6.1 Portability Issues

Sun’s graphics libraries are suited for many different applications

Sun workstations offer a wide range of graphics software libraries for software development. For a list of graphics software interfaces and standards, see Figure 6-1. For software portability and software-investment protection over heterogeneous graphics computer systems, ANS/IS-approved graphics standards such as the Programmer’s Hierarchical Interactive Graphics System (PHIGS), Computer Graphics Metafile (CGM), Graphical Kernel System (GKS), ACM Core System, and the working draft ANSI standard Computer Graphics Interface (CGI) can be used as a graphics software platform. All of Sun’s graphics standards software take advantage of graphics accelerators and floating-point accelerators. However, if the

Workstation	Graphics Software Interface				Graphics Standards				
	Pixrect/ Pixwin	GPSI	CG8	TAAC-1	PostScript Language	SunCore	SunCGI	SunGKS	SunPHIGS
Sun-386*i*/150	● 4.0				○		◕ 4.0	◕ 4.0	○
Sun-386*i*/250	● 4.0				○		◕ 4.0	◕ 4.0	○
Sun-3/50	● 3.5 4.0				◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-3/60	● 3.5 4.0				◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-3/110	● 3.5 4.0				◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-3/140	● 3.5 4.0				◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 4.0 3.5	◕ 3.5 4.0
Sun-3/160	● 3.5 4.0	◔ 3.5 4.0		◔ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-3/180	● 3.5 4.0	◔ 3.5 4.0		◔ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-3/260	● 3.5 4.0	◔ 3.5 4.0		◔ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-3/280	● 3.5 4.0	◔ 3.5 4.0		◔ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0	◕ 3.5 4.0
Sun-4/110	● 4-3.2 4.0		◔ 4.0		◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4.0
Sun-4/150	● 4.0	◔ 4.0		◔ 4.0	◕ 4.0	◕ 4.0	◕ 4.0	◕ 4.0	◕ 4.0
Sun-4/260	● 4-3.2 4.0	◔ 4.0		◔ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4.3.2 4.0	◕ 4-3.2 4.0	◕ 4.0
Sun-4/280	● 4-3.2 4.0	◔ 4.0		◔ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4-3.2 4.0	◕ 4.0

◕ All Models ◔ Color/CXP Models ◒ Color Only Models, no CXP ○ Will be available

3.5 SunOS 3.5 4-3.2 SunOS Sys4-3.2 4.0 SunOS 4.0

Figure 6-1. Graphics Software Interfaces and Standards

software interface is below the display list level and portable software is not an issue, then specialized hardware-dependent software libraries are available, such as Pixrect“ /Pixwin, Graphics Processor Software Interface (GPSI), and TAAC-1“ libraries in the SunWindows environment. Imaging and animation applications that need 24-bit color image display without acceleration can use the CG8 frame buffer. For applications that need a stencil/paint model, PostScript is available on the NeWS window system platform. Figure 6-2 shows the interaction between Sun® graphics software and hardware.

