NASA TM- 85726

NASA Technical Memorandum 85726

NASA-TM-85726 19840020426

CURRENT AND FUTURE GRAPHICS REQUIREMENTS FOR LaRC AND PROPOSED FUTURE GRAPHICS SYSTEM

NANCY L. TAYLOR
JOHN T. BOWEN
DONALD P. RANDALL
RAYMOND L. GATES

JUNE 1984

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

CONTENTS

1. INTRODUCTION

Following the general trend in both government and industry, there is a growing requirement for improved computer graphics capabilities at Langley Research Center (LaRC). There are more computer graphics users than ever before, involved in a wider variety of applications (each with its own set of unique requirements), requesting a higher degree of sophistication in their graphics displays. The computer graphics explosion has been fueled by the increased availability of graphics hardware now being offered at prices affordable to more installations.

As is the case with any relatively new and expanding discipline, the rapid growth of the computer graphics field has resulted in certain undesirable side effects. The increased demand for new and improved graphics capabilities (both hardware and software), subject to time constraints and other limiting factors, has necessitated that individual requirements be satisfied while the design and development of the overall <u>graphics system</u> be neglected. "A graphics system may be described as any collection of hardware and software designed to make it easier to use graphics input and output in computer programs." [NEWM79]

LaRC, with a large computing installation characterized by a diversity of computers, graphics hardware and software supporting various applications in aerospace and aeronautics research, has not been immune from the growing pains of the computer graphics field. Graphics capabilities at the Center have expanded dramatically during the last few years while the evolution of a graphics system has lagged behind. The Flight Software and Graphics Branch (FSGB) of the Analysis and Computation Division (ACD) recognized this void and formed a "Graphics Working Group" (GWG) to propose a solution.

The charter of the GWG was to assess the current and future graphics requirements of the LaRC researchers (with respect to both hardware and software) and to propose a graphics system designed to meet these requirements.

A four-phase plan was employed by the GWG to complete the investigation:

- Determine what LaRC researchers are <u>currently</u> engaged in with respect to all aspects of computer graphics (to include: hardware, software, batch graphics, interactive graphics, ..., etc.). Ask LaRC users to project their <u>future</u> graphics requirements over the next five years.

- <u>Analyze</u> their current and future graphics requirements with respect to current graphics capabilities at the Center.

- <u>Propose</u> an <u>"ideal"</u> graphics system to support both the current and future graphics needs of the user community.

- <u>Propose</u> a <u>"realistic"</u> graphics system showing the evolutionary growth from the Center's current capabilities to a system that will meet future needs.

The intent of this paper is to report the findings and to present the proposal for a future graphics system.

2. CURRENT COMPUTING ENVIRONMENT

Before embarking on the first phase in the four-phase investigation, it was necessary to examine the current computing environment at LaRC. The LaRC Scientific Computer Complex will be discussed in this section with special attention being directed towards the graphics hardware. The present graphics software will be identified, and its interfaces with the graphics hardcopy devices detailed. System limitations that impact computer graphics users will be outlined. Finally, an attempt will be made to define the role played by graphics at LaRC.

2.1 LaRC Scientific Computer Complex

The LaRC Scientific Computer Complex will be described with respect to: the various computers in the complex, the operating systems driving the computers, the user access to the numerous systems, the communication paths between systems, and the extent of "stand-alone" computing at the Center.

2.1.1 Computers

The LaRC Scientific Computer Complex (central site) is located in ACD.

The Scientific Computer Complex consists of three computer systems:

CDC CYBER 170 series
PRIME (interactive graphics)
CDC CYBER 203

The CDC NOS (Control Data Corporation Network Operating System) computer system consists of 7 CYBERS organized into two clusters of machines. Each machine is identified by a single character name, and each machine is designated to perform a primary function.

The computer name, model type, and primary functions are as follows:

	Computer	Model	Primary Use
Cluster 1	A	CYBER 173	Interactive Processing
	Y	CYBER 173	Interactive Processing
	C	CYBER 170-750	Batch Processing Data Reduction Tape Processing
	D	CYBER 175	Batch Processing PRIME to NOS
Cluster 2	R	CYBER 175	Real Time Simulation Remote Batch
	T	CYBER 175	Real Time Simulation Remote Batch
	Z	CYBER 170-730	CYBER 203 Access Station

Each cluster has a separate set of permanent files.

The PRIME computer system consists of four model 750's identified K, L, M, N and one model 850 identified J. The PRIME system is used primarily for interactive graphics. The five machines have access to the CDC NOS computer system via the "D" machine.

The CYBER 203 machine is a vector processor. It is accessed via the "Z" machine in the CDC NOS computer system.

The current LaRC Scientific Computer Complex is shown in figure 1.

2.1.2 Operating Systems

The CDC NOS modified for specialized LaRC subsystems is the operating system supporting the CYBER computers, Common Permanent File System, interactive terminals, remote job entry sites, and I/O devices. The current system version and PSR level are NOS 1.4 552.

The PRIMOS (PRIME Operating System) is the virtual operating system supporting interactive graphics on the PRIME computers. The current system version is Revision 18.2.

The CYBER 200 OS Version 1 is the operating system supporting the CYBER 203 computer. The current system version is 1.4.

It should be noted that each of the above-mentioned operating systems has been locally modified to meet specific LaRC requirements.

2.1.3 Computer Access

Interactive terminal access to the LaRC Scientific Computer Complex is through a central digital data switching system. This includes:

LaRC interactive terminals
Off-site interactive terminals (dial in)
Off-site computers
Local on-base computers (including PRIME)

The baud rates vary from 300 to 9600.

2.1.4 Inter-System Communications

The present configuration of the LaRC Scientific Computer Complex permits limited forms of inter-system communication. These forms of communication can be categorized as belonging to one of two types: communication between machines running the same operating system, and communication between machines running different operating systems.

The CDC machines running NOS communicate through a "satellite coupler" system (see figure 1.) linking the two clusters. The CYBER 203 is a special case whereby a 5 megabit PPU trunk adapter links the CYBER 203 to one of the other CDC machines (called the "access station").

The PRIME computers running PRIMOS communicate via the ring network, PRIMENET, with an "effective" transfer rate of approximately 8 megabits.

The communication links between different operating systems are in a primitive state. The PRIMOS and NOS machines communicate over a 9600 baud line. The communication software (called COMET) which resides on PRIMOS emulates the standard CDC 200 UT protocol. This form of data transfer is restricted to card image and printer output files.

A similar line exists between NOS and several PDP-11/XX class machines running RSX-11/M. The PDP resident software again emulates the 200 UT protocol and is called MUX200.

A VAX 11/780 is tied into NOS using the same line as the PRIME machines with the same limitations.

2.2 Graphics Hardware

There are many types of graphics hardware being used at LaRC. The following is a list of some of the types and models:

- Electrostatic
  - VARIAN 4115, 3310A (now called BENSON)
  - VERSATEC 8136A
- Dot matrix printer/plotter
  - PRINTRONIX 300
- Pen Plotters
  - CALCOMP 1036, 1055 drum plotters
  - TEKTRONIX 4662 flatbed
- Film Recorders
  - DICOMED D47 (color)
- Storage tube
  - TEKTRONIX 4014, 4114, 4052, 4054, 4081
- Monochrome raster
  - HEWLETT-PACKARD 2647
  - TEKTRONIX 4025
  - DEC VT-125
- Refresh
  - IMLAC Series 2
- Color raster
  - AED 512
  - RAMTEK 6212
  - CHROMATICS 7900
  - ISC 8001
  - TEKTRONIX 4113, 4115
  - APPLE+2

The graphics hardware configuration for LaRC is shown in figure 2.

The graphics hardware at LaRC is located in three different areas:

ACD production hardcopy (central site plotters)
ACD Open Shop
Local site graphics terminals

The ACD production hardcopy graphics hardware is located in the plotting area (restricted area) in ACD. This hardware is maintained and operated for the LaRC user by ACD and is accessed by the user by selecting a specific graphics hardware device driver (postprocessor) on the CDC NOS computer system. The production hardcopy is used for report quality figures, working copy figures, and film. There are approximately 20,000 plots per week produced on these plotters alone. The following are valid production device names:

On-line to CDC NOS Computers

VARIAN roll electrostatic
VARIAN fanfold electrostatic

TEKTRONIX preview

Off-line

VERSATEC electrostatic 35" drum
CALCOMP 11" drum pen plotter
CALCOMP 33" drum pen plotter

Since the TEKTRONIX graphics terminal can be used in an interactive mode, the TEKTRONIX can be used to preview (or debug) plots before submitting the plots for plotting on the other plotters, such as CALCOMP or VERSATEC. The function of the ACD Production Graphics Devices is shown in figure 3.

The ACD Open Shop graphics hardware equipment is available at ACD for a LaRC user to schedule. The user is responsible for generating the software and also for operating the hardware. The current ACD Open Shop equipment includes:

Color Monitor	DIGITAL TV
Color Film Recorder	DICOMED D47

The local site graphics terminals are purchased and maintained by the local sites. Graphics terminals can be connected to the PRIME computer system or to the CDC NOS computer system. Some of the local site graphics hardware includes:

AED 512, 767
APPLE+2
CHROMATICS 7900
DEC VT-125
HEWLETT-PACKARD 2647
IMLAC
ISC 8001
RAMTEK 6212
TEKTRONIX 4014, 4114, 4025, 4052, 4054, 4081
TEKTRONIX 4113, 4115
TEKTRONIX 4662 flatbed

The current ACD production equipment has not been replaced or updated periodically in the past. Also, our production devices are used 24 hours each day (three shifts), and this use rate puts an added strain of aging on the equipment parts; therefore, maintenance becomes a serious problem. The only equipment that has been updated in the 80's is two Varian roll plotters in 1980, and one CalComp 1055 pen plotter in 1982 and one in 1983. The rest of the equipment was purchased prior to 1979.

2.3 Graphics Software

The graphics software is presently configured as a set of independent graphics packages, each providing a set of distinct (but not necessarily unique) capabilities. Graphics software packages are usually installed on the CDC and PRIME computers (see figure 2.), and a special effort is made to minimize the external differences between the user interfaces on both systems. Historically, ACD has provided graphics software users with several choices and allowed them to select the software package best-suited for a particular application.

ACD has acquired graphics software from a variety of sources: in-house, commercial vendors, universities, and other government agencies. Each package obtained from outside sources has been locally modified to meet the requirements of the LaRC Scientific Computer Complex.

All ACD graphics software supports at least two classes of output devices:
(1) the central site plotters, and (2) the TEKTRONIX graphics terminal. The central site plotters are supported through a low-level, device-independent Plot Vector File (PVF), which each package is modified to write. Each central site plotter has a postprocessor which accepts the PVF as input and produces graphics hardcopy as output. Each graphics library is also capable of driving the TEKTRONIX storage tube terminal. The TEKTRONIX terminal is virtually an industry standard and has been the graphics device employed by a majority of the LaRC graphics users. Other graphics terminal users have usually acquired a TEKTRONIX emulator for their devices which permits the ACD graphics software to support them.

Despite the limited inter-package communication between a few of the graphics packages and the capability to generate a common PVF, the system cannot be characterized as "integrated"; there is no common interface between the software libraries, and the PVF carries very litle graphics information. The present system also does an inadequate job of supporting the modern graphics user who is demanding color and three-dimensional (3D) capability and does not wish to be limited to TEKTRONIX or even storage tube terminals.

