Chapter 9 Market Porting Scenarios

This chapter is divided into market segments and serves as a guide for vendors . The sections address some commonalities in the market segment, and gives comments Sun has obtained from vendors who have recently ported to the Sun workstation.

9.1 Electronic Design and Automation (EDA)

Key issues: graphics performance, many transactions, large databases, networking, floating point

The EDA segment covers a very wide range of applications and system requirements. Key issues are graphics performance, a high level of interactive operations (transactions), very large databases, networking, and floating-point performance.

Most EDA graphics operations involve 2-D integer databases with high levels of precision. Applications often use full 32-bit integer representation, but use it to represent fractional values. When this data is rendered to the screen, applications are prepared to do most of the transformations, including clipping, scaling, and rotation. This means that most applications cannot readily take advantage of some of the more advanced graphics features because the graphics software is simply tied very closely to the structure of the internal database, which is usually proprietary. Many vendors feel they get better performance overall (throughput) this way because they can fully optimize the database and graphics operations simultaneously. As a result, most graphics-intensive EDA applications use the lowest level of graphics interface offered.

Another characteristic of EDA applications is that the data is very Òregular.ÓAn integrated circuit or printed circuit board is represented using thousands of orthogonal polygons and rectangles.

Performance is critical in the use of dragging and rubberbanding, which is quite prevalent in EDA. Vendors often use the SunView ÒnotifierÓto correctly route the various input events.

EDA requires a high degree of user interactivity, and users are typically operating on databases of 100,000 - 1,000,000 entities. In the design process, they separate out very small subsets of this data to manipulate. Several applications have special ÒfiltersÓoperating on their databases. The use of commercially available databases is not yet significant due to the difficulty in making the transition from their custom, proprietary data bases and their size requirements. Increased CPU power has a very strong effect on interactive graphics performance because the navigation through databases uses many more CPU cycles than the actual rendering of the data.

Circuit analysis applications tend to be floating-point intensive, and are usually written in FORTRAN. These applications are highly complex; a ÒrunÓmay take hundreds of CPU hours, so compiler optimization is an issue.

large mixes of workstations, PCs, VAXes, and mainframes. Therefore, the use of networking products such as the Network File System, SunLink DNI, and the SunLink 3270 products is quite prevalent. Tight integration is not usually necessary.

9.2 Mechanical CAD (MCAD)

Key issues: data alignment, graphics performance, floating point

Long software quality assurance and testing processes are typical for the MCAD vendors because they comply with government specifications for certain analytical procedures. Even a Òsimple recompileÓrequires a minimum of three to six months of testing. For this reason, any information provided to the MCAD vendors that allows for fewer ports over time is important.

While MCAD applications tend to use a variety of features of the workstation, there is a set of functions that almost all applications use. Therefore, various MCAD applications tend to have common problems when they are ported. Some of these problems include:

¥ Differences in byte ordering and bit ordering between various architectures

¥ Differences in the alignment of bytes words and double precision words across architectures

¥ Structure alignments and the internal padding of structures

¥ Various subroutine calling conventions that affect the way parameters are passed on the stack. One should pay particular attention to passing structures and mismatched parameter types

¥ Bitwise operations such as arithmetic and logical shifts

¥ Initialization of variables (depending on the language and compiler uninitialized variables may be either set to zero or contain arbitrary values)

¥ Problems related to source-code control such as, dependencies, and the residence of source, object, and executable files. In this case, SCCS has to be used to provide the same structure as on the previous host.

In addition to the topics covered in this document, other Sun documents may be useful to your MCAD porting effort:

Hardware

¥ Sun-2, Sun-3, Sun-4, Sun386*i* hardware manuals Ñ particularly sections explaining alignments, precision, and subroutine calling conventions

¥ GPIC (graphics processor interface card) manual Software

- Programming Utilities (C language tools, Make, and SCCS)
- *SunView ProgrammerÕs Guide*
- SunView 2 directions statement
- *Porting Software to SPARC Systems*

9.3 Computer Integrated Manufacturing (CIM)

Key issues: graphics performance, networking, languages, realtime UNIX, standards

The CIM market covers a wide range of technical requirements, including graphics performance, networking, languages, realtime UNIX and standards. For additional information, please refer to the ECAD and MCAD sections of this chapter. The Catalyst Third Party program offers solutions for realtime UNIX.

