

Current technology permits a number of related solutions to these time-to-market issues. The first of these is the concept of team documentation. Operating in the context of a common database, engineers, technical writers and marketing specialists can make complementary versus redundant contributions to the product support documentation. A second powerful tool is the concept of hierarchical document design. Such a concept allows a document to be distributed among a team of contributors yet viewed simultaneously as a complete document with all section numbering, referencing and other structures appropriate to each level. The idea of referencing documents can then be extended to the referencing of graphics. If, for example, a schematic drawing in the design database is updated, its inclusion by reference allows each document using it to be updated automatically. To stay current with the state of the design database, implied, of course, is the integration of text and graphics both on screen and as output. Elimination of cumbersome cut-and-paste operations is one of the major contributions to improved time-to-market in the product literature phase. Close to this is the concept of automatic structure maintenance. Automatic maintenance of table of contents, list of figures, list of tables, references, tabulated indices, footnotes, sections and other structures eliminates major bottlenecks caused by last minute changes in product design. Finally, a documentation tool architecture which supports associations is ideally suited to the problem of configuration management.

Product quality is the second major area of product support documentation impact. Early in the proposal and specification stages, cumbersome documentation tools often result in limited review at the time of highest decision leverage. Because indices, tables of contents and other document structures are too difficult to create manually, they are generally missing from preliminary documentation such as that sent to product beta test sites. Everyone is familiar with last minute changes which leave management with a choice of holding product shipment or incorporating unwieldly documentation errata. All these phenomena reflect in some way the quality of the product of its literature.

Solutions to the quality issue begin with a system architected to make multi-author review easy. The document originator distributes the document to any number of reviewers



A Concept workstation.

electronically. As each reviewer notates the document, the system collects name and comments from multiple reviewers for feedback to the originator. The originator may then process the comments or send all comments back out for additional input.

As the production stages are approached, the value of What-You-See-Is-What-You-Get (WYSIWYG) display begins to add to literature quality. Multiple fonts and multi-column output improve the readability of the documentation. Laser printer output provides excellent graphics reproduction and near-phototypeset text legibility.

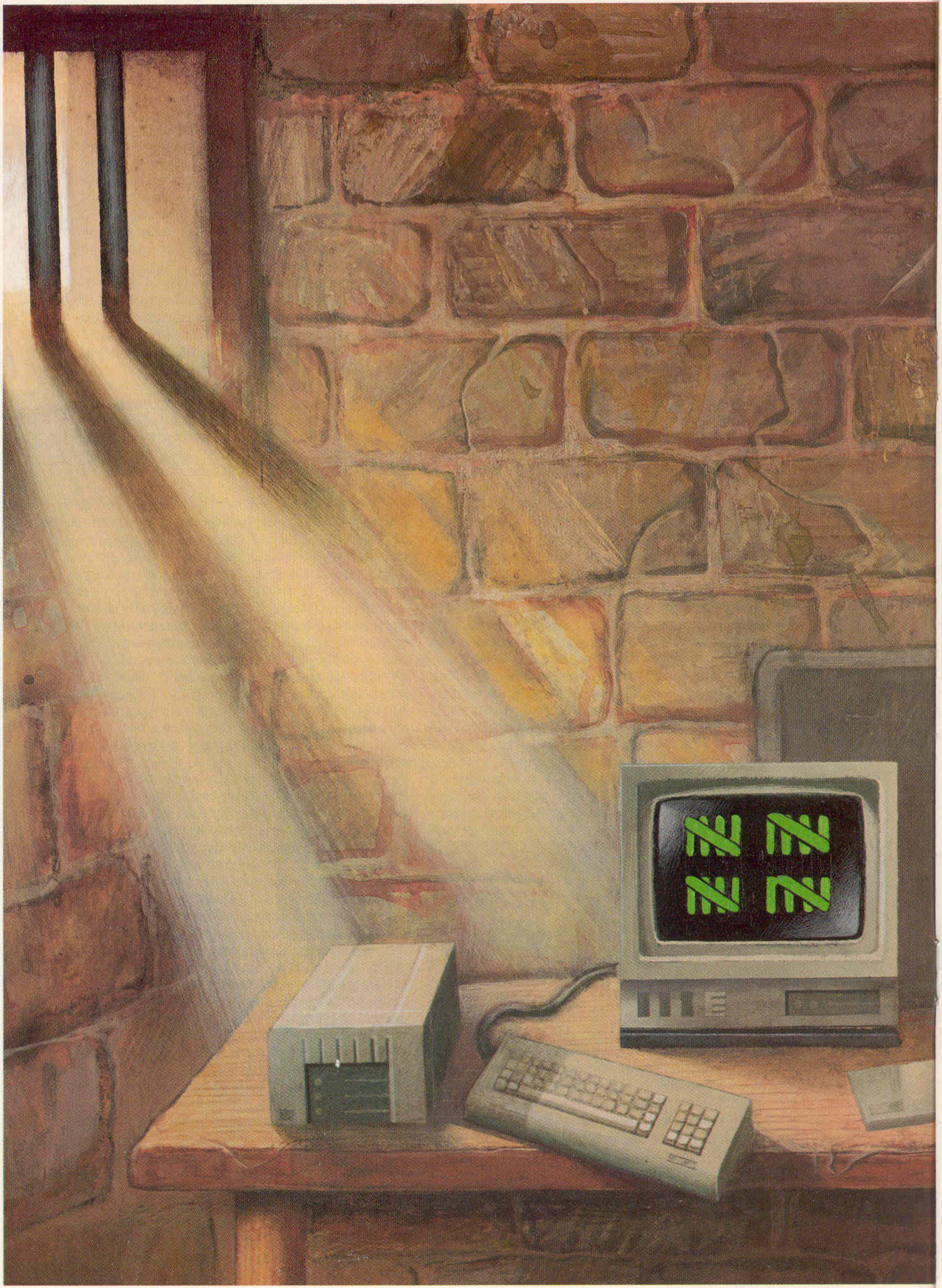
Automatic structure maintenance makes possible such quality features as table of contents, list of figures, tabulated index and others, not only at production time but also throughout the development and testing of the product. Such capabilities taken together improve both the product and the literature which represents it.

In addition to their impact on time-to-market and quality, the right documentation tools can have significant impact on development costs. In the current environment, product designs and their documentation are typically developed on separate systems.

Text and graphics, though closely related, are typically developed separately as well. If graphics creation is done by an illustrator, there is an added burden of information transferred between the originator and the illustrator. In many cases, significant information is lost. Revisions are particularly difficult in a non-integrated environment. Changes to graphics involve expensive, labour-intensive rework unless they are accessed directly from the design database. Even a computer-generated graphic which is copied instead of referenced into a document will need to be changed for each document in which it has been used. Similarly, last minute revisions to text often require changes to structured elements such as table of contents and indices. Of course, the situation is compounded for large documents in excess of 1,000 pages. Finally, in an effort to obtain visual quality, organisations often pay the control and overhead costs of using commercial publishing services.

Solutions to the cost problems begin with the documentation tool architecture which permits inclusion by reference of both text and graphics generated by design automation tools. As referenced elements change, automatic update of the documents that use them saves the cost of expensive rework. Similarly, integrating graphics and text, particularly with multi-column format, minimises the need to use outside commercial publishing services as well as the need to use internal illustrators for paste-up. Instead, illustrators can improve their productivity by using the illustration systems to produce graphics. Large document design becomes feasible when automatic structure and format take care of the document's context. Writers are allowed to focus primarily on content.