2.3.1 Supported Packages

2.3.1.1 LARCGOS

The LaRC Graphics Output System (LARCGOS) consists of an in-house library of subroutines and a set of postprocessors that drive the ACD production graphics plotting devices. The library may be used to build a picture by calling routines to scale the data, draw a grid, draw and annotate axes, and plot an array of data points. The library produces a PVF which contains only plotting commands and is device independent. The postprocessor formats the PVF for a particular graphics device and also allows modifications to be made to the PVF.

LARCGOS is essentially a passive graphics package oriented toward two-dimensional (2D) plotter output. It provides the basic graphics primitives and has added capability for doing arcs, circles, rectangles, and multiple sets of character fonts (e.g., HERSHEY). LARCGOS also provides a publication module which allows users to produce report figures which are well suited for the LaRC publication process.

The software is written in FORTRAN and is installed on both the PRIME and CDC computers. LARCGOS does not support color raster, 3D, segments, or image transformations.

2.3.1.2 PLOT10

PLOT10 is a TEKTRONIX graphics subroutine library that supports interactive graphics for the TEKTRONIX 401X series terminals. PLOT10 provides the capability to do moves, draws, dashed lines, windowing, viewporting, plotting, and crosshair cursor and graphics tablet input.

The user is presented with a 2D, CRT-oriented graphics package with a limited degree of interaction capability. Users may work in either screen coordinates or virtual space, or both, with interaction provided by the alphanumeric keyboard and the cursor positioning devices. Device independence is achieved within the TEKTRONIX 401X series of terminals. A PLOT10 program is capable of driving the TEKTRONIX 4010, 4012/4013, or 4014/15/16 terminals. Most other graphics terminal manufacturers include emulators which allow PLOT10 to drive their terminals.

PLOT10 has been modified locally to write a device dependent PVF (PLOT10 PVF) which may be saved and redisplayed at the user's convenience, or converted by local utilities and postprocessed by the LARCGOS postprocessors for a permanent hard copy on one of the central site plotters.

The software is written in ANSI FORTRAN and is installed on both the PRIME and CDC CYBER 170 computers. PLOT10 does not support color, 3D, segments, of image transformations, but it does provide an easy-to-use set of routines that give a good low-level introduction to computer graphics.

2.3.1.3 IGL

The Interactive Graphics Library (IGL) is the new TEKTRONIX graphics subroutine library which supports the TEKTRONIX 401X, 402X, and 411X graphics terminals. TEKTRONIX intends it to replace the current PLOT10 graphics software. In addition to the basic graphics features in the current PLOT10 software, IGL has 3D graphics primitives, color, multiple character fonts, paneling, image transformation, segments, and device drivers for all TEKTRONIX graphics terminals.

IGL represents a fairly complete definition of a general purpose graphics system designed to provide portability and meet the requirements of a graphics standard. It allows users to view the world from a 2D or 3D coordinate system with a high level of interaction. The package is modular and allows great flexibility in selecting modules to be installed.

IGL's main drawbacks are that it does not support non-TEKTRONIX terminals, and it requires large amounts of memory to do non-trivial graphics problems. IGL has been modified locally to produce a PLOT10 PVF for 401X devices. No PVF support is currently available for other device drivers.

IGL is written in MORTRAN (a structured FORTRAN) and must be preprocessed before being compiled by the FORTRAN compiler. It is already installed on the PRIME computers and is currently being installed on the CDC computers.

2.3.1.4 NCAR

NCAR is a graphics subroutine library which can do automatic graphic generation, contouring, three-dimensional plotting, map generation, and velocity displays. The software takes its name from the research center at which it was produced, the National Center for Atmospheric Research. NCAR allows users to work in 2D or 3D coordinate system to produce contours and map displays. NCAR produces a Metafile which can be postprocessed and displayed on a TEKTRONIX terminal or one of the central site plotters.

NCAR is written in FORTRAN and runs on the CDC computers.

2.3.1.5 MOVIE.BYU

MOVIE.BYU is a collection of FORTRAN programs (from Brigham Young University) for the display and manipulation of data representing mathematical, architectural, and topological models whose geometry may be described in terms of panels and solid elements, or contour lines. Working with various data files, the programs display the data as either line drawings or continuous-tone images, generate title or geometric model files, and convert contour definitions into polygonal element mosaics.

The user interacts with MOVIE.BYU through its interactive command processor which accepts forty commands. Two of these commands allow the user to select an output device. MOVIE.BYU has been modified locally to produce plots on the AED512, DICOMED, PRINTRONIX, and TEKTRONIX graphics devices. MOVIE.BYU produces color images, black and white line drawings with hidden lines removed, surfaces, and shaded images. MOVIE.BYU has been locally modified to produce a PLOT10 PVF.

The software is written in FORTRAN and runs on the PRIME computers.

2.3.1.6 Others

There are several other graphics software packages which exist locally, but are not supported by ACD. These are normally user-maintained and may be dependent on a particular mainframe or may support a single graphics device (e.g., AEDLIB for the AED512 or 767). These packages are normally obtained at a nominal fee from the device manufacturer or are developed locally and allow users direct access to the device's most powerful features. The software is usually low-level and not suitable for use as a general purpose graphics package.

Graphics users with local site mainframes have usually installed the ACD PLOT10 software on their computers. As new ACD graphics software gains acceptance, these users will probably want to install it locally, also.

2.3.2 Inter-Package Communication

The PLOT10 software has been locally modified to interface to the LARCGOS software. The interface allows LARCGOS users to generate TEKTRONIX plots and permits PLOT10 programers access to such LARCGOS features as the HERSHEY character sets. MOVIE.BYU and NCAR interface to LARCGOS through PLOT10 subroutine calls. All of the graphics packages generate a PVF which can drive the central site plotters, but they share no common interface. The current graphics software system is shown in figure 4.

2.3.3 Limited Device Independence

The ACD graphics software's device independence is limited to the central site plotters and the TEKTRONIX storage tube terminals. The software can also drive any device which can emulate a TEKTRONIX storage tube terminal. Only IGL, among the general purpose libraries, can drive non-storage tube terminals; and it drives only TEKTRONIX terminals.

2.4 Systems' Limitations of Concern to Graphics Users

Besides the obvious requirement for adequate computer graphics hardware and software, computer graphics (particularly interactive computer graphics) makes necessary demands on other critical computer resources. A system not explicitly catering to the unique requirements of the interactive user will not adequately support interactive computer graphics.

Although the CDC NOS operating system gives priority to interactive users, system limitations remain that are unacceptable to a scientific research community. The introduction of the "interactive design and graphics" (PRIME) computers (running PRIMOS) eased the burden on the interactive user, but difficulties still exist. The following paragraphs describe the system limitations that plague the interactive user at LaRC.

Interactive computing requires relatively high transmission (baud) rates between the "host" computer and the user's terminal. The 300-1200 baud rates available on NOS are unacceptable for a majority of interactive applications. The 300-9600 baud lines available under PRIMOS offer a significant improvement, but the PRIME minicomputers are not suitable for all applications.

The system "response time" is another critical issue in interactive computing. Both NOS and PRIMOS favor interactive users to varying degrees, but in each case the response time fluctuates significantly with respect to the system load at any given instant and is therefore unpredictable.

The current computing environment affords no centralized mass storage facility. Scientific applications programs can generate large volumes of data that must be saved on some medium. This problem is most severe on PRIMOS where users are continually filling the limited amount of disk space, but is also evident on NOS where automatic archiving is employed to conserve disk space.

Interactive jobs are not always small with respect to memory requirements. NOS imposes "field length" limits on all interactive jobs, thus severely restricting the interactive user. Although PRIMOS supports virtual memory, the PRIME is not suited for all applications.

Intersystem communication is often necessary to take advantage of the strengths of an operating system or special purpose machine hardware. For instance, the CAD user can utilize PRIMOS to construct a geometric model and to generate graphic displays, but the compute bound model analysis is better suited to NOS and the CDC machines. Although some intersystem communication is possible in the current configuration of the LaRC Scientific Computer Complex, the real need is the availability of a high speed local area network linking, not just the CDC and PRIME machines, but machines from all over the field.

The final limitation is the limited number of available communications ports. This problem may be administrative in nature, but nevertheless, it is often necessary to wait months for the installation of a communications line.

These system problems are of concern to the interactive graphics community at LaRC. The development of an adequate graphics system may well depend upon solving such problems.

3. USER REQUIREMENTS

The first two phases of the four-phase investigation involved interviewing LaRC graphics users with the intent of determining hardware and software requirements which will provide the basis for a "new" graphics system. This section presents the findings of these phases of the investigative process.

3.1 User Interviews

In order to assess the current and future graphics requirements of LaRC researchers, the GWG interviewed several groups of users involved in varied applications, who use computer graphics as an integral part of their research effort. These groups interviewed included CYBER 203 users, CAD/CAM users, users concerned with business graphics, users conducting wind tunnel data analysis, and graphics art users. During these interviews, the researchers were questioned on these points:

1. the extent of their current use of graphics hardware and software;

2. problems with, or complaints about the current graphics environment;

3. their future graphics needs/requirements.

Section 3.1.2 will summarize the current utilization of the graphics system. Current graphics problems and user complaints will be discussed in sections 3.1.3 and 3.2. Finally, future graphics needs and requirements will be discussed in section 3.3 of this paper.

3.1.1 Observations

From the interviews conducted by the GWG, several observations can be made. First, LaRC users are concerned, understandably, with solving their own particular graphics problems before solving the graphics problems of the Center. They are concerned with acquiring the necessary graphics hardware and software that will assist them in their particular research efforts.

Many of the concerns users presented to the GWG were perceived to be graphics problems, but in actuality, were limitations imposed by the system. These limitations have been outlined in section 2.5. Another complaint aired by users was the lack of a data base management system suited for scientific and engineering applications.

Hardware and software needs were not differentiable to many of the users interviewed, nor was the difference between analysis and graphics software. Additionally, new graphics hardware is perceived to be a panacea by many users.

Also, it was observed that users found it necessary to address their short-term graphics requirements before tackling the long-term graphics requirements of the future.

Additionally, the user community believes that ACD should provide a "clearing house" for information on computer graphics hardware and software, and also provide assistance in the development and implementation of graphics systems to provide for long-term graphics requirements.

In conclusion, many users are dissatisfied with the current graphics environment. Many users have ideas on how to improve this environment, and many more have questions as to this issue.

3.1.2 Use of Existing Graphics Hardware/Software

The importance of the interview process became apparent when several groups stated that they will be selecting new hardware and acquiring new software but will continue to use the central site graphics equipment and will rely on ACD support as long as the environment is satisfactory.

Currently, a wide range of graphics hardware is being used by LaRC researchers. The central site production and open shop graphics hardware (described in 2.2), together with local site terminals and plotters, are being used to varying degrees by the user community.

The graphics software being used falls into two categories: first, the ACD-supported software (LARCGOS, PLOT10, MOVIE.BYU, NCAR, IGL, etc.); and second, locally supported software (PLOT80-TEKTRONIX support routines for 4081 series equipment, AEDLIB-support routines for AED 512 color raster terminal).

Researchers are producing data analysis graphics through the use of LARCGOS an PLOT10. (For a description of these packages, see 2.3.2.1 and 2.3.2.2.) Contouring and mapping are produced with the NCAR graphics system and locally-written software. Publication-quality graphics is produced through the use of LARCGOS.

Several of the design and analysis packages (ANVIL4000, PATRAN-G, NASTRAN, etc.) in use at LaRC for computer-aided design and manufacture (CAD/CAM) produce limited, application-dependent graphics output.

The user community indicated interest and involvement in three-dimensional graphics software, color graphics, solid modeling, visual realism techniques, and movie-production/animation.

