AD-A265 471

RL-TR-92-345, Vol V (of seven)
Final Technical Report
December 1992

SYSTEM ENGINEERING CONCEPT DEMONSTRATION, Technology Assessments

DTIC
ELECTE
JUN 0 9 1993
S A D

Software Productivity Solutions, Inc.

Guy Ivie, Sharon Rohde

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

93-12881

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

93 6 08 079

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-345, Volume V (of seven) has been reviewed and is approved for publication.

APPROVED:

FRANK S. LAMONICA
Project Engineer

FOR THE COMMANDER:

JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1992	Final Feb 90 - Jul 92

4. TITLE AND SUBTITLE
SYSTEM ENGINEERING CONCEPT DEMONSTRATION,
Technology Assessments

5. FUNDING NUMBERS
C - F30602-90-C-0021
PE - 62702F
PR - 5581
TA - 18
WU - 54

6. AUTHOR(S)
Guy Ivie, Sharon Rohde

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Productivity Solutions, Inc.
122 4th Avenue
Indialantic FL 32903-1697

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3CB)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-92-345, Vol V (of seven)

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Frank S. LaMonica/(315) 330-2054

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This final technical report documents the objectives, activities, and results of Air Force contract F30602-90-C-0021, entitled "System Engineering Concept Demonstration." The effort, which was conducted by Software Productivity Solutions, Inc., with McDonnell Douglas Corporation - Douglas Aircraft Company and MTM Engineering Inc. as subcontractors, demonstrated and documented the concept of an advanced computer-based environment which provides automation for Systems Engineering tasks and activities within the Air Force computer-based systems life cycle. The report consists of seven (7) volumes as follows: I) Effort Summary, II) Systems Engineering Needs, III) Process Model, IV) Interface Standards Studies, V) Technology Assessments, VI) Trade Studies, and VII) Security Study.

This Volume (Volume V - Technology Assessments), provides detailed assessments on the viability of technologies related to peripherals, workspace software, framework software, and tools and their potential for use in developing the envisioned systems engineering automation.

14. SUBJECT TERMS
System Engineering, System Life Cycle Tools, System Life Cycle Environment

15. NUMBER OF PAGES
96

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

Preface

This is Volume 5 of the Final Technical Report (FTR) of the Systems Engineering Concept Demonstration, contract F30602-90-C-0021 for the Air Force Rome Laboratory (RL). The document is organized into six sections:

- Integrated Workstations and Software Tools
- Peripherals
- Workspace Software
- Framework Software
- Tools
- Conclusions

The conclusions from the technical assessments are reiterated in the FTR, Volume 1 (Effort Summary), while the detailed assessments appear in this document.

This document was originally written in 1990 and 1991 to report on the "state of the art" in various technology areas: what was in common use, what was being introduced, average costs, and availability. It was updated in July 1992 by the addition of audio and voice technology and pen-based computers. This document is the final version, July 22, of Volume 5.

In some cases, specific tools, manufacturers, and vendors are mentioned. Such mention is in no way a recommendation for or against the firm or the product; the names and other information are provided as examples only.

1. Integrated Workstations and Software Tools

There are a number of solutions to integrating heterogeneous workstations, depending on the ultimate goal. At the low end is the simplest connection to provide a path for ASCII file transfers between nodes. At the high end is a combination of tools, an operating system and a communications protocol(s) that treats the entire network as a single logical machine. This combination providing transparent access to files regardless of physical location and assigning computing tasks to any idle processor.

The International Standards Organization (ISO) Reference Model is a seven-layer communication model that defines standards for linking heterogeneous computers. "In the journey toward Open Systems Interconnect (OSI), communications products are becoming more specific in adherence to the requirements at each layer, and products are fitting more neatly into the layers. In the horizontal approach to networking, products operating at each layer can be upgraded as OSI compatibility comes online without affecting the layers below..."[1] The diagram below illustrates the current industry standards at each layer of the model.[2]

ISORM Layer	Dominant Standard		
Application	NFS	X Window System	PostScript
Presentation	XDR	X Window System	PostScript
Session	UNIX Facilities		
Transport	TCP		
Network	IP		
Link	Ethernet		
Physical	Ethernet		

Figure 1-1. The ISO Reference Model.

---

1 Harrison, Bradford T. "Workgroup Integration Strategies." DEC Professional, Vol. 9, No. 10, October, 1990. 47.

2 Harrison. 46.

This is not to say that other protocols do not exist; rather, these are the most commonly supported by tool vendors. However, Ethernet is being challenged at the link and physical layers by ARCNETplus. Standard ARCNET, operating at 2.5 Mbps, has 20% of the worldwide market. Part of its popularity, despite being 75% slower than Ethernet, is the ease of installation. "ARCNET's topology is extremely flexible and forgiving. With only a few simple interconnect rules, it's exceptionally easy to configure. A network can be wired in a star or bus layout. These simple wiring layouts can easily be combined into a complex, freeform topology which includes coax, twisted pair, and fiber optic products..."[1] Another feature is the ARCNET controller chip, which "enables PCs to join and leave the network without disrupting network activity. This, coupled with ARCNET's flexible topology, makes it easy to add or relocate PCs at will."[2] ARCNET is a token-passing protocol. "In addition to guaranteeing equal access to the network for all PCs, this protocol is deterministic, enhancing reliability in time critical applications. And as more PCs are added to the network, throughput degrades only in small predictable increments."[3] By contrast, Ethernet employs the CSMA/CD protocol; network loading "increases the frequency of collisions, degrading network performance exponentially."[4] Now ARCNET has been enhanced to 20Mpbs and renamed ARCNETplus. In addition to being twice as fast as Ethernet, ARCNETplus has a maximum packet size of 4096 bytes and can support up to 2047 nodes per single network. ARCNET and ARCNETplus can coexist on the same cable, thus protecting an investment in ARCNET.

At the physical and link layers, using an open-system approach may necessitate gateways if the users want to connect an existing LAN (e.g., an AppleTalk network) to the Ethernet-based network. Individual workstations can be connected directly to the Ethernet.

There are also proprietary solutions to linking heterogeneous workstations. Harrison presents a network of Macintoshes, IBM PCs, DECstations, and microVAXes which are linked using VAX/VMS and VAXcluster technology.[5]

---

1 Karlin, Geof. "ARCNETplus Is Twice As Fast As Ethernet." Computer Technology Review, Vol. X, No. 10, August, 1990. 8.

2 Ibid.

3 Ibid.

4 Ibid.

5 Harrison. 52.

The physical connections necessary for integrating workstations present no real problem. The current state-of-the-art provides an adequate solution, at least for the time being. As needs expand (e.g., a larger bandwidth for video transmission), solutions will arise that provide compatibility with existing de facto standards to satisfy the installed user base. In fact, one proposed video solution is an extension to X Windows which requires no change in the hardware connections between workstations.

The real problem in integrating workstations is software; more specifically, the available tools. This focuses on the application layer of the ISO model. Communications between platforms are necessary and important, but we also need communications between the tools operating on those platforms.

Software tools for systems engineering fall into several categories, including Development, Maintenance, Quality Assurance, and Project Management. Hitt, et al., identified more than 1,000 existing tools across all categories. "The majority of these tools (50%) are electronics design automation tools. Mechanical design and computer aided software engineering (CASE) tools represent another 20% each. The remaining 10% of the tools are project management, documentation, and visualization tools. Reliability and physics-of-failure tools comprise less than 1% of the tools identified."[1]

While these tools fall into categories according to their use, they also fall into categories according to the platforms that host them. Until recently, the bulk of the project management and documentation tools were available only on microcomputers, either the IBM PC and its clones or the Macintosh. More development tools have been available on workstations and minicomputers than on microcomputers. This has been especially true of CASE, CAM and high-end CAD tools. Those barriers are breaking down as tool functionality (and sometimes the tools themselves) migrates in both directions between microcomputers and workstations. This migration is a result of both more powerful microcomputers, which are better able to support workstation tools, and demand from users, who want to see the same functionality in tool types regardless of the platform.

---

1 Hitt, Ellis F., et al. "Systems Engineering Tools for Computer Aided Design of Ultra-Reliable Systems." Report No. R-6274, Columbus, Ohio: Battelle Tactical Technology Center, July, 1990. sponsored by DARPA. v.

As it stands now, the COTS arena provides two primary choices for inter-tool communications* across a network of heterogeneous workstations. The first choice is to use tools that are available on multiple platforms but use a common file format. The second choice is to use single-platform tools that can read and write to multiple file formats. Most tools fall somewhere between these two extremes. A given tool may be available on two or three platforms; it may read and write to its own proprietary format, which is readable on any of the supported platforms. A tool could also read and write to a few foreign formats. Frame Technology's FrameMaker is a good example of such a tool. It is available on platforms ranging from Sun workstations to PCs to Macintoshes and, in a networked situation, can open any FrameMaker file on any platform regardless of which version of the application the user is running. FrameMaker can also import and export data in other file formats, such as Microsoft Word.

Another example of a multiple-platform tool is a recently-announced product from Keyword Office Technologies. This product provides support for all UNIX platforms with its "mail-enabled" document interchange software. Using these utilities, plain text and multimedia documents can be shared among editors used on DOS, Mac, OS/2, UNIX and VMS machines... [The utilities include] support for many word processing and other popular applications programs. Conversion occurs via e-mail. A user mails a document to someone else or back into his own account, specifying the conversion that's to occur.."[1]

Translators are a third option for inter-tool communication. The aforementioned interchange software from Keyword Office Technologies is one such tool. Others are available for Macintoshes and PCs for both text and graphics.

Such tools may be of somewhat limited use in a large heterogeneous workstation network. Translators usually cover only the most "popular" tools and formats. As Hitt, et al., point out, "[Translation] is a very inefficient process since many tools can't work directly with a standard format, which necessitates running both input and output translations."[2] In addition, if a single company has a large installed base of some older or proprietary application, there may be no translator available for it.

---

* Although we refer specifically to standalone tools here, the same applies to integrated toolsets insofar as communication between disparate toolsets is concerned.

1 Harrison. 54.

2 Hitt, et al. 48.

There is also the question of what can be translated and to what extent. Changing a graphics file from object-oriented PICT to bit-mapped PCX, or a text file from WordStar to Microsoft Word, is relatively simple. Even so, there is sometimes a loss in the translation, such as typesetter's quotation marks becoming foreign accent characters. Based on a given methodology, converting the output of one CASE tool based for use by a different tool based on a different methodology could present larger problems. This problem is especially true since no standard for such conversions exists.

"This problem has been recognized by the CAE/CAD industry as well as users of the tools. The result has been efforts such as the CAD Framework Initiative (CFI). CFI has seven technical subcommittees that address different CFI standards. Microelectronics and Computer Technology Corp. (MCC) has announced the creation of the CAD Framework Laboratory (CFL) to support and evaluate the standards proposed by CFI. CFL may proposed selected framework interfaces for consideration as candidate standards. CFL also develops a Verification/Validation (V/V) software system for each standard adopted by CFI."[1]

Cadre Technologies has proposed a CASE Data Interchange Format (CDIF) as a standard for files created and used by CASE tools. As of July, 1990, CDIF "ha[d] not been adopted by any recognized group such as IEEE, ACM, or SAE."[2]

SPS addressed the general aspects of tool integration via some framework in the NADC-sponsored study "Environment Frameworks: A Technology Assessment." The report discusses four different forms of integration that a framework can use:

- information integrations – a framework's ability to support data sharing and the meaning of data
- control integration – a framework's ability to manage tool execution
- user interface integration – a framework's ability to provide a common user interface across tools
- method integration – a framework's ability to control the usage of tools to conform to a particular development method.

Section Four of the final technical report addresses the standards involved in using these forms of integration. The extent to which a given tool or set of tools

---

1 Ibid.

2 Hitt, et al. 55.

supports these standards determines the success between the communications and the heterogeneous tools. This success is measured using the networked workstations, and any existing environment/framework, as the medium of communication.

Figure 1-2 depicts the architecture that Hitt, et al., suggest for integrating the various types of tools, and the place that a unified database has in the architecture.

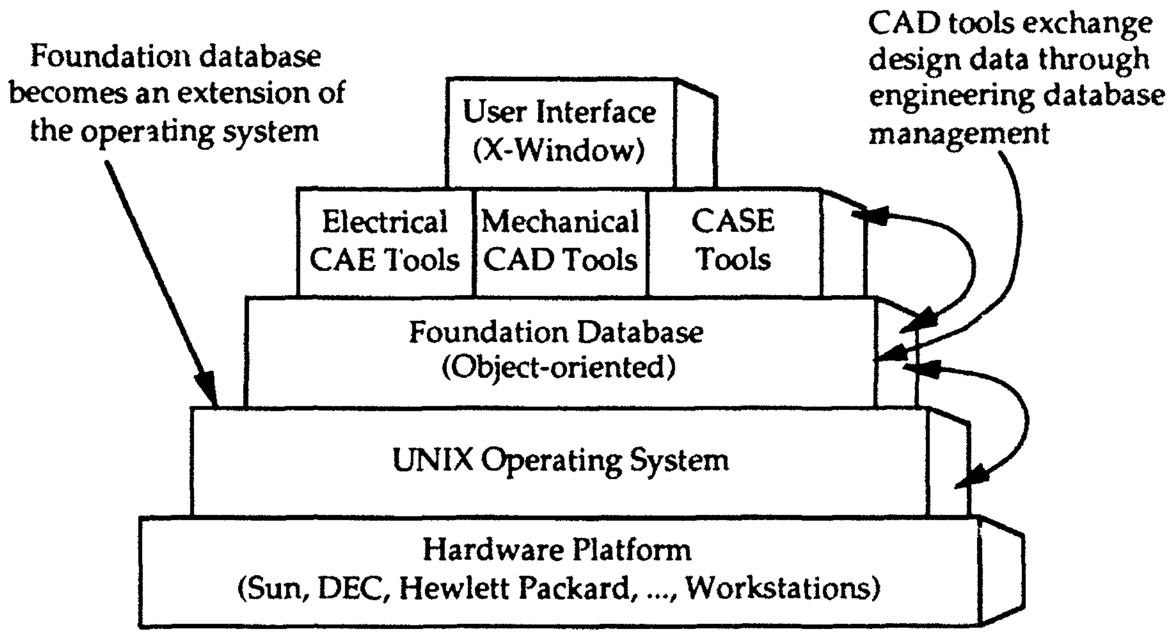

Figure 1-2. Architecture of Integrated Toolset for Design Automation.[1]

Currently available tools address the four integration areas to varying degrees. SPS evaluated three environments (Atherton Software Backplane, ESPRIT Portable Common Tool Environment, and SLCSE) and two framework components (Apollo Domain Software Engineering Environment and Sun Network Software Environment) as part of the NADC study. Although somewhat dated, the data obtained during that assessment still provides an interesting relative comparison. If desired, the assessment method could be applied to other current systems.

---

1 Hitt, et al. 23.

2. Peripherals

2.1. Storage Devices

In the storage arena, the competition is between optical and magnetic media. The ammunition is capacity and access time.

2.1.1. Optical Storage

There are three types of optical storage media available commercially: Compact Disc Read-Only Memory (CD-ROM), Write-Once, Read-Many (WORM), and Erasable Optical (EO). The three available storage media have little in common except that they can hold more than the average Winchester and are all read by laser.

2.1.1.1. CD-ROM

CD-ROM is a spin-off from audio CD technology. To the eye, there is little difference between the two other than labeling (this was mandated by Sony and Phillips in what is known as the Yellow Book, the standard for a CD's physical layout). As the name implies, and like its predecessor, CD-ROM is read-only. With its 550MB capacity, CD-ROM is ideal for distributing large databases that change very little over time. Examples of such databases are encyclopedias, bibliographic information, zip code directories or geological survey data (the latter has found a use in some prototype mobile navigation systems). The Grolier encyclopedia was one of the first publications moved to CD-ROM; it took up less than a quarter of the available space. Some companies (including Sun and Digital) also distribute their software and/or documentation on CD-ROM, thus saving reams of paper and shipping charges. Other firms offer clip art, photo libraries, and font libraries on CD-ROM, all of which can run to many megabytes per file. Drives and software drivers for PCs and Macintoshes have been available for a number of years. However, the workstation and minicomputer markets are opening up as well, with the offerings from Sun and Digital.

For some users, the lack of write capability is a major drawback to CD-ROM. However this usually isn't a problem because of slowly changing or unchanging reference materials. Some publishers do provide updates to their CD-ROM publications in the form of monthly or quarterly floppy disks and quarterly or annual replacement CD-ROM discs.

Access time is a possible problem, one which is shared by the other two types of optical drives. Winchester drives are available with access times of under 30ms. For a CD-ROM, access time can range from 400ms to 1 second. This is due to the fact that CD-ROM discs use the constant linear velocity (CLV) rotational mode. "Although the CLV format is ideal for storing large amounts of data, it is not ideal for retrieving data. For one thing, the long spiral track makes it difficult to find individual sectors. In addition, each head movement to find a particular sector must be accompanied by the mechanical process of speeding up or slowing down the disc..."[1] The data transfer rate is also slower: 625 KB/second for a small Winchester vs. 150 KB/second for CD-ROM. Given the single-user, interactive, reference nature of most CD-ROMs, this isn't necessarily critical.

In its early days, the lack of a standard storage format hampered the CD-ROM. However, in June of 1986 the High Sierra Group, composed of several companies, involved in optical storage technology, submitted a standard for the volume table of contents and directory structure of CD-ROM media to the National Information Standards Organization (NISO) and the European Computer Manufacturers Association ECMA).[2] The standard, which allows "all CD-ROM applications to read data from a CD-ROM system independent of the particular operating system or hardware used"[3] is in use today.

Production costs for CD-ROM can run as little $1500 per title for mastering, with subsequent copies going for a few dollars apiece.[4] Purchase costs depend on the contents of the discs, of course, but there are CD-ROM discs available for as little as $60. Drive prices are driven by the type of computer to which they will be attached. In the PC/Macintosh market, a basic data-only drive goes for as little as $400. Drives to be used for data or audio run higher. At least one company is offering a six-disc CD-ROM changer, based on its audio CD changer; the asking price is under $1500. In the workstation market, CD-ROM drives, like software, run quite a bit higher. Digital offers drives for the microVAX and the VAXstation; prices range from $1,614 to $2,812. Sun's drives are slightly less expensive, running roughly $1,000.

---

1 Einberger, John. "CD ROM Characteristics." *CD ROM, v. 2: Optical Publishing,* S. Ropiequet, et al., eds., Redmond, Wash.: Microsoft Press, 1987. 36.

2 Staff. "High Sierra Group Formulates CD-ROM Standards Recommendation." *Video Computing,* July/August, 1986. 18.

3 Ibid.

4 Eades, Brent. "Data Storage Concerns Network Manager." *Computer Technology Review,* Vol. X, No. 10, August, 1990. 12.

2.1.1.2. WORM

The first form of read-write optical media was the WORM drive. WORM cartridges are available in approximately 600MB or 1GB sizes, and offer the possibility of portability between drives from the same manufacturer. Graphics, medical and geographical imaging, and engineering and architectural applications can all take advantage of the capacity.

After their storage capacity, the most-cited advantage of WORM drives is the permanent audit trail left by files that cannot be erased. Audit trails are especially important in financial applications. "According to James Porter, president of...Disk/Trend Inc.: 'Write-once optical systems will do well in specialized applications like financial, insurance, and geophysical exploration, where you need a trail of data and you don't want to erase any records.'"[1] The same could hold true for large-scale software and hardware development projects, where versioning is important.