The graphics requirements for CIM applications fall into two categories. One is fairly simple; for example, 2-D and 3-D assembly drawings to be viewed but probably not edited. These drawings are often coming from a remote CAD section over a network. The ability to rapidly display various drawings is important. Depending on the interface from the CAD system, the graphics interface may be highly optimized; for example, using the lowest level graphics interface.

Other CIM graphics requirements may be animation-like and may require the lowest possible interface with highest performance to simulate robotic movement. Networking issues in CIM revolve around the use of standards on the factory floor, particularly MAP/TOP/OSI.

For additional information, refer to the SunLink section in Chapter 2.

9.4 Earth Resources

Key issues: graphics performance, networking, floating point, large databases, languages

Petroleum applications, such as seismic interpretation, processing, or modeling; well log analysis; contouring; reservoir simulation; fluid flow analysis; and expert systems, require responsive graphics and an intuitive, powerful user interface. Users want to view the information without waiting. The less time that professionals must wait for an answer or a view, the more they can interactively probe the data and receive responses to queries Ñ many times faster than on a mainframe. Similarly, if the application does not present an intuitive, menu-driven, ÒclickÓ and go interface, the learning curve is prohibitively flat. We are noticing oil customers valuing the simplicity, consistency, and efficiency of the user interface. A standard user interface, such as OPEN LOOK, dramatically increases user ability and desire to use new applications and provides a consistent style of operation across all software.

3-D subsurface models, rapid signal analysis on seismic panels, and other high-performance graphics are effectively driven by a balanced system of graphics accelerators, raw compute performance, and system I/O. Many earth resources software vendors will support the OPEN LOOK user interface. It provides the simplicity and consistent style for all corporate applications. For more information on NeWS, X.11, or SunView toolkits for this environment, please refer to Chapter 5.

Sun polled several vendors in the earth resources market segment and received the following feedback:

Vendor A had a FORTRAN application for image processing that he ported to the Sun. To do this, he used raster operation calls to load pictures into a SunView canvas. (SunView is the optimal choice for raster graphics.)

Vendor B had a 2-D graphics package that required a dual-headed monitor. With the help of SunÕs technical support, they ordered an OEM special product from Sun that allows two CG2 or CG3 color boards in a 12-slot VME system. CG5 color boards do not need any special configuration, because the user can configure the boards for dual-headed monitor use. This application used GKS heavily, and the vendor is being encouraged to complete the porting of whatÕs left of the CGI code to GKS. The application is primarily written in FORTRAN with some C.

9.5 Graphics Information Systems (GIS) and Mapping

Key issues: graphics performance, large databases, floating point, networking, languages

The mapping, GIS, and remote sensing markets are price/performance sensitive, while also being solutions-driven. The functional purpose of mapping application is to represent spatial data by manipulating them into easily understood graphic formats for topographic maps, profile and cross sections, natural-resource and land-use maps, and hybrid (derived) maps consisting of several different map overlays. DEC systems and IBM/PCs are popular in this market, but it is a compute-intensive market well suited to workstations. Cost conscious users are beginning to demand that software vendors offer their products on workstations. This market is typified by four factors that strongly influence porting issues:

¥ Price/performance Ñ To answer end-user demand for workstation power, software vendors typically port to UNIX from VMS or MS/DOS

¥ Heterogeneous computing environments Ñ Typical computing environments are composed of a variety of vendor platforms that must be utilized in the network and not isolated ment, typically requires the integration of several software packages to accomplish end-user objectives

- Peripheral-intensive Ñ The application requires a variety of input devices, output devices, and large disk and tape capabilities

One vendor found several ÒbugsÓ in his code that he could Òget away withÓ in VMS or Prime OS, but he found UNIX to be more strict. For example,

- In passing an array to another routine, he had to declare that array in the second routine
- In passing double-precision numbers to real numbers, the VAX would resolve the differences, UNIX did not
- In using SunÕs *dbxtool*, it would have been helpful to have known earlier about the ability to turn off case-sensitivity in order to find FORTRAN variables, which are not case-sensitive

FORTRAN code is often written in upper case; *dbxtool* only recognizes lower-case variables.

This vendor said he made good use of SunÕs online manual facilities and sample programs.

9.6 Finance

Key issues: networking, compute servers, window system integration

In the financial market segment, customers drive the platform choice, meaning software vendors must provide their products on a wide variety of hardware platforms. Finance vendors want portability between different vendorsÕ machines; this portability requires industry standards. Finance vendors look to Sun for options that offer the least risk of being tied to particular hardware and the most flexibility.