At present, color graphics, in a formative stage at LaRC, is produced either by the DICOMED film writer in conjunction with the locally-developed software which supports that system, or by local color graphics terminals/workstations that are supported by locally-written libraries (e.g., AEDLIB).

Solid modeling, geometric modeling, and visual realism are performed at LaRC through the use of the MOVIE.BYU graphics display system (for a description, see section 2.3.2.5) or locally-developed software. Today, movie production at LaRC is being accomplished through the use of the central site CALCOMP 1675 film writer or the open shop DICOMED film writer system and each system's accompanying software.

Business/management graphics and graphics art are not supported by the current graphics system.

As a result of the interviews of the user community, a better definition of the current use of existing graphics hardware and software was obtained. Furthermore, users could be delineated into groups based on their applications and subsequent needs.

A categorization in this manner produced five groups: scientific/engineering, computer-aided design/manufacture, business/management, simulation, and graphics art. Each group had a unique collection of specific problems and requirements, but all shared a commonality of basic graphics needs.

3.1.3 Summary of Current Graphics Environmental Limitations

Users indicated that there were many problems, in their opinion, with the current computer graphics environment at LaRC. Many of the concerns/complaints were not of a graphics-based nature but did influence the production of graphics. Various system limitations (listed in 2.5) were mentioned repeatedly by those interviewed.

Graphics users are concerned about the absence of a unified graphics software system. Many expressed dissatisfaction with having to learn several different graphics systems to fulfill their needs. Concern with the development of a "standard" graphics software system also was expressed by many users.

Additionally, users working in graphics arts and business-related areas have no support from the current graphics environment. Special-purpose, one-of-a kind software has been developed locally for specific applications, but this does not meet the current needs of these users.

Another prime concern among many users was the lack of communication between groups at the Center regarding the availability of graphics hardware and software. As a result of this communication failure, different user groups are forced to research similar hardware and software problems and, in some instances, duplicate development efforts.

Finally, many users feel that there is a deficiency of graphics capabilities at the Center. Those persons interviewed feel that there is a need for additional advanced capabilities in the various facets of computer graphics such as color production. These requirements will be detailed in sections 3.3.1 and 3.3.2 of this report.

3.2 <u>Non-Graphics (System) Requirements</u>

During the GWG interviews with LaRC users, it became obvious that many of the complaints voiced concerned non-graphics limitations. Graphics users at LaRC indicated to the GWG that these "system problems" must be solved to insure the adequate development of any graphics system. The following paragraphs describe the system requirements as outlined by the interviewed users.

First, certain system limitations hinder graphics production at the Center. These system limitations have been discussed in detail earlier in this paper (refer to section 2.5) and will not be repeated here.

Second, the absence of a data base management system is recognized by users to be a major problem. Typically, graphics users are required to manage large volumes of data. A usable data base management system would alleviate some of the problems involved in the management of this data, freeing the user to concentrate on the graphics portion of his (or her) research.

Finally, no geometry standard exists at the Center. Users wishing to use particular graphics programs on the field must transform their data (in whatever form it may exist) into a format compatible with the program desired to be used. This can lead, at times, to problems. The users must concern themselves with handling data pre- and post-processors in order to interface to the desired program. This program management leads to other assorted problems. A geometry standard could help to remedy some of the problems existing today.

As stated previously, many of the concerns voiced by the users interviewed pertained to non-graphics matters which, undoubtedly, influence the graphics environment at LaRC.

3.3 Graphics Requirements

3.3.1 Hardware

Based on the GWG interviews of LaRC graphics users, the present graphics hardware, for the most part, is inadequate for the current and future needs of the Center's researchers. Users have indicated that the production of quality multi-media hardcopy (both color and black and white) is a necessity. Users are also concerned with the turnaround time for the production of hardcopy at present.

From discussions with users, it has been indicated that high-quality, high-resolution plotters with the capability to produce full-size engineering drawings, color plotters (vector and raster), and a smaller, less expensive electrostatic plotter, similar to the VERSATEC currently in production at LaRC, would be extremely valuable to the research effort at LaRC.

Users have indicated that the Center's movie-production capabilities are extremely limited. They have expressed an interest in the acquisition, by the Center, of a better movie-production system--one that can be used in day-to-day work as well as production.

Users would like to have available, for research support, medium-to-high-resolution color workstations whereby graphics (and other) tasks could be downloaded from a host computer.

The capability to produce video tape output in addition to film was discussed by several users. This capability would provide an alternate output media for use when film facilities are not available or adequate (i.e., conferences, etc.).

The availability of quality computer graphics hardware is a prime concern to the graphics users at LaRC; and the feeling among users is that for LaRC to have an adequate graphics environment, the current graphics hardware must be upgraded, and in some cases, replaced.

3.3.2 Software

Of prime concern to graphics users is the state of the graphics software at LaRC. Many of those interviewed believe that the present graphics system does not provide them with either the necessary flexibility, or the capabilities to assist in their research. This section will describe the graphics software requirements as outlined by the interviewed user community.

Addressing the concept of flexibility, a graphics software system should be both portable, that is, capable of being implemented on different systems with minimal impact, and compatible between local and central sites. These factors would alleviate the problem of researchers having to learn a plethora of graphics routines (and/or systems) to produce desired graphics output. As a result, distributed computing problems would be minimized.

Additionally, the capabilities of the existing graphics software system must be expanded to provide researchers with additional, more powerful graphics "tools." As examples, there exists a need among the engineering and scientific segments of the user community to have available more data display formats. The computer-aided design sector of the user community has a need for advanced techniques for visual realism, geometric modeling, and three-dimensional data manipulation and display routines. These users have also indicated a need for drafting routines which would allow designers to move from the drawing board to the computer terminal.

The capability to support real-time, or near real-time, animation is a need of many users, particularly those researching systems simulation techniques. This capability could be used in conjunction with the movie-production equipment at the Center to produce valuable research output.

The business/management and graphics arts sectors of the user community desire support of their efforts by the graphics software system. Provisions to support their effort must be provided in the design of a graphics system.

Also, the users indicated the desire for the capability to produce graphics output via an interactive command language which would free the user from becoming involved with the mechanics of a graphics system.

In conclusion, the user community requires that any graphics software system have two attributes: (1) that it be flexible enough to support varied applications, and (2) that the system have the capability to support future needs of the research effort.

3.4 Dedicated Graphics Group

Many users interviewed believed that a "dedicated graphics group" would be invaluable at LaRC. According to the user community questioned, the purpose of this group would be twofold:

(1) the group would provide hardware and/or software evaluation as necessary. This could prove to be a great asset to those who are contemplating the acquisition of graphics hardware and/or software.

(2) the group could conduct graphics research which would help to keep the Center abreast, and perhaps, near the forefront, of developments in the graphics field.

4. FUTURE COMPUTING ENVIRONMENT

The role of the computer in supporting the LaRC research is growing in importance, and this trend is expected to continue in the future. The increased demand for more and "faster" computers, more flexible and powerful operating systems, and more sophisticated support software has exceeded the capability of the present configuration to provide such resources. Because of these present resource limitations, the computing environment at LaRC will undergo drastic changes in the future. Several enhancements have been proposed for the central computing complex, and local systems supporting specific applications are beginning to appear.

It is imperative that a graphics system be designed with the future computing environment in mind, because the changing role of the computer also infers a changing role for computer graphics.

4.1 System Configuration

4.1.1 Proliferation of Local Systems

One approach to alleviate the problem of limited computer resources is to purchase the hardware and software necessary to support a particular application(s). The recent reduction in hardware costs, the relaxation of ACD policies on new hardware purchases, and the increased costs of using central site resources have made this an attractive alternative to various organizations (individuals, branches, divisions, ...) at the Center. These local systems can range from powerful minicomputers with a full complement of peripheral devices to an intelligent workstation driving one (or more) graphics output devices. The resources of some such systems are occasionally shared by linking two (or more) compatible local systems with a local area network.

The local system is not a universal answer to all computing problems, because the system can generate its own inherent set of problems. Besides the obvious problems of maintenance and operations, a local system may not be able to share any of the resouces of the central site. Of particular interest here is the accessibility of the central site graphics software.

The relative merits of local site computing will not be debated here, except to point out that local computing will have an impact on the overall graphics system design. The impact is difficult to measure because the growth rate of local computing at LaRC is an unknown.

4.1.2 Proposed Enhancements to LaRC Scientific Computer Complex

The LaRC Scientific Computer Complex will undergo significant upgrades during the next few years. Several of the proposed enhancements will have a positive impact on the graphics environment at the Center. The individual enhancements will not be described in great detail in this report, but will be addressed with respect to their collective effect on graphics at the Center.

In the near future the central data switching system (TRAN) will be replaced. The new "terminal access switching system" will provide numerous advantages to the interactive user including a capacity for supporting up to 1500 terminals and data rates of up to 9600 baud.

Another area receiving considerable attention is the utilization of one (or more) local area networks. It is anticipated that a high speed local area network will alleviate many of the inter-system communication problems that currently exist. Present plans call for the installation of the CDC standard product LCN/RHF (Loosely Coupled Network/Remote Host Facility) sometime in the near future. The local area network will have an effective transfer rate of 20-40 megabits and will link the CDC machines (including the CYBER 203), the PRIME machines, several DEC (of the PDP 11/xx and VAX 11/7xx class) machines, and possibly other non-central site machines. Although the details concerning which "nodes" should be linked to the LCN and (in some cases) how the linkage should be accomplished are not yet finalized, the arrival at such a local area network will be an aid to both the central and distributed computing environments.

A third enhancement that will affect the graphics environment at LaRC is the introduction of an on-line central "Mass Storage System" (MSS). Each MSS cartridge storage unit (one for each "cluster") will have a capacity of 300 disk units (or over 52 billion bytes). In addition to increased capacity, the MSS will increase access time and improve recovery time. The MSS system will serve to support the growth of distributed computing at LaRC, as well as improve central site computing.

The combination of a local area network and the MSS will permit users Center-wide to transfer data (including graphics information) at high speeds to a central site for storage.

Other system enhancements with a more minimal (or localized) impact have been proposed. CDC's commitment to CAD/CAM has assured ACD personnel that newer releases of NOS will be more favorable to interactive users. Again the details of these operating system changes are unknown at the present time, but provisions for "fast" and predictable response times are vital to interactive graphics. Finally, ACD is planning to purchase a CYBER 855 computer in the near future to exclusively support the CADA (Computer Aided Design and Analysis) effort at LaRC.

Enhancements to the LaRC Scientific Computer Complex, such as those described above, will undoubtedly improve the computing environment at the Center. The graphics system proposed in this report is directed towards this new computing (and hence, graphics) environment.

5. PROPOSED GRAPHICS HARDWARE

A major component in any graphics system is the compliment of graphics hardware that is available to the user community. The Center's current graphics hardware capabilities were discussed previously (see section 2.2); and based on the criticisms expressed during the user interview sessions (see section 3.3.1), the current graphics hardware does not satisfy user requirements. In addition to satisfying user needs, it is necessary to develop a replacement strategy for obsolete equipment because hardware (unlike software) does wear out. Finally, any proposal for graphics hardware must take the computer hardware industry into account where new technologies in areas such as "chip" design are affording improved capabilities at lower costs.

5.1 Hardware Requirements

The graphics hardware requirements at the Center closely parallel today's trends in the graphics hardware industry. The increased demand for color; the need for multimedia hardcopy in the form of paper, film, microfiche, etc.; the desire for higher resolution (implying better quality); the increased use of graphical input; and the capability for local device intelligence are all concerns that currently are being addressed by the computer graphics community. In the recent past, such capabilities were merely aspirations, but competition in the computer graphics marketplace has permitted more graphics users to realize their desires.