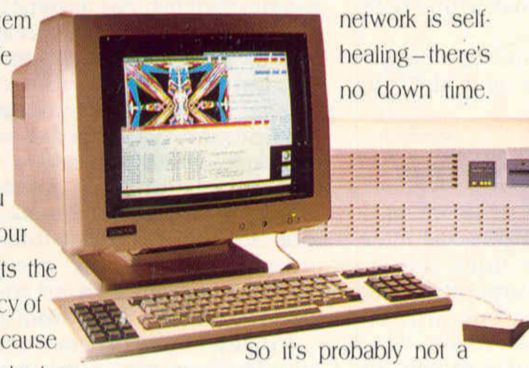
The document database is as much a part of the products as are the chips, printed circuit boards and mechanical structure it supports. Depending on one's perspective, it may actually be the product. Workstation, graphics, networking, and laser printing technologies have converged to make feasible solutions to documentation process issues. Solutions which link closely with design databases can provide high returns of time-to-market, product quality and development costs.



ARE YOUR WORKSTATIONS IN SOLITARY CONFINEMENT?

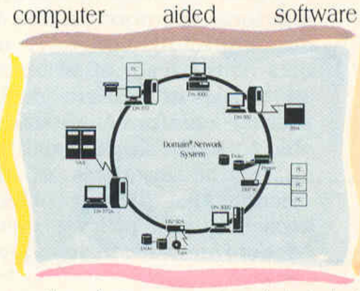
Apollo invented the workstation. At the same time we also produced the ultimate processing environment for high-speed local area networking – DOMAIN® We believe that creating a single computing environment for all workstation users is as important as the capability of the workstation itself. Choose the wrong system and your staff could face life sentences of inadequate performance. That's why the more workstations you add to your DOMAIN, the more you enhance the capabilities of your system, which in turn boosts the total productivity and efficiency of your work groups, because DOMAIN allows transparent sharing of information and resources across the network. Better still, DOMAIN connects to and enhances the functionality of other computer systems using industry standard protocols such as X.25 and ETHERNET.®

DOMAIN processing eliminates the need for multiple file copies thanks to total data integration across the network. In short, DOMAIN brings you optimised project time and thus far greater overall project control. Should a workstation require maintenance for any reason, the network is self-healing – there's no down time.



So it's probably not a great surprise that the majority of technical engineering workstations in use are from Apollo. Or that our product support and training are second to none.

Today more than 600 of the best application packages run on DOMAIN, covering electronic design, computer aided software



engineering, computer integrated manufacturing, artificial intelligence, and much more. Naturally, DOMAIN supports UNIX® too.

If your company has a need for high performance technical computing Apollo brings you the power – and the freedom to use it.

Judge for yourself by contacting:
Jeanne-Marie Richards
Apollo Computer (UK) Limited
Aegis Park, Bramley Road,
Bletchley MK1 1PT. Tel: (0908) 366188

apollo

Workstations for CAD/CAM

Part Two of Bob Henson's evaluation of workstation technology, looking at Networking and UNIX

by R E Henson
Apollo Computer (UK) Ltd.

The design engineer who is seated in front of a 'top-of-the-range' CAD/CAM workstation is in a fortunate position. But how, you will ask, does the design engineer use and store the data of the team of which he is a part? The answer lies in the network. Users of big time-sharing mainframe systems have always been able to access large shared databases and use all of the resources of the system. It is essential that the networked workstation approach should provide these same facilities. Modern network technology makes this possible.

Workstation networks fall into two distinct categories: the bus structure and the ring structure.

The bus structure

Networks which use the bus structure most commonly use Ethernet as the physical interconnect. Examples of vendors using this approach are Daisy Systems Corporation and SUN Microsystems. Using the bus structure of network, each workstation normally has a discrete operating system which usually resides on a disk local to each workstation. The executable code also resides on a local disk and demand paging takes place between the individual workstation and the local disk. Communal files may be stored on remote disks with network software transferring data at record or file level to the local workstation disk. These local disks are referred to as 'swapping disks'. System resources, such as plotters, may have files transferred to them using the same approach.

The SUN Microsystems 'Network File System' is an example of a network system which provides file services over Ethernet. Network File System is built on top of the native operating system of the workstation (currently the UNIX variant SunOS on SUN). The SUN Network File System also allows individual workstations to operate without local disks (diskless operation), significantly reducing the overall cost of a network.

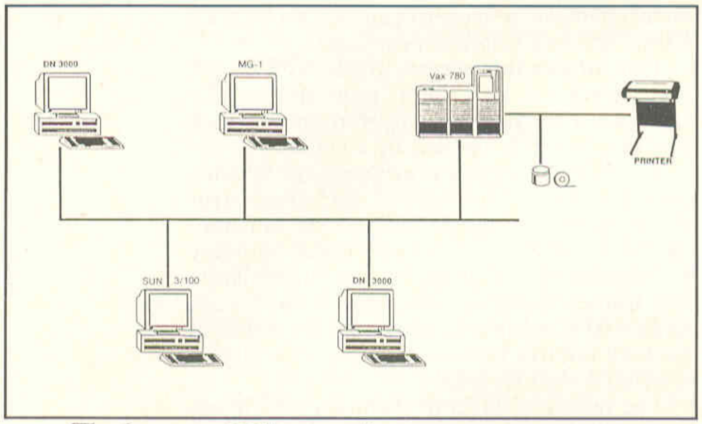
Other software systems are available to provide file transfer capabilities between UNIX-based workstations; these include the 'Newcastle Connection'.

One of the reasons for specifying Ethernet as a physical interconnect is that it is, to a degree, an industry standard (IEEE 802.3). Many manufacturers either use Ethernet as their standard interconnect or offer gateways from proprietary systems to an Ethernet.

Ethernet was designed as a means of passing data at high speed (10 Mbit/sec) on an occasional basis between different machines. Ethernet operates in contention mode with random re-try times for busy or clash conditions. This may be likened to a non-prioritised crossroads, where vehicles cross if the route is clear and wait if it is not. Since the basis of arbitration is random, transfer speeds may not be predicted. Experience shows that transfer times via Ethernet degrade rapidly as the utilisation exceeds 30 per cent of the theoretical bandwidth. Once serious contention occurs, the 'sort-out' time is long and unpredictable, rather like the aftermath of a motorway accident. Ethernet is also restricted in overall length and over 2Km is reported to be unsatisfactory. Ethernet systems perform well when the various workstations are well supported with local disk facilities and the traffic on the network is not unduly high. There is no technical reason why demand-paging systems, for instance, cannot be implemented over Ethernet, it is simply that the lack of predictable response renders the system unsuitable.

The IEEE 802.3 standard defines the physical and data link levels of Ethernet; it does not define network or transport protocols, thus the Ethernet bus connecting a community of like workstations cannot necessarily interface directly with a 'foreign' machine. In order to achieve this interface, it may be necessary to configure an additional Ethernet gateway on one of the workstations.

In addition, a network and transport layer of software is necessary, which must be supported by both machine types. Currently there is no standard. An emerging standard, however, is the US Department of Defense Transmission Control Protocol/Internetworking Protocol (TCP/IP). Implementations of the TCP/IP exist for many operating systems, on hardware ranging from super-computers down to personal computers. With higher level protocols, file transfer and other user level services are possible between heterogeneous systems.



The bus structure of network.

The ring structure

Ring structure networks are one-way systems with all of the workstations linked in a 'daisy-chain'. Examples of ring networks are the Cambridge Ring and the Apollo Domain.