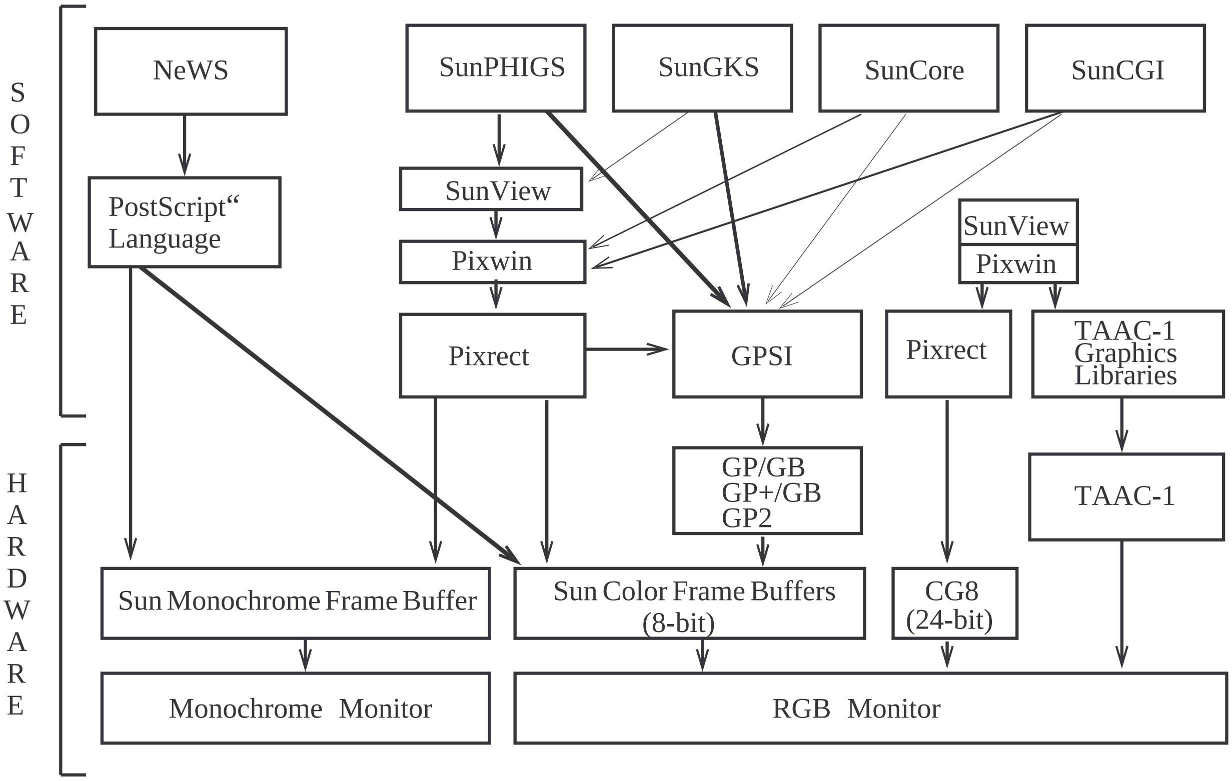

Figure 6-2. Interaction between Sun® Graphics Software and Hardware

Several factors are involved in choosing the right graphics package for software development.

Choosing the right graphics package for software development

One factor depends on whether the graphics are 2-D or 3-D. Every graphics software library produces 2-D graphics. Only SunPHIGS“ , SunCore¤ , GPSI, and TAAC-1 software libraries generate 3-D graphics. All standard graphics interfaces can be displayed in color or monochrome or high-resolution monochrome.

Another factor is whether the application should display graphics in a window in SunWindows or NeWS. Sun is committed to graphics in windows, and all future software will be integrated in windows. All the standard graphics software libraries display graphics in a window. Pixrect, GPSI, SunCGI“ , and SunCore can display on the raw frame buffer as well as draw in a winside other software applications in SunWindows. One advantage is that graphics in a SunWindows window can employ the locking function. Locking temporarily locks the window system manager and allows the graphics to exclusively take control and draw onto the frame buffer. When the drawing is completed, the window is unlocked, and the window system manager regains control. Locking provides synchronization for an application in the window system. Graphics standards packages such as SunPHIGS, SunGKS“ , SunCore, and SunCGI, have the locking built into their libraries.

Display list graphics may also be a factor in choosing a graphics package. A display list is a buffer of graphics commands organized to optimize both screen redrawing and the communications bottleneck to the host. The SunPHIGS, SunGKS, and SunCore graphics standards packages all have display list interface capabilities. Only SunPHIGS has an editable hierarchical display list. Display lists are especially useful for applications that have large amounts of data that need to be graphically displayed and are often updated. Storing all the data in a display list saves repainting the entire picture each time an update is made or when window damage is detected.

Another factor may be vector graphics versus raster graphics. All of the graphics software libraries display vector graphics. In particular, Pixrect/Pixwin, GPSI, SunPHIGS, SunGKS, SunCore, and SunCGI are tuned for higher performance in vector graphics through graphics and floating-point accelerators. Some of the packages can display raster images; however, the TAAC-1 applications accelerator is especially designed for imaging applications. It quickly displays vector graphics or a combination of vector and raster graphics. It has its own processor that includes a floating-point processor, graphics processor, and on-board memory that contributes to its high performance. The CG8 24-bit frame buffer does not have graphics acceleration. However, it is dedicated to displaying high-quality 24-bit images in a color monitor. Monochrome images can also be displayed in this frame buffer through its 1-bit overlay plane.