The hardware requirements at LaRC include both "passive" and "interactive" graphics devices. The "passive" devices are characterized by the traditional pen and/or electrostatic plotters, but may also include a color capability and the facility for varied output media. The "interactive" devices are characterized by the typical vector or color raster terminal with facilities for interactive input and some degree of local intelligence.

The user requirements for passive hardcopy devices are very specific. These requirements include facilities for: publication quality black and white line drawings, full-size (up to size E) engineering drawings, gray-scale and/or color raster digital images, a production COM system capable of producing both color or black and white raster and/or vector movies, and inexpensive rapid turnaround first draft hardcopies. Note that the requirements do not identify any of the available technologies for graphics hardcopy devices, but are concerned only with "end use."

Consequently, if the "end use" user needs are satisfied, the competing technologies of pen, electrostatic, impact printer, ink jet, photographics, or laser xerography are, to some extent, irrelevant.

On the other hand, the requirements for interactive graphics hardware devices are not so specific. One reason for the vague requirements for interactive devices is that experience is limited. For the majority of users at the Center, interactive graphics experience has been restricted to the TEKTRONIX 401X series (or an emulator of this series) terminals. Another reason for vague requirements is that such requirements are driven by the ever-changing applications.

The so-called "graphics workstation" is a proposed solution to satisying the demands of the interactive graphics user. In contrast to the passive hardcopy devices, which are characterized by competing technologies, the graphics workstation is characterized by competing features. Screen resolution, number of simultaneous colors, firmware functions, processor "speed," and additional local intelligence are examples of capabilities that differentiate one workstation from another. Therefore, for interactive devices the problem reduces to matching features with application requirements.

5.2 Hardware Selection

For the "ideal" graphics system, satisfying the hardware requirements of the user community is the only major criterion. In reality, hardware selection must also be based on other concerns such as: rapidly changing technologies, hardware vendor support, system integration requirements, anticipated device workload, maintenance requirements, and cost constraints.

The responsibility for operating and maintaining the graphics hardware is also a major consideration at the Center. The three existing divisions for hardware responsibility were discussed earlier (see section 2.2) and are listed below:

ACD production
ACD Open Shop
Local site

With the expanding facilities for local site computing, the emphasis on local site graphics will also increase. If the graphics system implementors are to remain responsive to the user community, they must acquire experience pertaining to the local site graphics efforts. For this reason, a fourth category of responsibility, ACD FSGB, was added to the original list.

An ideal graphics hardware configuration is shown in Figure 5. The hardware breakdown was performed by device type and area of responsibility. Adhering to the requirements described previously, the device types are given generic names, void of the technology used and vendor name. Because of the need for operator coverage and maintenance constraints, several hardware devices were included only in the area of ACD production. In other instances, because certain devices could be utilized for varying purposes by different users, they were placed in more than one category. The intent of this figure is not to provide a purchase list, but instead a classification of graphics hardware consistent with LaRC user needs and current trends in the computer graphics community.

6. PROPOSED GRAPHICS SOFTWARE

The third phase in the four-phase investigation consisted of proposing an "ideal" graphics system to satisfy the current and future graphics needs of the Center. The structure and limitations of the present graphics software system (see section 2.3), and the current and future graphics requirements as seen from the user viewpoint (see section 3.3) dictate that the present graphics software system be replaced. In addition to local concerns, the phenomenal growth of the computer graphics industry in areas such as graphics hardware capabilities, and the recent proposals for national and international graphics software standards, substantiates the argument that the LaRC graphics software system is obsolete.

6.1 Graphics Software Design

The need for an overall system design is magnified by the diverse requirements of a computing environment such as LaRC's. An environment characterized by a large number of users performing a wide variety of applications running on diverse hardware makes life difficult for the software designer. The design is further complicated by the dynamic nature of the computer graphics field, where dramatic advances can render a graphics system obsolete almost overnight.

Before examining the graphics system design, it is worthwhile to list some of the graphics software system goals. First, the system must obviously satisfy the current requirements of the LaRC user community. Second, the software must be designed to permit future growth. The future growth of the system could be driven by a variety of sources including: new user requirements, changes in the Center's computing environment, and new technology (hardware) and new techniques (software) in the graphics field. Next, because of the large volume of user code currently in existence which uses the present graphics software, it will be mandatory to minimize the impact of the new system on users of the current system. Fourth, although it is desirable to design an "ideal" system free from cost, time, or other resource constraints, in the final analysis the new graphics software system must be realizable from the current graphics environment. Finally, the graphics software system must be attractive to a potential user. Borrowing terms from software engineering, the system should be "user friendly" and "reliable." A user will easily become discouraged when utilizing a software system that is unforgiving and uninformative with respect to error handling. The system will require considerable auxiliary support in the form of extensive documentation, training, and consultation. Insuring the use of a software system is not possible, but allowing prospective users to be active participants in the design process is clearly a step in the right direction.

The graphics systems goals described above are general in nature and could apply to any large software development project. Therefore, we could apply a software engineering "software life cycle model" to the graphics software system. The stages in the software life cycle as identified by Zelkowitz [ZELK78] are: requirements, specification, design, coding, testing, and maintenance. The requirements and specification stages were completed earlier through understanding the current graphics system (both capabilities and limitations), determining user requirements (via the interview process), and surveying the current literature for applicable information on similar efforts. The design philosophy being employed is a combination "top-down" and "bottom-up" approach with the high level design outlined later in section 6.2 and the lower levels detailed in later sections (see sections 6.4-6.8). The coding and testing stages will be overviewed in section 7. In addition to accounting for the most time-consuming stage in the software life cycle, maintenance is extremely important to the graphics software system because future growth is a certainty and not simply a possibility. Ease of maintenance will be a recurring theme throughout the software design process. Although rigid controls may not prove worthwhile, the software life cycle model will be considered during the entire development process.

The general design goals alluded to above can be further refined to more directly reflect the design of a graphics system. "Portability" will be one of the major objectives in the software design. The graphics software should be installation, host, and device independent. When a program using graphics output is transferred from installation to installation, the result is often an extensive conversion effort on the receiving end because the graphics software is coupled to the local environment of the sender. Although complete installation independence is probably unrealistic, an effort toward standardization is necessary. Following the current trends, LaRC is moving in the direction of a "distributed computing" environment.

This implies that graphics software could be exercised on a variety of computer systems offering varying capabilities. Conversion from host to host is unacceptable, necessitating that the software must be "host" independent. Finally, the graphics software system must be device independent. A program requiring graphics output should be free to choose any graphics output device ranging from an inexpensive paper plotter to a sophisticated workstation with no software modifications. Another objective of the proposed graphics system is that it supports both passive ("batch") and interactive graphics. Although the current interests emphasize interactive graphics, there will always be the requirement for high quality production and publication graphics that are more suited to a passive environment. The graphics system should be capable of supporting multi-media graphics output. Paper has been the primary medium for graphics output in the past, but now there is an ever-increasing demand for alternative media such as film, microfiche, and video tape.

The need for software portability has previously been discussed. The only viable means of achieving portability is to adhere to the proposed national (ANSI) and international (ISO) standards for computer graphics. A brief history of the graphics standardization efforts is outlined in [BONO82]. A brief description of the proposed GKS (Graphics Kernel System) is contained in [SCHO82], and a more detailed explanation can be found in the latest ISO draft [GKS 82]. The GKS standard will not be explicitly described in this report but its functionality will be addressed in later sections (see section 6.7-6.8). For definitions of the graphics terms used in this report, see reference describing the CORE system [GSPC79].

Throughout the design process, the "ground rules" for graphics software design,

simplicity
consistency
completeness
robustness
performance
economy

identified by Newman and Sproull [NEWM 79], will be followed.

6.2 Graphics Software System Overview

A graphics software system supporting the researchers at LaRC must satisfy two very important criteria. First, the system must support a wide range of graphics applications varying in complexity from the passive rendering of a simple graph to the real-time rendering of a visually realistic image. Second, the system must recognize the sophistication of the graphics user which can range from the novice with little or no exposure to computers or computer graphics to the experienced applications programer desiring some "high level" tools for hidden surface removal and shading. Designing a single system to support all applications and all levels of users may not be feasible. On the other hand, as pointed out earlier in the report, accumulating individual packages may afford a short term solution in some cases; but in the long term, the absence of communication between the packages creates more problems than it solves. It is evident that a compromise is required, and it is anticipated that the proposed system provides such a compromise.

The proposed graphics system will be a "single system" comprised of a collection of distinct (but related) components. The components are arranged in a hierarchial fashion where the levels of the hierarchy correspond to the so-called "levels of sophistication" at which a prospective user could interface to the system.

The nucleus of the graphics system is called the "Kernel" (named after the GKS Kernel mentioned previously). The Kernel provides the complete functionality of a stand-alone graphics system (such as that provided by an implementation of either the ANSI CORE or ISO GKS standards). The functionality of the Kernel will be described in detail in section 6.7 of the report. Employing a central Kernel will alleviate the problem of functional overlap caused by the current multi-package system and will also serve to eliminate inter-package communication links.

The graphics Kernel, although rich in funtionality, provides only a primitive interface for the applications programmer. In order to adequately support the research work being performed at the Center, a higher level interface providing applications dependent graphic tools is required. Because of the diversity of applications, a single user library is inappropriate; instead, a collection of user libraries each tailored to a graphics applications area is proposed. The categorization of a graphics application into a particular area remains an open question, but a tentative subdivision based on LaRC user requirements yielded the following five areas:

Science/Engineering
CAD/CAM
Business/Management
Simulation
Graphics Art

The applications dependent libraries will be discussed in detail in section 6.4.3.1.

There will undoubtedly be overlap in functionality between the individual libraries in the applications oriented level. Also, the Kernel may not provide for certain graphics capabilities peculiar to LaRC. For these two reasons, a provision has been made for a level between the Kernel and the Applications Library called the "Common Library." The Common Library will contain capabilities not resident in the Kernel, but common to more than one applications area.

The need to accommodate users of varying degrees of sophistication was expressed earlier. The levels of the proposed graphics software system previously addressed provide interfaces (via subroutine calls) only to applications programers. Such user interfaces are useless to the engineer or manager who desires a quick graph or pie chart. For this level of user a "menu driven" or "command driven" interface is required. Naturally, the applications programer is encouraged to construct such an interface for his particular group of users. In addition to these user-developed programs, the proposed system will provide a menu or command driven interface to the appropriate applications dependent library.

The requirement for device independence is still of the utmost importance. In order to achieve total device independence, another level called the device dependent/independent (DD/DI) level is required. The mechanism providing the device dependent interface (DDI) level is the "Metafile." The organization, content and uses of the Metafile will be detailed later in the report (see section 6.4.5), but for the present the Metafile can be thought of as merely the interface between the Kernel and the device drivers for the individual graphics output devices.

The above discussion has presented the proposed graphics software system as a hierarchial structure of six distinct levels to include:

User-Developed Program Interface
Supplied Menu/Command Driven Interface
Applications-Oriented Libraries
Common Library
Kernel
Device Drivers

The multi-level structure is shown in figure 6. The content of the individual levels will be detailed in the subsequent subsections.

6.3 Distributed Computing Environmental Assumptions

As alluded to earlier, the computing environment at LaRC is evolving into the era of distributed computing. Any software system proclaiming to support the Center must be designed to interface with a variety of computer systems ranging from the "super" computer and large mainframe to the minicomputer and microcomputer, and even down to the personal computer and graphics workstation.