The Apollo Domain network is currently based upon a 12 Mbit/sec token-passing ring. The system can be likened to a circular underground railway with a continuously moving high-speed train, onto which packets of data may be placed for transmission to other stations around the circuit. The advantages of token-passing rings are that they can span greater distances and transmit at greater data rates; furthermore, heavy utilisation of the system does not impair the transfer rates. Apollo workstations can be located up to a maximum separation between active workstations of 1Km, using 75 ohm coaxial cable, and over 2Km, using fibre optics. There is no physical maximum on the number of workstations that can comprise a network.

The network interface into a workstation is normally via a removable plug, which facilitates maintenance. The plug contains a relay which is powered from the workstations. If the power is switched off or the plug removed from the workstation, the relay switches to complete the network. In order to increase maintainability of ring networks, many users adopt a star-type configuration. Using this approach, each sublocation has a network which returns to a central point, thus the network may remain as a single network or subnetworks which can be switched out for fault correction.

General Motors is also committed to the token-passing approach, with its Manufacturing Automation Protocol (MAP), which is designed to be a high-speed factory floor data interconnection system. The indications are that token-passing network speeds will increase in the coming years by at least an order of magnitude.

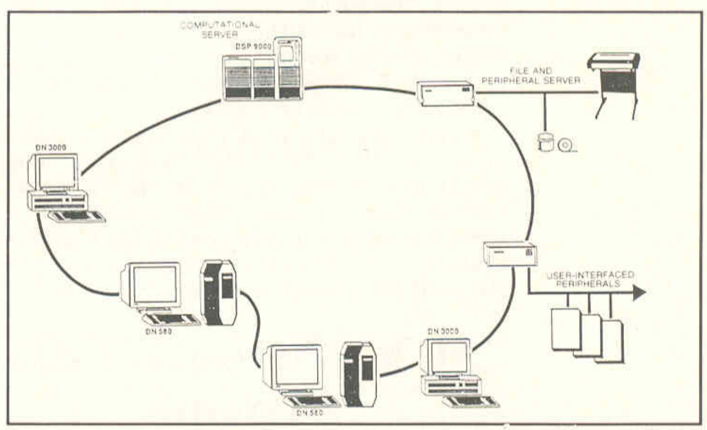
Most workstation vendors utilise a discrete operating system per machine with file transfer facilities between these machines. However, this approach has limitations, since quite severe restrictions are placed on the flexibility with which users can access files. New systems are being developed to overcome these restrictions. An example is the Apollo Domain system.

The Apollo Domain System

Apollo offers a single operating system environment which is network wide. This is a virtual memory operating system which provides demand paging around the network. The advantage of this approach is that users of the network as a whole have the same facilities as the users of a timesharing mainframe computer. A user logging in at any workstation can use code and access files on any disk storage device in the network, likewise the user can use all the system peripherals, computational server processors, attached processors or gateways, with complete transparency. This design allows users and their programs to view the system as an integrated whole, rather than as a collection of individual workstations.

Because the Apollo Domain system integrates virtual memory with a network-based storage system, an individual Apollo workstation within a network does not require its own disk storage in order to operate, indeed many Apollo networks operate from centralised disks with as many as 12 workstations per disk.

Currently the type of global operating system that we are discussing is limited to single local area networks. The future, however, has much in store. There is no reason why such systems should not operate over heterogeneous networks, for instance, between two ring topography networks linked via an Ethernet, which could also be part of (say) a VAX cluster. Equally, there is no reason why such systems should be restricted to local area networks. The practical implementation of demand paging remotely over public networks only awaits circuits with sufficient speeds.



The ring structure of network.

UNIX

Amongst workstation vendors, the most commonly used operating system is UNIX. The UNIX operating system was originally developed through the effort of a research group at AT&T Bell Laboratories. At the time, the system provided facilities that were not generally available on minis; these included a hierarchical file system, compatible file, device and interprocess input/output and a powerful set of programming tools. Further developments, many from the University of California, Berkeley, added virtual memory management, and aimed at turning UNIX into a portable system; to allow portability of code between different machine types.

The really thorny issue is that UNIX was developed before LAN technology became common. To use UNIX in a network environment, it is necessary to add additional software. The facilities offered by different manufacturers in this respect vary greatly. A number of systems are discussed.

AT&T's Remote File System

One of the primary goals of AT&T's RFS is to provide transparency between remote and local file systems. Another goal is to provide sufficient mechanisms to allow local system administrators to assure the confidentiality and integrity of their own data, and it also aims to maintain UNIX file system semantics and concurrent file access.

RFS is a 'stateful' system, RFS keeps a count of how many local remote programs have a particular file open, and it ensures that data written from one program in a single write request are not intermingled with data from another program on a different machine.

RFS uses the streams I/O (input/output) system for intermachine communications. This system allows the implementors to plug in any one of several network protocols and makes RFS independent of any one kind of network hardware or protocol.

A major advantage of RFS is that it maintains the UNIX file system semantics. This means that, remote and local operations behave in exactly the same way. For example, a server knows how many times a file has been opened, so it can safely decide when the file can be deleted after an unlink operation.

SUN's Network File System

The design goals of Sun's NFS are similar to those of AT&T's RFS and include transparent file access, reliability in the face of imperfect networks and machines, and maintenance of UNIX file system semantics. NFS, however, attempts to achieve the more ambitious goal of providing transparent file access among machines that might be running operating systems other than UNIX systems.

The NFS protocol is a set of primitives that define the operations that can be made on a distributed file system. In contrast to AT&T's RFS, NFS is a stateless protocol.

This implies that servers under NFS do not keep track of any past requests — for example, a server does not even know which files are currently opened by a client.

The NFS approach to network file systems provides several advantages, including error recovery, system independence, and availability. NFS makes error recovery quick and easy by eliminating the saving of state information that tremendously complicates error recovery. Unlike RFS, users may not notice intermittent network failures because no state information is saved or lost by failure.

The failure of NFS to maintain UNIX operating system semantics for remote files is a weakness. For example, guaranteed append mode and Berkeley advisory locks are not supported under NFS for remote access. Additionally, a file being used by one user can be deleted by a second user. NFS's failure to adhere to the UNIX file system semantics prevents users from trusting a shared file.

Apollo's 'Domain File System'

Apollo's implementation of the UNIX operating system extends virtual memory beyond the local disk and throughout the entire network. Any workstation or server can demand page from anywhere in the network, making local disks optional. The token passing ring topology of the Apollo network is especially suited to this task, since it allows thousands of network accesses each second. Demand paging across the network means that data is always left in place, and is only transferred, pages at a time, when needed.

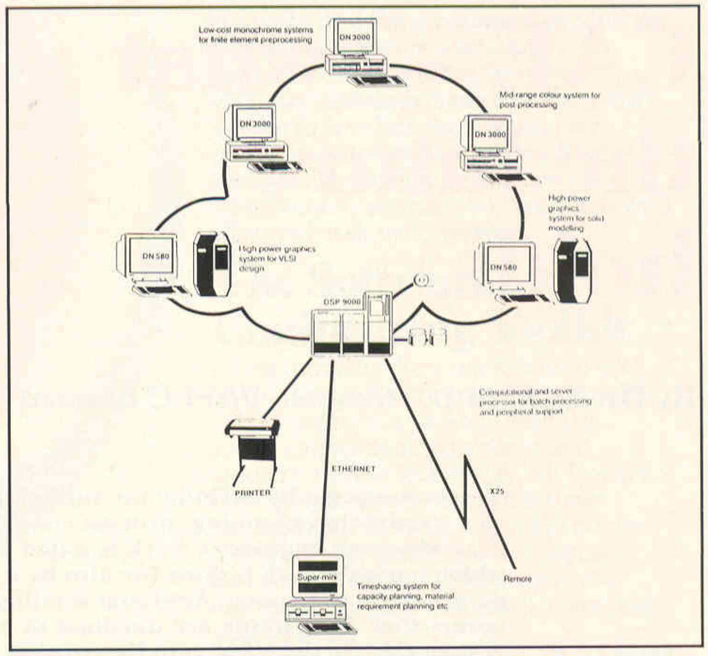
The combination of network-wide demand paging and a distributed file structure provides the UNIX operating system with the power of distributed processing, while maintaining the data sharing capabilities previously found only on large timesharing superminis and mainframes.