The financial marketplace is heterogeneous, and end-users need to access data from all types of systems (IBM mainframes, System 36/38s, PCs, DEC systems, fault-tolerant systems, and others). SunÕs heterogeneous networking products, both ONC and the SunLink family, are important. Peer-to-peer networking among all types and sizes of systems is critical as the data could reside anywhere. In addition to heterogeneous networking, financial vendors choose Sun workstations for their superior display management of realtime data and cost-effective compute power.

A typical example of a financial application is tying in to a ticker service over RS-232, then using X11/NeWS to display that data graphically over a network to several heterogeneous clients (See Figure 9-1.)

(Remember that a X11/NeWS server controls a display.) This data is now available to all X11/NeWS servers. X11/NeWS servers communicate over Ethernet to a Sun or non-Sun client

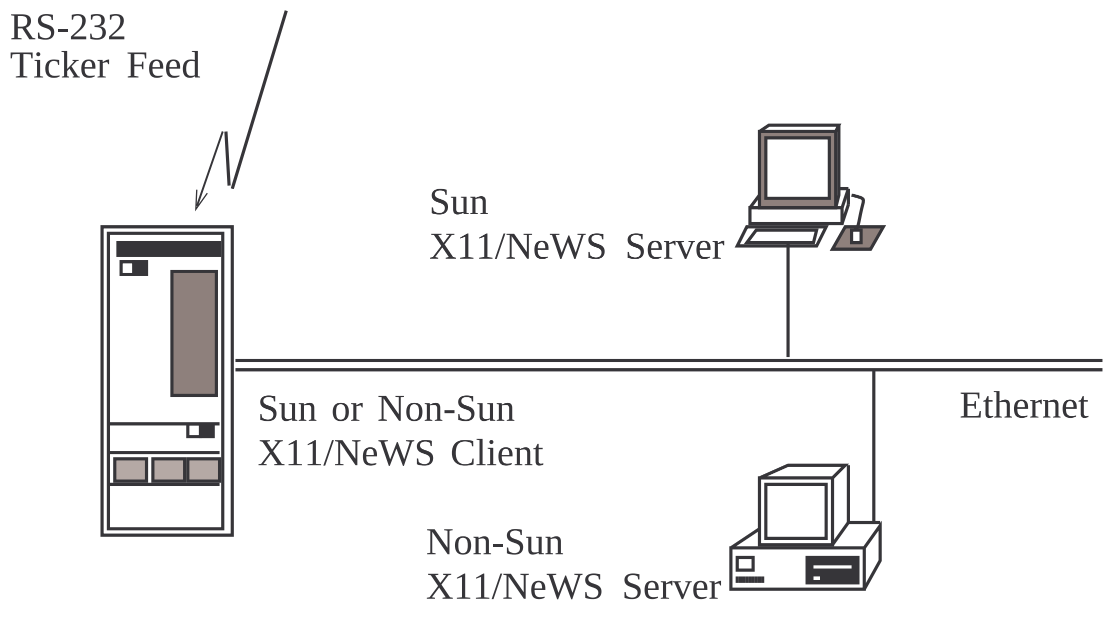

Figure 9-1. Financial Application Example

and display graphical data at a trader's desk. Traders can send buy/sell orders while processing analytical computations that display output (graphs, charts, etc.) on both the X11/NeWS server and client.

In this application, the availability of X11/NeWS on various platforms means an application can be used from any node on the network that uses X11/NeWS; it is not necessary to port that application. X11/NeWS takes care of the hardware, device, and resolution-specific characteristics of the platform, so the application doesn't need to know the end-user's location or what type of machine is used.