Before describing the components of the proposed graphics software system in detail, it is necessary to define how such a software system can be integrated into a distributed computing environment. In order to accomplish the integration, two assumptions about the future computing environment must be made. Supporting distributed computing requires the existence of one or more local area networks capable of transferring information from "host" to "host" at relatively high speeds. Second, a centralized mass storage facility, characterized by high volume and fast access rates, is necessary to manage the information to be transferred. The proposed graphics system will take advantage of both features when feasible.

The ideal configuration would require that a copy of the entire graphics software system be resident on every "host" and that every "host" be directly linked to every available graphics output device. The requirement for redundant graphics hardware is certainly cost prohibitive. Fortunately, the availability of a local area network, capable of transferring graphics information, negates the need for such a requirement. A graphics user has access to any graphics device (for which a device driver is available) either directly or indirectly linked to the network. The transferring of graphical data across a network implies a "host" (as well as device) independent data format. From the previous discussion, such a format is provided by the Metafile generated by the Kernel. Therefore, to utilize the proposed graphics system in a distributed computing environment, the minimum software subset must contain the Kernel.

For the mainframes and large minicomputers, it may be desirous to implement all levels of the graphics software system. On the smaller systems some system dependent compromise, including the Kernel, must be made based on user need and available resources.

It is conceivable that all computer systems will not be linked to the central site via a local area network. The proposed graphics system can support such sites in a manner analogous to that described above. Some subset of the software system, again including the Kernel, must be implemented with the additional requirement for the implementation of the necessary device drivers.

In summary, the proposed graphics software system can be exercised in a distributed computing environment provided a software subset (including the Kernel) be implemented on each "host." Such a restriction dictates that the software be written in a high level/highly portable language(s), thus minimizing conversion efforts from system to system.

6.4 Graphics Software Levels

6.4.1 User-Developed Programs

The basic entry level into the graphics software system is a User-Developed Program. This level allows for two different types of entry and also allows for different types of users, ranging from the sophisticated user to the novice or non-programer.

The User-Developed Program is a program that is designed, written, modified, and maintained by an applications programer. The applications programer can select different graphics capabilities from the lower levels in creating the program. For example, a programer can select capabilities from the Applications-Oriented Libraries (discussed in section 6.5), the Common Library (discussed in section 6.6), and/or the nucleus of the system, the Kernel (discussed in section 6.7). These lower levels provide for all basic graphics capabilities that may be needed by the users; therefore, it must be supported and maintained at the Central site. Since the applications programer is solely responsible for his User-Developed Programs, there is no limit to the number of programs he can generate.

6.4.2 Menu/Command Driven Interface

The Menu/Command Driven Interface allows the user to invoke programs which achieve the desired graphics output through a series of preselected program prompts or menus. This capability is well-suited to the person unfamiliar with the software system or with graphics, but will also benefit the experienced user who wants to experiment with different graphics elements and layouts. Some categories of graphics applications might not need a Menu/Command Driven Interface such as the CAD/CAM or the Simulation, while the user might benefit from a Menu/Command Driven Interface for categories such as Science/Engineering, Business/Management, and Graphics Arts. The Menu/Command Driven Interfaces are basic graphics capabilities that are needed and used by many users; therefore, it should be supported and maintained at the Central site.

The User-Developed Program and the Menu/Command Driven Interface allow for different types of user interaction. The User-Developed Program could require intimate knowledge of the program to produce results while the Menu/Command Driven Interface provides a guided type of user interaction to produce results.

6.4.3 Libraries

6.4.3.1 Applications-Oriented Libraries

Since the User-Developed Programs, written by many applications programers, and also the Menu/Command Driven Interface might require some of the same graphics capabilities, a library of general applications-oriented capabilities should be supported and maintained at the Central site.

Applications-Oriented Libraries are sets of capabilities in user-callable subroutine format that the programer can access to satisfy his applications specific graphics functions.

The GWG compiled individual applications requirements into five broad categories. The five broad Applications-Oriented Library categories are:

Science/Engineering
CAD/CAM
Business/Management
Simulation
Graphics Arts

There is an additional category called Other to allow for growth. The CAD/CAM and Simulation applications libraries are to support existing capabilities and are not intended as stand-alone libraries.

The initial content of the Applications-Oriented Libraries has not been determined. Some representative features that might be included in the library contents are discussed briefly here.

Science/Engineering
The scientific and engineering requirements were combined because both of these usually involve manipulating a large amount of data. Some representative capabilities that might be included are more flexible contouring routines, better 3D algorithms with hidden line removal, some tools for fluid flow visualization, and some tools for conveniently building non-standard formats.

CAD/CAM
This category is available to provide some support to CAD/CAM users. Some representative capabilities that might be included are geometric modeling tools, tools to assist drafting applications, techniques for constructing more usable man-machining interfaces, tools for rendering visually realistic images, and some geometry standard for communicating between different CAD/CAM programs.

Business/Management
The business and management requirements were combined into one category because the type of output is similar, and usually both of these require small amounts of data to be manipulated. Some representative features that might be included in a general purpose package are to draw pie charts and bar charts, generate quick output for management decision making, and provide some statistical capabilities.

Graphics Arts
The graphics artists at the Center are not currently using the computers to generate their presentation graphics. The graphics artists are interested in using the computers to replace their hand-drawn graphics output. Some representative capabilities that might be included are an interactive "paint" program to replace the hand-drawn output and the capability to integrate graphics and text easily.

Simulation

This category is available to provide some support to the simulation graphics area. Some representative capabilities that might be included are a library of tools for constructing man-machine interfaces and a real-time subset of the Kernel.

Since the user requirements were arbitrarily grouped together by GWG, it might be necessary for a user to use the capabilities from more than one library, therefore the libraries can be used together. Guidelines for the libraries' design will be established and followed, therefore all libraries will look alike to the user and be modular.

Since this is a system designed specifically for the user, it is imperative that his requirements are met, therefore the user is an active participant in determining the contents and acceptability of the Applications-Oriented Libraries.

Since the needs will change as the software and hardware change, these libraries will be dynamic in growth. The state-of-the-art software and hardware create new and greater graphics possibilities on a graphics system, enforcing the concept that it is imperative that the system be designed for growth.

6.4.3.2 Common Library

As in the case of the User-Developed Program, Menu/Command Driven Interfaces, and Applications-Oriented Libraries there are some capabilities that are needed by all. These capabilities will be supported in a Common Library along with some unique NASA requirements that need to be available to all users.

The Common Library is a set of capabilities in user-callable subroutine format containing LaRC local features and common requirements that are not supported by the Applications-Oriented Libraries or the Kernel. This library will be dynamic in growth as dictated by the user requirements.

The initial contents of the Common Library has not been defined, but will be defined by FSGB. Some representative features that might be included in the Common Library are discussed briefly here.

NASA standard symbols

NASA has a unique set of NASA symbols that are used in NASA technical reports.

NASA standard line patterns

NASA has a unique set of NASA ordered line patterns that are used in NASA technical reports.

NASA publication standards

NASA has a unique set of publication standards for NASA technical reports.

The Common Library, like the Applications-Oriented Libraries, will be dynamic in growth and determined by the user requirements.

6.4.4 Kernel

The Kernel provides the functional interface between the Applications-Oriented Layer of the proposed graphics system (see figure 6) and the configuration of graphics devices to be supported. As such the Kernel is the nucleus of the system and contains all required functions for performing interactive and passive graphics tasks. Thus, there is no need to duplicate the graphics functions elsewhere in the proposed system.

The Kernel must also control all graphics devices uniformly; this is called device independence. Device independence implies that a single applications program will produce similar, perhaps identical, images on more than one graphics device see [WARN81]. The applications program must then target its graphics output commands to some virtual graphics device, and the device dependent layer of the graphics system must interpret the virtual commands for the device being driven.

The Kernel should adhere to a recognized methodology--a standard. Graphics standardization efforts are now underway within the X3H3 Committee of the American National Standards Institute (ANSI). ANSI's efforts are aimed at developing a methodology and a set of device-independent functional capabilities for graphics programers.

Since the standard is still evolving and many details are not yet known, the Kernel should bear proximate resemblance to the standard and provide all of its functionality.

6.4.4.1 System and Device Control

The applications program references one or more graphics devices. At program load time the user nominates one or more physical graphics devices to replace the virtual device. Each physical device is supported by a device driver. A device driver translates the device-independent commands into the appropriate device-dependent instructions to generate the required graphics on the selected physical device.

A virtual device must be both initialized and selected. Initialization binds the virtual device to the applications program, and selection opens the communication path from the Kernel to the device driver.

6.4.4.2 Graphics Primitives

Primitives are the fundamental commands that define objects in a 3-D world coordinate system. These primitives define moves, lines, polylines, polygons and markers. All positioning and nontext primitives can be defined as 2-D or 3-D, absolute or relative, world coordinates. Primitive attributes determine the general characteristics of output primitives . They include color, intensity, line style, line width, and marker symbol. Text primitives define character strings that are output as graphics primitives on a graphics display device. Text primitives are produced in one of four quality levels, where the quality level determines how closely the text will adhere to its actual attributes. Text attributes include character path, font, justification, size, and gap.

6.4.4.3 Workstation Concepts

A workstation represents a unit consisting of zero or one display surfaces and zero or more input devices, such as keyboard, tablet, and lightpen. The workstation presents these devices to the applications program as a configuration of abstract devices thereby shielding the device peculiarities [GKS 82].

6.4.4.4 Segment Concept

A segment is a named collection of output primitives and primitive attributes. They are the units for display manipulation and change. Manipulation includes creation, deletions, and renaming. Change includes transforming a segment, making a segment visible, and highlighting a segment. Segments also form the basis for workstation independent storage of pictures at run time. The appearance of segments is controlled by segment attributes, which include segment transformation, visibility, highlighting and detectability.

6.4.4.5 Coordinate Systems and Transformations

The applications program can specify transformation of subsequently created segments in their own coordinate systems before they are mapped onto a virtual display device. Translation, scaling, rotation, and shearing transformations can be defined in a 4 x 4 transformation matrix. Individual transformations can be merged to form a composite modeling matrix.

1) World coordinates, (WC) defined by the applications programer.
2) Normalized Device Coordinates, (NDC) used by the Kernel to define a uniform coordinate set for the virtual graphics device (all workstations).
3) Device Coordinates (DC), one coordinate set per workstation, representing the actual dimensions of the workstation.

Modeling transformations manipulate an object in WC to form one or more desired objects. Viewing transformations map WC to NDC. Image or segment transformations map NDC to NDC. Workstation transformations map NDC to DC for graphics output to the workstation, and DC to NDC for graphics input to the program.

6.4.4.6 Graphics Input

Most interactive graphics display devices support one or more graphics input devices. Some input devices (e.g., a crosshair cursor or a lightpen) are part of the graphics device. Other input devices (e.g., a tablet) may be interfaced to the device as a peripheral.

Six virtual input functions are required:

1) BUTTON returns a positive integer value.
2) LOCATOR returns a single virtual coordinate (X,Y) pair.
3) VALUATOR returns a floating point value in the range [0.,1].
4) KEYBOARD returns a string of characters.
5) STROKE returns a stream of virtual coordinate pairs.
6) PICK returns the name of a visible segment.

6.4.4.7 Inquiry, Error Processing, Special Device Functions

The applications program can inquire about the state of the graphics and the state of the device. The Kernel will return such values as the current position, the borders of the window, current color, text gap, and line style.

The device drivers will return various characteristics of the initialized display device.