Graphics viewing systems versus modeling systems is another factor to consider. Viewing and modeling are different methods of implementing graphics solutions. SunGKS, SunCore, and SunCGI are viewing systems, while SunPHIGS is a modeling system. A viewing system can be imagined by using the analogy of a virtual camera where the data for graphics objects are pushed through a viewing pipeline then stored in the segmented display list. Display list editing to modify the data is not allowed in viewing systems because viewing transformation occurs before the display list is built. Segment highlighting, transformation, and visibility are features of viewing systems

tems are more flexible and expensive in terms of additional overhead because they have fully editable display lists that constantly feed information through the graphics pipeline. Modeling systems are compute-intensive, and high-performance workstations are a suitable hardware platform. A graphics standard based on modeling systems should only be used if the graphics data need to be edited in the display list. Otherwise, a graphics standard package based on a viewing system should be used.

The remainder of this chapter describes each graphics software library available today on the Sun workstations. For information on programming in SunWindows, refer to Chapter 5, *Window Systems on Sun Workstations.* Programming examples are available in Appendix A of this guide. For more in-depth information, manuals are available in SunÕs standard operating system release documentation or can be ordered from Sun. Graphics software libraries that are distributed at no additional charge with a Sun operating system release are: Pixrect/Pixwin, GPSI, SunCore, and SunCGI. SunPHIGS, SunGKS, CG8 frame buffer, and TAAC-1 graphics software libraries are unbundled software products that are purchased and released separately from the operating system. The PostScript langauge is available with the NeWS software.

6.2 Graphics Standards

SunCore, SunCGI, SunGKS, SunPHIGS, and the PostScript Language comprise SunÕs graphics standard offerings.

6.2.1 SunCore

3-D presentation graphics software package

SunCore is a 3-D presentation graphics software package based on the ACM Core System. Core is a viewing system, not a modeling system like PHIGS. Some of SunCoreÕs graphics features include segment operations, 2-D and 3-D translation, scaling and rotation, synchronous input, and 3-D primitives. SunCore works with all Sun frame buffers including monochrome, high-resolution monochrome, and 8-bit color. SunCore graphics can be accelerated by the addition of Graphics Processor and Graphics Buffer boards. Floating-point operations are accelerated by the addition of a Floating Point Accelerator board on Sun-3 workstations or a Floating Point Unit on Sun-4 workstations. Programs can be written in C, FORTRAN, or Pascal.

There are advantages and disadvantages of using SunCore for application development. Core was proposed in 1979 when window systems and bit-mapped terminals were nonexistent and graphics were displayed on vector displays. SunCore was developed to draw graphics on one display or view surface and is not well integrated into SunWindows. For example, an appli-

SunCore graphics application. One reason to use SunCore is for some limited degree of similarity of an existing Core-based application, but remember that SunCore will be supported but not enhanced after SunOS Release 4.0 and will be available at the next major release in binary format in a ./old directory. SunCore will have minimal future device support and will only run on the Pixrect/Pixwin graphics platform. SunCore was developed for C programs, and support for FORTRAN programs is through a nonstandard FORTRAN binding. Support for Pascal programs is also available. SunPHIGS is the logical replacement for SunCore-based applications. For additional information about SunCore, refer to the *SunCore Reference Manual*.

6.2.2 SunCGI

2-D graphics package

SunCGI, a 2-D graphics package that provides access to low-level graphics functions, is an implementation of the 1985 draft of the proposed ANSI Computer Graphics Interface. CGI has not yet been approved as a standard. Some SunCGI features include output primitives, attribute selection, and input device management close to the graphics device driver level. In contrast to SunCore, SunCGI is at a lower level, and complex graphics functions such as display list, transformations, and segmentation are not available. Similar to SunCore, SunCGI supports monochrome, high-resolution monochrome, and 8-bit color displays. SunCGI graphics speed can be accelerated by the addition of Graphics Processor and Graphics Buffer boards. Programs can be written in C and FORTRAN.