WORM drives also offer advantages for archival storage. While standard nine-track magnetic tape has a life expectancy of less than 10 years, WORM discs are rated for up to 30 years. In addition, data on tape is accessed sequentially; optical discs can be accessed randomly, skipping between sectors easily. Tape back-up is usually a manual procedure, but at least one company offers an optical system that performs back-up automatically, transferring files to optical storage if they haven't been accessed in a certain amount of time. The Epoch-1 Infinite Storage Server is aimed at the workstation market and supports NFS.[2]

Once again, access time can be an issue: 100-200 ms as opposed to a Winchester's 14-48 ms. Throughput is about the same as current Winchesters, between .5MB and 1MB per second. However, improvements are appearing in access time throughput: Maxtor claims a seek time of 35 ms for its new Tahiti line of optical drives, with a throughput of 10MB per second for both read and write operations,[3] and Maximum is offering a 1GB capacity 5.25-inch WORM system, with a 28 ms seek time.[4]

---

1 Radoff, David. "New Choices in Storage." *UNIXWorld*, Vol. VI, No. 4, April 1989. 72.

2 Radoff. 72.

3 Radoff. 71.

4 advertisement. *Computer Technology Review*, Vol. X, No. 10, August, 1990. 12.

Compatibility is also an issue. Not being constrained by the Yellow Book, various physical and logical formats using different rotational modes have proliferated among WORM drives. In general, cartridges written on one manufacturer's drive are not readable on another manufacturer's drive.

Cartridges run from $150 to $250, depending on the vendor. NeXT Computers seems to subsidizing optical drives, however. The company resells cartridges for their Canon drive at $50 each, less than Canon charges NeXT. Like CD-ROM drives, prices differ between the PC and workstation markets. At least one discount house is offering an 800MB drive for $3,495—for a PC. Digital's RV-20 WORM drive, for the microVAX and the VAXstation, ranges from $39,885 to $42,563. A jukebox containing multiple drives handling up to 64 cartridges ranges from $232,795 to $334,698.

2.1.1.3. EO

When initially released, EO drives and cartridges offered roughly the same storage capacity as their WORM counterparts, with a difference: data could be erased using a combination of magnetics and optics. A laser pulse and an applied magnetic field working in tandem allowed erase and write/rewrite capabilities. The first commercial shipments of these drives came in 1988.[1] There has also been considerable research into phase-change media and dye-polymer media. Matsushita released a commercial phase-change system in Japan in 1989. The first drives were released in the U.S. in 1990, and permit users to interchange write-once and erasable media in the same drive.[2]

At the time of their release, EO cartridges offered up to 650 MB of storage on a 5.25-inch disc for approximately $250 per cartridge (approximately $.38 per MB). Drives ran in the $5,000 to $6,000 range. Access time was anywhere from two to six times slower than a magnetic hard drive, but the high rotational speed (up to 3,600 prm) kept throughput roughly equal. Part of the reason for slow access time with magneto-optic drives is the need for two passes, one to erase and one to write. Additionally, the power of current lasers limits the write rate.[3]

---

1 Radoff. 70.

2 United States. Department of Commerce/International Trade Administration. *1991 U.S. Industrial Outlook.* GPO, 1991. 28-7 – 28-12.

3 Hoagland, A.S. "Storage [R]evolution." UNIX Review, Vol. 8, No. 10, October, 1990. 52.

Increases in access time and decreases in size and price have been realized. As of this writing, Pinnacle Micro is posting a four-month backlog of orders for its 3.5-inch EO drive, which has a 28 ms seek time. The $129 cartridge can hold 128 MB of data and has a shelf life of ten years. Maxtor's Tahiti EO drive was released in October of 1989 at $2,500 in OEM (Original Equipment Manufacturer) quantities. Cartridges for the Tahiti were priced at $175, also in OEM quantities. Retail price for 5.25-inch cartridges are still holding in the $200 to $250 range. "Increased sales of 3.5-inch optical drives are anticipated when IBM introduces its version of the device to the marketplace in late 1990... Several critical technologies including dye-based media...should migrate from R&D status to practical applications in the erasable optical disk drive industry. This should improve the cost/performance of erasable drives, making them serious contenders for displacing high capacity magnetic drives..."[1]

2.1.2. Magnetic Disk Storage

Despite this prediction, however, magnetic disk drives are far from obsolete. The technology has not been standing still while optical blossomed. Platter size has decreased and capacities have increased. "The first commercial disk drive (IBM Ramac, 1956) used 50, 24-inch diameter disks, which provided a non-volatile direct-overwrite store with a capacity of 5 megabytes and an access time of almost one second. Today the volume leader at the low end is 3.5-inches in diameter. Some models with a height of one inch can store up to 210 megabytes with an access time of less than 20 milliseconds..."[2] In fact, Seagate is offering 3.5-inch drives with a 426 MB capacity and 5.25-inch drives with a 1.2 GB capacity. Notebook computers are sporting 2.5-inch drives with a 20 MB (soon to be 40 MB) capacity and a 28 ms seek time. Costs have dropped between 1989 and 1990 as well. "[T]he factory cost per megabyte dropped...from $3.96 to $3.11 for 5.25-inch disks, and from $5.86 to $4.65 for 3.5-inch disk form factors... [W]orldwide sales of 3.5-inch drives...increased nearly tenfold, from 40,000 in 1989 to 373,000 units in 1990.. Sales of 5.25-inch drives...increased 203 percent, from 158,000 to 480,000 units."[3]

---

1 United States. Department of Commerce/International Trade Administration. *1991 U.S. Industrial Outlook*. GPO, 1991. 28-7 – 28-13.

2 Hoagland. 50.

3 United States. Department of Commerce/International Trade Administration. *1991 U.S. Industrial Outlook*. GPO, 1991. 28-7 – 28-13.

Removable cartridge drives are also popular, especially in the PC and Macintosh markets. These drives provide better security for sensitive data: the cartridge can be stored in a safe when not in use. One of the first units marketed was the Bernoulli Box, so named for the cushion of air (the Bernoulli effect) that supports the special high-density floppy inside its sealed cartridge. PLI and Syquest both market removable cartridge hard drives for the Macintosh, at prices under $800 for the drive and under $100 for a 44 MB cartridge. Syquest has announced the spring 1991 release of a new drive and cartridge with double the capacity.

2.1.3. Magnetic Tape Storage

Various forms of magnetic tape are also in common use, although for back-up and archival storage rather than on-line storage. In addition to the traditional nine-track reel tape, there are various cartridges in use for back-up and distribution: half-inch cartridges (similar to VHS video cassettes), quarter-inch cartridges, 8mm cartridges (based on Sony's 8mm videotape cartridges) and 4mm cartridges (based on digital audio tape—DAT—technology).

"The 3480 IBM tape drive uses 1/2-inch tape, an 18-track head, and due to the parallel transfer of data, operates at 3 MB per second. The IBM 1/2-inch cartridge has become a de facto standard."[1] The cartridge stores 200 MB at 32 tpi. The simplest of the 1/4-inch cartridges "uses a single magnetic head for data transfer and steps the head across the tape as it reverses in what is called *serpentine recording*... [T]he technology is being aggressively pushed and a 1.3 GB cartridge has been announced, a dramatic increase in capacity beyond previous offerings."[2] A multi-track head assembly could allow higher data transfer rates.

Both 8mm and 4mm tape are used in rotary-head, or helical-scan, devices. "These helical-scan devices offer very high track density (900 to almost 2000 tpi) in addition to high linear density... This head/tape configuration favors high track density (and hence high capacity), but uses a single active-head channel. The high relative speed of the spinning head assembly gives respectable data rates, although they are far less than that provided by parallel-track longitudinal recording."[3]

---

1 Hoagland. 50.

2 Ibid.

3 Ibid.

A recent discount-house advertisement listed 450 MB, 3480 drives for $4,295; 350/520 MB, quarter-inch drives for $1,495; 2.3 GB, 8mm drives for $3,695; and1.2 GB, 4mm drives for $3,195, all aimed at PC users.

2.2. Input/Output Devices

2.2.1. Input devices

2.2.1.1. Audio and Voice

Audio plays an important role in multimedia applications. When a service representative adds a voice note to a credit record, when executives hold a video conference, when travelers listen to voice mail or have their email read to them over the phone, or when new employees complete training modules, they all use desktop audio.

Audio can be used in many applications, including voice annotation, voice conferencing, voice mail, training and presentations, text-to-speech, and speech recognition. The following paragraphs describe applications in voice annotation, voice conferencing, voice mail, training and presentations, text-to-speech, and speech recognition.

A voice annotation application enables audio comments to be added to documents, data base records, and so on. These applications can be built using fairly simple record and playback capabilities. For example, someone records a voice comment and attaches it to a certain spot in a document. The document then displays an indicator, showing that an audio note is attached. While viewing the document, the reader can select the indicator to play back the audio recording. Voice annotation applications may also provide some simple audio editing capabilities for message creation.

Voice conferences enable people to speak to each other in real-time over the network. Voice conferencing is an alternative to video conferencing since both parties do not always have to see each other. Often, voice conferencing would be used in a collaborative environment, where two people can sit at their workstations, look at the same document at the same time, and make verbal comments about it.

Voice mail provides a way to send and receive voice-quality audio recordings in a multimedia email message. Audio messages can be recorded, sent as attachments to a multimedia email message, and played back by the recipient. With the integration of telephony into the desktop, messages could be recorded automatically by a telephone-answering application, and forwarded as email to the telephone owner.

As of June 1992, Command Corporation Inc., supplies a voice-command system for Sun SPARCstations. The voice-command system is used to control Sun's OpenWindows user interface or any application that runs on a SPARCstation. It enhances productivity when used to replace mouse commands by letting users keep their fingers on the keyboard.[1]

In OpenWindows, which uses the X Windows windowing system, the voice command system can quickly switch between applications in separate windows. By using a voice command to launch or bring an application forward, users can stack full-screen windows on top of one another and switch between them without having to move or resize them first.

Voice commands can also be used to activate hidden desktop accessories, such as the calendar or calculator. Audio can be very effective in training applications. It can provide a soundtrack for video segments, or for illustrations of any form. It can also provide help and feedback to the student with a more personal feel, and without interrupting the student's focus. Audio can also provide richer, more interesting, and more effective presentations. Audio in combination with other multimedia technologies enables authors to create presentations of a quality that meet the expectations of today's consumers.

Text-to-speech technology enables information stored as text to be converted to speech - effectively to be read aloud. Applications can use text-to-speech to provide verbal help, or to read your calendar, address file information, email message or other information. When integrated with telephony capabilities, text-to-speech technology can provide remote access to information. For example, you could telephone your computer and have it read your mail or appointments. This technology can also provide spoken desktop messages, such as a reminder of a pending appointment, without imposing a pop-up window into the middle of your current work. These desktop alerts could include announcing an incoming email message or, with telephone integration, an incoming phone call. They can also provide spoken status messages or warnings from the system.

AT&T Bell Laboratories and Telefonica Investigacion y Desarollo are developing a prototype Voice English/Spanish/English Translator (VEST) that performs subject-domain-limited spoken-language translation. VEST combines speech recognition of naturally spoken language and text-to-speech synthesis of translated sentences in a second. The powerful host processor needed by VEST is

---

1 Corcoran, Cate. "Command Readies Voice System for Sparcstation." *News/Software* June 8, 1992 p. 31.

a Sun Microsystems SPARC-based workstation for sentence analysis, translation and synthesis.[1]

Speech recognition enables you to speak to your computer through a microphone or telephone. The workstation translates the voice input into text that the system can understand. This technology enables the use of voice as an additional input channel to supplement the keyboard and mouse. For example, radiologists save valuable time by dictating their x-ray reports into the computer for immediate viewing. They can be edited by voice or by keyboard. You could also give commands to open and close windows or start applications without moving away from your current work. With telephony integration, speech recognition would enable you to give commands to your workstation verbally over the telephone, for example, "Read the headers of any mail messages from William Tell." Speech recognition technology in the near future will be limited in the size of its vocabulary, and will typically require you to train your system to your own voice. Eventually, these restrictions should disappear.

"Two decades of R&D have established the importance of the following dimensions in understanding the properties of a given speech-recognition device:

- Speaker dependence versus independence: A speaker-dependent system is trained to recognize only a single voice. A speaker-independent system can recognize anyone's speech, but with less accuracy.
- Discrete words versus continuous speech: A discrete-word system (also called an isolated-word system) requires pauses between words. Continuous-speech recognition lets you speak in a natural manner, but it is more complex and error-prone.
- Vocabulary size and grammar complexity: A system vocabulary defines the set of recognizable words, and the grammar defines the types of sentences (i.e., word sequences) allowed or preferred. Small vocabularies and restrictive grammars are easier for speech recognition, but systems with large vocabularies and loose grammars are more useful.
- Speech recognition versus speech understanding: A speech-recognition device produces a sequence of words, while a speech-understanding system tries to interpret the speaker's intention."[2]

---

1 Bindra, Ashok. "English/Spanish Translator Works in Real Time." *Electronic Engineering* Times April 27, 1992.

2 Lee, Kai-Fu, et al. "The Spoken Word." *Byte*, Vol. 15, No. 7, July, 1990. p. 225-226.

One of the most common uses of voice input converts a single spoken word into a keyboard command or macro. A single word could translate into several hundred keystrokes. This lets users add voice input to most COTS applications. Other uses for voice control or voice input include aids to the physically challenged, voice-dialing for mobile phones, and controlling robotics.

Commercially, a number of voice interfaces are available for any PC-compatible, in a wide range of capabilities and prices—from $150 to more than $26,000. At least one vendor provides a system for the Sun SPARCstation—priced at $34,000—and another provides a system for the Macintosh—this one runs $1,300. Only one card required no system memory; the others need between 64 KB and 10 MB. Some also require up to a 140 MB hard disk. Maximum active vocabularies range from 13 words to 40,000 words.

The middle-of-the-road devices tend to fall into two categories: speaker-dependent, continuous-speech systems or speaker-independent, discrete-word systems. The choice of system depends on the application. There are also some speaker-dependent, discrete-word systems on the market, often with extra features such as a telephone interface with Touch-Tone recognition, voice storage and playback, and text-to-speech translation. "While individual products tend to vary, we expect that this group will provide the highest accuracy recognition, since its approaches are more restrictive and better researched."[1]

The two most important elements of a microcomputer-based voice-processing system are the voice card and the development software package. When purchasing a voice card, users must be aware the microcomputer's bus architecture, processor type and speed. Operating systems that support multitasking such as Unix are better suited to voice processing.[2]

Verbex Voice Systems Inc., Emerson and Stern Associates and Sun Microsystems each recently introduced products for speech recognition and/or synthesis. Many of the offerings are targeted at multimedia application, but potential users should be aware of the limits of the current technology. Sun Microsystems for example is shipping microphones with all SPARCstations desktop workstations.[3]

---

1 Lee, et al. sidebar to "The Spoken Word." p. 228.

2 Robins, Marc. "Roll your own: how to choose hardware and software when building a voice-processing system." Teleconnect. Vol. 10 April 1992.

3 Wolfe, Alexander. "Talking up audio boards: many offerings aimed at multimedia market." Electronic Engineering Times. April 27, 1992.

Research in speech-understanding continues in a number of research laboratories. "For over 10 years, researchers at IBM have been working on a natural-language dictation system that would let you say anything you want."[1] Although the system has attained better than 95 percent accuracy on a 20,000-word discrete-word task, and better than 90 percent with a 5,000-word continuous-speech task, it requires users to train it, speaking 100 words for the discrete-word tasks, and 2,000 words for the continuous-speech tasks. AT&T Bell Laboratories researchers are working on recognizing continuously spoken digits over the telephone. "In a field trial on real speech and real credit card numbers, a Bell system correctly recognized over 98 percent of the number strings—that's about 99.9 percent digit accuracy. This system is the best of its kind..."[2] Researchers at Carnegie Mellon University have developed Sphinx, "the first large-vocabulary speaker-independent continuous-speech-recognition system. Sphinx can achieve a word accuracy of 96 percent on the 1,000-word Naval Resource Management benchmark task."[3] This effort, and others, is sponsored by the DoD. Some of these efforts involve speech understanding as well as recognition.

Voice input technology found 1991 as the year to make rapid development. Articulate Systems, based in Cambridge, MA, commercialized the Voice Navigator II, and won the MacUser Editor's Choice Award for the most significant hardware/software product for the Macintosh for the year. Ivan Mimica, founder and CEO of Articulate Systems, Inc. states "The next logical standard is voice interface."

Voice Navigator and Voice Record features recording the users voice for messages and annotations as well as voice commands for a collection of popular Macintosh software. As many as 10 sets of voice trainings can be stored on disk in individual voice files, and the user's personal voice file is selected by the Macintosh control panel. The voice recognition technology recognizes any voice, any accent, and any language using a discrete utterance recognizer with real-time recognition performance. Voice commands can launch and switch applications, cut and paste between applications, and access windows and desk accessories. Basic voice commands are included to minimize setup time and additional useful voice commands can be added to customize the predefined language files.

---

1 Lee, et al. 231.

2 Lee, et al. 232.

3 Ibid.

Voice Record provides voice annotation to clarify comments to a spreadsheet, memo or report, and voice messaging requires no typing, just "say it" and "send it." A desktop microphone is included with each Voice Navigator II This custom microphone was developed to provide high quality voice control in a sleek desktop design. It operates effectively in most office and home environments where there is moderate ambient noise. A headset microphone is available for those environments that have more than moderate ambient noise, and records only those sounds within the distance of telephone receiver range. Voice Navigator hardware and software sells for less than $700.

In April 1992, John Scully of Apple Corporation showed off a computer program dubbed Casper, which follows spoken commands and replies in pretty good English. In a tradeshow demonstration[1], both Sculley and inventor Kai-fu Lee were able to schedule an appointment and program a VCR using voice commands. The remarkable advancement of this technology is that the speech-recognition program is not tailored to a singe user. Casper can converse with just about anybody. Previously, programs responded to single words; Casper hears whole sentences. Casper isn't confined to supercomputers, it runs on a personal computer, the top-of-the-line Quadra. Casper does not need the level of voice training necessary to operate Articulate's Voice Navigator II.

The power of synthesized speech is coming sooner with software developed by San Diego-based Emerson & Stern Associates Inc. The company's Sound Bytes program, being marketing to systems integrators and value-added resellers, will give a natural-sounding artificial voice to the Macintosh as it synthesizes speech from text[2]. Sound Bytes converts English sentences into intelligible synthesized speech through a process called diphone concatenation, which is the linkage of units of speech consisting of two sounds and the transition between them.

Sound Bytes also incorporates knowledge of sentence structure to emulate the intonation and rhythm of natural speech. Other voice synthesizers usually work word by word, such as the recorded woman's voice of telephone directory assistance - and they sound robotic by comparison. Like the phone company, Emerson& Stern also will bring gender to the Macintosh; its artificial voice is female. Products incorporating the software should be available before the end of the year, according to Louise Courtier, sales and marketing director.

---

1 Schwartz, John. "The Revolution." *Newsweek*. April 6, 1992. p. 43-48.

2 Hollis, Robert. "Voice technology beckons." *MacWeek*. 03.16.92. p. 33-34.

A new wave of technology is coming as the fields of computers, consumer electronics and telecommunications blend together. The result is an explosion of new supergadgets and services that could change all our lives. The year of 1992 promises to be a period of rapid development with applications due that will create more natural sounding artificial voices. At the same time, information systems managers say they see growing although still limited uses of voice-control technology emerging from independent software and hardware vendors. Yet for information systems, it remains unclear whether voice technology will become a relatively common Macintosh tool with broad applications in corporate settings.