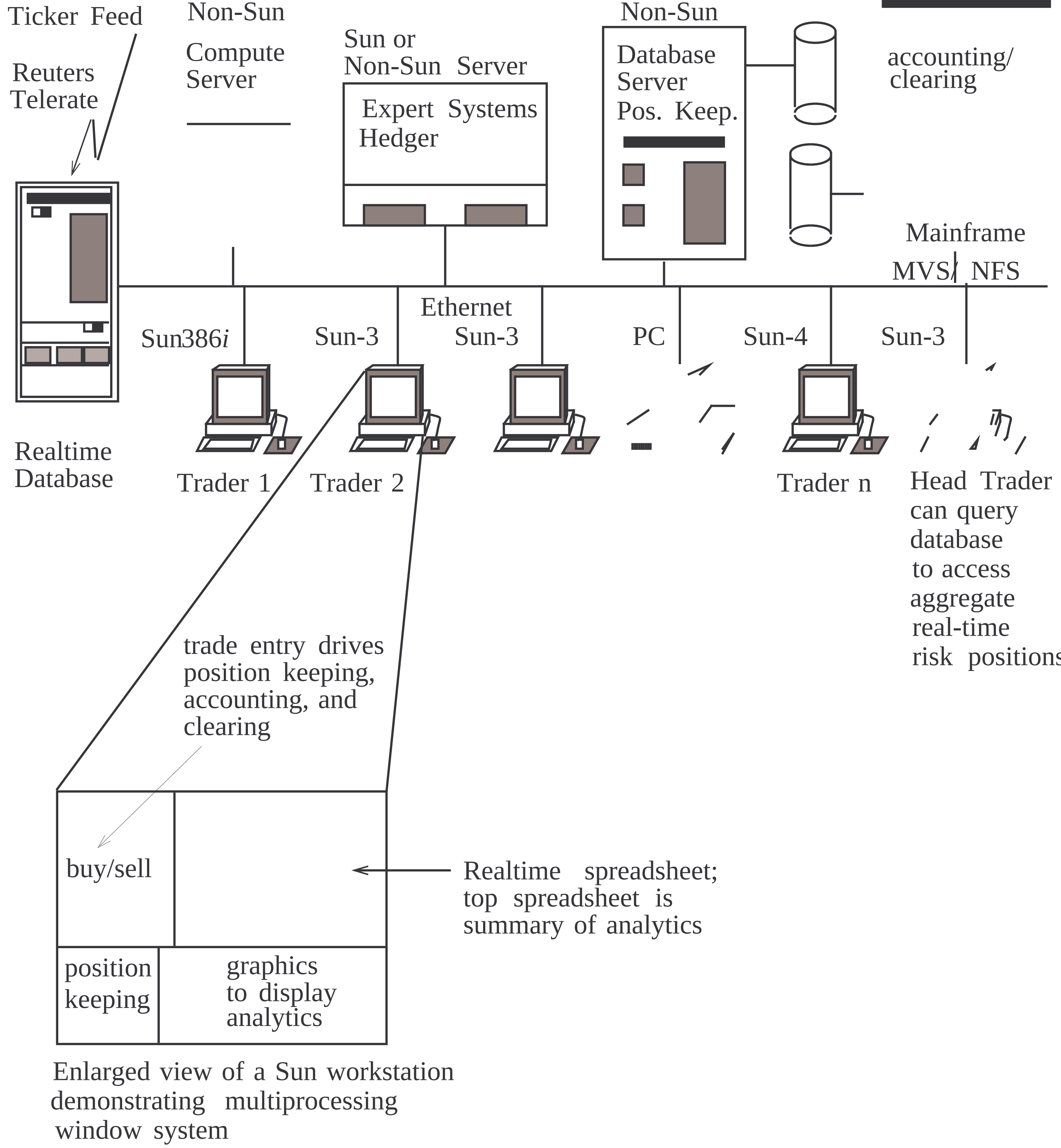

Figure 9-2. Typical Trading Floor Solution

Figure 9-2 shows an example of a typical trading floor. An enlarged Sun display shows multiple applications running concurrently in multiple windows. This represents a significant advantage over PCs that can run only one application at a time. Additionally you can also run analysis on the data received, servers. With the recent announcements of NFS support for MVS and VM systems, traders workgroups can now be easily integrated into existing back office systems. For example, a trade entered in the buy-sell window updates the accounting and clearing systems on the mainframe as well as a shared database. This allows the head trader to look at aggregate risk positions, and makes this information available to expert systems that can recommend appropriate hedges.

9.7 Add-On Hardware/Input and Output Peripherals

Key issues: device drivers, hardware interfaces

For input/output peripherals, vendors are required to provide both hardware and software interface. Examples of digital input peripherals are digitizers, flat-bed scanners, and slide scanners. Cameras with frame grabbers convert analog input to digital. A typical digital interface for input peripherals could be RS-232C, Centronics Parallel, GPIB, SCSI, or Custom VME. Output peripherals consist of printers/plotters, film recorders; display devices; display emulators; projection devices; and VCR/TV with a digital, analog, video, video converter, or software interface. For output devices, the digital interface could be RS-232C, Centronics Parallel, GPIB, SCSI, Custom VME Board, DR11-W, or Versatec.