When an error is detected, the Kernel builds an error message and logs the message on an error report file. All errors are assigned a severity level, and the applications program controls which error levels cause the program to abort. The application also sets the debug level to determine the verbosity of traceback messages.

The Kernel should support device-independent capabilities that allow the applications program to utilize special features available on many graphics display devices. Some examples are the pause function, sending messages, or downloading the color look-up table. Other special hardware features of a specific display device may be accessed using escape functions. Common escape functions are conic generators, program-defined fonts, and bit plane selection on a raster device.

6.4.5 Metafile Interface

The Metafile is a system for filing graphics information for the purpose of external long-term storage and exchange. It is a sequential file and may be thought of as the "audit trail" of device-independent picture information that would normally be sent to the graphics display. The Metafile is treated as any other graphics device and has its own device driver which communicates with the Kernel.

The interface consists of the Metafile device driver and a Metafile translator. The Metafile translator is an interactive program that will postprocess a Metafile to any supported graphics device. It facilitates such applications as picture archival, editing, transfer, hardcopy and previewing.

The Metafile is host independent and provides a common interface to the device drivers.

7. IMPLEMENTATION OF FUTURE GRAPHICS SYSTEM

The implementation of a graphics system can be viewed as two parallel processes: one of hardware implementation and the other of software implementation and integration. In both processes the replacement of obsolete or non-functional components must be effected.

In the case of hardware, equipment must be identified which can no longer meet the requirements issued by the users of the graphics system. Once this identification is complete appropriate action, such as replacement and/or upgrade of equipment, can take place.

Once replacement equipment has been acquired and equipment upgrades made, appropriate measures must be taken to determine certain operating factors such as location, maintenance, and access of this equipment. This integration of hardware components would complete the hardware portion of the implementation of a future graphics system.

The software process closely parallels that of the hardware implementation and integration. As in hardware identification of components of the existing graphics software that need replacement and/or upgrade must be made. Once identified, this software can be either replaced or modified with software that will satisfy the users' requirements.

In order to effect this the selection of appropriate software must be undertaken (refer to section 6.2). Once the selection of this software has been made, integration of those components into the graphics system must be accomplished. With the completion of this step the software process is completed. Hence, the two processes are completed and the graphics system can be implemented.

The successful implementation of the graphics system is dependent upon two requirements. First, there must be an orderly transition from the current graphics system to the future graphics system; and second, minimization of impact on current users of the graphics system as the transition takes place. This applies to both the hardware and software portions of the evolving graphics system.

Therefore, a stepwise implementation is proposed to (1) effect an orderly transition from the current graphics system to a future system, and (2) minimize the impact on present graphics users.

The implementation of the proposed graphics system can be partitioned into five phases. These five phases represent logical steps in the implementation of such a system:

Phase 1

In phase one of the implementation three concurrent tasks will be undertaken. The first task is to evaluate and select the Kernel software, the "nucleus" of the graphics software system. The selected Kernel must meet requirements previously stated (refer to 6.4.4.1); (1) it must be host-independent and device-independent, and (2) it must provide the necessary software foundation on which to build the remainder of the graphics system.

It is important to determine the source of the Kernel software. Two alternatives exist: (1) develop the Kernel software in-house, or (2) obtain the software from existing commercial and/or public domain packages. In-house development could be cost-prohibitive and counter-productive in that repetition of previous efforts will be undertaken in the development of the software. Obtaining the Kernel software from existing packages would prevent "reinventing the wheel" and provide support for the Kernel software, freeing in-house developers to concentrate on applications.

Concurrently, the initial content of the Common Library (refer to section 6.4.3.2) would be determined during this phase by the appropriate parties. Additionally, applications-specific user committees would be formed during this phase so that the initial content of the applications-specific software libraries could be determined.

Phase 2

Phase two of the implementation involves three tasks. First, having chosen the Kernel software and the initial content of the Common Library, the implementation of this software would be undertaken. During this implementation rewrites of any device- or host-dependent code would be accomplished so that the Kernel and Common Library software could be implemented on any host using any available device.

Second, device drivers would be implemented at this phase so that the integration of the "nucleus" of the graphics software system with the hardware components could be effected. The third task of this phase is the development of interfaces to allow the current graphics user to function productively while the integration of the new system is taking place. In other words, existing graphics software should function as it did prior to the integration of the future graphics system. The idea is to cause minimal impact to the user community while this system is being put in place.

The integration of the current graphics software with the future graphics system could be accomplished by one of three possible methods. First, the existing graphics software could coexist with the future system. This approach would lead to many problems such as maintenance of two or more separate systems, lack of interfaces between the systems, and multiple documentation, to name just a few.

Another approach is to develop interfaces at the device driver level for the existing graphics software so that all devices can be utilized by this software. A final approach would be to develop "look-alike" routines which would utilize calls to the Kernel software rather than existing software primitives. This approach implies use of the Metafile to insure device independence. These "look-alike" routines would function transparent to the user and would give the user access to the full range of devices addressed by the Kernel software.

With the completion of phase two, a basic graphics software system would be available to the user community to an extent that many graphics requirements dictated by the users could be met. All Kernel and Common Library software and the DD/DI level would be in place for program development.

Phase 3

Phase three of the implementation of the future graphics system would deal with the formation of the applications-specific layer software. Working from phase one to this phase the individual committees responsible for deciding the initial content of each of the applications-specific layer libraries have made a determination of that content. Once this has been completed commercially-available and/or public domain packages would be evaluated for possible inclusion. It is important to note that based on system design goals stated previously (see section 6.1) any routines, programs, etc., selected must be compatible with all other portions of the system.

Once the evaluation and selection of these packages have been accomplished, the amount of necessary in-house development will be defined. Appropriate measures can be taken at this point towards the development of the necessary software.

Phase 4

Once the content of the applications-specific layer has been identified and the source of the software components determined, phase four can begin. This phase involves the implementation of the applications-specific layer graphics software. This applications-specific software will be implemented using the Kernel software as the "building blocks" for the higher-level graphics routines. Each of the graphics software routines in the individual applications-specific layer libraries will conform to design goals stated in section 6.1 and use the Kernel software as a guide to the function of the library routines in such items as error handling, and parameter passing. It is anticipated that the respective user committees responsible for this layer's libraries will coordinate testing and validation of their appropriate library.

Phase 5

The final phase of the implementation will involve the design and subsequent implementation of the Menu/Command Driven Interfaces. These interfaces will be "attached" to the appropriate applications libraries (see section 6.4.2 for further details on the interfaces) to provide a method for users to access the graphics software without developing a program. During this phase existing software packages that have not previously been interfaced to the new system could be interfaced at this point since the Kernel, Common Library, and Applications-Oriented Libraries have been implemented.

GLOSSARY

ACD	Analysis and Computation Division
ANSI	American National Standard Institute
Applications-Oriented Library	A subroutine library which produces graphics output for a category of applications.
Audit trail	Metafile
CDC	Control Data Corporation
Central site	Facilities located in the Analysis and Computation Division.
COMET	Communication software (residing on PRIME) between the PRIME computers and CDC computers.
Common Library	A subroutine library which satisfies LaRC local features and common requirements not supported by the other libraries.
CORE	A proposed graphics standard developed by the ACM Special Interest Group on Graphics (SIGGRAPH).
DC	The device-dependent coordinate system which represents the limits of the display device.
DD/DI	Device dependent/device independent.
DDI	Device dependent interface.
Device driver	A device-dependent program that supports a graphics device. The device driver generates device-dependent output from device-independent input and handles device-dependent interaction.
Device Independence	The ability to control all graphics devices uniformly.
FSGB	Flight Software and Graphics Branch of ACD.
GKS	A graphics standard approved by ISO.
GWG	Graphics Working Group of FSGB.
IGL	A library of graphics subroutines written by TEKTRONIX that supports the interactive graphics of the TEKTRONIX 401X, 402X, and 411X graphics terminals.
Interactive graphics	Graphics which allow dynamic modification of the display through graphics input devices.
ISO	International Standards Organization

Kernel	A subroutine library which contains all required functions for performing interactive and passive graphics tasks.
LaRC	Langley Research Center
LARCGOS	A library of graphics subroutines written in-house and a set of post processors that drive the ACD production graphics devices.
Local site	Facilities located at sites around LaRC other than at the central site.
Menu/Command Driven Interface	A collection of programs which prompts the user to provide the appropriate inputs to get the desired graphics.
Metafile	A sequential file which contains the device-independent picture information normally sent to the graphics device.
Metafile Translator	The program which interprets the device-independent Metafile commands for specific physical devices.
MOVIE.BYU	A graphics display system written by Brigham Young University, implemented at LaRC.
NCAR	A library of graphics subroutines written by the National Center for Atmospheric Research.
NDC	The addressable surface of the display defined in device-independent coordinates in the range from 0 to 1.
Network (LaRC)	A Center-wide medium-speed (50K baud-30M baud) data network will be implemented at LaRC to support minicomputers, work-stations, and personal computers in a distributed environment, allowing responsive resource sharing among network devices.
NOS	Network Operating System
Open shop	Facilities located in the Analysis and Computation Division that are available for scheduling.
Passive graphics	Graphics requiring no dynamic interaction with the display.
PLOT10	A library of graphics subroutines written by TEKTRONIX but modified locally that supports interactive graphics for the TEKTRONIX 401X series terminals.
PLOT10 PVF	PLOT10 device-independent Plot Vector File.
PRIMENET	PRIME ring network.
Primitives	The basic graphical entities that define objects in a 3-D world coordinate system.
PRIMOS	PRIME Operating System.

Postprocessor	A device-dependent program that drives an ACD production graphics device.
PVF	Device-independent Plot Vector File.
SCS	Satellite Coupler System that links the CPFS and WFS file clusters.
Segment	A collection of output primitives which can be named.
Transformation	A function which modifies the display by introducing rotation, scaling or translation.
User-developed program	A program written by an applications programer.
Virtual graphics device	The union of capabilities available on all supported graphics devices.
WC	A device-independent coordinate system used to define objects meant for display.
Workstation	A unit consisting of zero or one display surfaces and zero or more input devices.
2D	Two-dimensional
3D	Three-dimensional

REFERENCES

[NEWM79] Newman, W. M., Sproull, R. F., Principles of Interactive Computer Graphics, McGraw-Hill Book Company, 1979, pp.79-81.

[ZELK78] Zelkowitz, M. V., "Perspectives on Software Engineering," Computing Surveys, Vol. 10 (2), pp. 197-216, June 1978.

[BONO82] Bono, P. R., Encaracao, J. L., Hopgood, F. R. A., ten Hagen, P.J.W., "GKS-The First Graphics Standard," IEEE Computer Graphics and Applications, Vol. 2 (5), pp. 9-23, July 1982.

[SCHO82] Schoenhut, Juergen, VIM Conference Proceedings for VIM-36/ECODU 33/VIM Australasia-19, St. Paul, Minn., pp. 100-105, April 1982.

[GKS 82] Draft International Standard ISO/DID 7942, "Information Processing Graphics Kernel System (GKS) Functional Description," August 1982.

[GSPC79] Graphic Standards Planning Committee, "Status Report of the Graphic Standards Planning Committee," Vol. 13 (3), August 1979.

[WARN81] Warner, J. R., Principles of Device-Independent Computer Graphics Software, IEEE Computer Graphics and Applications; Oct. 1981; pp. 1-8, pp. 99-100.