Sonitech has introduced two DSP-based interfaces that allow SPARCstations and VME-based systems to run speech processing, telephone, audio and multimedia applications. The Stereo Audio/Telephone Interface Box (SAIB) is a 16-bit stereo A/D and D/A converter with a dynamic range of up to 80 dB. The SAIB includes a built-in input anti-aliasing filter and output low-pass smoothing filter. It is software programmable for input and output sections for gain, sampling rate (8 to 48 Khx) and word size (16-bit linear, 8-bit u-law, 8-bit A-law). The product has three options for input-line level, microphone 1/4-inch phone jack and telephone handset- and four options for output-line level, headphone with 1/4-inch stereo phone jack, telephone handset and speakers.[1]

Another example of a multimedia authoring system/application development software is GainMomentum from Gain Technology of Palo Alto California. GainMomentum runs on the SPARCstation and combines text, audio, video, and animation using object technology. It focuses on the corporate market and offers a way for users to link relational databases by posing SQL-based queries. It is written in C++ and comes with an extension language for creating links to SQL text in relational databases, graphics, image, audio, and animation objects.

Although there are several voice-based applications that provide real benefits for business and users, as discussed above, the applications do not exist in large numbers at the moment. Unless information systems managers see real solutions evolving in the form of useful applications, the future of the voice recognition and voice synthesis product for the Unix platforms and the Macintosh will depend on how the market drives the technology to create such applications.

"Most major computer manufacturers recognize that accessible voice input is the next frontier in interface technology. You will see the necessary hardware appear

1 "Sleek SAIB for Sun." *SunExpert Magazine*. July 1992. p. 78.

as standard equipment within five years. You can also expect to see the more advanced research systems become available commercially."[1]

2.2.1.2. Graphics Tablets

Graphics tablets are frequently used by CAD operators/designers. Although many CAD programs can be operated with a menu/window interface and a two-button mouse, a tablet maximizes performance.[2] Graphics tablets are not only used for sketching, drawing, and tracing, however; they are now sometimes considered an alternate pointing device. Small graphics tablets are available for PCs and Macintoshes that allow users to move icons and windows with a fingertip rather than with a mouse. The same tablet may also come with "template" software that lets the tablet become an extra set of application-specific, programmable, function keys. Larger tablets (active areas of 8.5-inch x 11-inch to 11-inch x 17-inch) may provide a similar feature, but need a mouse-like cursor device or a stylus for input; some tablets come with a cordless cursor or stylus. These tablets are aimed more strongly at graphic artists and designers. Positioning accuracy for PC/Mac graphics tablets varies from 200 to 1,000 dots per inch (dpi). Prices range from $200 to $1,000.

2.2.1.3. Pen-Based Computers

A pen computer is characterized by an integrated display and stylus-oriented entry device and does not depend on a traditional keyboard[3]. The operating systems for pen computers must therefore support the translation of handwritten characters into text, the recognition of gestures (stereotyped movements for editing and navigation), and inking (tracking a stylus on the display and turning on pixels under the point as though the user were writing on paper).

To implement the pen metaphor, developers brought the tools even closer than their desktop: into their hands. Why does the pen work so well as a writing tool? Because it's so light and versatile. You hold it in one hand; you can write with it, draw with it, scratch things out, fill things in, do quick calculations, doodle. You can even point and gesture with it. Most important, you can take it anywhere. The pen moves computing off the desk out into the open. Hardware

---

1 Ibid.

2 interview with Randall Rosier, a CAD designer (PC boards and electromechanical hardware) at Harris Corporation.

3 Duncan, Ray. "Power Programming." PC Magazine. January 28, 1992. p. 335-338.

technology has made it possible to pack the contents of an entire desk into a briefcase computer, and the pen makes it possible to work without the desk itself.

According to pen developers, applications should capitalize on the pen's virtues: easy, direct entry of data; point-to-point, cursorless movement across an "electronic" page; the ability to write with only one hand, forsaking the keyboard; the ability to draw on any size screen and to combine drawings with text; and the ability to do it all with gestures and hand movements.

The market for pen computers is wide. Not only are pen computers good tools for jobs that require time out of the office (e.g., meetings, travel) considered as white-collar productivity jobs, but also for workers who must stand or walk during much of their day (e.g., inspectors, sales representatives, police officers, nurses, survey takers, and utility crews).[1]

Developers are taking two approaches to creating the pen interface. First, to protect the vast investment in the thousands of programs that were created for keyboard and mouse, mouse-aware desktops have been modified to become *pen-aware*, even if the result is less than optimum. At the same time, to exploit the power of the pen more fully, developers will build new *pencentric* systems that are designed around the pen[2].

As of 1992, the four leading environments for pen-based computers are the following products:

- PenPoint from GO
- Application Development Environment (MADE) from Momenta
- Pen DOS from Communications Intelligence Corporation's (CIC)
- Windows for Pen Computing (MWPC) from Microsoft

PenPoint is an object-oriented environment that will run on third party, 386-based computers. These computers will be IBM-compatible only insofar as they implement the Phoenix BIOS or a similar one for switching from PenPoint to DOS or other systems.

---

1 Kuhn, Thomas. "Pentop Paradigm." *Pentop*. Volume 1. Number 1. November/December 1991.

2 Miley, Michael. "Pen-Centric and Pen-Aware: Two Ways to Make the Pen Work." *Pentop*. Volume 1. Number 1. November/December 1991.

Momenta's MADE is an object-oriented operating environment that runs on Momenta's 386-based "pentop" computers. These computers are fully IBM-compatible and will also run PenWindows and DOS.

PenDOS is an operating environment for 386-based computers ranging from notebooks to desktop models running with 1MB of RAM. It provides Pen support for standard mouse-aware DOS applications designed specifically for the pen. Any program which is Microsoft 'mouse-aware' can run under PenDOS without modification.

Microsoft Windows for Pen Computing is a set of extensions to the Windows 3.1 graphical user interface that allows existing Windows and mouse-aware DOS applications to use the pen without modification. The extensions are an assortment of special device drivers and dynamic link libraries (DLLs). Pencentric applications designed specifically for the pen can also be supported. Windows applications will run under MWPC without modification.

CIC and Microsoft are pen-aware applications while Go and Momenta were created pencentric from the ground up. Since the latter are start-up companies, they have no investment in mouse-based software. Both maintain that menus, windows, and dialogs, the usual machinery of the mouse interface, inhibit the use of a pen as a pen. The pen-awareness is achieved by overlaying a new layer of pen tools onto the applications' environment to add pen "events."

Two underlying software technologies make pen-aware and pen-centric applications work: a pen interpreter, which translates pen events into mouse events; and a handwriting recognition module, which translates hand-printed characters into digital text that the computer can read. In both cases, electronic "ink", the trail left by the pen, can either be left as it is or translated into some digital equivalent.

These new technologies work reasonably well, but sometimes are awkward. GO is trying to change that by making it more natural to use the pen. GO uses a metaphor that is called the Notebook User Interface: folders are arranged on a page like a table of contents, with sub-chapters (files) that can be hidden or shown, as in some outliner programs. Each program has it own version of what GO call "stationery," with specific capabilities like drawing or word processing. To work on a document, you merely have to open a piece of the proper stationery.

Instead of a notebook, the Momenta computer has a small, expandable panel, like a menu, at the bottom of the screen. This is where files and applications reside. Users can rotate through revolving "shelves" or expand the panel to display shelves that are kept open all the time. The rest of the screen is blank, except for small command icons that you tap for faxing, printing, using a modem, etc.

A good pen-based metaphor would ultimately fail without good handwriting recognition at the heart of it. Currently, *character* recognition has launched into the market, but all four current pen-based computer systems may have problems, at times, recognizing characters when printed quickly. True, real-time handwriting recognition will most likely make its debut in 1993. When it is developed, it can be easily substituted in existing systems.

The pentop interface is still in the making. While each of these developers has made progress developing the pentop metaphor in their own way, they still have a long way to go. Momenta and GO have perhaps gone the furthest, but pencentric programs can be created on any operating system, whether it's DOS, Windows, PenPoint, or MADE. Applications under PenDOS and MWPC achieved the point of pencentricity and are impressive. So the new companies don't have much of a head start, but are to be commended for a good first cut at this revolutionary technology. Pen computers could become the ubiquitous personal computer that the industry has typed for decades, the devices that are really, to borrow the phrase Apple computer founder Steve Jobs, "the computer for the rest of us."[1]

2.2.1.4 Scanners and OCR

Scanners digitize the image of pages of text, photographs, and graphics so that they can be manipulated by graphics programs. Optical character recognition (OCR) software converts the image of text into actual characters that can be manipulated by word processors.

Low-end and mid-range scanners are the best sellers right now. "Many designers and publishers are using these scanners to bring art into document layouts for positioning while sticking to high-resolution scans brought in from expensive color pre-press houses for final output. Many users are finding that the low-cost scanners provide acceptable images for computing and archiving and as the basis for more-creative image manipulations."[2]

---

1 O'Connor, Rory. "Pen Players Boost the Ante." *Pentop*. Volume 1. Number 1. November/December 1991.

2 Taub, Eric. "In-house scanners becoming routine." *MacWeek*, Vol. 5, No. 10, March 12, 1991. 47-48.

Such scanners fall into several categories. There are hand-helds, flatbeds, and overheads. Original image sizes can range from 4 x 6 inches to 4 x 14 inches, to letter-, legal-, and tabloid-size (and overhead units can scan 3-D objects). Depending on the unit, there are modes for black-and-white line art, halftones, shades of gray (ranging from 16 to 255 shades), and color (up to 16.8 million colors in 24-bit mode). Prices range from $150 to more than $5,000.

DEC offers a scanner for the VAXstation called the VAXimage Scanning Subsystem. It is compatible with software such as DECwrite, VAX Document, and DECpage. This is a sheet-fed scanner; it can hold up to 50 sheets and scans at six pages per minute. Resolution is user-selectable and runs in six increments from 75 dpi to 300 dpi. Total cost of the subsystem, including application software license and documentation, is $7,232; this is considerably more than a PC or Macintosh scanner offering similar capabilities.

HSD Microcomputer offers an 8-bit, 1,500 dpi scanner, the Scan X Professional, for Sun's SPARC platforms. The retail price is $2,995, which is more in line with PC/Mac scanner prices.

Most scanners are intended to be used with opaque paper; without a special backlighting attachment, overhead transparencies or photographic slides do not come through very well. Scanners for those media, as well as for X-rays, are now appearing in the mid-price range, starting at roughly $3,200.

It is almost impossible to use a scanner without plenty of memory and hard disk space; scanned images run from the merely large to the overwhelming. Color scans take up much more space than gray-scale, which take up more than black-and-white, and 24-bit scans take up a great deal more space than 4-bit. The size of the scanned area also figures in the size of the file. Multi-megabyte color scans are not uncommon. Many times, users manipulate the scanned image's complexity and depth to reduce the storage requirements. According to Phil Rogan of Genigraphics Corp., a San Francisco-based service bureau, "For slides, there's little apparent difference between 24-bit and 16-bit."[1]

Scanners are also one of the subsystems in document imaging systems. Such systems can be based on workstations or PCs, and use sophisticated software to control the scanner, compress the images, and file the images (as opposed to the *text*) in a database for later retrieval.

---

1 Taub. 48.

Document imaging systems are best used as part of a streamlined package of services, rather than solely for saving storage space and providing fast access to documents. "A successful imaging system provides users with a variety of functions. Users can analyze alternatives, make decisions and execute those decisions within the same system. Imaging must be integrated with other important systems, such as transaction processing or decision support systems... If [a user] makes a decision while looking at a particular image, and then must fill out a form or turn to an entirely different system to execute his decision, the image piece has become more trouble than it's worth."[1] Given that such systems can run into the tens of thousands of dollars per installation, it makes sense to avoid implementation strategies likely to end up in failure.

OCR packages are popular with organizations that regularly hand-key paper-based text and data into computers. Several companies offer OCR software for use with PC- and Mac-based scanners; some bundle their software with the scanners. At least one vendor offers such a combination for less than $600. One vendor is offering packages for both half-page, hand-held scanners and full-page scanners. Since a half-page scanner has a scan-width of approximately 3.9 to 4.1 inches, the software "stitches" two passes together in memory.

The various OCR packages vary in their capabilities, but none are restricted to the special OCR-A and OCR-B typefaces. Some OCR packages recognize no fonts when delivered, but can be "taught" to recognize any font. Others have a limited recognition set, with or without a learning capability. Still others offer a wide range of typeface recognition, including typeset, proportionally-spaced, and kerned characters. Prices range from under $200 to $700 and up for PC- and Mac-based OCR packages, depending on the features.

At least one company is offering OCR for Sun workstations. OCR servant "recognizes a wide variety of fonts and character sizes ranging from 6 to 36 points in all type styles at resolutions ranging from 200 to 400 dots per inch..."[2] The OCR software retails at $995, slightly higher than retail prices on PC/Mac OCR packages.

---

1 Carter, Joe. "Improving Your Image." *Workstation News*, Vol. 2, No. 2, February, 1991. 27.

2 staff. "OCR Tool for Sun Bows." UNIX Today!, January 21, 1991. 41.

2.2.2. Output devices

2.2.2.1. Graphics Terminals

In the workstation and high-end PC market, graphics terminals, as opposed to character-based terminals, have become the norm. This is due at least in part to the proliferation of graphical user interfaces (GUIs) with their overlapping windows and icons. In the CAD workstation arena, graphics displays have always been the norm. CAD workstations generally come with monitor and processor packaged together; it is mainly in the IBM PC clone and Macintosh markets that monitors are priced and sold individually. Low-resolution monitors can be had for a few hundred dollars or less, but are not considered adequate by most for intensive. "Adequate" monitors measuring an average of 14 inches diagonally, with a 640 x 400 pixel resolution, range between $400 and $800. Multisync monitors boost the price, as do the larger "full-page" or "two-page" monitors. A 19-inch, color, two-page monitor can run more than $3,000; it may or may not come with a controller card, depending on the vendor. Most manufacturers sell a variety of monitors, and don't necessarily need to sell a large volume of the higher-price large color monitors; prices seem to be fairly stable for now.

Another driving force in graphics display devices is the upswing in image processing applications. "Image processing has been identified as a key technology in medical, document processing, Geographic Information Systems, electronic publishing, desktop presentations, industrial inspection, machine vision, command, control and communications, and military simulation and training markets."[1] Revenues in the image processing market for 1990 were projected to reach $1.3 billion.[2] By 1994, revenues are expected to reach $12.2 billion.[3] Depending on the application, resolutions can range from 512 x 512 pixels to 32,000 x 32,000 pixels, with intensity resolutions from 1 bit-per-pixel, single-channel black and white to 16 bits-per-pixel, three-channel true color. Of course, the display device isn't the only part of the image processing system. "In the future, advanced hardware components such as image compression/decompression chips and parallel pixel processors, when combined with the high megahertz RISC...CPUs of the workstation, will make standard desktop

---

1 Harmon, Scott. "The Image Makers." *Workstation News*, Vol. 2, No. 2, February, 1991. 22.

2 Ibid.

3 Carter. 28.

machines capable of truly interactive performance... In the interim, workstation vendors are embedding specialized high-end image computing devices with standard workstations unified under a single, transparent application programming interface (API)."[1]

2.2.2.2. Printers

Although there are many impact printers still in use, and many more still being sold, laser printers are more and more becoming the "standard" printer in most businesses. Adobe's PostScript and Hewlett-Packard's format are the de facto standards for imaging on laser printers, and 300 dpi is considered the minimum standard resolution. While HP LaserJet series printers are more commonly found attached to PCs, PostScript-compatible printers are generally found attached to Macintoshes, or to networks with a mixture of platforms that includes Macs. Several firms, including HP, offer cartridges that allow users to send PostScript output to HP laser printers and compatibles. Most HP-compatibles also offer emulations for the major dot-matrix and daisy-wheel printers (e.g., Epson and Diablo). Small, "personal" laser printers can be had for well under $2,000; those aimed at the working world's higher duty cycle range from $3,000 to $6,000 depending on features and discount rate.

Some vendors are offering plain-paper laser printers with higher resolutions. LaserMax Systems offers printers with resolutions of 1,000 x 1,000 dpi, and 1200 x 800 dpi, with print speed of up to 20 pages per minute (ppm). Printware, Inc., offers a 1200 dpi laser printer with 4 MB of RAM and a 20 MB internal hard drive for font storage. The major advantage of higher-resolution laser printers is smoother output (i.e., the image is less "fuzzy" on curves and diagonals) without using expensive photochemical imagesetters.

Methods of color output in the office range from multi-color ribbon impact to inkjet to thermal-wax-transfer. In expensive impact printers are generally sold for personal use, and are rarely used in the office these days. Inkjet printers aimed at the business market range from $1,500 to $3,000. Their resolutions range from 180 dpi to 216 dpi. Most are not PostScript-compatible, requiring the use of software-based PostScript emulators.

There is a wider selection in the thermal-wax-transfer market; many of the printers offer PostScript or one of its clones. All offer 300-dpi resolution and can handle letter-size paper. Depending on the printer, some also handle legal and/or tabloid-size paper. Prices range from $5,000 to $13,000. One possible

---

1 Harmon 23.

problem with these printers is the life of the print head. A service bureau in Portland, Maine, goes through one per year, at a cost of $900 per print head.[1] The owner of the shop also noted that the "four-color dithering process reduces the effective resolution by one-fourth."

Some manufacturers are experimenting with solid-inkjet technology to create a plain-paper color printer. "This technology uses heat to melt crayon-like cylinders of plastic onto the media."[2] Howtek, Inc., markets a printer based on this technology, but it has not been well-accepted due to the thickness of the inks. Spectra, Inc., has developed a process that "provides for a smooth finish, as well as a simpler, more reliable mechanism..."[3] They will be licensing their process to OEMs, with resulting products priced in the $7,000 to $10,000 range.

There are other options for plain-paper color printers, but they are much more expensive. Colorocs Corp. offers a 300-dpi printer based on electrophotographic technology. It has a PostScript clone interpreter, handles paper sizes up to tabloid-size, and sells in the neighborhood of $29,000. Canon USA, Inc., sells a color laser copier/printer. The unit provides near-photographic quality output for $49,000. Canon is developing a PostScript interpreter for it.

2.2.2.3. Imagesetters and film recorders

Although 300 dpi is sufficient for most business use, it falls short for real publishing use. When viewed through a 10x loupe, the edges of individual dots are visible, especially on curves and diagonals. When used as camera-ready output for offset printing, the final product delivered from the printer has a fuzzy appearance. Imagesetting equipment uses the output of the computer to create an image on photographic paper at resolutions starting at 1,270 dpi. Imagesetting equipment is expensive—a low-end Linotronic printer costs roughly $30,000—and it can be messy, as well—photographic paper means photographic chemicals. Most companies don't have the volume of output to justify the cost of the equipment and send their output to service bureaus for imaging.

---

1 Waltz, Mitzi. "Color-printer choices expand as prices fall." *MacWeek*, Vol. 5, No. 7, February 19, 1991. 57.

2 Waltz, Mitzi. sidebar to "Color-printer choices..." *MacWeek*, Vol. 5, No. 7, February 19, 1991. 58.

3 Ibid.

Film recorders are used to record image data directly to film, rather than paper. The result is high-quality, high-resolution color slides and transparencies. Again, cost and chemicals are usually a factor in the decision to purchase or use a service bureau.