Note: Developing an SCSI driver is not currently recommended. However, Sun Consulting is a good source for such a development. As an example, a printer with an RS-232C connection can be ported to all Sun platforms with minimal development effort, when the UNIX driver is offered by the vendor.

9.8 Graphics and Imaging

Key issues: device drivers, hardware interfaces

The Graphics and Imaging marketplace is comprised of 1) Graphics, which includes charting and graphing, graphics subsystems, standards, libraries, and tools, 2) art, animation, and video, and 3) imaging. In the area of graphics, charting and graphing and standards, libraries, and tools are software products and developers find it generally simple to port to Sun platform. Graphics subsystems consist of both hardware and software that need to be ported to Sun platform and the vendors need to provide and develop the interface to Sun. They need to determine their need for any additional card slots or VME based Sun Workstations. As an example, some of these Graphics subsystem vendors have provided 24-bit frame buffers for Sun and they have been very successful in porting to Sun platform. In some cases they have developed their product compatible with Sun VMEbus, with the use of Sun's pixrect library.

Key issues: window system integration, networking, operating system

Office automation application categories include word processors, spreadsheets, databases, personal information managers, outliners, business graphics, decision support systems, accounting software, manufacturing tools, document search and retrieval aids. This is a very broad category where vendors can be porting to Sun from a wide range of bases Ñ from PCs to mainframes.

SunÕs NeWS windowing system fits well with many office automation applications. Since the PostScript Language model is the standard interface to printers as well as to NeWS, it is easier to write WYSIWYG applications using NeWS than with other windowing systems.

Many vendors coming from the DOS world find that their programs run in a DOS window on the Sun386*i* without modification. However, taking full advantage of SunÕs multitasking UNIX operating system and window system environment often requires restructuring the application.

One vendor porting a Macintosh program split off the user interface section of the application from the computational part. By doing that, much of the code remained the same between the Macintosh and the Sun versions of the program, and it was easier to maintain both versions of the program.

Another vendor ported a 150,000 line program written in FORTRAN 66 to a Sun-3 and to a Sun386*i*. He noted the following:

- In calling a C routine from FORTRAN, C represents the subroutine as Òa_Ó rather than Òa.Ó Sun and HP are similar in this respect, while DEC (Ultrix) and Apollo represent the routine as Òa.Ó This is because FORTRAN and C generate the same object code, therefore it is necessary to distinguish the FORTRAN ÒaÓ from the C ÒaÓ.
- Byte ordering differences for the Sun386*i* were reconciled and the program was ported in about three days. This was a relatively simple FORTRAN 66 application without character declaration statements or FORTRAN I/O.

Another vendor did his port to Sun from System V code. In this case, it was necessary to change the directory access code that was hardcoded into his application. That was easily done with a Ò*scandir*Ó command. He was then able to compile with the System V library. Once TTY setting calls were changed, he compiled with the Berkeley 4.2 libraries in Sun OS 3.2. This vendor noted he had also ported to Pyramid, which was more difficult without the System V library routines.

Key issues: window system integration, databases, operating system

The CASE marketplace is comprised of three segments, commercial (DP/MIS), defense/aerospace/industrial, and scientific/engineering. Porting issues frequently cross market segments although they differ widely by the type of CASE application. Software development efforts frequently require software that runs on multiple hardware architectures networked together. In order to make the job of porting to multiple systems easier, many of Sun's Catalyst vendors are driving the development of standards.

It is difficult to generalize among CASE products. In order to demonstrate the range of issues that may need to be addressed, the porting examples in this section fall into three categories: compilers, analysis and design (front-end tools), and realtime OS/in-circuit emulators.

Compiler vendors are primarily concerned with the machine instruction sets associated with the SPARC RISC, 680X0 family and 80X86 architectures; the coprocessors that work with them, whether they be 68881, floating-point Weitek or other chips; the output object format; and byte ordering/ data alignment issues. Porting from one hardware vendor to another when both use the same chips may take as little as two days of work. Porting a compiler from one CPU architecture to another may take six months or longer. A port using Sun's assemblers/linkers and optimizers, especially the peephole optimizer, saves a considerable amount of porting time.