DOMAIN/IX is Apollo's twin port of the two standards of the UNIX operating system: Berkeley 4.2 and System V Release 2. Users can run applications in either operating system, or both simultaneously, from a single workstation. DOMAIN/IX provides users with all the benefits of the UNIX standards integrated into a distributed processing environment.

The one problem with Apollo's DFS is that it does not provide transparent file access across the diverse set of computers found in today's data processing environment. Consequently, Apollo and other vendors have been extending their systems to support heterogeneous networks. Creating heterogeneous file system that can perform and function well is a difficult task. Lack of co-operation between vendors often leaves network designers with no choice but to use certain communication protocols that are common to all machines but can lack sufficient functionality and performance.

Network configuration

The internals of CAD/CAM workstations have been discussed in some depth. It is now important to understand how a network may be configured in practice. A network of workstations serving a community of users is illustrated.



To meet the needs of mechanical engineers, a number of low-cost monochrome workstations with mid-range computational power are configured for finite element preprocessing, using an application package such as PATRAN-G or SUPERTAB. The solution phase is passed for batch processing to a computational server processor, a network resource configured for this purpose. The post-processing phase could then be performed on a mid-range colour system, using, for instance, colour shading to show different areas of stress. The needs of the solid modelling group in the community of users are met by a high power/complex graphics workstation. The community includes an electronics computer-aided design group which is concerned with producing custom-designed VLSI circuits; this group is also using high power/complex graphics workstations.

Various other resources are available as part of the illustrated network. These include server processors, attached to which are large file stores. Eight-inch, 500 Mbyte Winchester technology disks are available and a single cabinet can contain four of these systems, giving units of up to 2 Gbytes each, thus very large engineering databases can be stored using networked systems. Also illustrated are plotters, printers and communication gateways to other systems, such as capacity planning, material requirement planning and purchase ledger systems running on separate computers.

Care must be taken in correctly configuring a network. The relationship between the number of workstations and the number of disk drives must be calculated in order to ensure that disk accessing does not create a bottleneck. The considerations here are very similar to those for a mainframe or super-mini environment, except that disk traffic in the workstation environment is likely to be less, due to the presence of dedicated computational processors. This significantly reduces the disk traffic, since, in a timesharing environment, much of the traffic results from the operating systems swapping the priority of tasks sharing the single processor. The server processors used in the network environment usually configure at least 2 Mbyte of cache memory; this enables network operating systems to transfer directly to users those records which have already been accessed and are available in the cache. Performance improvements arising from this facility can be considerable.

In order to keep a network of workstations well tuned for performance, some system administration is necessary. However, this usually proves to be a less onerous task than keeping a mainframe system well tuned. In the mainframe environment, not only does the operating system soak up a considerable number of CPU cycles, but so do the systems programmers. A recent survey showed one site where the systems programmers actually consumed 27 per cent of the available prime shift CPU cycles to tune the operating system. Needless to say, the engineers for whom the facility was provided normally had to content themselves with an overnight service.

Conclusion

In summary, the style of computing offered by networked workstations has numerous advantages when compared to traditional forms of computing such as graphics terminals linked to timesharing super-minis or mainframe computers. In the first place, each workstation may be a stand-alone system capable of solving complex engineering problems. The initial cost of such a system may be low, since no central components need to be purchased. Users can expand the system in increments which meet their particular needs, at a cost that is affordable. In addition, users enjoy a predictable and constant level of performance that allows them to schedule their time without regard to computer availability. Thus, the workstation approach integrates the advantages of a mainframe computer with high-performance, local area networking and graphics capabilities at a cost per user that is well within the reach of engineering and graphics applications.

This article was first published in State of the Art Report, Pergamon Infotech, 1985.

CAE — what the future holds

By Dr. Richard D. Henshell, PAFEC Limited

Let me begin by defining my subject. By CAE I mean any design draughting, pre-production planning, process control, stress analysis and geometric modelling where an engineer's work is aided by a computer. I do not include activities in which a man's work is done for him by a computer. Such things are talked about but do not seem to happen. Artificial Intelligence (AI) may change that somewhat, but it seems that AI systems are destined to aid rather than replace man's thought processes. CAE is the most rapidly growing area in computing today. We can therefore expect lots of innovation and change.

The workstation versus mini or mainframe argument has largely dried up. Today the cost of processing power has fallen so much that it is almost irrelevant when considering a good quality graphics screen. It now makes sense to have a powerful workstation to handle the large graphics needs of engineering even if it is coupled to a mini or mainframe. The price of disk systems on workstations remains very high. The likelihood is, therefore, that we shall see workstations with sufficiently large memories so as to make disks irrelevant. When workstations have 100Mb — and that day is not far off — the need for paging and scratch disks will have vanished.

Back in the days of storage displays I used to think that serious resolutions needed to be 4096 square. It is now clear that this was wrong. Screens with 1500 square pixels are not so markedly better on the eye than those with 1000 square. What I should like to see for engineering use is a much larger screen before bothering with higher resolution. 30″ diagonals with 2000 square pixels would be super, but hardware people tell me such things are impossible. What a pity!

We all know that processors are getting faster or cheaper or both. It seems to me that there is another trend. Looking at my own graphs for MIPS/£ or MIPS/salary over the last ten years I see a steady increase for the first seven or eight years and, recently, a dramatic change in the slope upwards. Is this a step change which will not necessarily be repeated, or are we on a different curve now? Looking at the nature of the trends, I think that pace is accelerating. One might think that the price of the hardware will fall at such a rate and the performance will go up so quickly that hardware itself will become irrelevant compared with software. The conclusion of this is that software suppliers will rule the world.

However, there is another factor to consider: the hardware suppliers do not only sell hardware. Most of their effort and value goes into software, in the form of the operating systems, user-environment, design of software and firmware functions such as bit-map handlers, display list processors, communications protocols and so-on. The list is long and getting longer all the time. These are the factors which separate one so-called hardware manufacturer from another.

It is the software houses who have therefore to watch out. The hardware companies are buying or redeveloping their software products, putting them onto chips and giving them away free with a screen. Those software houses which expect to stay around must move on and provide those functions which the manufacturers are either not interested in or unable to market successfully. But software houses cannot afford to slow down, because as they become successful in a new field that too will find its way into chips.

So how does all this affect the engineer trying to do a job. I believe that he will not see the rate of change which computer hardware and software companies experience now or in the future. The engineer requires a more stable environment and will seek more permanent standards to work with. This is where the software house can score. Through all the frenetic activity in the computer scene they *must* provide the permanency and standards which keep the real world revolving.

It is very easy to develop software products to fill a niche today. It is much harder to create software which will not only fulfil current needs but act as a foundation for the next and future generations.