2.2.2.4. Plotters

Plotters are standard output equipment in CAD-oriented organizations. At one time, plotters were pen-based devices that used rollers to move the paper along one axis while the pen was moved in the other. Now there are a number of other technologies available for plotting, including electrophotographic, electrostatic, thermal transfer, and liquid ink-jet. Designers often use standard 300-dpi laser printers, as well, for proofing designs before imaging on more expensive systems.

"The advantages of pen plotters are price—$6,000 to $20,000 will get a decent E-size plotter [prices can go under $1,000 for smaller sizes]—and on-site accessibility. There is always a need for check plots, or a hard copy of what's on the screen. Another advantage is the ability to plot multiple media (paper, Mylar, or vellum). The main disadvantages are fine definition and speed of operation (this last is not a major drawback)."[1]

Electrophotographic plotters are toner-based devices, using a light source to charge the selected areas of the paper. Although these devices sometimes use LEDs and liquid crystal shutters, lasers are the most common light source.[2] "The advantages of the laser photo-plotter are the precision and fine definition of the image. The major disadvantage is price, of course. For smaller operations, it is usually more cost-effective to have an outside photo-plotting service [similar to the service bureaus used by DTP operations]. For this service, you'll need Gerber output format, which may require additional software, depending on the CAD program."[3] Lasers can also be used to expose film stock, at resolutions of up to 2,000 dpi. Prices for laser plotters can range from $4,000 to more than $20,000 for high-resolution monochrome; with color it can jump to as much as $120,000.[4]

---

1 Rosier interview.

2 Diehl, Stanford, and Steve Apiki. sidebar to "Plotters in Perspective." *Byte*, Vol. 13, No. 13, December, 1988. 164.

3 Rosier interview.

4 Diehl.

Electrostatic plotters use nibs to charge areas on the paper, which then attract toner. Both monochrome and color versions are available. These plotters are expensive: from $12,000 for B media (11 x 17 inches) to as much as $120,000 for larger formats.[1] They also require special paper, an additional cost.

Thermal transfer plotters use heating elements to melt colored wax and fuse it to the paper. "Resolution typically falls in within the 160- to 300-dpi range, while prices range from $300 to $9,000. The major disadvantage here is the high cost of supplies."[2]

"Reliability problems have thwarted the promise of ink-jet technology. Newer models belie that reputation, but low resolution and slow throughput are still legitimate gripes... Prices for ink-jet printers typically range from $700 to $7,000, although some specialized large-format models can cost up to $75,000. Supply costs are moderate."[3]

2.2.2.5. Videotape

More and more, companies are using 'desktop video' as part of their output. Video is an excellent way to disseminate information in a form nearly everyone can use and understand. By combining computers, video images, and VCRs, very complex subjects can be examined and explained. Common uses are demonstrations of software products, or walk-throughs of architectural designs. Another is professional video production. "[T]here's an excellent chance that the lines and arrows on the weather maps on your TV news were produced with a desktop-video system."[4]

The VCR can be used both as an input and an output device. When used for input, the VCR (or other video source) is under computer control. Special software responds to commands in the presentation, displaying specific segments of the videotape at specific times. One example of such use is in interactive training. The user chooses a selection from a menu; in response, the computer searches the videotape for a specific segment. After locating that spot

---

1 Ibid.

2 Ibid.

3 Ibid.

4 Cook, Rick. "Desktop Video Studio." *Byte*, Vol. 15, No. 2, February, 1990. 229.

on the tape, the computer displays it on its own monitor, or on a separate television screen.

Special hardware makes it possible to have the video image appear on the computer screen, with computer-generated text or graphics appear overlaid on it. A genlock device synchronizes the computer's output with the video signal. You can also send the computer output, with or without the video picture, to a standard television or a VCR. An encoder converts the computer's RGB output to an NTSC (or PAL, the European standard) analog signal for use with those devices. Many vendors offer video processor cards for PCs or Macintoshes that combine the two functions. The cards often accept multiple types of video input, including videotape, videodisc, video camera, or broadcast signal. Some cards allow still-frame capture of video images; others handle full motion.

Putting these productions together requires some sort of editing software, in addition to the applications used to put together text overlays or animation. Editor packages offer varying features, at a varying prices. The editor may be little more than a sequencer, putting animation frames in the right order and sending them out to the screen or the VCR, or it might provide sophisticated features such as wipes and fades between frames, digitally synchronized with stereo sound.

Several film and special-effects studios, most notably Lucasfilms' spin-off, Industrial Light and Magic, use Sun or Silicon Graphics workstations for their video work. Both the equipment and the software are expensive and specialized, and take specialized training to use. By contrast, desktop video is much less expensive and often aimed at the less-than-expert user. Granted, the quality is not the same as that achieved with a studio full of professional equipment, operated by professionals, but it is usually adequate to the purpose. "Desktop video is good enough for many kinds of video in the same way that 300-dot-per-inch laser printing is adequate for a lot of published material."[1]

Equipment costs vary according to the host platform and the quality of the output. For example, encoders for the Macintosh range from $395 for a simple NTSC encoder to $2,995 for a "broadcast quality and flicker-free" encoder that will provide both NTSC and PAL full-color composite or S-Video output.[2]

---

1 Cook. 234.

2 Guglielmo, Connie. "VideoLogic box prints to video." MacWEEK, Vol. 5, No. 10, March 12, 1991. 18.

"Desktop video is still in its infancy. Desktop publishing took off not with the invention of the laser printer, which made the technology possible, but with the release of the Macintosh and the Apple LaserWriter printer, which made it easy. Today, desktop-video technology is possible, but it's not yet easy. It's still waiting for its equivalent of the Mac. Probably the biggest need is to integrate the systems and make them easier to use... Some progress has occurred in that direction, however. There is a strong trend toward standardization in file formats for video software... A number of companies, especially in the Amiga market, are working on video authoring systems to make the whole process easier... The hardware is improving as well. Genlocks and editors are getting better, if not necessarily cheaper. New formats, like Super VHS, offer better picture quality at lower prices. And video equipment manufacturers are adding features to make their products more useful for amateur and desktop-video production."[1]

---

1 Cook. 234.

3. Workspace Software.

3.1. Operating Systems

Workstations have long been the province of scientists and engineers and, since being rewritten in C in 1973, UNIX has been the popular operating system for them. No longer just a scientific and engineering tool, the workstation is moving into the financial, manufacturing, and services sectors, and UNIX is moving along with it. According to the Department of Commerce, the workstation market is expected to grow rapidly, increasing 40 percent in 1990 to nearly $9 billion; in the next five years, it should reach $24 billion. "Electronic publishing, animation, and image processing for mapping and for medical and retail imaging will become the principal graphics applications for these systems."[1] As the workstation market grows, so does the installed base of UNIX users.

There is no question that UNIX is a presence in the marketplace. It is familiar, powerful, and pervasive. There is a version of UNIX for nearly every processor on the market, and a plethora of tools that run under it. However, UNIX is not the result of a focused, controlled development and marketing effort. UNIX was "unlike other operating systems of the day, because it was small, was written in a high-level language, encompassed some new useful ideas, and was not kept secret."[2] Although UNIX was originally developed at Bell Labs, users wrote utilities as they were needed, and those utilities became part of the operating system. Many versions of UNIX proliferated before the first "official" release of Version 7 in 1978. A commercial version, Version III, was released in 1982. There are now more than a dozen versions of UNIX available, many on multiple platforms, and none are completely compatible with all the others. According to a survey performed by *UNIX Today!* in January of 1990, there is a "standard desire for standards..."[3]

Bill Demmer, manager of the development of the IBM System/360 Models 65 and 85 and now vice president of Digital's Mid-range Systems Business Group, agrees that a standard UNIX is a necessity: "...If AT&T would sell off its control

---

1 United States. Department of Commerce, International Trade Association. *U.S. Industrial Outlook.* Washington: GPO, January 1990. 30-5.

2 Thomas, Rebecca, and Rik Farrow. *UNIX Administration Guide for System V.* Prentice Hall: Englewood Cliffs, N.J., 1989. 3.

3 Krill, Paul, et al. "What Do You Want From Unix?" *UNIX Today!*, January 8, 1990. 1.

over UNIX, that move could lead to the one thing I've been hoping would happen since we jumped on the UNIX bandwagon—that we would see a single standard UNIX become available to the industry that would not be under the control of one company."[1] When asked about the value of competing versions in pushing technology forward, Demmer replied, "Two is better than twenty, but certainly not as good as one. If there were one standard, I suspect that many different parties would try to make enhancements to it. And I think the competition would be in developing and promoting those enhancements to be incorporated into the standard..."[2]

There are other complaints about UNIX and its capabilities. Joel Birnbaum, corporate vice president of HP Laboratories and developer of the first RISC computer, listed some specific problems with UNIX. "...It doesn't yet provide fault tolerance, its network-level error recovery is less ambitious than what MVS and VMS provide today, its real-time services are primitive, and it has a limited developer interface. It now does better what it was originally designed to do, but in order for it to be the micro-kernel to which all the value can be added by database, user-interface, or network-management developers, UNIX is going to have to grow."[3]

UNIX was not specifically designed to take advantage of the distributed networks that are common today. Operating systems under development now could prove to be more efficient at putting idle CPUs to use.

Bell Labs, in the persons of Brian Kernighan, Ken Thompson, Dennis Ritchie, and Rob Pike, is working on Plan 9, an experimental research operating system. Plan 9 is "designed for the highly distributed environments that will emerge in the 1990s... Although Plan 9 is an 'experimental' system, Bell researchers [hope] the software could be licensed as source code within one to two years. To users, Plan 9 resembles UNIX, but the system is not UNIX-compatible... AT&T is believed to be considering the operating system as one of the options for its own multiprocessor machines."[4]

---

1 Chandler, David. "Interview: Bill Demmer." 71.

2 Ibid.

3 Chandler, David. "Interview: Joel Birnbaum." *UNIX Review*, Vol. 8, No.1, January, 1990. 57.

4 Faden, Michael. "Plan 9: It Came From Bell Labs." *Unix Today!*, July 23, 1990. 1.

Amoeba is another operating system aimed at multiprocessor machines. "The Amoeba software is based on objects. An object is a piece of data on which well-defined operations can be performed by authorized users, independent of the user's and object's locations... To bridge the gap with existing systems, Amoeba has a UNIX emulation facility consisting of a library of UNIX system call routines that make calls to the various Amoeba server processes."[1] Under development for nearly ten years at the Free University and the Centre for Mathematics and Computer Science in Amsterdam, and partially funded by grants from the Open Software Foundation and DEC, among others, Amoeba is now in its fourth version. It currently runs on VAXs and Motorola's 68020 and 68030 processors. A port to Intel's 80386 is underway.

Mach, an OS for both uni- and multi-processors, is a product of research at Carnegie Mellon University (CMU). Among other features, it provides a network-transparent communication subsystem. "The Mach kernel incorporates compatibility code derived from 4.3BSD UNIX that provides complete binary compatibility. Mach runs on a variety of uni-processor and multiprocessor architectures, including the DEC VAX, Sun 3, IBM RP3, and Encore Multimax. Mach is available and supported as a product by a number of hardware vendors, including Next, Encore, and Omrom. It is also the base technology for the OSF/1 operating system from the Open Software Foundation."[2]

Existing non-UNIX operating systems must be considered, as well. MVS and VMS, among others, are alive and well, and more systems that use them are being sold every year. Microcomputers running MS-DOS and the Macintosh OS often sit next to a workstation on an engineer's desk. Rather than lose their investment, most companies choose to network new and old together. Software tool vendors have begun to port their products across the gap, easing the transition and providing capabilities that were not available previously. Such applications as Lotus 1-2-3, WordPerfect, Wingz, and FoxBASE have been ported to various UNIX platforms. FrameMaker and Interleaf have been ported to the PC and the Macintosh. Ported applications offer file compatibility across platforms, but the porting process can be long and tedious. Any machine- or operating system-specific calls must be changed or eliminated. In addition, the newly-acquired file compatibility applies only to the specific tool, not to all other tools.

---

1 Mullender, Sape J., et al. "Amoeba: A Distributed Operating System for the 1990s." *IEEE Computer*, Vol. 23, No. 5, May 1990. 36.

2 Black, David L. "Scheduling Support for Concurrency and Parallelism in the Mach Operating System." *IEEE Computer*, Vol. 23, No. 5, May 1990. 36.

In a modern distributed computing environment, many operating systems must be able to work together. We need a means of ensuring some form of compatibility among the tools that work under these OSs, especially if heterogeneous tools are to access each others files and data across the network. The IEEE POSIX (Portable Operating System Interface) standard for operating systems may provide one avenue for the necessary compatibility.

The POSIX standard specifies a set of external calls and interfaces for operating systems. Applications developed for a POSIX-compliant operating system, making only POSIX-compliant calls (i.e., no calls to special, local OS or hardware features), can be recompiled on another POSIX-compliant OS and will run unchanged. POSIX-compliance could eliminate the bulk of the work involved in porting applications across several platforms. POSIX is getting a strong push from the federal government; POSIX-compliance is "mandated" in future information systems contracts, according to John Cox in *Digital News*.[1]

John Keller reported in *Military&Aerospace Electronics* that the standard operating system for the Navy's Next Generation Computer Resources program will be based on POSIX. "The Navy action followed by just three months the release of 'Air Force Communications-Computer Systems: 1990 Planning & Architecture Guidance,' an Air Force directive that all but mandated POSIX." [2] Keller also quoted Roger Martin, manager of the software engineering group at the National Institute of Standards and Technology: "'What we see in all the services is a move to make POSIX mandatory,' NIST's Martin says. 'The Army is moving toward POSIX also.'"[3]

However, Donn Terry, chairman of the IEEE POSIX committee, "advises companies supporting the interface standard to hold the champagne. 'Across DoD, it's not clear that POSIX is emerging as a standard,' he says. 'DoD has a mind of its own, and each service has a mind of its own, too.' But he admits, 'It will be hard for DoD to ignore it.'"[4]

---

1 Cox, John. "VMS/Posix edges by UNIX." *Digital News*, June 11, 1990. 68.

2 Keller, John. "POSIX Picks Up the Pace." Military&Aerospace Electronics, Vol. 1, No. 8, August, 1990. 1.

3 Ibid.

4 Keller. 35.

November 1990.[1] Unisys has been awarded a two-year DoD contract to supply 75,000 of its PW2 microcomputer systems, complete with a POSIX-compliant version of UNIX. [2]

However, Terry is right when he advises not opening up the bubbly. Although P1003.1 was published late in 1990, it was only the first of several standards dealing with POSIX. The rest are still in committee. Two of the standards, P1003.2, the POSIX standard for shell and utilities standards, and P1003.9, the FORTRAN bindings, might complete balloting this year. Terry said, "We're in the process of going out to bid to have someone execute the editorial content of a draft for a language-independent version of P1003.1..."[3] The existing P1003.1, which is based on the C language, would be the basis of the new version and was available for comment in April 1991.

There is not as yet any agreement on a 'standard' GUI for POSIX.[4] According to one attendant, "The problem with [the standard GUI committee] is that every meeting you go to, Open Look and Motif proponents will pack the meeting. They are each willing to adopt their own user interface, but without change. They are not willing to compromise and take the best of each. It makes it impossible to agree on anything."[5] There have been other proposals, including a shift away from selecting a particular GUI and instead choosing a common application programming interface (API).[6] [7] Peter Janecek, manager of connectivity strategy for X/Open, summed it up: "As of March 1991, we have been unable to find a consensus position."[8]

---

1 Cox. 1.

2 Staff. Currents column. *UNIX Review*, Vol. 8, No. 1, January 1990. 11.

3 Faden, Michael. "Standards Efforts Taking Shape." *UNIX Today!*, March 18, 1991. 59.

4 Faden. 63.

5 Wagner, Mitch. "POSIX Group Fails to Agree on GUI." UNIX Today!, August 6, 1990.

6 Ibid.

7 Faden. 63.

8 Ibid.

3.2. Repositories

3.2.1. Data Models

There have been three generations of data models: the primitive data models, classic data models, and semantic data models.[1] Current research is focused on developing fourth generation data models which are object-oriented or frame-based. In a primitive data model, objects are represented in records that are grouped in files. Relationships can be represented using directories implemented as indexes or inverted lists. The classic data models are the hierarchical, the network, and the relational data models. The hierarchical data model represents all data in the form of trees, whereas the network data model represents data in networks of records with pointers showing relationships among records. The underlying structure of a relational data model is the tree. These data models are described by Date.[2]

Classic models did not contain enough semantic information about the data; as a result, semantic data models were developed. Semantic models can be classified into the following categories:

- Direct extensions of classic data models
- Mathematical data models
- Irreducible data models
- Static semantic hierarchy models
- Dynamic semantic hierarchy models[3]

Direct extensions of classic data models include the Structural Model[4] which restricts relations in the relational data model to five specific types; the Object-

---

1 Brodie, Michael, "On the Development of Data Models," Chapter 2 of *On Conceptual Modelling*, eds. Michael Brodie, John Mylopoulos, Joachim Schmidt, Springer-Verlag, 1984.

2 Date, C.J., *An Introduction to Database Systems*, Addison-Wesley Publishing Company, 1982.

3 Brodie, 1984.

4 Wiederhold, G., and R. El-Masri, "The Structural Model for Database Design," *Proc. International Conference on the Entity-Relationship Approach to Systems Analysis and Design*, Los Angeles, CA, December 1979.

Role Model[1] which extends the network model by adding the concept of role where one object may play many roles in an application and may have different attributes for each role; and the Entity-Relationship (ER) Model[2] which combine features of the relational and network models. Many extensions to the ER Modeling technique have been developed (see Teorey, et al.,[3] for a summary); one such extension was used to develop the ER model of the Software Life Cycle Support Environment.[4]

The mathematical foundations of the relational model (e.g., set theory and predicate calculus) have been extended by several researchers to formally define and extend the relational data model. Some of these extensions are based on set theory,[5] some on First Order Logic,[6] [7] [8] and some on the universal instance assumption.[9] The mathematical models provide formal notations and definitions for concepts presented in other models.

---

1 Bachman, C.W., "The Role Concept in Data Models," *Proc. 3rd International Conference on Very Large Databases*, Tokyo, Japan, 1977.

2 Chen, Peter, "The Entity-Relationship Model-Toward a Unified View of Data," *ACM Transactions on Database Systems*, Volume 1, Number 1, March 1976.

3 Teorey, Toby J., Dongqing Yang, and James P. Fry, "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model," *ACM Computing Surveys*, Vol. 18, No. 2, June 1986.

4 Grau, J. Kaye, *DoD-STD-2167A Entity-Relationship Schema* for RADC /COEE, Contract #F30602-86-C-0206, Software Productivity Solutions, Inc., January 13, 1989.

5 Childs, D.L., "Extended Set Theory," *Proc. 3rd International Conference on Very Large Databases*, Tokyo, Japan, October 1977.

6 Reiter, Raymond, "Towards a Logical Reconstruction of Relational Database Theory," Chapter 8 of *On Conceptual Modeling*, eds Michael Brodie, John Mylopoulos, Joachim Schmidt, Springer-Verlag, 1984.