SunCGI was implemented with window system interaction. SunCGI has the advantage of allowing a SunWindows application to incorporate SunCGI routines to draw graphics in a SunView canvas. The SunCGI routines that are used in a SunView application are called CGIPW. For a detailed description of the function calls refer to Appendix E of the *SunCGI Reference Manual* . SunCGI will not be enhanced after SunOS Release 4.0, and SunCGI will be available in binary format in a ./old directory in the next major release. SunCGI will only run on the Pixrect/Pixwin graphics platform and will have minimal device support. SunCGI has a poor FORTRAN binding. Although it can be used to port existing CGI programs, it is not a recommended porting solution. Today, 2-D graphics standards are moving toward the approved ANS/IS GKS international standard, and existing CGI applications should be ported to SunGKS. For additional information about SunCGI, refer to *SunCGI Reference Manual*.

2-D interactive computer graphics standard

SunGKS, an application software interface standard designed for 2-D interactive computer graphics, is an implementation of the ANS/IS Graphics Kernel System (GKS). SunGKS conforms to level 2C (Workstation Independent Segment Storage and full asynchronous input) of the GKS standard. GKS is the first international standard for programming 2-D computer graphics applications. GKS is a viewing system with functions for display list, picture generation, picture presentation, segmentation, transformation, immediate-mode input/output, input, output, and storage.

SunGKS supports basic output primitives and attributes and allows them to be collected into groups called segments, which can be scaled, translated and rotated. SunGKS users can view a segment with different transformations on different display surfaces. Display surfaces are called workstations. SunGKS uses SunWindows windows to represent OUTPUT and OUTIN workstations. Metafiles are available for storing SunGKS generated graphic images. Computer Graphics Metafile (CGM) output will be available in the next release.

Both C and FORTRAN bindings are available for software development. Porting existing GKS applications to SunGKS depends on whether standard GKS functions were used. Each GKS package will be slightly different because the interpretation of the standard determines the implementation of the functions. The integration of a GKS application into the SunWindows does not require an understanding of SunWindows or the SunView user interface toolkit unless the use of SunView subwindows or input/output packages is necessary. SunGKS is the logical replacement for the older SunCGI graphics package. Porting from SunCGI to SunGKS is not a difficult task, and comparable functions are available in both packages.

Graphics performance can be accelerated with the addition of the Graphics Processor board with automatic GP+ and GP2/CG5 acceleration. Floating-point operations are accelerated by a Floating Point Accelerator board on the Sun-3 workstations or a Floating Point Unit on the Sun-4 workstations. SunGKS supports hardcopy output to HPGL plotters and HPLaserjet printers and output to CGM metafiles. For more information about SunGKS, refer to *SunGKS Reference Manual*.

6.2.4 SunPHIGS

3-D hierarchical interactive-graphics system standard

SunPHIGS is an implementation of the ANS/IS, Programmer's Hierarchical Interactive Graphics System (PHIGS). PHIGS is a graphics standard providing dynamic, interactive modeling for the display and modification of 3-D graphical objects. The PHIGS standard is patterned after GKS for its primitives and input model. Performance is key to the concept of PHIGS, both

PHIGS can also be used for 2-D hierarchical applications, especially when editing and rapid updates are needed. PHIGS+ is the proposed extension of PHIGS with single and multiple source lighting, Gouraud and Phong shading, depth cueing, NURB curves and surfaces, direct RGB color, and complex geometric primitives.

SunPHIGS is integrated in the SunWindows environment and has several key features. Programs can be written in both C and FORTRAN with or without SunWindows/SunView code. Graphics performance is accelerated by the Graphics Processor, and floating-point operations are accelerated by the Floating Point Accelerator on the Sun-3 workstations and by the Floating Point Unit on the Sun-4 workstations. Computer Graphics Metafile (CGM) output is available for storing SunPHIGS-generated graphic images.