CASE developers of Analysis and Design (front-end) tools, on the other hand, have completed ports to a new CPU architecture in a day or two. However, if a new window system, user interface, compiler (substitute a Sun compiler for a non-compatible non-Sun compiler), and database are included with the port to the Sun, it will take considerably longer. Porting between Sun architectures is generally simple for front-end vendors when these other elements are not issues.

Realtime OS developers and developers of in-circuit emulators have brought up totally different concerns. They are primarily interested in understanding the impacts on system software of adding more hardware devices to the Sun. Refer to Section 7.4 for characteristics of Sun bus architectures, how boards are able to communicate with each other and Sun's policies regarding porting devices to the SCSI. Sun device driver issues are discussed in Section 7.5. Hardware changes and operating system upgrades may have major impacts on development schedules.

Other issues that may concern at least one CASE vendor include:

- Hardware (CPU, coprocessors, other boards, bus, memory, disk space) communication, priorities, permissions, directories, open/close system calls, environmental variables used, shared libraries and shared data)

- Languages (i.e., tools written in C are more portable than those written in Pascal whose implementation varies widely from vendor to vendor)
- Databases
- Windowing systems (CASE applications on Sun workstations typically run on SunWindows using SunView, although some run on X and NeWS.

9.11 Artificial Intelligence

There are more than 75 Artificial Intelligence (AI) vendors in SunÕs Catalyst program. A few of them specialize in hardware, but the majority provides software.

Key issues: window system integration, Lisp

Hardware AI vendors mainly provide special add-on boards for specific functions such as a Prolog processor.

Software vendors provide languages for AI application development, environments for AI application development, specialized image processing, diagnostic applications, and realtime reasoning. Most of them are written in Lisp, C, and Prolog; they have complex data structures, graphics-oriented user interface, and capabilities of handling a large amount of data.

Of all the porting concerns AI vendors have, the windowing interface is the most important. Some packages make use of Sun Core; some provide an interface to SunView. X11 window-based software is also common. X11 has been endorsed by many major computer vendors. X11 is highly portable across computer platforms, is network-based, and has a device-independent graphic programming interface. However, since X11 is public domain software, Sun does not support it. X11 window source for all Sun platforms (including Sun 386*i)* can be obtained from MIT. For details of future window systems on Suns, please read the specifications of OPEN LOOK and the X11/NeWS merge.

For porting software from SunOS 3.X to 4.0, the following 4.0 changes may affect the design of a software and the port: the introduction of shared libraries (shared code in memory, which causes less paging), the redesign of virtual memory (e.g., use of Ò*mmap*Ó to map files to user address space), greater compatibility with the System V Interface Definition (SVID), and support of NETdisk. Other significant differences are: the introduction of the automounter and the loopback filesystem, and changes in utilities and the layout of the filesystem.

Software written in C and making use of the standard library differences in hardware architectures. Changes in machine instruction set, processor, and memory management may have drastic effects.

Two porting problems are mentioned below for reference.

One vendorÕs software is a language for AI application development. The port involved changing the executable files dynamically (copying header information from the running executable, modifying the headers to account for the new data size, and then saving a running program in a form that can be executed again). On SunOS 4.0, the default use of shared libraries is incompatible with environment saves; therefore, the vendor had to specify the -Bstatic (static in place of dynamic linking) in the compilation of the software.

Another vendor discovered that SunOS 3.2 was more forgiving about wrong information in the header of an a.out file than SunOS 4.0. In this case the given data segment size was too small. Under SunOS 3.2 the file was read and the program executed anyway; under SunOS 4.0 the program ran successfully on a local disk, but generated a Òsegmentation violationÓ error message when read remotely.

Some packages require high-computing power and a large swap space to run. Minor adjustments like increasing swap space, or making a custom kernel (increasing MAXUSER) can improve the environment under which the application runs. A vendor can take these steps to fine tune the port.

To run Sun Common Lisp 3.0, you need at least 8 Mbytes of physical memory and 35 Mbytes of swap space. However, we recommend 12 Mbytes of physical memory and 60 Mbytes of swap space. The nature of your Lisp application may also cause you to need additional memory or swap space. Most Lisp applications tend to be big (5 - 12 Mbytes); therefore, a 141Mbytes disk is the minimum requirement to fit a kernel, swap space, and /usr space.

Note: Most AI vendors use the Emacs editor besides vi. GNU Emacs is running under SunOS 4.0.