7 Jacobs, B.E., "On Database Logic," *Journal of the ACM*, Vol. 29, No. 2, April 1982, pp. 310-332.

8 Gallaire, H., and J. Minker (eds.), *Logic and Data Bases*, Plenum Press, New York, 1978.

9 Ullman, J.D., *Principles of Database Systems*, Computer Science Press, Potomac, Maryland, 1980.

Irreducible data models are based on the representation of information as "atomic" facts rather than as complex groups of facts. Atomic facts cannot be further decomposed; thus the name "irreducible." Examples of this kind of model include the Binary-Relationship model,[1] a restriction of the relational model in which relationships are binary rather than *n*-ary; the Irreducible Relational Model,[2] in which relations cannot be decomposed into two or more relations without loss of data; and functional data models. Functional data models combine the relational data model with functional programming so that relationships are represented as functional mappings between objects. Such a representation is irreducible because each attribute is related to its associated object by a function. DAPLEX,[3] FQL,[4] [5] and FDM[6] are three functional models.

---

1 Bracchi, G., P. Paolini, and G. Pelgatti, "Binary Logical Associations in Data Modelling," in J.M. Nijssen (ed.), *Modelling in Database Management Systems*, Proc. IFIP TC2 Conference, Freudenstadt, North-Holland, Amsterdam, The Netherlands, 1976.

2 Hall, P., J. Owlett, and S.J.P. Todd, "Relations and Entities," in J.M. Nifssen (ed.), *Model,* 1979.

3 Shipman, D.W., "The Functional Data Model and the Data Language DAPLEX," *ACM Transactions on Database Systems,* Vol. 6, No. 1, March 1981.

4 Buneman, O.P., and R.E. Frankel, "FQL—A Functional Query Language," *Proc. 1979 ACM SIGMOD International Conference on the Management of Data,* Boston, MA, May 1979.

5 Buneman, Peter, and Rishiyur Nikhil, "The Functional Data Model and its Uses for Interaction with Databases," Chapter 13 of *On Conceptual Modeling,* eds Michael Brodie, John Mylopoulos, Joachim Schmidt, Springer-Verlag, 1984.

6 Housel, B.C., V. Waddle, and S.B. Yao, "The Functional Dependency Model for Logical Database Design," *Proc. 5th International Conference on Very Large Databases,* Rio de Janeiro, Brazil, October 1979.

Static Semantic Hierarchy models integrate relational data model concepts with four relationships from AI based semantic nets: classification, aggregation, generalization, and association. These relationships are used to support data abstraction in which specific details are suppressed and those pertinent to the problem or view of information at hand are emphasized. Static semantic hierarchy models include SHM,[1] [2] ADD,[3] LGDM, [4] [5] RM/T,[6] SAM,[7] and SDM.[8]

---

1 Smith, J.M., and D.C.P. Smith, "Database Abstractions: Aggregation and Generalization," *Communications of the ACM*, Vol. 20, June 1977.

2 Smith, J.M., and D.C.P. Smith, "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems*, Vol.2, No.2, June 1977.

3 Roussopoulos, N., "ADD: Algebraic Data Definition," *Proc. 6th Texas Conference Record on Computing Systems*, Austin, Texas, November 1977.

4 Brodie, M.L., "Specification and Verification of Database Semantic Integrity," Ph.D thesis (Computer Systems Research Group Technical Report No.91), Univ. of Toronto, April 1978.

5 Brodie, M.L., "Axiomatic Definitions for Data Model Semantics," *Information Systems*, Vol. 7, No. 2, 1982.

6 Codd, E.F., "Extending the Database Relational Model to Capture More Meaning," *ACM Transactions of Database Systems*, Vol. 4, No. 4, December 1979, pp. 397-434; IBM Research Report RJ2599, San Jose, Calif., August 1979.

7 Su, S.Y.W., and D.H. Lo, "A Semantic Association Model for Conceptual Database Design," *Proc. International Conference on the Entity-Relationship Approach to Systems Analysis and Design*, Los Angles, Calif., December 1979.

8 Hammer, M., and D. McLeod, "Database Description with SDM: A Semantic Database Model," *ACM Transactions on Database Systems*, Vol. 6, No. 3, September 1981.

Dynamic semantic hierarchy models are extensions of static semantic hierarchy models which integrate concepts for modeling dynamics with those for modeling statics. These models integrate a number of programming language concepts with database concepts. Examples include SHM+,[1 2 3] TAXIS,[4 5] and the Event Model.[6]

Object-oriented and frame-based data models represent the fourth generation. Object-oriented concepts provide useful abstractions for relational database design. These abstractions are intuitive, expressive, and extensible. Further levels of abstraction include details such as defining attributes and tables, and specifying rules to guarantee the integrity of the tables. Databases based on the object-oriented model lend themselves to better evaluation of performance, integrity, understandability, and extensibility. These databases have a better theoretical foundation than network and hierarchical systems and are the focus of intense commercial activity.[7] The frame-based or knowledge-based data model is based on an abstract data type of knowledge representation languages. A formal definition of a frame data model is presented in terms of a denotation semantics approach . For example, the notion of a frame is defined by a mapping

FRAMES - Fnames --> SLOTS

---

1 Brodie, M.L., "On Modelling Behavioural Semantics of Data," *Proc. 7th International Conference on Very Large Databases,* Cannes, Frances, September 1981.

2 Ridjanovic, D., and M.L. Brodie, "Semantic Data Model-Driven Design, Specification and Verification of Interactive Database Transactions," Computer Corporation of America, Cambridge, Mass., April 1982.

3 Brodie, 1984.

4 [MBW80]

5 Mylopoulos, J., "A Overview of Knowledge Representation," *Proc. Workshop on Data Abstraction, Databases, and Conceptual Modelling,* M.L. Brodie and S.N. Zilles, eds. *SIGART Newsletter,* No. 74, January 1981; *SIGMOD Record,* Vol. 11, No. 2, February 1981; *SIGPLAN Notices,* Vol. 16, No. 1, January 1981..

6 King, R., "A Semantics-Based Methodology for Database Design and Evolution," Ph.D. thesis (Technical Report), Computer Science Dept., Univ. of Southern California, October 1982.

7 Blaha, Michael R., William J. Premerlani, James E. Rumbaugh, "Relational Database Design Using an Object-Oriented Methodology," *Communications of the ACM,* Vol. 31, No. 4, April 1988.

Fname denotes the set of frame identifiers and SLOTS denotes the set of slots. The notion of a slot is given by the mapping

SLOTS = Sname --> SENTRY

Sname denotes the domain of slot identifiers and SENTRY denotes the domain of actual and permitted slot entries. The actual and permitted entries for a slot are given by the mapping

SENTRY - Type --> Entries

This formal semantic specification avoids the troubles that can be encountered with databases containing large amounts of rapidly changing data.[1]

3.2.2. Database Technology

Database technology has become very sophisticated since the 70s. It has gone through three generations– file systems, network and hierarchic systems, and relational systems– and is working on the fourth. However, there is some dispute over what actually constitutes the next generation. Many tout object-oriented systems (OODBMSs) as the next step, but at least one group sees object-orientation as just a part of a larger group of features.

Relational database management systems (RDBMSs) are the current state-of-the-practice, and examples are found on every computer platform. Among the more popular in the workstation market are Oracle, Informix, and Ingres. At least part of Oracle's popularity is the fact that it is available for nearly all workstations as well as the Macintosh and the IBM PC and its clones. Despite their popularity and coverage of the market, RDBMSs "have drawbacks for use in a concurrent engineering environment. RDBMSs force all information to fit two-dimensional tables. They do not meet user requirements when either the data or the applications become too complex."[2] An object-oriented database, on the other hand, "has the ability to accept attributes without changes in its structure."[3]

---

1 Reimer, Ulrich, and Udo Hahn, "A Formal Approach To The Semantics Of A Frame Data Model, *Proc. 8th Annual International Joint Conference on Artificial Intelligence,* Vol. 1, Karlsruhe, West Germany, August 1983.

2 Hitt, Ellis F., et al. "Systems Engineering Tools for Computer Aided Engineering of Ultra-Reliable Systems." Report No. R-6274, Columbus, Ohio: Battelle Tactical Technology Center, July, 1990. sponsored by DARPA. 23.

3 Hitt, et al. 22.

Mary Loomis attributes the appeal of object-oriented database management systems (OODBMSs) to two factors:

1. The data model more closely matches real-world entities. Objects can be stored and manipulated directly; there is no need to transform application objects to tables. Data types can be defined by the user, rather than being constrained to certain pre-defined types.

2. The database language can be integrated with an object-oriented programming language, meaning that the programmer deals with a single uniform model of objects.

The first factor is instrumental in bringing DBMS capabilities to applications that have complex data (e.g., CAD, CAM, CASE, CAP, etc.) and have not been able to meet their data type or performance requirements with relational DBMSs. The second factor is instrumental in improving the productivity of application developers...[1]

Stanley Zdonik and David Maier agree with Loomis on object-orientation as the future of database technology. They also believe that object-oriented systems will develop much more rapidly than did relational systems. "...There are several reasons for the more rapid emergence of OODBs [object-oriented databases]. For one, database technology as a whole is more advanced now that it was in the 1970s, and the better understanding of transactions, recovery, memory management, indexing schemes, and so forth makes designing the architecture for any database system a less daunting problem. Second, in many cases, OODBs are supporting applications where no database system was in use formerly. Relational systems for the most part went head to head with hierarchical and network databases that were already in use. Third, object-oriented languages are rapidly gaining acceptance, and OODBs are seen as better able to take care of the persistent data needs of these languages than are record-based models. In addition, much of the conceptual and language-design work from OOPLs carries over easily to OODBs."[2]

---

1 Loomis, Mary E.S. "The Basics." *Journal of Object-oriented Programming*, Vol. 3, No. 1, May/June 1990. 80.

2 Zdonik, Stanley, and David Maier. "Fundamentals of Object-Oriented Databases." *Readings in Object-Oriented Database Systems*. Stanley Zdonik and David Maier, eds. Morgan Kaufmann Publishers, Inc.: San Mateo, Calif., 1990. 28-29.

Parsaye, et al., agree that object-orientation is desirable, but not as an end in itself. They advocate that "...the relational approach to databases should be combined with object-oriented, deductive, and hypermedia approaches."[1] The result would be what they call an intelligent database, the next step in database technology evolution as they see it. An intelligent database has three main features:

1. Provides high-level tools for data analysis, discovery, and integrity control, allowing users to both extract knowledge from, and apply knowledge to, data.

2. Allows users to interact directly with information as naturally as if they were flipping through the pages of a standard text on the topic or talking with a helpful human expert.

3. Retrieves knowledge as opposed to data and uses inference to determine what a user needs to know.[2]

They have developed a general model, called FORM (Formal Object Representation Model) for intelligent databases. As yet, only one test application, Fortune Finder, has been developed using the model.[3]

Parsaye, et al., also see distributed database technology as having a major impact on the future of database technology. A major feature in a distributed database management system (DDBMS) is transparency; the users "specify high-level queries stating what they want rather than how they want to get it."[4] They offer four examples of necessary transparencies:

> *Location transparencies* – After a user presents a query to the system, the system must determine an efficient strategy for obtaining the data from various sites and combining them to answer the query.

---

1 Parsaye, Kamran, et al. *Intelligent Databases.* John Wiley & Sons: New York, 1989. 91.

2 Parsaye. 25.

3 Parsaye. 425.

4 Parsaye. 92.

> *Fragmentation transparencies* – Data may be stored in a fragmented way, for example, records in the file may be stored in multiple locations in the distributed network. The user need not be aware that the fragmentation occurs when using the system.
>
> *Transaction transparencies* – The users must be protected from side effects of interference between their own transactions and other transactions currently in the system.
>
> *Failure transparency* – Site or communication-link failures should affect only applications that currently use these resources and should be transparent to all other applications in the system.[1]

The SPS/RADC Automated Access Experiment touches on these aspects of transparency.

Zdonik and Maier "...expect the first wide use of OODBs as part of delivered applications will be embedded OODBs as part of CAE and CASE environments. These systems are likely to be persistent versions of, or heavily biased toward, a single object-oriented programming language, such as C++. OODBs that are more like current systems—supporting linkages to multiple programming languages and ad hoc querying by users—will emerge strongly a little later, probably for office applications, such as document processing, task management, and organization modeling. Eventually, OODBs will appear that are slanted toward the engineering-simulation and scientific-computing markets—areas characterized by their extensive use of ordered and graph-structured data; such as vectors, tensors, matrices, finite-element grids. We also expect to see many object-oriented features brought into record-based systems—support for complex structures with record-valued fields, hierarchies of record types, a wider variety of base types and type constructors, and more procedural material stored in the database."

"The research and development work on OODBs is far from complete... Some of the work to be done is conceptual; such as gaining a better understanding and better formalization of persistence, inheritance, and typing. Other work is system engineering; such as determining how to merge features, or how to make OODB implementations more efficient and reliable. Finally, there is still much to learn about how to map applications onto OODBs."[2]

---

1 Ibid.

2 Zdonik and Maier. 28-29.

Loomis identifies these five vendors as being the primary providers of OODBMSs in 1990: Object Design, Versant (formerly OBJECT-Sciences), Objectivity, Ontologic, and Servio Logic.[1] Object Design's product, ObjectStore, contains bindings for C and C++.; it runs under UNIX and X-Windows. Versant markets a suite of tools, including their own versions of C and C++. The tools support such standards as OSF/Motif, NIH's C++ libraries, and TCP/IP protocols, among others. Ontos is Ontologic's object database product. Available for OS2 and VAX/VMS as well as Sun3, Apollo and VAXstation workstations, it provides persistence to C++. Servio Logic is the developer of the GemStone OODBMS, which runs on the VAX under VMS using any of the following as workstations: IBM PC under Windows or Smalltalk/V, VT100 terminals, the Sun3 with Smalltalk-80, and the Tektronix 440X with Smalltalk-80. GemStone supports the C language.

Zdonik and Maier offer some predictions for active areas of OODB research over the next three to five years. In the conceptual area, types, query languages, and version and configuration models are in the early stages; more research is required. "An active area in database-systems research is deductive or logic databases. These are relational-like databases with query languages based on logic-programming languages or other logic formalisms, particularly the Prolog language..."[2]

In system engineering, optimization technology, storage management, and parallelism are all important research areas. The emergence of non-standard architectures for database systems will also have an impact on OODBs.[3]

Under applications, Zdonik and Maier discuss application-development tools and design methodologies. They describe tools as "minimal" and methodologies as "a black art."[4] They also point out that although hypertext and OODBs share some features, there is no direct mapping between the two. "Hypertext links are also more personal than are shared data; they indicate the connections that a particular individual thinks are interesting or important. Two users might want

---

1 Loomis. 80.

2 Zdonik and Maier. 29.

3 Ibid.

4 Zdonik and Maier. 30.

to maintain entirely different linkage structures on the same corpus of text."[1] However, like hypertext, OODBs are good candidates for cooperative work environments. "They support sharing of artifacts at multiple levels of granularity, appear to be able to support versions and alternatives reasonably well, and can capture the mass of management information that goes along with any multi-person project..."[2]

It would appear that the differences between the two views of the future of database technology are semantic rather than real. Both touch on the same combinations of technologies, and both agree that object-orientation is, at the least, a major part of the next generation of DBMSs.

3.2.3. Reuse and Component Libraries

Reuse of both code and design information is a worthy goal in both government and industry. For government applications, reuse involves Ada, and represents a possible savings of dollars as well as time. "...Ada reuse could save billions of defense dollars each year, says Ralph Crafts, president of Software Strategies and Tactics Inc., Harper's Ferry, W. Va. Craft is editor of Ada Strategies, a newsletter focusing on competitive strategies in the Ada marketplace... 'Congress could meet the Gramm-Rudman deficit reduction target simply by establishing a national software engineering policy with Ada as the Standard Language,' Crafts insists. Estimating that '75 percent of the government's software budget goes to maintain old code,' Crafts says those costs 'would drop drastically under an Ada standard.'"[3]

But before extensive reuse becomes a reality, we have to change the way people think. Programming "culture" is a stumbling block for reuse. Programmers are used to *developing* solutions to problems, not borrowing someone else's solution. At a software industry panel discussion in 1987, Andy Hertzfeld, developer of the Macintosh operating system, made this statement: "'How can you care about your program if you use someone else's code?' He continued, 'I consider myself an artist. If I were another kind of artist, a writer for instance, would it be right for me to go out and buy a paragraph here, a chapter there, and include them in my book? Would it still be mine?' [Scott Kim, author of the book *Inversions*] added that it usually takes him longer to read and understand another person's

---

1 Ibid.

2 Ibid.

3 Pappas, T.L. (Frank). "The Trials and Tribulations of Ada and Reuse." *Military & Aerospace Electronics*, Vol. 1, No. 8, August, 1990. 31.

code than it does to write it himself in the first place. All the programmer panelists, however, said that over the years they had put together their own subroutine libraries, which they regularly use."[1]

Personal subroutine libraries are very common and are widely used on an individual basis: the programmer knows where to find the routines and is aware of their weaknesses and strengths. The same is not true of other subroutine collections, which makes categorizing and cataloguing very important. To get programmers to change their habits, reuse must be fast and easy. What incentive is there for reuse when a programmer can develop a component faster she can locate it in a reuse library?

Supporting reuse is not simply a matter of saying, "From now on, we will support reuse of our code." Anthony Gargaro, lead scientist at Defense Systems Division, Computer Science Corp., told *Military & Aerospace Electronics* that it "takes considerably more time, planning, and expertise to write reusable software than most people realize. It's not until the software is reused that savings are actually realized."[2]

The fact that there are many platforms and many Ada compilers also contributes to reuse difficulties. Not every compiler implements all features of the Ada language in exactly the same way. This is particularly true of Ada's Run Time System, "the hardest portion of the language to adapt from one hardware target to the next."[3] Programmers have to solve their particular system problems with the available compiler. "Sometimes programmers may even have to write their code so that it depends on the particular task scheduling algorithm used by a given vendor—just to meet performance requirements. The downside to this approach, however, is that it reduces the potential to reuse the software in another system."[4]

The Software Engineering Institute (SEI) at CMU has developed an alternative to the Ada RTS. The Distributed Ada Run-time Kernel (DARK) "is especially useful for distributed applications in avionics and command and control systems." DARK is available for Motorola 68020-based systems and the VAX, but could be ported to another platform in as little as 8 labor months. Although DARK is a

---

1 Staff. "Programmers Debate Languages, 'Corporate' Programming." *Byte*, June, 1987. 38.

2 Pappas. 33.

3 Pappas. 33.

4 Pappas. 33.

prototype, Judy Bamberger, DARK's transition project leader at SEI, says that "a standard—whether it's DARK itself, a standard based on DARK, or some other standard—is imperative for reuse to occur."[1]

There are a number of legal issues to consider when reusing someone else's software, including warranties, liability, copyright, contractual clauses, and proprietary information. "For example, does a contractor need to provide a warranty for reusable software? If a system fails due to a flaw in a reusable software component, who is legally liable? The developer? The contractor who decided to reuse the software? DoD, if it provided the component?"[2]

Security should be considered, as well. In his acceptance speech for the 1983 Turing Award, Ken Thompson addressed the security issue. He provided an excellent example of a Trojan horse that could be inserted into a C compiler and would replicate itself with no trace in source code. Thompson followed up with this statement: "The moral is obvious. You can't trust code that you did not totally create yourself. (This is especially true for code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code... As the level of program gets lower, these bugs will be harder and harder to detect. A well-installed microcode bug will be almost impossible to detect."[3]

There have been a number of federal studies of reuse, including the Air Force's CAMP and RAASP, the Army's RAPID and ARCS, NASA's Eli, and the joint services STARS program. SPS has been involved with ARCS, Eli, and STARS, as well as several other reuse efforts.