SunPHIGS has several Sun-specific features:

¥ CG5 hardware double buffering

¥ Generalized Structure Element (GSE)

¥ Generalized Drawing Primitive (GDP)

¥ Integration into window system

¥ Transparent acceleration through graphics processor (GP+/GB, GP2)

¥ Z-buffer hardware on GB, GP2 for hidden surface removal

The ease of porting an existing PHIGS application to SunPHIGS depends on the number of nonstandard PHIGS functions used. The closer the code is to standard PHIGS, the easier the port. Existing SunCore-based application should be moved toward SunPHIGS. Functions available in SunCore are also available in SunPHIGS, with better graphics performance. SunPHIGS is a good platform on which to develop new 3-D applications. Many tools are available to help programmers. For more information, refer to the following documents:

¥ *Getting Started with SunPHIGS*

¥ *SunPHIGS Reference Manual*

Graphics standards offer portability over heterogeneous computers

Graphics Standard	Capability	Typical Application	Sun Product
PHIGS+	¥ Gouraud/Phong shading ¥ Multiple/spot/diffuse lighting ¥ NURB surfaces, meshes ¥ RGB color models ¥ Depth cueing	¥ MCAD/solid ¥ Biomed ¥ Simulation ¥ Animation	Planned development
PHIGS	¥ 3-D hierarchical display list ¥ Application data, name sets ¥ Modeling system, input	¥ MCAD/wireframe ¥ Simulation	SunPHIGS 1.0
Core	¥ 3-D, linear display list, input ¥ Viewing system,Gouraud/Phong	¥ General 3-D	SunCore
GKS	¥ 2-D, linear display list, input ¥ Viewing system, flat shading	¥ 2-D mapping	SunGKS
CGI	¥ 2-D, no display list, ibgnput ¥ Viewing system, flat shading	¥ 2-D general charts	SunCGI
CGM	¥ 2-D picture storage format	¥ Hardcopy transfer	Output from SunPHIGS/GKS
Postscript Language	¥ Stencil/Paint imaging model ¥ Transformations ¥ Text, lines, arcs, Bezier splines	¥ WYSIWYG applications ¥ CAP, CASE,OA	NeWS

Figure 6-3. Standard Graphics Software Libraries for SunOS

6.2.5 PostScript Language

The PostScript imaging model is one of the most successful-graphics standards. PostScript language offers great power and flexibility in 2-D text and graphics. It is a graphics language rather than merely being an interface. This power and flexibility has been available to Sun users since 1987 in the form of the NeWS windowing system. In NeWS, the PostScript language can be used to draw directly into a window. Thus, the same imaging model used by many printers can be used in a Sun workstation window. With the availability of the X11/NeWS merged window system the PostScript language will become an increasingly important part of the Sun windowing environment.

The PostScript language has a sophisticated stencil/paint imaging model. Rendering goes through a transformation pipeline using a transformation matrix. This means that anything drawn in the PostScript language can be arbitrarily rotated and translated. Forms are drawn on the screen by first building up their outlines using primitives such as lines, arcs, and Bezier splines. That outline is then either drawn or filled using a specified color. The PostScript language's stencil/paint model extends to text as well as graphics. Thus, rotated and scaled text is a deeply imbedded part of the PostScript model.

PostScript imaging model graphics standard

What You Get) applications. Because of PostScript’s ability to output the same image to both the screen and the hardcopy printers, it is widely used for developing electronic publishing (EPUBS), computer-aided software engineering (CASE), and office automation tools.

For additional information on PostScript, refer to the following books offered by Adobe Systems Incorporated:

- *PostScript Language Tutorial and Cookbook*
- *PostScript Language Reference Manual*
- *PostScript Language Program Design*

6.3 Graphics Software Interfaces

Pixrect/Pixwin, the Graphics Processor Software Interface (GPSI), CG8, and TAAC1 comprise Sun’s graphics software interfaces.

6.3.1 Pixrect/Pixwin

Lowest-level graphics interface

The Pixrect graphics library is the current lowest level graphics interface that manipulates rectangular arrays of pixel values on the screen and in memory. The graphics library routines are called RasterOps and are common to all Sun workstations. The Pixrect library does not support windows or input devices; these functions are handled at the SunWindows window manager level through the Pixwin graphics library. Many Pixrect functions have a corresponding Pixwin function. The only difference between the two is that a Pixwin draws rectangular arrays of pixels to a window, and a pixrect draws to anywhere on the screen or memory. Pixrect and Pixwin graphics are accelerated with the Graphics Processor and Graphics Buffer hardware.