Everyone recognises the growing importance of software in industry and research. And the economical production of practical systems is absolutely fundamental. Too often, however, everyday pressures mean that there isn't time to fully refine the system; user interaction suffers and the resulting compromises in quality and cost-effectiveness can make you wish that we'd all stayed with steampower . . .

The Software Engineering Toolkit

Fortunately, there is a solution; with PA's unique software engineering toolkit (SET) you have a package based on sound applications experience and industry-proven technology. SET offers flexible user interface and database management features which are not just cosmetic; they accelerate development and production functions to give you more user facilities at a much lower cost.

SET: Proven in Action

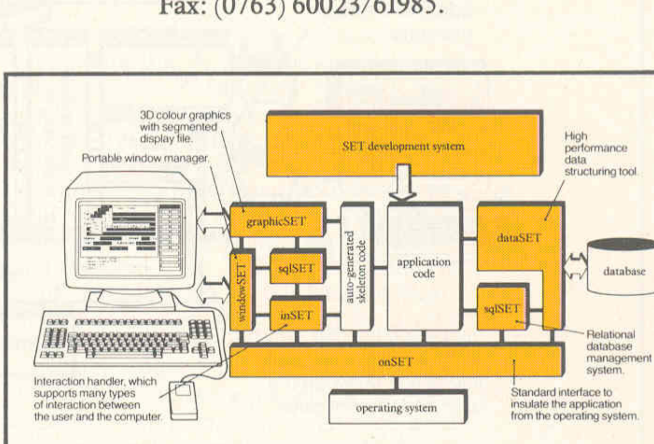
Our current users (from CAD/CAM to graphics vendors and from mapping to astronomical research) are already using SET to develop better systems than the competition and they're doing it faster: prices are low, payback high and – perhaps most significantly of all in cost terms, the resulting software is portable and maintainable. We support SET on popular displays, 32 bit workstations and time-shared computers.

If you want the latest advances in technology working for you, contact PA. We'll provide everything from a single module in the toolkit to a complete SET system, – or we'll write your applications for you . . . using SET, of course. For details of SET and our bespoke software services, contact: Tony Bishop or Norman Schofield on (0763) 61222.

PA Computer Aided Engineering,
Cambridge Laboratory, Melbourn,
Royston, Herts SG8 6DP. Telex: 81561.
Fax: (0763) 60023/61985.

DON'T WASTE TIME ON SOFTWARE ENGINEERING...

...PA's SET users don't!



PA

Fast Friendly User Interfaces

Technical applications need highly interactive, graphic user interfaces. PA has pioneered methods which enable all software engineers to prototype and build such systems at low cost.

by Dr Tony Bishop,
PA Manufacturing Services

Computer software has always been notoriously difficult to produce. The problem has been made worse with the advent of modern interaction techniques, graphics and complex data structures. PA saw the need for a new generation of software tools at a time when many organisations were trying to build highly interactive systems and port them from one computer to another. PA's Software Engineering Toolkit (SET) was developed to meet this need.

The background to the development of SET was the requirement of PA (an international consultancy group) to satisfy the many diverse needs of its clients in acquiring, developing and integrating technical software, mainly in the CAD, CAM and CAE fields. This necessitated being able to develop one-off software systems quickly and cost-effectively, but at the same time ensuring that the end result would be portable, robust and maintainable as easily as packaged software.

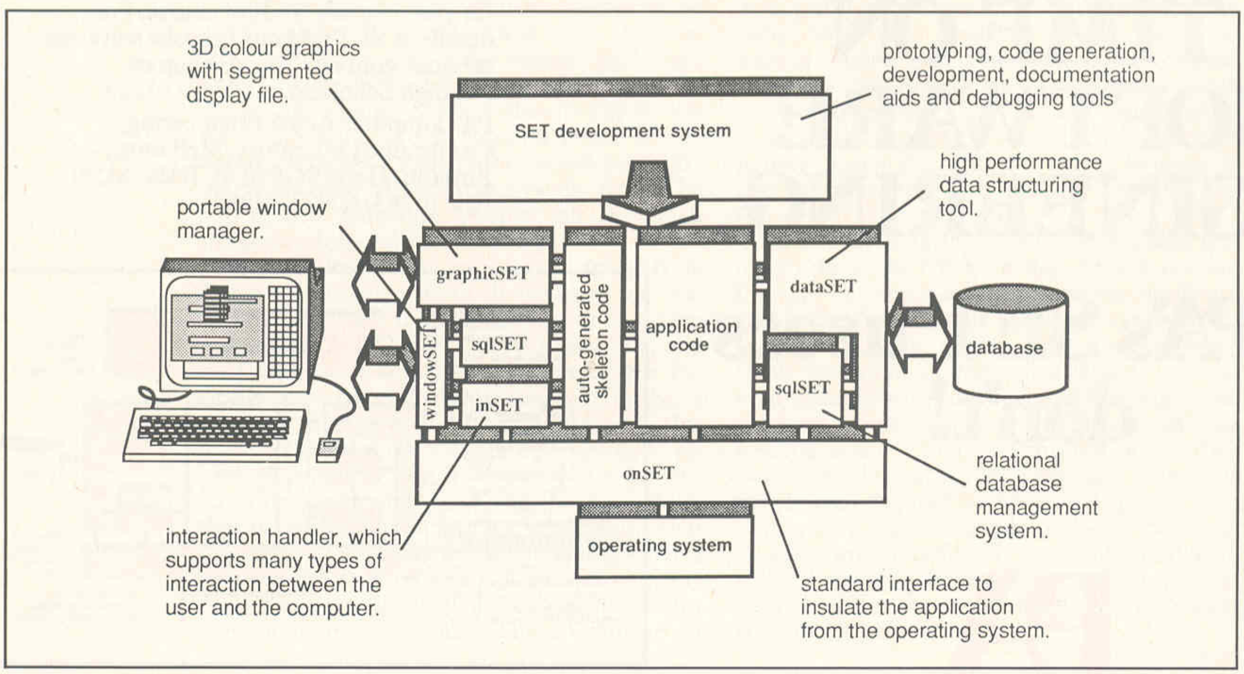
New techniques in software engineering were studied at the outset. There are many approaches to software development and each have their merits but, with most, the major benefit is due to a discipline in project organisation. The key aspects are to think before doing and to specify before writing. With a good structural approach to software development, the merits of one 'system' against another make little effective difference.

Having defined a strict project control mechanism, attention was turned to software development aids; those that help generate code (development utilities) and those that can be used as part of the final package (building blocks).

The Complete SET

SET consists of a series of compatible tools, which are now marketed world-wide to those producing technical software. Users include CADCAM vendors, manufacturing companies and research organisations. The tools are supplied separately, or in combination, and currently include:

* inSET — an interaction handler which can transform the user interface. It also contains powerful tools for prototyping the user interface.
* graphicSET — a comprehensive graphics package which includes 2D and 3D,



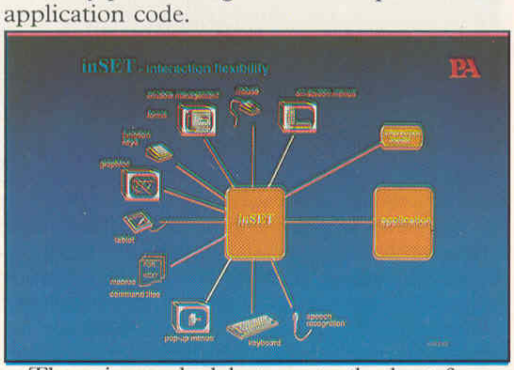
The complete SET development system and run-time modules.

together with lines, surfaces, text and colour.
* windowSET — a portable window manager which gives Apollo-like facilities on a wide range of text and graphics terminals.
* onSET — which insulates the application from the operating system. An application can thus normally be ported between computers in a matter of days.
* dataSET — a data structuring system which provides high performance whilst supporting large, complex data structures.
* sqlSET — a relational database management system, which is a comprehensive implementation of IBM's Structured Query Language.