Intermetrics developed a Reusable Software Library (RSL) prototype in 1987. "[T]he RSL comprises the RSL database and four subsystems

(1) Library Management

(2) User Query

(3) Software Component Retrieval and Evaluation (SCORE), and

---

1 Pappas. 33.

2 Pappas. "Procurement, Legal Issues Hinder Reuse." sidebar to "The Trials and Tribulations of Ada and Reuse." 33.

3 Thompson, Ken. "On Trusting Trust." UNIX *Review*, Vol. 7, No. 11, November 1989. 72.

(4) Software Computer-Aided Design (Soft CAD)."[1]

Although components in the RSL can be in any language, the company emphasized Ada "because many of Intermetrics' applications are Ada-related, and because Ada's generic packages show great promise for reuse—they are natural candidates for entry into the RSL."[2] Prior to its inclusion in the library, a component undergoes evaluation by the RSL librarian, who evaluates its "structure, functionality, complexity, level of testing, quality of documentation, and other issues..."[3]

The RSL has some interesting features. One is the natural language front end for user query. "As expected, [it] was significantly easier to use than the database query language. It was also significantly slower."[4] Another is the SoftCAD tool, "a graphical design and documentation tool that has been integrated with the RSL prototype to aid the user in the high-level design of software systems."[5] The designer draws object-oriented diagrams "that are interpreted by SoftCAD and automatically translated into Ada PDL."[6] The designer can create new RSL components with SoftCAD or modify existing ones.

The Software Technology Support Center, part of the Ogden Air Logistics Center-TISAC at Hill AFB, Utah, listed a number of sources for reusable software components in its newsletter, Crosstalk:[7]

- Ada Language System/Navy (ALS/N) (SYSCON Corp.)
- Ada Software Repository (tape from DECUS Program Library, disk from Advanced Software Technology, Inc., and CD-ROM from ALDE Publishing)
- Booch Components (Wizard Software)

---

1 Burton, Bruce A., et al. "The Reusable Software Library." IEEE Software, July, 1987. 26.

2 Ibid.

3 Ibid.

4 Burton. 28.

5 Burton. 30.

6 Ibid.

7 Levine, Trudy. Reusable Software Components column. *Crosstalk*, STSC, Hill AFB, UT, December, 1990. 6-8.

- Cosmic (University of Georgia)
- C2MUG Software Catalog (AMSEL-RD-SE-BCS-MC (C2MUG), Ft. Leavenworth, KS)
- Generic Reusable Ada Components for Engineering (GRACE) (EVB Software Engineering, Inc.)
- HALO (Media Cybernetics, Inc.)
- Math Pack (MassTech, Inc.)
- PragmAda Reusable Components (PragmARCs) (PragmAda Software Engineering)
- Ada Math Advantage (Quantitative Technology Corp.)

4. Framework Software

4.1. User Interfaces

Because more work is being done with object-oriented tools and techniques, tools are moving away from command line interfaces to graphical, object-oriented interfaces. The Macintosh desktop metaphor, based on work done at Xerox PARC with Smalltalk and the STAR system, has become the de facto standard for graphical user interfaces (GUIs). Customers demand it and other developers try to emulate it as closely as possible, so much so that Apple filed suit against Microsoft and Hewlett Packard for emulating it a bit *too* closely. Still, it would be difficult to buy a workstation today that did not have some sort of GUI with windows, icons, and a mouse or some other pointing device.

A drawback of GUIs is the amount of system overhead they introduce. Constantly updating and redrawing several graphical windows takes up processor time as well as memory. At times, users wait through noticeable delays as the screen is refreshed, as anyone familiar with the lower-end Macintoshes will attest. However, engineering workstations, such as Suns or Apollos, have enough horsepower to get around this glitch.

Although many workstations have proprietary window managers and their own GUI "look-and-feel," the growth of networks of heterogeneous workstations has made standards a necessity in the marketplace. In the UNIX world, there are three separate parts to a GUI: one handles the mechanics of the windows, another provides the appearance of on-screen items, and a third provides the Mac-like portion (pull-down menus, icons, etc.). At present, M.I.T.'s X Window System, a window manager, seems to be the de facto standard for windowing systems. Many existing tools have been converted to work with X, and nearly all tools being developed will work with it. There is some contention, however, about a standard for the look-and-feel portion. The two main contenders are Open Look and Motif. Many tool developers are hedging their bets by supporting both. There are several choices for the desktop manager, the part that provides menus and icons. Among them are Looking Glass, X.desktop, and Visual User Environment (VUE).

Looking Glass is strictly a UNIX product, "designed to give users the full functionality of UNIX in a graphical format."[1] X.desktop's goal is to provide

---

1 Wagner, Mitch. "Computing Made Safe and Easy." supplement to *UNIX Today!*, March 1991. 10.

"the same look-and-feel for any hardware platform, from a PC to a VMS mainframe to a Cray supercomputer, so that users don't have to sit down and learn a new way of computing every time they sit down at a new computer."[1] VUE "is designed with the network in mind—to give users access to all the resources of the network from a single screen, including PCs, VMS mainframes, and UNIX boxes all the way up to a Cray supercomputer.

As a window manager, X has been and continues to be an evolving standard. Two growing needs are video and audio. The Video Extension for X, better known as VEX, was proposed in 1988. "VEX is an attempt to provide basic abstractions for the operations a client—an X program that works on behalf of the user—would want to perform with video input and output: placing a video picture on the screen, obtaining the digitized pixels from a frame of video, controlling the blending of video and graphics pixel by pixel, and capturing a portion of the screen from some particular video recording device. The extension also includes a basic framework for controlling external devices, but VEX leaves unspecified the semantics of each device and all its knobs and dials for the very simple reason that there are potentially so many different kinds."[2] At least one vendor is not waiting for a standard to appear: RDB Spectrum is offering a $750 package called X.TV "that allows a video image, contained in a standard window, to be repositioned anywhere on the screen or to be scaled up or down in size. The product runs only on RGB/View hardware via VMEbus, SCSI, or RS232 port..."[3]

There is no similar standard proposed for extending X to include sound, although there have been a number of architectures developed over the past ten years for integrating audio with workstations. In 1984, Xerox PARC developed the Etherphone server, which consists of a "centralized server with distributed control and user interface... Etherphone is...still in daily use."[4] The Modular Integrated Communications Environment (MICE), developed at Bell Communications Research from 1985 through 1988, "provided a full range of audio and telephony functions from a UNIX-based centralized server. An application programming environment provided distributed access control to this server over Ethernet; voice was transported using conventional analog

---

1 Ibid.

2 Brunhoff, Todd. "Pleasing the Eye." *UNIX Review*, Vol. 7, No. 10, October, 1989. 71.

3 Staff. Product Briefs column. *UNIX Today!*, March 18, 1991. 46.

4 Schmandt, C. "Audio Servers: A Brief History." Sidebar to "Getting the Word." *UNIX Review*, Vol. 7, No. 10, October, 1989. 56.

telephony [via a computer-controllable Redcom PBX]."[1] Northern Telecom's Meridian Mail "gives PCs network-based interfaces to voice mail and, indirectly, to the PBX... Local PCs send commands over the network to the centralized voice storage server via a bridge, allowing for a distributed user interface..."[2] USC's Phoneserver "provides basic telephone interfaces between UNIX-based workstations over Ethernet to a Rolm PBX. The server itself is a PC with a special PBX interface card; workstations transmit requests via UDP datagrams, and a local user interface (Phonetool) runs under...SunView... Phonetool has recently been ported to the X Window System using auto-dial modems..."[3] The latest research development is the VOX Audio Server from Olivetti Research Center. "The VOX server provides a device-independent interface for audio functions, including play, record, and telephony; there are also plans to include extensions such as speech recognition and synthesis. In addition the VOX server will provide support for audio routing and mixing... VOX is based on a network-transparent client/server architecture heavily influenced by contemporary window systems..."[4]

The developers of Motif and Open Look each hope to have their product become a standard, but that doesn't seem likely. The competition has taken on the overtones of a religious war, and neither product's followers seem likely to convert; nor does there appear much chance of a compromise between them.

The desktop managers are not necessarily trying to become standards. Each is aimed at a slightly different audience, and has a slightly different mission. As with any other products, users are best served by choosing the one that suits their needs.

Current technology in user interfaces is stable, particularly within a homogeneous environment. Some difficulties may arise in a heterogeneous environment, particularly with audio applications, but current research is addressing the problems.

---

1 Schmandt. 57.

2 Ibid.

3 Ibid.

4 Schmandt, C., and B. Arons. "Getting the Word." *UNIX Review*, Vol. 7, No. 10, October, 1989. 61.

4.2. Hypertext/Hypermedia

Hypertext, hypermedia, and multimedia are related technologies; the differences between them are more of degree than of kind. Definitions are less than exact or agreed upon. Those provided here have been assembled from various sources, and fit the way we view the use of the technology.

Multimedia is the simplest of the three. It is a presentation technology based on computer control and integration of a variety of electronic media, such as CD-ROM discs, speech/audio synthesizers, VCRs, videodisc players, and computer monitors. Sound, text, and visuals are mixed together from the various sources and displayed on the computer monitor. Uses range from marketing to education.

Hypertext is a more interactive technology, but it is limited to text and computer graphics stored on a standard computer storage device, such as a hard drive. Hypertext is also known as nonlinear text or threaded text. That is, rather than appearing sequentially, like chapters in a book, nodes of information are connected via machine-supported logical links (threads). The user is not constrained to turn page after page; instead, he or she can jump from node to node following the various links. The first generation of hypertext systems (e.g., NLS/Augment, FRESS, and ZOG) "were all originally mainframe-based, focused primarily on text nodes, and used display technologies with little or no graphics capabilities."[1] Second-generation systems are similar in concept to the first generation but the workstations and PCs on which they operate allow more sophisticated user interfaces. Text is fully-formatted, and graphics and animations are possible; some sort of windowing system is standard (see figure 4-1). The primary difference between the two generations is that second-generation systems are aimed mostly at single users or small workgroups. They provide less support for collaborative work than in the earlier systems.

---

1 Halasz, Frank. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems." *Communications of the ACM*, July 1988. 840.

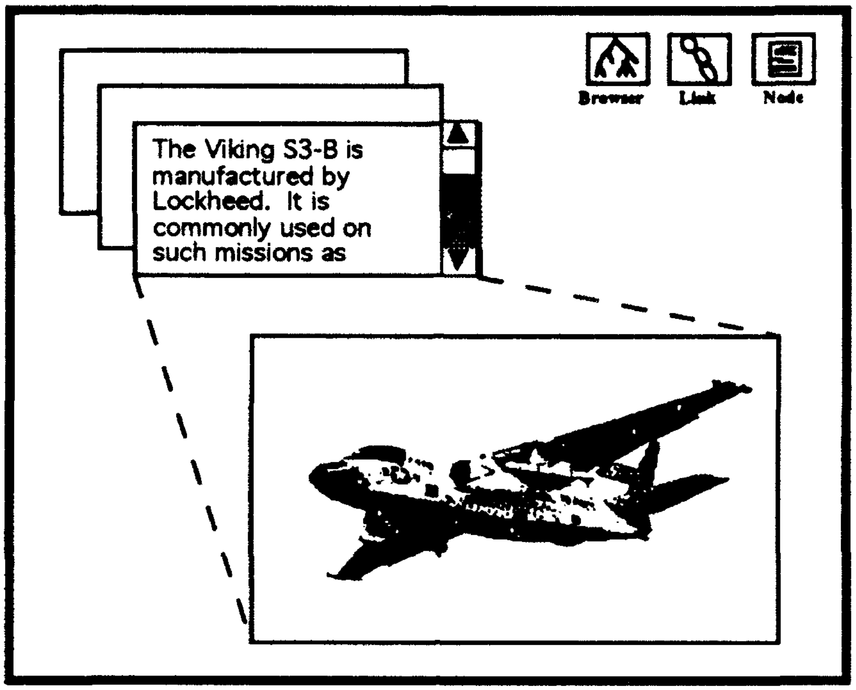

Figure 4-1. Sample hypertext system display.

Most of the current hypertext systems have been designed for specific purposes. According to Jeff Conklin, hypertext systems have been developed in four broad application areas:

- *macro literary systems*: the study of technologies to support large on-line libraries in which interdocument links are machine-supported (that is, all publishing, reading, collaboration, and criticism takes place within the network);
- *problem exploration tools*: tools to support early unstructured thinking on a problem when many disconnected ideas come to mind (for example, during early authoring and outlining, problem solving, and programming and design);
- *browsing systems*: systems similar to macro literary systems, but smaller in scale (for teaching, reference, and public information, where ease of use is crucial);

- *general hypertext technology*: general purpose systems designed to allow experimentation with a range of hypertext applications (for reading, writing, collaboration, etc.).[1]

Table 4-1[2] provides a listing of some of the available hypertext systems and their features. (Note: We disagree with some of Conklin's inclusions; specifically, outline processors.)

Hypermedia is an amalgam of hypertext and multimedia. The hypertext "engine" retrieves data, sound, and visuals from a variety of electronic media, which may be attached directly to the computer or accessed over a network. The interactive interface can range from limited to rich, depending on the purpose of the system. Those which are mainly informational, or browsing, systems, such as touch-screen museum exhibits, offer limited menus from which users select choices by touching the appropriate part of the screen. No new links can be forged. The system SPS built for the RL, the Automated Access Experiment (AAE), uses multiple computer platforms equipped with keyboards and mouses and numerous peripheral devices. The platforms are all linked via Ethernet.[3] This system does permit individual users to create new links, albeit in a limited fashion, thus extending the functionality of the system.

Frank Halasz identified seven issues for hypermedia development:[4]

1. Search and query in a hypermedia network
2. Composites—Augmenting the basic node-and-link model
3. Virtual structures for dealing with changing information
4. Computation over networks
5. Versioning
6. Support for collaborative work
7. Extensibility and tailorability

---

1 Conklin, Jeff. "Hypertext: An Introduction and Survey." IEEE Computer, Vol. 20, No. 9, 1987. 20.

2 Conklin. 21.

3 Contract No. F30602-89-0003

4 Halasz. 841-850.

How each of the issues is addressed depends on the use to be made of hypertext/hypermedia in the system.

Table 4-1. Hypertext systems and their features.

Hypertext System	Hierarchy	Graph-based	Link Types	Attributes	Paths	Versions	Procedural Attachment	Keyword or String Search	Text Editor	Concurrent Multi-users	Pictures or Graphics	Graphical Browser
Boxer	Yes	Yes	Fixed[1]	No[1]	No	No	Yes	Yes	Emacs	No	Yes	Yes
CREF	Yes	Yes	Yes	No	No	By link	No	Yes	Zmacs	No	Yes	No
Emacs INFO	Yes	No	No	No	No	No	No	Yes	Emacs	No	No	No
IBIS	Yes	Yes	Yes	No	No	By link	No	No	A basic text editor	Yes	No	No
Intermedia	Yes	Yes	Yes	Yes	No[2]	No	No[2]	Yes	custom	Yes	Yes	Yes
KMS	Multiple	Yes	Fixed	No	No[1]	Yes	Yes	Yes	Text/ graph. WYSIWYG	Yes	Yes	No
Neptune	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Smalltalk-80 editor	Yes	Yes	Yes
NLS/Augment	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Custom	Yes	Yes	No
NoteCards	Multiple	Yes	Yes	Nodes	No	No	Yes	Yes	Interlisp	Yes	Yes	Yes
Outline Processors	Yes	No	No	No	No	No	No	Yes	Various	No	No	No
PlaneText	Unix file sys.	Yes	No	No	No	No	No	Unix/grep	SunView text ed.	Yes	Yes	Yes
Symbolics Document Examiner	yes	Yes	No	No	Yes	No	No	Yes	None	No	No	No
SYNVIEW	yes	No	No	No	No	No	No	No	line ed./Unix	No	No	No
Textnet	Multiple	Yes	Yes	Yes	Yes	No	No	Keyword	Any	No	No	No
Hyperties	No	Yes	No	No	No	No	No	No[2]	A basic text editor	No	Yes	No
WE	Yes	Yes	No	Fixed	No[2]	No[2]	No[2]	No	Smalltalk-80 editor	No[2]	Yes	Yes
Xanadu	No	Yes	Yes	Yes	Yes	Yes	No	No	Any	No	Yes	No
ZOG	Yes	No	No	No	No	No	Yes	Full text	Spec. Pur.	Yes	No	No

[1] Can be user programmed.

[2] Planned for next version.

In this table, each column represents one possible feature or ability that a hypertext system can provide. The negative or affirmative entries in the table indicate whether the corresponding hypertext systems meets the standard criteria for a specified feature. These criteria are listed below.

Hierarchy: Is there specific support for hierarchical structures?
Graph-based: Does the system support nonhierarchical (cross-reference) links?
Link types: Can links have types?
Attributes: Can user-designated attribute/value pairs be associated with nodes or links?
Paths: Can many links be strung together into a single persistent object?
Versions: Can nodes or links have more than a single version?
Procedural attachment: Can arbitrary executable procedures be attached to events (such as mousing) at nodes or links?
String search: Can the hyperdocument be searched for strings (including keywords)?
Text editor: What editor is used to create and modify the contents of nodes?
Concurrent multiusers: Can several users edit the hyperdocument at the same time?
Pictures or graphics: Is some form of pictorial or graphical information supported in addition to text?
Graphics browser: Is there a browser which graphically presents the nodes and links in the hyperdocument?

4.3. Access Control/DIS Security

Computer security is an important issue in both the commercial and government arenas. For standalone, single-user workstations, security might be as simple as controlling physical access to the workstation itself. However, multi-user systems and networks of many workstations and servers is more the norm these days. Nearly all multi-user systems provide some sort of access-control, usually in the form of passwords and/or read-write permissions, to protect users' data from casual scrutiny. However, as the number of publicized "break-ins" indicates, such mechanisms can be gotten around; the Internet worm proliferated itself into as many as 15,000 computers.[1] In the commercial sector, the loss or compromise of data may not be as important as the costs arising from such loss or compromise. Compromise is a bigger issue with the government, although costs are considered as well. Government systems cannot rely on such a low level of security, and commercial operations are beginning to feel the same way.

The Department of Defense has set up requirements for computer security in DoD 5200.28-STD, *Trusted Computer System Evaluation Criteria,* also known in the industry as "the Orange Book." Vendors submit both hardware and software, including operating systems, for evaluation against the requirements. There are seven levels of trust outlined: D (no security), C1, C2, B1, B2, B3, and A1 (the greatest security). The National Computer Security Center (NCSC) publishes guidelines to help developers understand the requirements.