The Pixrect graphics library can be used for several applications. If a third-party hardware vendor has an add-on frame buffer board for a Sun workstation and runs the SunWindows window system manager on it, someone has to write a Pixrect device driver for the add-on frame buffer. Memory pixrects are useful for storing an image temporarily in memory for later use. With the addition of the CG5 color frame buffer board, hardware double buffering is available for animation applications and is programmable through Pixrect and Pixwin functions. A raster image may be stored into a file in Sun raster file format through the use of Pixrects. For additional information on Pixrects and Pixwins, refer to the following documents:

- *Pixrect Reference Manual*
- *SunView Programmer’s Guide, Pixwin chapter*

take advantage of windows. Unless the application absolutely needs to use Pixrects, use the corresponding Pixwin used so that the application runs in SunWindows/SunView. The only advantages of using Pixrects are for writing a Pixrect device driver, using memory pixrects, and storing raster images in Sun raster file format. Hardware double buffering and all the vector and raster operations are available in Pixwins. Through Pixwins, the application is controllable in the window system with input/output devices, and the application runs with other window-based applications. Pixrects applications take over the whole screen, and only one application can run at a time.

6.3.2 Graphics Processor Software Interface (GPSI)

Programming interface to graphics processor and graphics buffer (GP+/GB, GP2) hardware

The Graphics Processor Software Interface is the programming interface to the Graphics Processor and Graphics Buffer (GP/GB, GP2/CG5) hardware. GPSI is also commonly referred to as Graphics Processor Command Interface (GPCI). This programming interface can only be used if the GP/GB hardware is resident on the local workstation. The workstations that currently accommodate the GP/GB are the 6-12 slot models. GPSI runs on several versions of the Graphics Processor hardware (all 2-board sets): GP/GB, GP+/GB, and GP2/CG5. For the GP2/CG5 hardware combination, the Graphics Buffer is combined with the GP2 on the same board, and the CG5 is a color frame buffer that adds hardware double buffering.

The GP hardware accelerates 2-D and 3-D vectors and flat and Gouraud-shaded polygons. The GB hardware is used for Z-buffer hidden surface removal and as general-purpose memory. Floating-point transformations are also built into GPSI. Rotation, 2-D and 3-D translation, and scaling are available. For additional information on the GPSI, refer to *GPSI Programmer's Guide*.

Programs using GPSI must be written in C. Applications that use the graphics libraries Pixrect/Pixwin, SunPHIGS, SunGKS, SunCore, and SunCGI have GPSI code built-in, and graphics are accelerated if the GP is present on the workstation. Therefore, it is not necessary to program in the low-level GPSI programming code. Programming in GPSI is not an easy task and depends on sending commands in the correct sequence to the GP hardware's graphics pipeline. Debugging is also difficult since the UNIX tools such as *dbx* and *adb* cannot be used. *dbx* and *adb* cannot read the commands downloaded in the GP hardware. Therefore, programming development for an application would be a series of compilations and executions to determine if the correct commands were downloaded and correct graphics were displayed.

able. The macro packages are supplied on the SunOS release software and are in source code so that users can modify the routines for any application. A GPSI macro package, VIEWPORT, has display lists as well as viewport and drawing routines. Another GPSI macro package, DEV_GP1, is simpler than VIEWPORT and does not have display lists. DEV_GP1 contains graphics functions that draw polylines, shaded polygons, vectors, and setup matrices and viewports. For programming examples and source code for two GPSI macro packages, refer to /usr/demo/GP1 and Appendixes A and B in the *GPSI ProgrammerÕs Guide*. Sun is not responsible for maintaining or making bug fixes to the GPSI macro packagesÕ source code. This source code can change without notice.

6.3.3 CG8: 24-bit frame buffer

24-bit frame buffer with 24-bit Pixrect calls used for displaying true-color images

The CG8 frame buffer consists of three planes; a 24-bit plane to represent true color images, a 1-bit overlay plane for high-speed display of monochrome images, and an enable plane which is used to select between the other two planes. The frame buffer is currently only available on the Sun-4/110TC (True Color) workstation, and in the future will run on other workstations as well. Graphics acceleration is not available for the CG8 frame buffer. The graphics processor and graphics buffer do not function with the CG8 frame buffer. Its primary purpose is to display 24-bit true color images and monochrome images on one frame buffer on a workstation.