The most innovative area of SET is the integrated set of man-machine interface tools and it is on these that the remainder of this article concentrates.

Interaction handling

An interaction handler provides a buffer between an application user and the code specific to the application. Given a set of interaction rules the system should be capable of handling input through any one of a number of input devices, carry out error handling and only pass through verified requests to the application code.

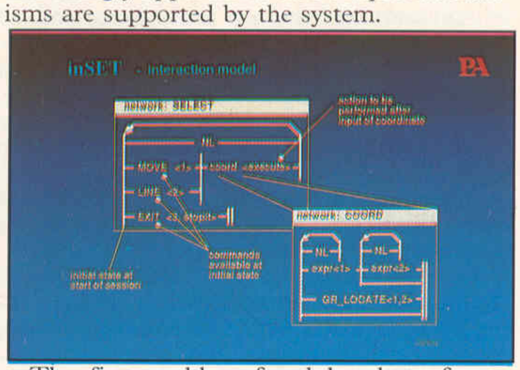


Many forms of user interaction must be supported.

There is much debate as to the best forms of user interaction. There are strong exponents of keyboard, touch screens, function buttons, forms, tablets, light pens and the mouse to mention but a few. In each case an example can be found that would back up the argument. However, in most applications there are usually different parts that are suited to different approaches. The eventual choice should lie with the application user. Where possible many interaction methods should be available concurrently and others upon demand. The layout of an application 'screen' can also be relied upon to cause debate; some like errors to appear at the top, others to the left whilst others prefer not to see them at all. Once again the application user should have the final say even though the application may suggest an initial layout. These were two goals for the design of the interaction handler.

In order to cater for many forms of input a common mechanism was sought for the analysis phase. It is to this common form that input from any device is converted prior to checking the interaction model. Various levels were considered but eventually a buffered command line was adopted as the standard. All the input devices supported would be mapped to a command line equivalent for processing. There are many attractions to this approach:
* It can be tested easily, since a keyboard can simply load the buffer.
* Any input can be trapped at the command line level and saved as a recognisable trace of the session.
* Command files (macros) can be accommodated simply.
* The definition of other input devices is simplified to a level of describing the equivalent command line for a given operation.

The flexibility of this approach has become increasingly apparent as more input mechanisms are supported by the system.



The interaction mode 1 defines the commands available to the user and the code to be executed.

The first problem faced by the software engineer is the definition of the interaction between man and machine. After studying various notations for syntax definition a system was adoted based on finite state transition networks. These could be represented graphically allowing an application designer to readily convey the options available to the users. The graphical representation of these networks shows the application states as vertical bars whilst the horizontal connections indicate the available transitions at each state.

inSET recognises several forms of 'transition' and from this basic set many more complex forms can be constructed. At the lowest level a transition can be defined for Numbers, Characters, Text, Graphics and some special functions.

From these a network can be constructed. It is soon apparent that to describe any large application by means of a single network is a major task. inSET therefore provides the ability to define sub-networks and call these as if they were themselves transition types. The calling sequence can be deep and totally recursive if necessary. The numeric transition types can also be constrained to a specific range of values. The NULL transition is worthy of mention due to the level of flexibility it provides. Any state linked by a null transition to another has the options of both states available concurrently. A technique that has proved successful in the design of networks involves a spiral approach. At each level the user can specify parameters to the latest commands whilst having all other command options available. This is a form of interaction that usually feels natural to the application programmer and user.

Given the command line structure of the system it was a logical extension to provide automatic abbreviation analysis. For the users selecting to type at a keyboard this allows for commands to be issued using only that portion of a command that uniquely identifies it. Other facilities were added at this level as they are then applicable to all the input forms. The first is the ability to assign a security level to each transition. When a transition is being checked against the valid options only those that are available to the users security level are scanned. Any transitions requiring a higher security level simply do not appear to exist for that user.

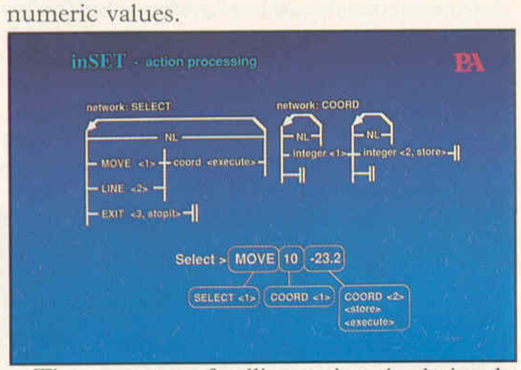
A second facility is the provision of parallel networks. Within a parallel network, commands can be specified that are equally valid throughout a section of the application (e.g. ZOOM or PAN). If the application user enters a parallel network all the data related to the current position within the main network is maintained so that upon exit the application can continue as if nothing had occurred in between.

Several networks are pre-defined for use in an interactive application. At one level a common set of sub-networks have been defined, effectively providing extensions to the transition types, such as a full expression handler. A full macro language has also been implemented as a parallel network providing many features of the BASIC language, including the use of real, integer and string variables. With these facilities a user can write command files requesting user input following conditional test and branching, open files, print output etc. Finally an 'open ended' parallel network is included that can be defined at run time. Into this network the application user can add his own commands (synonyms) along with the expanded command line he wishes to be executed whenever it is identified. Implementing it as a network allows the system to treat synonyms in the same way as other transitions, including security checks and minimum abbreviation analysis.

An important part of any interaction handler lies in its ability to provide guidance to the user. The use of the network structure immediately allows for the provision of context sensitive guidance. When waiting for input a particular state is current. A prompt string can be defined for each state and this is displayed on these occasions. Typically a prompt could be 'specify authors name' or some clue as to the next, most likely, input. Note that these prompts appear when the system is waiting for input. An experienced user could complete a command line in one go and hence be spared the prompts associated with the intermediate starters traversed. In areas of the application not used so often the prompts can be helpful to lead the user through the required input.

In case the prompts are not sufficient, or perhaps not defined within a network, the user can request notification of the available options at a state. The resulting display includes a list of all the options along with a simple description; 'transition help text'; of each as it has been defined in the network description. At a higher level still the system can generate a filename based on the position in the current network and state and display the contents of this file. Typically, parts of the user reference guide for an application can be split up into files for this purpose.

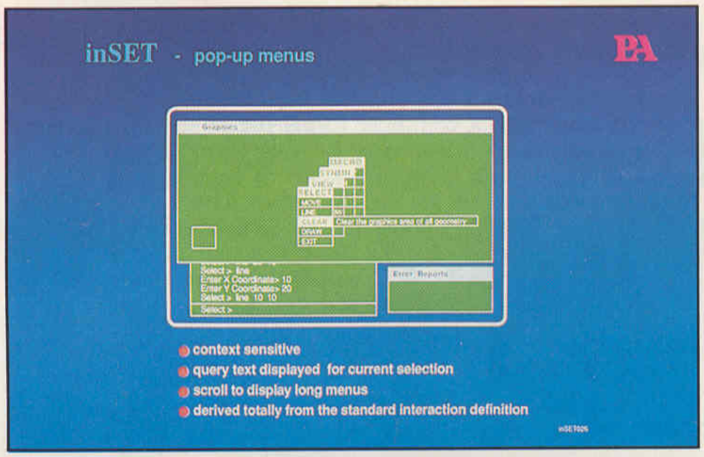
The software engineer can link the application to its defined interaction model through the definition of actions, or blocks of application code that can be associated with the network transitions. As a sequence of transitions is verified an associated stack of 'action' calls is accumulated. Once cleared the actions are called in sequence. From the 'action code' the application can interrogate the associated transitions to obtain more information such as numeric values.