"The main goal of [the NCSC] is to encourage the widespread availability of trusted computer systems... One of the features...that is required of a secure system is the enforcement of discretionary access control (DAC). DAC is a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a user or process given discretionary access to information is capable of passing that information along to another subject...[2] This basic principle of discretionary access control contains a fundamental flaw that makes it vulnerable to Trojan

1 Fisher, Sharon. "UNIX Security and Government Secrets." *UNIXWorld*, Vol. VI, No. 4, April 1989. 36.

2 United States. National Computer Security Center. *A Guide to Understanding Discretionary Access Control in Trusted Systems*. NCSC-TG-003. Ft. Meade, Md.: GPO, 1987. 1.

horses.[1] On most systems, any program which runs on behalf of a user inherits the DAC access rights of that user."[2]

"Discretionary Access Control cannot deter hostile attempts to access sensitive information. DAC mechanisms can be designed to eliminate casual browsing through the storage system, to prevent accidental disclosure, modification, and destruction of data, and to provide for the sharing of data between users..."[3]

Trojan horses are not the only means of breaking into computer systems. The Internet worm took advantage of 'holes' in UNIX to gain access to other systems. "The Internet worm broke into systems in three main ways:

- The sendmail program, which routes mail in a network, had its debugging bit on, meaning the worm could issue a set of commands to the mailer;
- The finger utility, which allows users to obtain information about other users, didn't check for buffer overflow, meaning that another set of commands could be added to the UNIX stack;
- Through an internal dictionary of words, which it used against user passwords."[4]

UNIX has long had a reputation for being not very secure:

"Users can easily make their files available to other users, and users with root privileges essentially have access to any file in the system. Moreover, any UNIX wizard worth his or her salt knows a half-dozen ways to get root privileges. But UNIX isn't necessarily any less secure than any other operating system, say experts. It can be as secure as the system administrator makes it.

'UNIX itself is not the problem,' says Russell Brand, a computer scientist math programmer at the Lawrence Livermore National Laboratories in Livermore,

---

1 "a computer program with an apparently or actually useful function that contains additional (hidden) functions that surreptitiously exploit the legitimate authorizations of the invoking process. An example is a program that makes a 'blind copy' of a sensitive file for the creator of the Trojan horse." NCSC-TG-003. 4.

2 NCSC-TG-003. 5.

3 NCSC-TG-003. 28.

4 Fisher. 39.

Calif. 'There are lots of utilities and practices in UNIX that are insecure, and there are lots on every other system that are insecure. There are just as many problems with VMS and with the IBM operating systems, but nobody will talk about them.'" [1]

One reason for that silence is that many times commercial ventures will write off a loss caused by breached computer security rather than publicize the break-in; they stand to lose more in prestige than the actual value of the loss.

According to Brand, Lawrence Livermore National Laboratories has—locked in a vault—books of security holes in various operating systems. 'The books on UNIX aren't any thicker than [for] anything else,' he says. 'They're all big books.'

'UNIX can be a very secure operating system,' says Dennis Rears, a computer scientist for the U.S. Army Armament Research Development and Engineering Center in Picatinny Arsenal, N.J. 'However, unlike other operating systems, it can be either made or broken by the system administrator. One small mistake can blow all the security.' System administrators are personally responsible for educating users in how important security is and how to make their systems more secure, Rears states...

Putting a computer on a network automatically makes it less secure, according to the experts... According to Rears, 'Anything on a network is inherently insecure. When you're passing packets back and forth, any system can grab those packets. Any network programmer can break into almost computer system that passes data through his computer, until we start using encryption.'

In the end, says [John Romkey, chairman of the board of FTP Software Inc.], 'the only truly secure network is one that's locked up and physically isolated from the rest of the world.'[2]

The government does exactly that with computers containing classified data. Brand says that "'classified machines [are put] inside a fence, and they have no connections to other systems. They're perfectly safe, assuming nobody gets over the fence and past the guards.' Rears says that 'at all government levels, you will

---

1 Fisher. 39

2 Fisher. 39.

not see any systems networked that process classified information. Never.'"[1] Since all software must be checked out and approved before it is allowed on the system, worms and viruses cannot attack classified systems.

There are several efforts underway that address security from several different standpoints, ranging from secure distributed file systems to secure operating system kernels to secure user interfaces.

In 1983, Carnegie Mellon University (CMU) and IBM began a joint project to create Andrew, a distributed UNIX file system. The project had two critical constraints: the resulting file system had to be scalable to an eventual network size of more than 5,000 workstations, and it had to be secure in order to ensure the privacy of users' files. The Andrew File System (AFS) is now in its third revision.

The "information-sharing backbone of the system" is called Vice and consists of a collection of trusted file servers and untrusted networks... A process called Venus, running on each workstation, mediates shared file access. Venus finds files in Vice, caches them locally, and performs emulation of UNIX file system semantics. Both Vice and Venus are invisible to workstation processes, which only see a UNIX file system, one subtree of which is identical on all workstations... The servers in Vice are physically secure and run trusted system software. No user programs are executed on servers. Encryption-based authentication [a variant of the Needham and Schroeder private key algorithm] and transmission are used to enforce the security of server-workstation communications.

When a user logs in on a workstation, his or her password is used to obtain tokens from an authentication server. These tokens are saved by Venus and used as needed to establish secure RPC connections to file servers on behalf of the user... Because tokens typically expire after 24 hours, the period during which a lost token can cause damage is limited.[2]

AFS also provides an audit trail for actions taken by system administrators. For file protection, AFS uses an access list mechanism that can specify negative rights (the denial of specific rights). In case of conflict, denial of a privilege overrides

---

1 Fisher. 40.

2 Satyanarayanan, Mahadev. "Scalable, Secure, and Highly Available Distributed File Access." IEEE Computer, Vol. 23, No. 5, May 1990. 9-22.

possession. "Although Vice actually provides protection on the basis of access lists, Venus superimposes an emulation of UNIX protection semantics."[1]

At the 1984 IEEE-sponsored Symposium on Security and Privacy, Steven Kramer reported on his efforts in developing LINUS IV, "a UNIX-like system that preserves the UNIX environment while greatly increasing the protection within the system."[2]

"Linus IV enhances some protection features at the top levels of the operating system to prevent disruption of implementation dependent design. LINUS IV radically changes the authentication mechanism, strengthens the existing discretionary access control, creates mandatory access control, creates a flexible auditing mechanism, and provides for better data integrity..."[3] Where UNIX has a single superuser, LINUS IV has three separate special users: the Security Officer, the Operator, and the Administrator. Each has different functions to perform, none of which can be performed by either of the other special users. A single special user "can still subvert the system but not without a good chance of recognition by the remaining special users. A total system penetration that in UNIX could be done completely and quietly by the superuser may take *multiple* LINUS IV special users and may not be so quiet... The separation of powers among the special users guards more against mistakes and simple penetrations rather than complex covert attacks by the truly malicious and determined special user."[4]

In 1987, NCSC formed the Trusted UNIX Working Group (TRUSIX) "to provide guidance to vendors and evaluators involved in the development of Trusted Computer System Evaluation Criteria (TCSEC) class B3 trusted UNIX systems. The NCSC specifically targeted the UNIX operating system for this guidance because of its growing popularity among the government and vendor

---

1 Satyanarayanan.

2 Kramer, Steven. "LINUS IV - An Experiment in Computer Security." *Proceedings of the 1984 Symposium on Security and Privacy*, Silver Spring, Md.: IEEE Computer Society Press, 1984. 24.

3 Kramer. 27.

4 Kramer. 29.

communities..."[1] Their first report described alternate methods of implementing access control lists for UNIX. Later reports will address other aspects of security for UNIX.

Varadharajan and Black are addressing information flow security in distributed object-oriented systems. In their model, security labels are attached to the containers of information—slots, or variables—rather than to the information itself; "the label of the container reflects the sensitivity of its information content. Hence controls that deny access to information are based on the labels of the containers."[2] Their model satisfies four groups of security properties: system set-up, message-passing, operations of methods within an object, and modification of security labels and creation/deletion operations. Varadharajan and Black point out that their work only addresses mandatory security mechanisms, and that it also requires authentication and discretionary access control mechanisms. They intend to formalize both the object model and the security model, and address multi-labeled methods, context-dependent label pairs, and the effects of labeling on classes and inheritance.

Security issues affect the user interface as well as the operating system itself. This is especially important today, with graphical user interfaces such as X riding on top of the operating system. TRW Systems Integration Group is working on the Trusted X Window System (TX/WS),"a prototype B3 implementation of the X Window System for the TMach (trusted Mach) operating system. Goals of this research effort include studying visible labeling issues, developing a formal model of the X Window System, examining covert channels, and applying a TRW process model for high performance trusted systems in Ada."[3]

"In TX/WS, each top-level window on the screen may belong to a different application. These applications may be running at a variety of sensitivity labels. Each window is visibly labeled with the maximum sensitivity label of the application which created the window. To assist the user of a multi-level

---

1 United States. National Computer Security Center. *Trusted UNIX Working Group (TRUSIX) Rationale for Selecting Access Control List Features for the UNIX System*, NCSC-TG-020-A. Washington: GPO, 1989.

2 Varadharajan, Vijay, and Stewart Black. "A Multilevel Security Model for a Distributed Object-Oriented System." *Proceedings of the Sixth Annual Computer Security Applications Conference*, Los Alamitos, Calif.: IEEE Computer Society Press, 1990. 71.

3 Epstein, Jeremy. "A Prototype for Trusted X Labeling Policies." *Proceedings of the Sixth Annual Computer Security Applications Conference*, Los Alamitos, Calif.: IEEE Computer Society Press, 1990. 221.

windowing system with overlapping windows, a visible labeling policy is required to avoid inadvertent misclassification of information."[1]

Security of computer systems is an issue at the international level, as well. One group, the European Computer Manufacturers Association (ECMA), "has been developing a standard (STD 138) entitled 'Data Elements and Service Definitions – Alice in Wonderland.' This standard proposes a set of security services and security information for use in the application layer of the OSI model. Protocols for accessing the services and a transfer syntax for the security information make this standard applicable to distributed systems. This standard...avoids placing unnecessary constraints upon the internal design and implementation of information processing systems that process and exchange security related information... The ongoing work is targeted for a standard on 'authentication and privilege service' by July 1991. Future services to be defined include 'secure association service' and 'authorization service.'"[2]

It's important to remember that people, not computers, are the linchpin of any security effort. In a 1984 lecture, Ken Thompson, one of the developers of the UNIX operating system, pointed out a method by which a C compiler can contain self-replicating trojan horse that leaves "no trace in source anywhere. The moral is obvious. You can't trust code that you did not totally create yourself. (This is especially true that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code. In demonstrating the possibility of this kind of attack, I picked on the C compiler. I could have picked on any program-handling program such as an assembler, a loader, or even hardware microcode. As the level of program gets lower, these bugs will be harder and harder to detect. A well-installed microcode bug will be almost impossible to detect."[3] Security depends on people.

---

1 Epstein.

2 Presttun, K. "Open Systems Security Framework, Alice and Beyond." *Proceedings of the Sixth Annual Computer Security Applications Conference*, Los Aiamitos, Calif.: IEEE Computer Society Press, 1990. 201.

3 Thomspon, Ken. "On Trusting Trust." UNIX Review, Vol. 7, No. 11, November, 1989. 73.

5. Tools

5.1. CAD/CAE

The Massachusetts Institute of Technology (MIT) introduced the concept of computer aided design (CAD) in the 1950s... At first, CAD tools were primarily used for mechanical design and provided 2D design representation in a plane. These tools were then enhanced to provide an interface with computer numerically controlled (CNC) machines to control machine tools in two, and then three, axes. This was followed by expansion to [3D] views using perspective and wire frame models. Solid models are now becoming more prevalent with the advances in workstation and computer technology providing higher computational rates.

Solid models have been developed to provide a graphical representation of the molecular structure of chemicals, and compounds, as well as individual parts of a design and the final product.

CAD systems now capture design attributes, rules, and data structures that can generate the geometry of a part or a machine through the use of parametric design. The application of object-oriented programming and expert system shells to CAD has resulted in object-oriented databases storing objects that represent design rules and the geometry of a part can be generated from the specifications and design rules. ICAD produced the first system that stores product information as a data structure of rules and requirements rather than as a geometric construct. Design information is extracted to create drawings and the program generates a parts list, bill of materials, process plans, and other documentation.

Computer Aided Engineering (CAE) marked the evolution to using computers for simulation of the design to verify design decisions prior to building a prototype. CAE is considered by many to encompass CAD and Computer Aided Manufacturing (CAM). Others use the term CAD to include all aspects of computer aided design and engineering.[1] CAE has also come to have a more specific connotation in some circles, where it refers to the use of computers to aid

---

1 Hitt, Ellis F., et al. "Systems Engineering Tools for Computer Aided Design of Ultra-Reliable Systems." Technical Report No. R-6274, Columbus, Oh.: Battelle Tactical Technology Center, July, 1990. 2-3.

in the design and manufacture of electronic components, circuits, and circuit boards.

Many of the existing CAD tools were developed by small innovative companies and are meant to capture a specific segment of the overall system design process such as schematic capture/editor, logic synthesis, digital simulation, analog simulation, board layout, board routing, thermal analysis, or mechanical design. Interfaces between tools developed by different vendors have been compared with the early railroads in which every company used a different track gauge. Ideally, a change made in one phase of a design such as schematic capture would be automatically reflected in the simulation, board layout, routing, and thermal analysis. Vendors offering an integrated set of tools handle this problem with a shared database. Unfortunately, this requires third party vendors offering tools which support the larger vendors to interface their tools to many different integrated toolset databases...

This problem has been recognized by the CAE/CAD industry as well as users of the tools. The result has been efforts such as the CAD Framework Initiative (CFI). CFI has seven technical subcommittees that address different areas of the problem. Coordination slows the process of development of the CFI standards. Microelectronics and Computer Technology Corp. (MCC) has announced the creation of the CAD Framework Laboratory (CFL) to support and evaluate the standards proposed by CFI. CFL may propose selected framework interfaces for consideration as candidate standards. CFL also develops a Verification/Validation (V/V) software system for each standard adopted by CFI. [1]

5.2. CASE

Computer-aided software engineering (CASE) tools are intended to be to the software engineer what CAD tools are to the draftsman or architect: tools that automate the job to some degree, making a complex task easier to accomplish. In its broadest definition, a CASE tool can be any software tool used by a software engineer in the course of developing a software product. This includes documentation tools, project management tools, and compilers. However, CASE has come to refer primarily to requirements, analysis, and design tools, such as Statemate, Software through Pictures, and Teamwork.

CASE tools of this variety are typically designed to support a specific method, and this factors into their purchase. The Barton Group's recent survey of 119

---

1 Hitt, et al. 47-48.

individuals, typically project managers from 55 organizations and all early CASE adopters, indicates that methods are in place and in use before CASE tools are selected.[1] Many reported that more than one tool was in use, and that the features used most heavily were those which supported the methods most used by the organization. The top three features were data flow diagrams, entity relationship (ER) diagrams, and functional decomposition diagrams. However, several respondents indicated that certain tools were discarded because they did not support the organization's methods.

This lends credence to Larry Constantine's contention that "many CASE vendors do not fully understand the application of existing software design method, but have merely implemented a drawing editor and some simple rule checkers to conform to a methodology description."[2] There are "arbitrary variations in modeling notations implemented in CASE tools, usually because tool implementation is given higher priority than notational clarity."[3] Constantine also points out that few CASE tools include software metrics such as coupling and cohesion as part of the standard design analysis, and "maintains that relatively few CASE vendors use their own methods and tools to support on-going development of their product lines."[4]

Constantine makes several points about what CASE tools need: The tools should not assume top-down design. They should be both more flexible and more integrated; the "models must be interconnected so the relationships between them are governed by explicit design rules."[5] The user interface needs attention; Constantine "suggests...more sophisticated input devices...and better graphical user interfaces similar to those used for mechanical and electrical CAD."[6]

Constantine also addresses CASE and reusability:

"• Define standards and conventions for defining object classes, interfaces, and names

---

1 Sullivan, Patrick J. "A Study of CASE Early Adopters." *C/A/S/E Outlook* 90, No. 1. 18.

2 Constantine, Larry. "What's Wrong with CASE Tools." *C/A/S/E Outlook* 90, No. 1. 12.

3 Ibid.

4 Ibid.

5 Constantine. 13.

6 Ibid.

- Organize libraries with the proper tools and disciplines for efficient accessing and updating
- Have a process for evaluating, refining and polishing modules before addition to a reusable library
- Periodically review and refine the library contents"[1]

Constantine's suggestions for capabilities of CASE tools to support these actions are similar to the capabilities of SPS' Automated Reusable Component System (ARCS) tool, now under development for Army CECOM and NASA.

Constantine also feels that metrics are an important part of CASE. SPS' QUality Evaluation System (QUES) is one step in that direction.

Respondents to the Barton Group survey were asked to name the least satisfying characteristics of the CASE tools employed on their projects. Quality of hardcopy output, text processing capabilities, performance (speed), and interfaces (integration capability) were the four most frequently listed problem areas.[2]

Despite the above-mentioned—or any other—problems, the Barton Group study shows that early adopters of CASE perceived the tools as having a positive impact on project success. However, the researchers clearly indicate that the surveyed users were considered to be sophisticated in the use of CASE tools and structured methods. Less sophisticated users may have more difficulties, and thus a lesser degree of perceived success. The researchers also point out that tools alone are not the solution to achieving better quality and higher productivity:

> [T]he first step to improve application development productivity is to implement and enforce the use of development methods within the organization. Once these methods are implemented, the use of tools provides greater productivity and quality gains. Finally, combining the methods and tools in a [Joint Application Design]-like development environment produces even greater gains. Even at this point, productivity and quality gains remain to be achieved through added experience of the developers and users with this entire process... [3]

---

1 Ibid.

2 Sullivan. 19.

3 Sullivan. 23.

...[P]roductivity is largely a function of the organization and the culture. If the goal is to produce systems more quickly and productively (from a cost perspective), addressing the environmental factors identified by DeMarco and Jones is probably the most effective approach. To produce systems with the same effort but of higher *quality* design and documentation, the best approach seems to be the use and enforcement of Structured Methods with CASE and JAD. Productivity from a value-added perspective is enhanced in this manner. [1]

5.3. Configuration Management

Millradt defines configuration management (CM) as "the ability to identify, manage, and control software and software-related components, such as requirements specifications, documentation, test suites, etc., as they change over time."[2] Forte, et al., take a general look at CM, listing these general requirements:

- version control (multiple versions of source code files)
- configuration control (multiple versions of grouped sets of files)
- system building (e.g., the UNIX *make* utility)
- reporting and query capabilities (interactive on-line querying and printed reports)
- traceability (requirements tracing)
- development process control

Forte, et al., also point out that CM tools must be able to handle the "increased number of machine-readable objects that must be maintained in synchronization within the development environment"[3] as a result of the use of CASE tools. "Many CASE tools guarantee consistency among the representations which they create and use explicitly; however, they do not guarantee consistency with external objects created by other tools, code created by hand, and so on. Consequently, a general-purpose CM facility is needed to serve in the role of a

1 Sullivan. 23.

2 Millradt, Bob. "Configuration Management: How Much Do You Need?" *CASE Outlook 90*, No. 2. 6.

3 Forte, Gene, et al. "Configuration Management Survey." *CASE Outlook 90*, No. 2. 24.

project-wide coordinator that extends across all tools and all phases of the life cycle."[1]

CM is also an important part of any reuse strategy, at both the code and the design levels. "Clearly, we would like a CM facility that not only recognizes that multiple versions exist and makes them accessible, but one that also provides information about the *content and purpose* of the variants."[2]

Millradt points out the importance of selecting the right CM system for the project's needs. "CM systems range in functionality from basic systems that provide fundamental version control, to sophisticated systems that enforce an engineering process within an environment. Identifying your own level of need, and the appropriate system to satisfy your need, is a difficult but vital task to any software engineering project."[3] Not every team and project need the most sophisticated CM tools and, at the lower end, some may not need tools covering all of Forte's general requirements.