Software developed for the CG8 frame buffer must be written in C, and graphics and images are drawn through the use of Pixrect and Pixwin graphics libraries calls. The Pixrect and Pixwin calls are extensions to the existing calls that are standard to SunOS. The software accompanies the CG8 hardware and must be installed on SunOS release 4.0 and above.

Applications using Pixrect or Pixwin for 8-bit images need source-code modifications in order to display in 24-bits. Color lookup tables are accessible through the Pixrect library calls. This allows an application to fine tune the intensity of the colors in an image. Monochrome applications will run unchanged on the 1-bit monochrome plane.

For additional information on the CG8 frame buffer refer to the following documents:

¥ 4.0-CG8 READ THIS FIRST

¥ 24-Bit Color Release Notes

User-programmable applications accelerator with graphics and image-processing libraries

The TAAC-1 is a user-programmable applications accelerator for compute-intensive graphics, imaging, or visualization applications. The hardware addresses a broad class of high-performance graphics and imaging requirements. Full-color, 8- and 24- bits/pixel with an 8- bit/pixel overlay plane are available. The TAAC-1 has fast floating-point processors and multiple integer processors that use a unique very long instruction word (VLIW) architecture. Eight Mbytes of on-board memory is used for fast pixel and data access. The TAAC-1 software links the Sun workstation CPU with the TAAC-1 hardware.

You can program the TAAC-1 in several ways. A program can be written in C with TAAC-1 library function calls, in-line assembly code, pure C code, or microcode. The TAAC-1 has its own development environment including the C compiler, linker, loader, and profiler. A debugger is also available that allows programmers to single-step and set breakpoints in the code. The program structure for a TAAC-1 program involves running portions of the program on the local CPU and portions on the TAAC-1. (see Figure 6-4.)

Sun Modules

¥ Disk I/O routines
¥ User interface routines
¥ Host system libraries
¥ Host TAAC-1 libraries
¥ Window management routines

TAAC-1 Modules

¥ Compute-intensive routines
¥ TAAC-1 application libraries
¥ TAAC-1 system libraries
¥ Image generation and manipulation routines

Figure 6-4. Sun and TAAC-1 Modules

The TAAC-1 has extensive graphics and image-processing software libraries, including source code for fast application development. The libraries currently available are the Graphics library, the image processing library, and the 3-D volume toolkit. Software that uses these libraries are portable to future hardware releases, whereas software that uses TAAC-1 microcode is not guaranteed to be portable in the future. It is recommended that the TAAC-1 software libraries be used in application development.

Figure 6-5 shows how the TAAC-1 software is used in an application program.