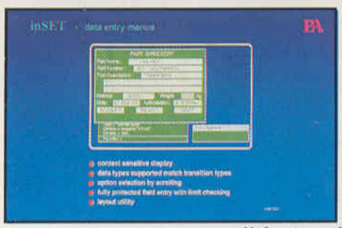


The application routines executed for a particular command.

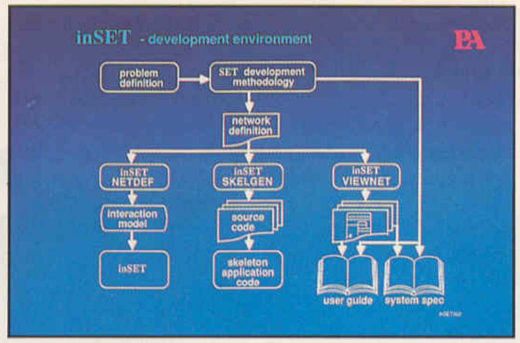
The sequence of calling actions is obviously crucial to an application and this can be traced along with many other aspects of the interaction handling. Simply by issuing a macro command the act of calling 'actions' will also cause a trace message to be issued to the screen or named file. This provides a quick mechanism for evaluating a user interaction model even with the actions being empty, i.e. at the prototype stage.

As described previously the buffered command line common format provides a route for many input facilities to be supported. Examples are the 'pop-up menus', data entry menus and command menus. For certain devices a trigger (e.g. a mouse button) for pop-up menus is defined. When recognised, this key causes a scan of the available command options to be made and loaded onto screen maps which are then overlayed onto the screen by the device driver. Movement of cursor keys, mouse etc, cause various parts of the menus to be selected and a second trigger (perhaps the release of the key) forces the selected option to be loaded into the command line for execution. The benefit of this form of interaction is that options are displayed at the point of interest. It is not necessary to move to a separate area to issue commands or scan options. Due to the close coupling of these pop-up menus to the networks it is also





Interaction using pop-up menus, data entry via menus, and command input via static menu.



The Inset development method.

possible to show the transition help text for the transition currently selected. This appears if the user does not change selection in under half a second. Again, this prevents an experienced user from having many prompts flashing up but provides support to the less experienced user.

Data entry and command menus are also provided as a closely coupled extension. Generally known as forms, they can be defined and associated with states in a network. However, one form may span many states if the resultant command line passes through many states. In the case of forms, new information has to be added in order to provide a flexible capability for layout and presentation. This information is held in a compressed format and scanned when the interaction handler requires input and the user has selected the forms mode of operation. Even when using forms the command line can still be made visible and data entered through this medium. It is the application user who decides his preference for the form of interaction at any particular time.

Application development

An interaction handler is a useful 'building block' providing consistent style of interface and form of interaction that often could not be justified for a single software project. However, there is a further spin off. The interaction model defining the rules of interaction has a lot of information regarding the modules and actions of the application itself — not what each action does but rather when each is called and how many exist. This has led to the development of an inSET module called SKELGEN which automatically generates skeleton application code based on the interaction model. By scanning the interaction definition a series of skeleton procedures can be generated and commented by the inclusion of details regarding the calling situation. The resultant output is capable of compilation and immediate execution when linked with the interaction handler library.

This method of partially automating the generation of code offers considerable flexibility.

* Initial skeleton code production is totally automated and will compile and execute at an early stage in the development process, providing a rapid route for prototype testing.
* The code generated conforms to a known structure which can be tailored by the software manager to suit his particular development environment and subsequent maintenance requirements.
* Application specific code can be added action by action with testing possible at each stage.
* The overall result can be highly optimised for its purpose, rather than accept the compromise resulting from a totally automated process.
* The techniques have very wide application.
* Existing applications can be converted rapidly to use a new structure of user interface.
* The software engineer remains in control at all times but can produce more sophisticated applications in a *predictably* shorter timescale.

Portability

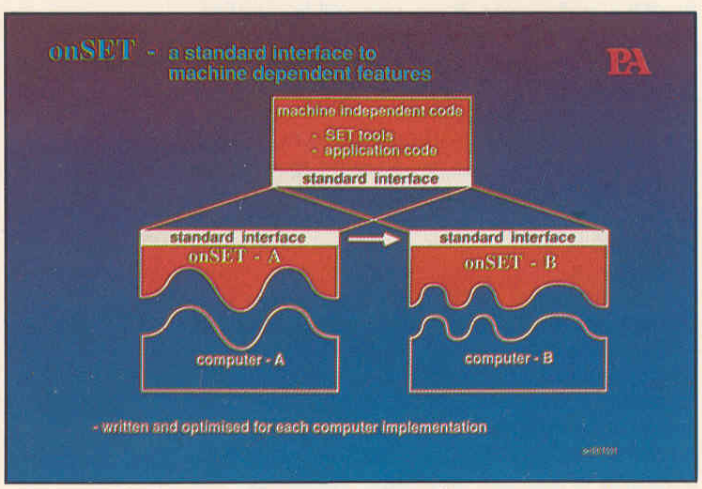
The interaction handler, like most portable software tools, has a readily identifiable machine dependent interface called onSET. Unusually the interface to this machine dependent layer is fully documented and its use by application programmers encouraged. Features such as message handling, file handling, input/output, character manipulation are performed by onSET.

Another crucial aspect of portability is the display, and in particular window management. windowSET provides a consistent interface to the multitude of window managers appearing on workstations today. Until now application programmers wishing to use these capabilities have had to lock themselves to a particular piece of hardware. The standardised interface allows the feature to be exploited in a hardware independent manner. The interface specification covers many of the more useful features of current window managers and exceeds the capabilities of some. In these cases the machine dependent code simulates the absent features to maintain the interface. On a VT100, for example, the code has to completely simulate a window environment whereas on an Apollo the driver is relatively small.

Application experience

The inSET system has been in commercial use for three years. Both internal projects and applications developed for clients have benefited from the rapid development process possible and the wealth of features provided. Now that the system is available for use outside PA there are already indications that external users are experiencing similar improvements. Two major benefits are recognised:

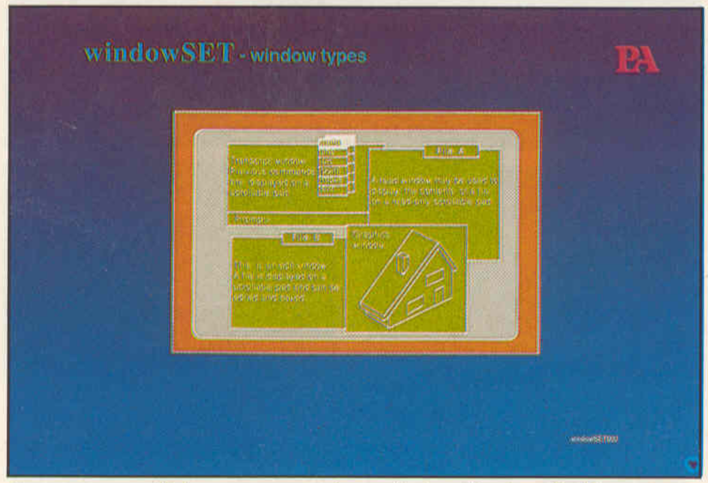
* Improvements in development process efficiency due to the use of a building block, the supporting utilities generating skeleton application code and the maintenance ability of the resulting code.
* Features offered to the users of the resulting application. In many cases each application could not afford to implement all the input mechanisms and features supported by inSET, if written in isolation. Centralising development, maintenance and support of such vital parts of any interactive program allows such a comprehensive range of facilities.