A small team, developing small-to-medium-sized applications for only a few targets and/or releases, needs easy-to-use tools that don't interfere with the development and maintenance process already in place. This group has several problems to solve:

- maintain change histories and provide access to any particular version
- modify previous release while developing new ones, and merge the changes into the new product
- handle concurrent access to files without losing any changes
- perform automated, reproducible builds of particular modules

The revision control system (*RCS*) and *make* utilities found on most UNIX platforms provide this basic level of CM. The Code Management System/Module Management System (CMS/MMS) from Digital Equipment offers similar capabilities.

A medium-sized team, developing large applications for several targets and/or releases, has different problems revolving around a need for increased productivity:

---

1 Ibid.

2 Forte, et al. 25.

3 Millradt. 6.

- alleviate the bottleneck in the edit/compile/debug cycle by performing incremental/parallel builds
- use the right versions of the right source modules for a given target, and recreate the build, if necessary
- transparently access modules distributed across networked platforms

Apollo's Domain Software Engineering Environment (DSEE) meets the needs of this group. "Tightly integrated with tne Domain Operating System (Domain/OS), DSEE is one of the few configuration management products that does not require that the data files under its control be stored in a database. Instead, Domain/OS understands the difference between an ordinary file and a DSEE versioned file. When a process...requests a file, if the file is under the control of DSEE, Domain/OS will invoke DSEE to retrieve the file... Furthermore, DSEE uses Domain/OS to support transparent access to remote files across the network..."[1] DSEE also supports branching. Changes in a branch can be merged with the main line of descent or another branch. DSEE can also display a graphical map of all versions and branches of a source file, to help the user visualize the change history. DSEE's best-known feature is performing parallel, distributed builds across a maximum of 20 nodes using a "least-busy" algorithm. DSEE uses a rule-based specification to build the system, allowing beta versions to be assembled base on whether a particular file is present in a particular place.

The Aide-De-Camp Software Management System from Software Maintenance & Development Systems, Inc., provides similar capabilities.

Large teams, building large applications under a formal process need the greatest amount of support from a CM system. Their problems are related to the formal process:

- "...ensure that all (and only) approved changes are included in system builds..."[2]
- prove that contractual requirements have been met
- "...automate and enforce a process [and] tailor the process to the specific needs of the organization..."[3]

---

1 Millradt. 10.

2 Millradt. 8.

3 Ibid.

Softool Corporation's Change and Configuration Control (CCC) helps solve these problems, keeping all files in a central database. The database is divided into systems, each of which is subdivided into configurations. "CCC controls the engineering process by allowing the user to make a copy of a configuration for each state in their software engineering process...and to define transitions between states... Only the changes between a parent configuration and a child configuration are actually stored in the child configuration."[1] A macro facility allows engineers to "create higher-level operations and...tailor CCC to enforce their engineering process..."[2] Security controls allow access restriction to both data and operations/commands on a user-class or individual user basis.

Expertware, Inc., offers Configuration Management Facility, which provides similar capabilities.

Regardless of the development effort's level, it is important that the CM system or tools be available on multiple platforms, under multiple operating systems, if possible. Networks of heterogeneous workstations are becoming the norm in software development; a CM system is not useful if it runs on no more than two platforms in a six-platform network. Forte, et al., provide an extensive survey of various tools and systems that support differing levels of CM and their supported platforms in their article "Configuration Management Survey," which appeared in *CASE Outlook 90*, No. 2.

5.4 Software for Computer-Supported Cooperative Work (CSCW)

Computer-Supported Cooperative Work (CSCW) is a multi-disciplinary field that "looks at how groups work and seeks to discover how technology (especially computers) can help them work."[3] A slightly different definition says that CSCW "deals with the study and development of systems that encourage organizational collaboration."[4] Research in CSCW has resulted in a breed of tools commonly known as "groupware." These tools "support groups of people

---

1 Millradt. 12.

2 Ibid.

3 Ellis, C.A., et al. "Groupware: Some Issues and Experiences." *Communications of the ACM*, Vol. 34, No. 1, January, 1991. 39.

4 Engelbart, Douglas, and Harvey Lehtman. "Working Together." *Byte*, Vol. 13, No. 13, December, 1988. 245.

engaged in a common task (or goal) and that provide an interface to a shared environment."[1]

The capabilities of groupware tools cover a wide range. Message systems—e-mail and bulletin board systems—are the most familiar example. Not all of them are "simple," however. The Information Lens uses a rule-base that "lets users specify rules that automatically file or reroute incoming messages based on their content."[2] MIT accomplished this by partially representing the information content of a message in frames.[3] The Imail systems lets users supplement messages with a script that will execute in the addressee's environment.

There are also a number of multi-user editors available. The ForComment ($195 for a single author, $995 for a 16-author network version) editor is an example of asynchronous editors; users make comments that are attached to the original document, but the original text is unchanged. Shared Book and Quilt are examples of real-time group editors. Multiple users can have concurrent read access to any segment of the document, but only one writer is permitted per segment. The Mercury editor, aimed at programmers, takes the concept a step further; it notifies users when changes made by other programmers will have an impact on their code.

Group decision support support systems (GDSSs) and electronic meeting rooms are another form of groupware. CompuServe's real-time forum gatherings are examples of electronic meeting rooms. GDSSs enhance the meeting by providing decision-making aids such as alternate ranking and voting, and aids for idea generation or issue analysis.[4] Argnoter, an experimental system at Xerox PARC, uses an AI technique (non-monotonic reasoning) to support a group evaluating proposals. "Individuals suggest arguments for or against each proposal and compare proposals...[stating] their underlying assumptions and evaluation criteria... [T]he system quickly evaluates proposals under different sets of assumptions."[5]

---

1 Ellis, et al. 40.

2 Ellis, et al. 42.

3 Crowston, Kevin, and Thomas W. Malone. "Intelligent Software Agents." *Byte*, Vol. 13, No. 13, December, 1988. 268.

4 Ellis, et al. 42.

5 Crowston and Malone. 272.

Groupware is also available for several types of conferencing: real-time computer conferencing, computer teleconferencing, and desktop conferencing. The Caucus system ($350 for PC version) also runs on workstations and mainframes) is one example of a real-time conferencing system. Running on an 80386 machine with multiple serial ports, Caucus "can handle up to 16 simultaneous users, which equates to a population of several hundred occasional callers. It's this capability and the ability to set separate discussions by topic that set real conferencing systems apart from bulletin board systems, with which they are often confused."[1]

"The most familiar examples of teleconferencing are conference calls and video conferencing. Teleconferencing tends to be awkward, requiring special rooms and sometimes trained operators. Newer systems provide workstation-based interfaces to a conference and make the process more accessible. Xerox, for example, established an audio/vide link between Portland and Palo Alto. Most video interactions occurred between large Commons areas at each site, but project members could also access video channels through their office workstations."[2]

"A third type of computer-supported conferencing combines the advantages of teleconferencing and real-time conferencing while mitigating their drawbacks. Dubbed *desktop conferencing*, this method still uses the workstation as the conference interface, but it also runs applications shared by the participants. Modern desktop conferencing systems support multiple video windows workstation. This allows display of dynamic views of information, and dynamic video images of participants."[3] Rapport is an example of this type of system. Rapport "supports various forms of interaction, from simple telephone-like conversations to multi-party shared-display interaction."[4]

Some systems use artificial intelligence techniques to generate their own "participants" in a group, or provide users with "assistants" to make their work easier. MCC has developed "a groupware toolkit that includes an agent named Liza. One of the tools in the toolkit displays the pictures and locations of all session participants. When Liza joins a session, a picture of an intelligent-looking

---

1 Opper, Susanna. "A Groupware Toolbox." *Byte*, Vol. 13, No. 13, December, 1988. 278.

2 Ellis, et al. 43.

3 Ibid.

4 Ibid.

android is also displayed, indicating to the group that Liza is *participating*."[1] Liza represents a set of rules that are active when she is present. The rules monitor session activity and suggest content or form changes at appropriate times.

The Carleton Office Knowledge Expert System (COKES), an experimental system at Carleton University, uses a rule-based frame system to provide each user with an assistant "that can provide details about office procedures, other staff members, or available resources. Other servers store shared organizational knowledge. These systems can communicate with each other, for example, to request or supply information. New managers could use the system to find which reports they need to write, when they are due, and who should get a copy. The 'assistant' could then help collect the information necessary for each report."[2]

Coordination systems are used to integrate individual work efforts toward the completion of some larger goal. "Forms-oriented models typically focus on the routing of documents (forms) in organizational procedures. These systems address coordination by explicitly modeling organizational activity as fixed processes... Procedure-oriented models view organizational procedures as programmable processes... This approach was first applied to coordination problems in the software process domain... Conversation-oriented models are based on the observation that people coordinate their activities via their conversation. The underlying theoretical basis for many systems embracing the conversation model is speech act theory..."[3]

---

1 Ibid.

2 Crowston and Malone. 270.

3 Ellis, et al. 43-44.

LIfE (Linked Information Environment) is a set of four products from Motorola based on the forms model. "LIfE has the following features:

- Forms produces electronic forms that look like paper ones already in use... LIfE
- Works provides high-end data entry for back-office activities. LIfE
- Plans offers high-speed, high-capacity workgroup spreadsheets. In addition, LIfE
- Lines is a workgroup E-mail system..."[1]

Polymer is a procedure-oriented system developed at University of Massachusetts. "Polymer uses planning ideas to help carry out the steps in a complex office procedure. When you state a goal, the system creates a plan—that is, a sequence of actions to achieve the goal. As each action is carried out (e.g., by assigning it to someone to do), the system checks to see if it worked as expected. If something goes wrong, the system can reevaluate the situation and, if necessary, make a new plan..."[2]

The Coordinator was one of the first commercial products to implement the conversation-oriented model. The system "is based on a set of speech acts (i.e., requests, promises, etc.) and contains a model of legal conversational moves (e.g., a request has to be issued before a promise can be made). As users make conversational moves, typically through electronic mail, the system tracks the requests and commitments."[3, 4]

There is certainly overlap between the categories listed here. Many expect that, as the demand for integrated systems increases, there will be more merging of the functionalities.

---

1 Opper. 282.

2 Crowston and Malone. 270.

3 Ellis, et al. 44.

4 Winograd, Terry. sidebar to "Where the Action Is." *Byte*, Vol. 13, No. 13, December, 1988. 256D.

6. Conclusions.

This final section offers our opinion on the usability, availability, and viability of the various technologies overviewed in this document.

6.1. Storage devices.

Magnetic media in general is a very stable technology, and is not likely to be entirely eclipsed by other media, such as optical. Disk drives should continue the current trend toward greater capacity and throughput. The smaller form factors will become more popular, partly due to the spread of notebook-type computers. Tape will likely continue to be the favorite form of back-up and archival media. The newer, smaller form factors—8mm and DAT—will likely gain more adherents.

Optical media should continue to grow in popularity. CD-ROM is relatively stable, and has been since the introduction of the High Sierra format standard. More and more software and/or data is being distributed this way, and some vendors—notably, Apple—discount the prices of CD versions of their products. There could be improvements in access time and throughput as R&D contributes to lighter, less bulky mechanical components in the read arm.

WORM drives, with their lack of rewrite capability, may end up a niche technology, popular primarily with those industries that need unalterable audit trails; examples are the financial, insurance, and medical professions. The lack of format standards will continue to prevent interchange of cartridges between drives from different manufacturers. This may also prevent some companies from buying; they won't want to become locked into a single manufacturer's product and pricing/support strategy. As with CD-ROM, R&D in smaller components may improve the access time and throughput.

EO technology is still in flux. Magneto-optical drives are available now, and format standards are in place. However, R&D into alternate methods, such as phase-change and dye-polymer media, could put competing technologies on the market. Standards may permit cartridges to be used on the drives of different manufacturers, but they certainly will not permit magneto-optical cartridges to be read in drives that use dye-polymer cartridges.

6.2. Input devices.

Although there is a great deal of R&D in alternate input technologies, the keyboard will be the mainstay for a number of years to come. The mouse—and its alter-ego, the trackball—will continue to be the most popular pointing device, although some users may move to graphics tablets. That move depends on the price of the tablets, currently much higher than the cost of a mouse, and whether the users have other uses for the tablet. Graphics designers and CAD operators will probably continue to be the largest consumers of graphics tablets. All of these technologies are stable, although it is possible that R&D could create mouses and graphics tablets with resolutions higher than those currently available. The pen-based computer technology is currently testing the market to see if users are ripe for a new input device.

Voice input is getting plenty of press, and thus plenty of attention from the marketplace. As prices come down and capabilities increase, these devices may come into more common use. For now, they are a significant first-cut technology, with a great deal of R&D still going on. Expect to see great strides in audio and voice technology in the next 5-7 years.

Scanners are growing in popularity, especially since more capabilities are appearing at lower prices. The primary users will be those who need to capture images and include them in their data/documents, and those who will benefit from OCR applications. Scanner technology is stable, and OCR is relatively stable. However, R&D in neural nets and other areas of artificial intelligence could bring great improvement in OCR's recognition capabilities. At present, the error rate keeps OCR from being any faster than typing.

6.3. Output devices.

As a whole, the technology of the current generation of output devices is stable. However, R&D can bring many improvements to them all. It remains to be seen whether the improvements will make the current devices obsolete.

We could see improvements in display quality at all price levels; certainly the displays on most users' desktops would have once cost as much as or more than an entire computer system; color is available at what monochrome once cost. With the growth of imaging applications, the market will be looking for higher resolution. Larger displays are likely to become more common in the office market, as they have in the workstation market.

Laser printers have not completely replaced other printer technologies, but they have become the "standard" against which others are compared. Coming improvements include higher resolution and less expensive color. Prices are already dropping, and may continue to do so. Larger companies with a large volume of printing will be watching the developments with Canon's color laser printer/copier.

Imagesetters and film recorders will probably remain the province of high-volume users, with others willing to make use of a service bureau. However, that could change if costs and ease-of-use should drop down to the high end of laser printers.

Plotters are likely to remain primarily CAD/CAE tools; their throughput is a disadvantage for other types of graphics producers, who also demand more versatility in their output devices. The newer technologies may drop somewhat in cost, but budget-conscious, smaller operations will keep the market for pen-based and electrostatic plotters alive and well.

The technology for getting digital images onto videotape is stable, but remains fairly expensive. As desktop video takes off, the demand for equipment could drive prices down, as is already happening with some of the video add-on boards. As always, though, the more sophisticated features you want, the more you have to pay.

6.4. Operating Systems.

There are conflicting opinions about the future of operating systems. UNIX has been around for years, and many pundits expect it will stay on for many more. However, there is much research into alternatives, some based on UNIX but offering desired capabilities, such as greater security. Distributed operating systems will likely become more commonly used, due to the growth of distributed systems. UNIX will not disappear, but it will be an option, rather than the only choice.

If parallel computing becomes more than a niche, parallel operating systems may take up a larger share of the market.

6.5. Repositories.

The relational database seems to be the most common in use, according to the products on the market. Object-oriented databases may be the coming thing, given the move toward object-oriented programming and modeling techniques; both need storage and management for objects that do not fit well into relational structures. Relational is not likely to disappear soon, and in fact may be used in conjunction with object-oriented forms of storage. The work being done in federated databases is an indicator of this.

6.6. User Interfaces.

Applications are becoming more complicated, while there are more and more users, both sophisticated and otherwise. The sophisticated users demand easier-to-use interfaces so as to increase their productivity, and to make complex operations easier to control. Less sophisticated users also want easier-to-use interfaces, so as to make computers less intimidating and to shorten their learning curve, as well as make complex operations easier to understand. The graphical interface seems to be here to stay. The questions still being worked on are ones of detail rather than of concept. There will probably continue to be several "standards" for graphical interfaces, but object-oriented tools developers will be able to create a single interface and quickly port it to any desired platform and GUI. The only real concern is whether every GUI vendor will wind up in court, paying legal fees, fines, and royalties to Apple for using icons and movable windows, thus raising costs to the end user.

6.7. Hypertext/Hypermedia.

The current generation of hypertext/hypermedia tools is fairly stable, and offers many usable capabilities. R&D continues, however. The first result will likely be better interfaces among different media sources, and tools that are easier-to-use. The main problem will continue to be design and implementation. Hyper-whatever is not the panacea that some people want it to be or that some vendors claim it is. For some problem domains, and with good up-front design, hyper-technology will solve problems and make systems easier to manipulate. For others, it will be the wrong choice.

6.8. Access Control/DIS Security.

A great deal of R&D has been and is being done in this area. That will continue as long as one group wants to protect information and another wants to get at it. The complexity of modern distributed systems, especially those that are geographically distributed overs hundreds and thousands of miles, make it a difficult problem to solve. There are those who believe the only guarantee of security is to stay off the network. The current generation of commonly-used operating systems appear to be only as secure as their users; not much is done to enforce security measures. File servers such as Andrew could add some enforced security to an existing operating system such as UNIX. Operating systems that are under development seem to have security in mind from the start, and offer enforced security features as a matter of course. This fact may be a major reason for some companies choosing an OS other than vanilla UNIX.

6.9. CAD/CAE.

As design tools, CAD/CAE packages are stable. More sophisticated features, such as solid modeling or interfaces to spreadsheets for estimating and billing, continue to appear. Features once found on more expensive workstation tools continue to migrate down to the PC tools, and the price/feature ratio continues to drop. Research in virtual reality has provided some benefit to this market, as well, with some packages now offering "walk-through" and load simulation capabilities.

CAD/CAE will benefit by greater connectivity between different types of software tools—such as the afore-mentioned interface to spreadsheets, for example. As various committees come up with standards for interfaces and data format, we may see a single system handling design, simulation and testing, analysis, estimating and billing.

6.10. CASE.

R&D in CASE tools continues. Most such tools are used for analysis, for design, or to develop requirements. Unfortunately, most support a single method, while most organizations use several for different purposes. Another problem is that many tools provide no metrics or cost-estimating support. Rather than develop an all-purpose tool, however, the industry is more likely to continue to support efforts such as this, allowing them to use any tool in their arsenal from a common point, with access to all the design data from any tool.

The CASE industry has accomplished a great deal during its first decade, but there is still much more to be done. In, fact, the more software engineers and systems professionals become dependent on CASE, the greater the demands on the technology. [1]

6.11. Configuration Management.

Configuration management tools are somewhat in flux right now. Systems are becoming more complex, and there is more to CM than just keeping a copy of the latest file. It is difficult enough to track thousands of software components through all their versions for a single platform; multiply the problem times X platforms and Y supported versions on each of them. CM tools need visibility

---

1 Norman, Ronald. "Automating the Software Development Process." *Communications of the ACM*. Volume 35. No. 4. April 1992. p. 27.

into all the sources of software objects, and throughout the life cycle. Again, rather than build a single all-encompassing CM megalith, it seems more likely that "environments" such as this one will incorporate several different types of CM tools that will share a central data repository.

6.12. Groupware.

Groupware, as software for computer-supported cooperative work is known, is the latest buzz-word. For the most part, it is still an R&D field, despite the availability of commercial products. In some people's minds, groupware means that you have multi-media access to any- and everybody and their data from your workstation. In reality, the commercial products are more limited in scope. R&D systems offer more of those fantasy features, but given the communication bandwidth needed for combined audio/video/data transmission, such systems are not likely to see common use in the near future. However, collaborative writing tools and multiple-person editors, group conferencing and decision-making systems, and the like should continue to penetrate the market.

*U.S. GOVERNMENT PRINTING OFFICE

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.