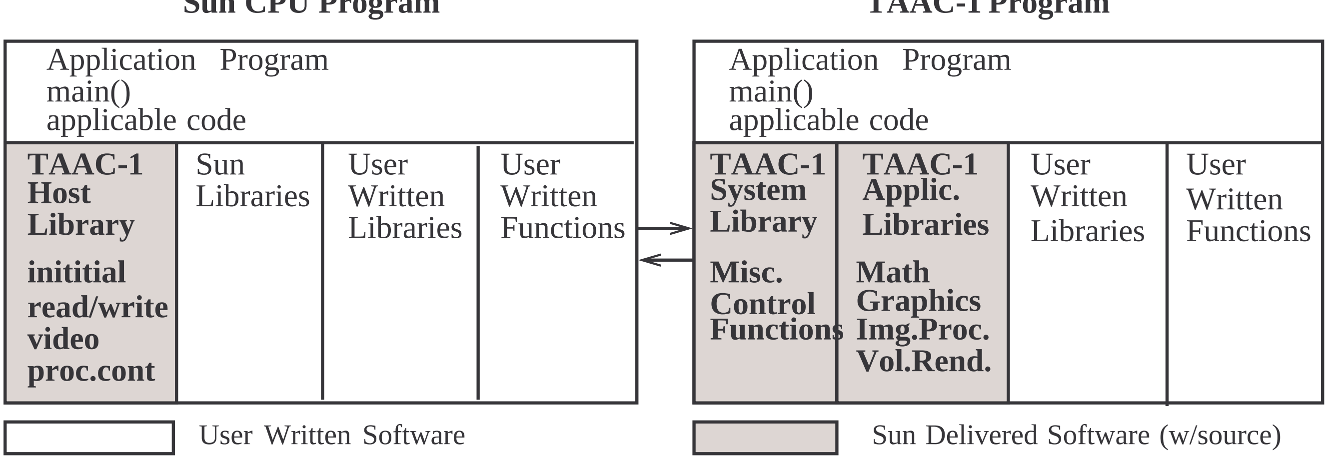

Figure 6-5. TAAC-1 Software Use

TAAC-1 video can be displayed in two ways. It can be piped through a SunWindows window or can be independently viewed on a separate monitor. SunWindows can only run on the host computer, but can act as a user interface for all interactive operations on the TAAC-1.

The TAAC-1 can display 2-D and 3-D vector graphics quickly, but a Graphics Processor (e.g., GP2) is more suited for accelerating vector graphics. Applications that need graphics standards (PHIGS, GKS, Core, CGI) or 8-bit color should not be using the TAAC-1, but rather a Graphics Processor, such as GP2.

For more information, refer to the following documents:

- *TAAC-1 Application Accelerator: User's Guide*
- *TAAC-1 Application Accelerator Technical Notes, Architecture and Application Development.*

For a summary of graphics software interfaces, see Figure 6-6

Graphics software interfaces are used in specialized graphics applications

Graphics Software Interface	Capability	Typical Application
Pixrect/ Pixwin	¥ 2-D, vectors, flat-shaded polygons ¥ RasterOps, locking ¥ Pixwin: window system interaction ¥ Runs on all Sun workstations	¥ General vector/polygon graphics ¥ Pixel-based graphics
GPSI	¥ 2-D, 3-D transformations ¥ 2-D, 3-D vectors, flat and gouraud-shaded polygons ¥ Z-buffer hidden surface removal ¥ Runs on Sun workstations with GP/GB, GP+/GB, GP2/CG5 hardware	¥ Complicated wireframe graphics ¥ MCAD, ECAD ¥ Low level interface to GP/GB, GP+/GB, GP2
CG8 24-bit frame buffer	¥ 24-bit plane, true color image display ¥ 1 bit overlay plane, fast mono image display ¥ Color lookup tables to adjust color intensity ¥ Program with Pixrect/Pixwin calls ¥ Runs on Sun-4/110TC	¥ General vector/polygon graphics w/24-bit color ¥ Imaging
TAAC-1	¥ 2-D, 3-D, transformations ¥ 8-24 bit color, fast raster graphics, vector, polygon ¥ Own floating-point processor ¥ Runs on Sun workstations with TAAC-1	¥ Image Processing ¥ Medical Imaging ¥ Animation

Figure 6-6. Graphics Software Interfaces

6.4 Future Directions

"Graphics over a network"

The future of graphics is distributed processing, which is the ability to do Ògraphics over a networkÓwithin network-based window systems like X11 or NeWS. Graphics will no longer depend solely on the application being on the same workstation as the graphics accelerator hardware. Acceleration will be gained by executing application software on a client computer and sending the graphics portions to the server computer to display graphics. A server computer with graphics accelerator hardware will accelerate graphics.

The current networked graphics solution is through NeWS which uses PostScript commands to draw graphics. PostScript commands are interpreted by NeWS to display graphics on the screen or input to PostScript printers, such as the LaserWriter, to generate hardcopy output. For additional information on NeWS, refer to Chapter 5 *Window Systems on Sun*.

PHIGS extensions to X (PEX) is a step towards networked graphics standards running in distributed window systems. PEX is a proposed protocol extension to X11 based on PHIGS, offering client/server 3-D graphics across the network. It is basically a mapping of PHIGS and PHIGS+ calls into a defined protocol suitable for network transmission.

software based on hardware-dependent software packages is a good choice if fast performance is more important than portability. Hardware dependent software packages have a tendency to change when the hardware changes and does not run across all workstation models. Using ANS/IS graphics standards, such as GKS, CGM, and PHIGS, benefit application developers because their code will run on all Sun workstations, and can be easily ported to other computers.