Providing portability across operating systems.

Initially the aim was to achieve a reduction in development times of around 40 per cent for an application development project. In reality the extensions to provide skeleton code generation along with the facilities offered by the system have resulted in savings of around 70 per cent.

In one particular case an estimating package was developed for a client. In this instance the user interface was completely developed through the use of the networks following a requirements specification stage. These networks were processed by SKELGEN to create a prototype environment for the user interface. At this stage a complete forms presentation was also implemented as the users were unused to operating computer systems. Due to the capability within the system to pass data entered in one form across to subsequent



Fully portable window management facilities.

forms, a credible representation of the final interface was then demonstrable. With this prototype running the client was able to view the eventual interface at an early stage in the development process and provide informed input to improve its ease of use.

As development of the application specific action code proceeded, the full use of the macro facilities allowed many of the aspects of the system to be made highly flexible yet require little programming effort. Certain user actions would cause a macro to be processed performing complex tasks simply by combining a number of lower level steps. As the mechanism for estimating changes in the future, it will only be necessary to alter these macros to select the changes. It is also expected that as users of the system become more conversant with the system they can be introduced to more of the features of inSET. Gradually they may begin to write their own macros, find and use the command line entry window and pop-up menus. Also, as inSET develops the application along with all the others incorporating inSET can, at minimal effort, benefit from the new features supported.

The final point concerns portability of application software — a key issue for many companies who recognise that their investment in data, software and user training far outweighs their hardware investment. If the client decides to move to another computer supplier as corporate policy changes or because local circumstances dictate, it is possible to lift and reimplement the application with ease and yet preserve the user interface understood by all the existing users.

ADVANCED GRAPHICS Magazine

Second computer generated film from Apollo

After the considerable success of their "Quest" film in 1985, Michael Sciulli and Jim Arvo formed the Midnight Movie Group and began work on a follow-up. During spare time from their research and development roles at Apollo Inc. Michael and Jim take over the corporate system resources of Apollo, consisting of many hundreds of networked workstations to develop sophisticated computer graphics films.

"Quest" won first place in the State of the Art category of the 1985 Computer Animation Film Festival; the Pan Pacific Computer Art Contest, Australia; and Eurographics Computer Art, Nice. The film was also featured at SIGGRAPH '85 in San Francisco and won acclaim at Computer Graphics '85 at Wembley.

The new film, Fair Play, shows a robot walking through a fairground complete with big wheel, dipper, hall of mirrors and fireworks. All the images and music of the film have been generated on a distributed processing network of Apollo DOMAIN workstations. Using the DOMAIN system, Sciulli, Arvo and other Apollo personnel have fully exploited its graphics capabilities, to support the advanced image generation technique of ray tracing.

Ray tracing is the most sophisticated 3-D rendering of materials. It allows a true surface representation to be achieved, with shading, shadows, highlights, reflections and refractions, texturing, patterns, and structure hierarchy. Combining these attributes allows for an accurate realistic computer-generated image.

CT... CONTACT... CONTACT... CONTA

Alliant Computer Systems Corporation —
(c/o Apollo Computer (UK) Ltd) .... (01) 948 6055
Amazon Computers Ltd, Milton Keynes .... (0908) 664123
Apollo Computer (UK) Ltd, Milton Keynes .. (0908) 366188
Context Corporation —
(c/o Mentor Graphics) .... (0344) 482828
PA Manufacturing Services, Royston .... (0763) 61222
PAFEC Ltd, Nottingham .... (0602) 292291
Selenia-Autotrol Ltd, Birmingham 021-455 7277
Shape Data Ltd, Cambridge .... (0223) 316673

Apollo adds the DSP9000 Mini-supercomputer to its DOMAIN Family

Apollo Computer (UK) Limited launched a powerful new addition to its DOMAIN family at the Café Royale on 30 September. The DSP9000 series of mini-supercomputers will provide workstation users with the capability of processing very large computational problems using parallel processing techniques.

The DSP9000 will be of most value to those working on the computationally intense applications found in engineering and science; including finite element analysis, electronic simulation, image processing and advanced problems in physics, chemistry and mathematics.

The new machine incorporates the industry's most advanced parallel processor and is the fastest compute server available, giving users access to a more powerful computing resource than has been available previously in this price range. The bottom end of the series will be capable, in applications that lend themselves to vector processing, of delivering two times the performance of the DEC VAX 8600 for less than half the price.

An engineer doing finite element analysis will be able to design a product on a workstation, analyse the design on the remote, high performance compute server and access the results without changing the user environment or interface.

The introduction of the DSP9000 represents a strategic move by Apollo to create a new class of DOMAIN products which give significant added value and additional power for compute intensive applications to Apollo users.

answerboard

Q

Dear Sir,

I am about to develop a package which uses graphics as an aid to data capture. I would like the package to be portable and adaptable to a wide variety of display devices. In respect of this I foresee the need for international standards to facilitate the production of off-the-shelf device drivers with a standard interface to the applications package.

My question to Answerboard is therefore: "What is the current status of international standards concerning graphics systems?"

R Molesworth
ICL Network Systems

A

The answer to this question is quite involved, and further reference should be made to the various standards committees for the most up-to-date information. In outline the current scenario is as follows.

Three main standards are presently accepted in relation to computer graphics, these are GKS, IGES 3.0, and NAPLPS. Other standards are under discussion including PHIGS, CGI, PHI-GKS and ISO/TC/184. Of the accepted standards; NAPLPS is concerned with a graphics to videotex interface, IGES 3.0 is a definition for interchange of graphics information at a file level and GKS (the Graphical Kernel System) is the only one involving the writing of graphics software. GKS is primarily a 2-D system at present and uses the concept of a 'workstation'. This is defined as "the logical interface through which the application program controls physical devices". Six types of workstation are defined and can be made to allow for a certain amount of device independence. Work is currently under way to add 3-D functionality to GKS.

The main competitor to GKS in the arena at present is PHIGS (the Programmers Hierarchical Interactive Graphics System). This is a system based on a structured data hierarchy that is close to the modelling that takes place in many CAD/CAM applications. This proposed standard includes 3-D and is incompatible with GKS. This incompatibility has led to an effort in Germany to produce a hierarchical 3-D version of GKS called PHI-GKS. This new initiative may well produce a standard that is much closer to the user needs than currently exists.

There are at present no real standards for screen interaction in the environment of windows, icons and menus; although there is an ANSI working committee. At this end of the spectrum the only standards that exist are for the codes controlling cursors (e.g. ANSI X3.64) and of course for character codes such as ASCII.

At present there is really no internationally accepted standard that will fulfil your requirement for a standard interface between the device drivers and the application package; although work has been carried on in this area for about the last fifteen years.

Further information can be obtained from the various standards bodies such as BSI, ANSI and ISO. Information is also available through organisations like Eurographics and the BCS.

Write to: Answerboard, Advanced Graphics Magazine, 32-36 Little Horton Lane, Bradford BD5 0AL with your questions on graphics and technical computing.

Printed in England by Optichrome Limited, Woking, Surrey