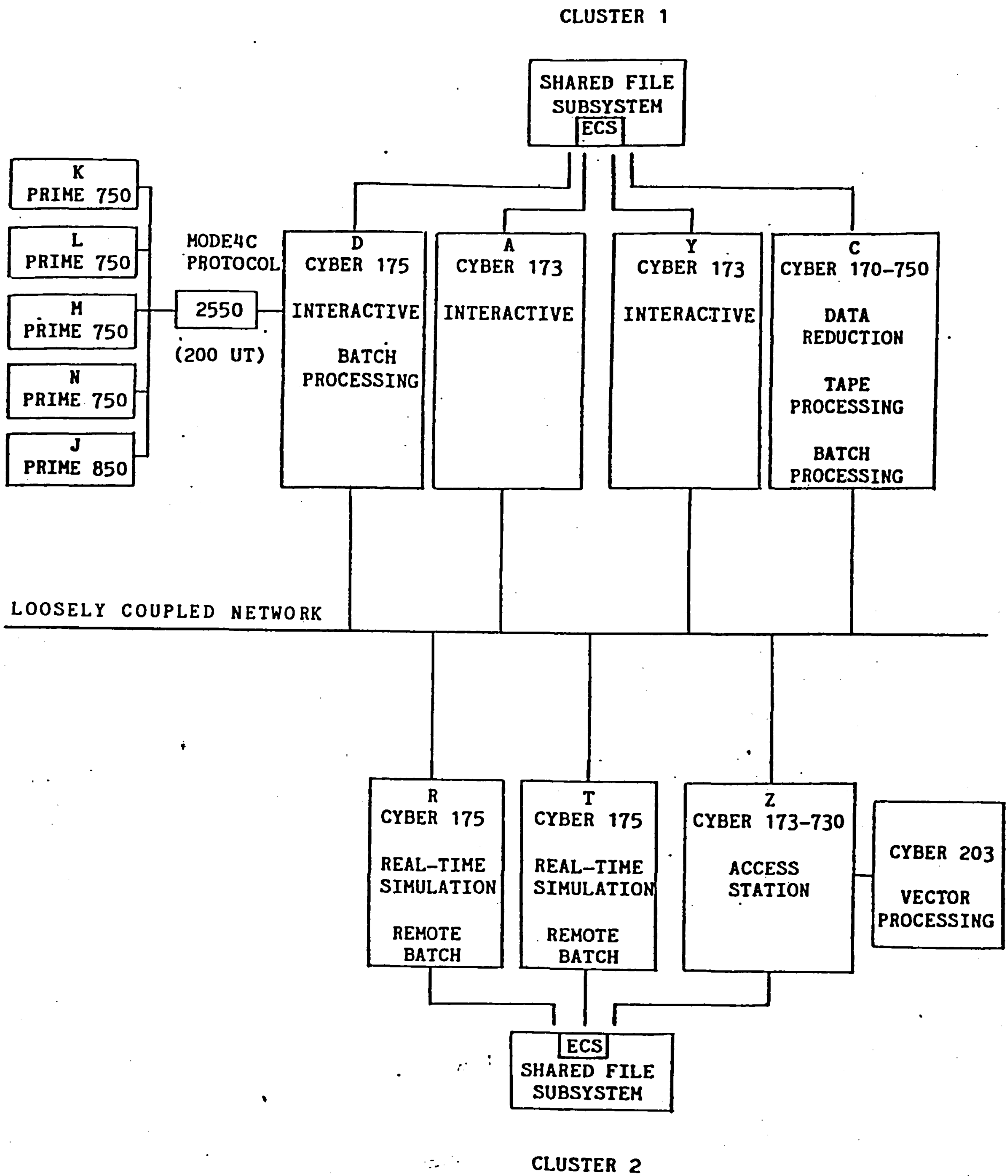


Figure 1. - Current LaRC Scientific Computer Complex

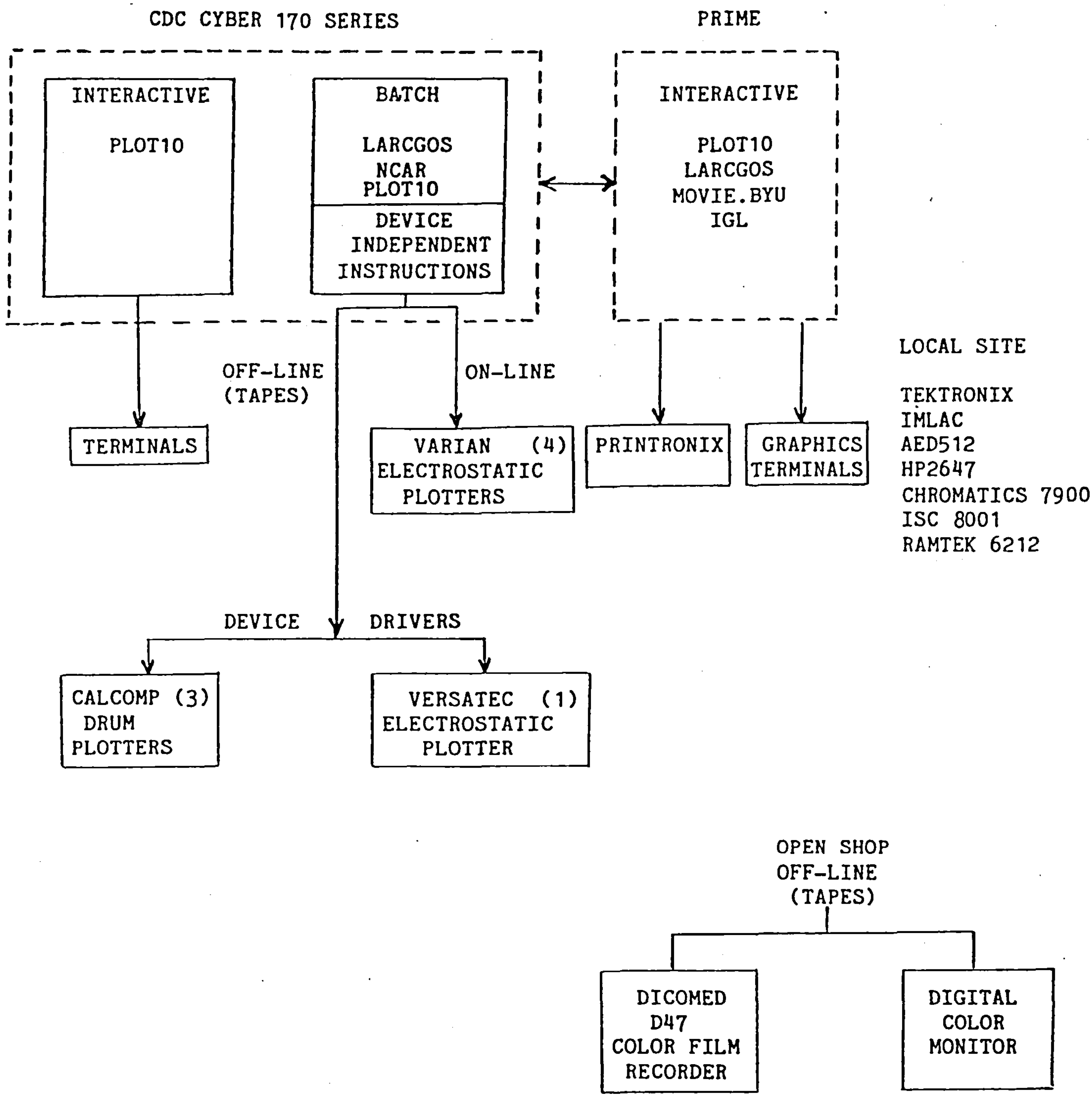
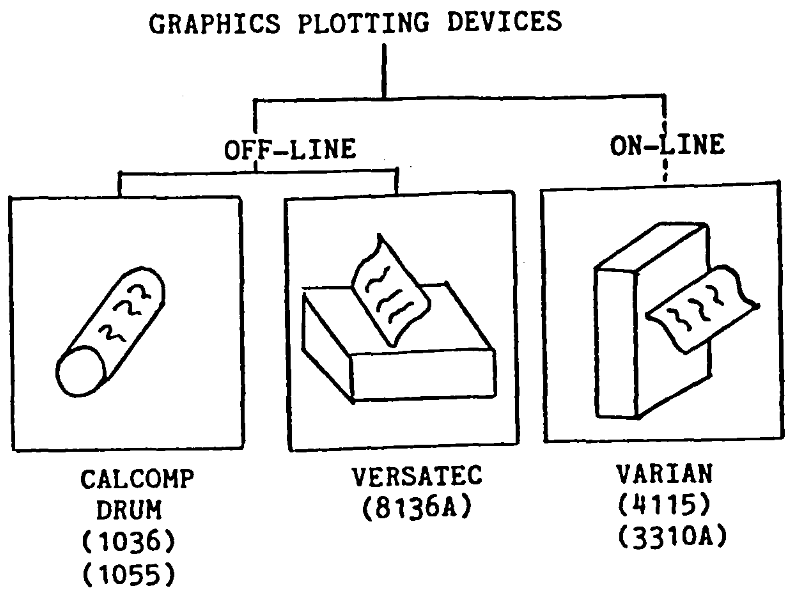


Figure 2. - Current Graphics Hardware Configuration



	CALCOMP DRUM	VERSATEC	VARIAN
TYPE	PEN	ELECTROSTATIC	ELECTROSTATIC
RESOLUTION	.002" (1036) .0005" (1055)	.01"	.01"
MAXIMUM PLOT SIZE	33" x 110' 11" x 110'	35" x 110'	13.91" x 28.79"
USES	FINAL PLOTS PREPRINTED GRID PAPER 33" x 110' 10" x 110'	FINAL PLOTS	DEBUG AND WORKING PLOTS

Figure 3. - Function of the ACD Production Graphics Devices

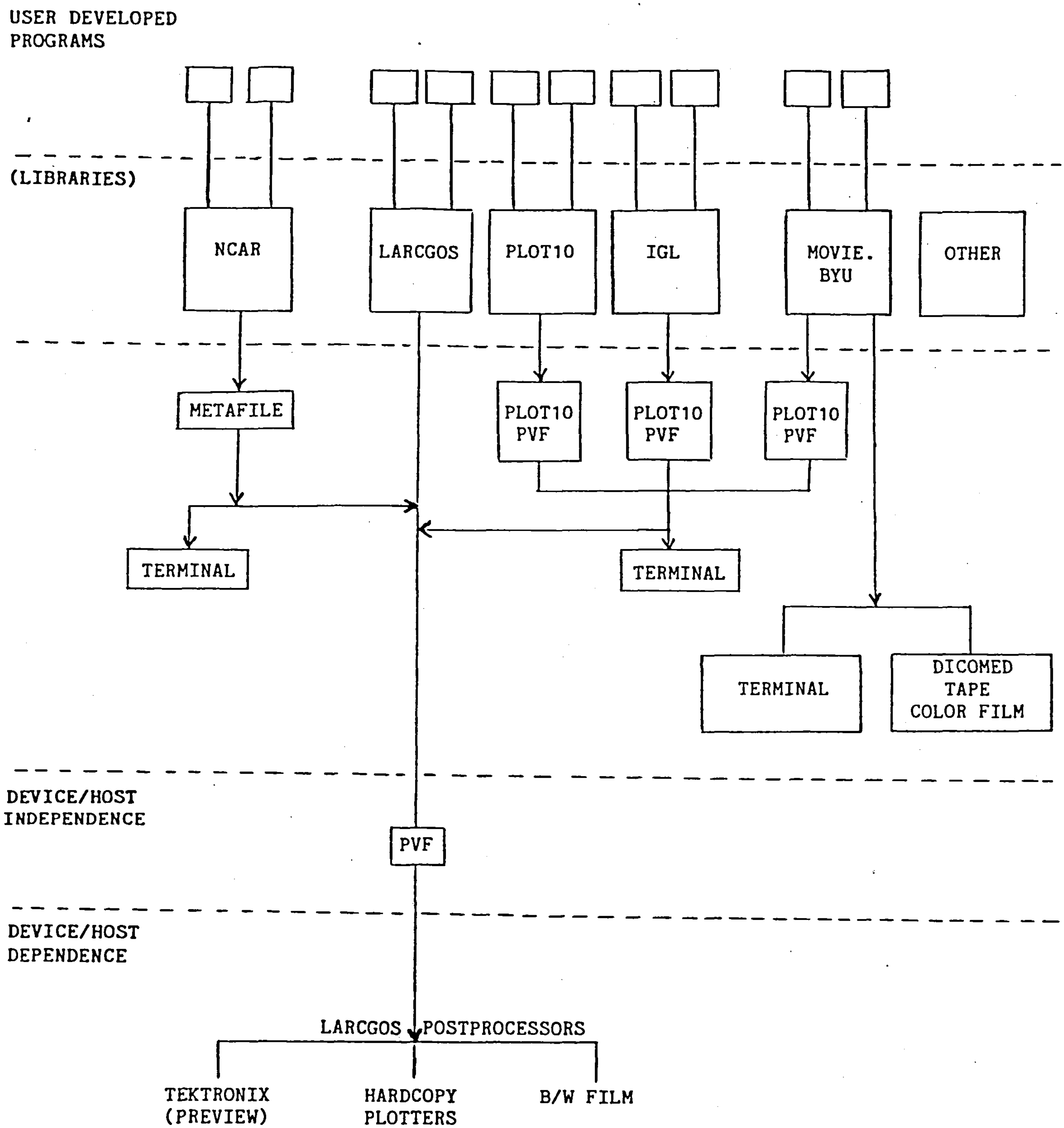


Figure 4. - Current Graphics Software Systems

	ACD PROD	ACD OPEN SHOP	ACD FSGB	LOCAL SITE
PRODUCTION COLOR FILM/SLIDE/MICROFICHE	X			
VIDEO OUTPUT	X	X		
HIGH RESOLUTION HARDCOPY (MULTIMEDIA)	X			
(TRANSPARENCY,PAPER)				
COLOR AND B/W				
FILM		X		
LOW RESOLUTION HARDCOPY	X		X	X
COLOR AND B/W				
PLOTTERS				
CAMERA				
DOT MATRIX PRINTERS				
HIGH PERFORMANCE LOCAL WORKSTATION		X	X	X
REFRESH VECTOR SYS				
COLOR RASTER				
MID PERFORMANCE TERMINALS		X	X	X
VECTOR REFRESH				
COLOR RASTER				
LOW LEVEL TERMINALS			X	X
MICROCOMPUTERS				
B/W MONITORS				
INPUT MECHANISM				
WRITING TABLETS		X		X
CAMERAS		X		X
3D DIGITIZER		X		X
DISPLAY				
LARGE SCREEN DISPLAY SYSTEM (ON-LINE)				
PROJECTORS (OFF-LINE)				

Figure 5. - Proposed Hardware Configuration

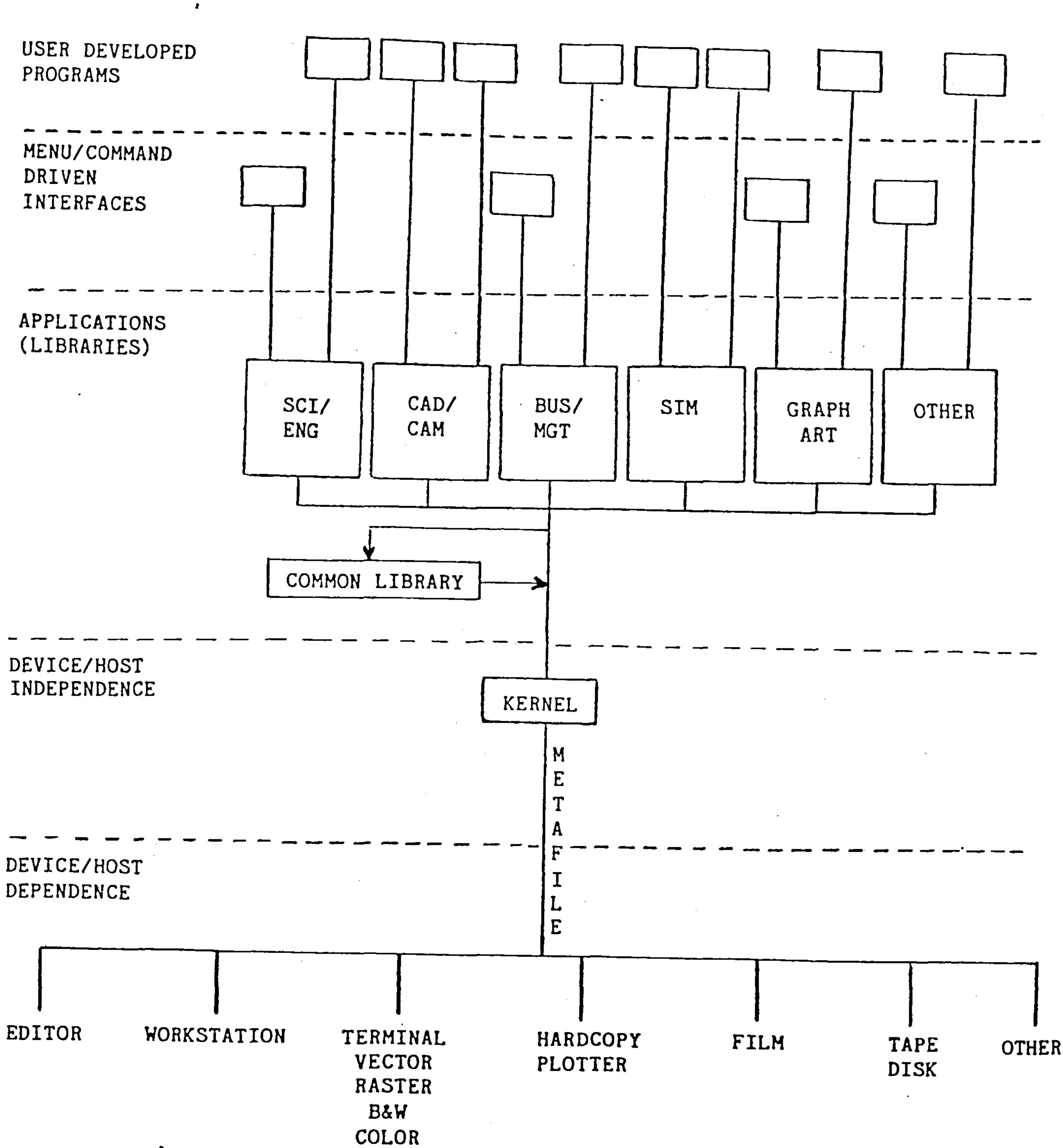


Figure 6. - Proposed Graphics Software System

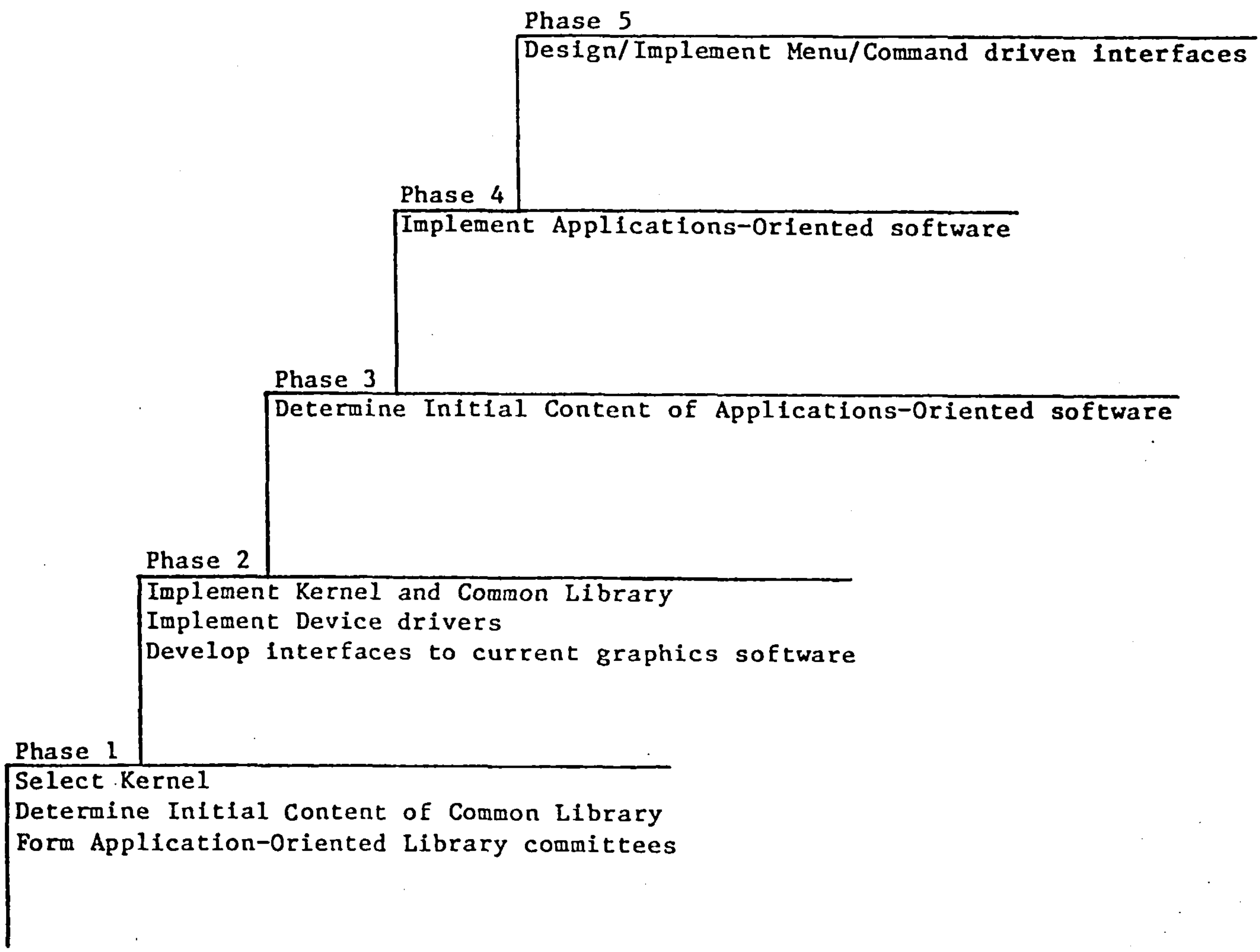


Figure 7. - Stepwise Implementation of Graphics System

1. Report No. NASA TM-85726	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle CURRENT AND FUTURE GRAPHICS REQUIREMENTS FOR LaRC AND PROPOSED FUTURE GRAPHICS SYSTEM		5. Report Date June 1984
		6. Performing Organization Code 505-37-23-01
7. Author(s) Nancy L. Taylor, John T. Bowen, Donald P. Randall* and Raymond L. Gates*		8. Performing Organization Report No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Virginia 23665		10. Work Unit No.
		11. Contract or Grant No. NAS1-16078
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Technical Memorandum Nov. 1982--Nov. 1983
		14. Sponsoring Agency Code

15. Supplementary Notes

*Computer Sciences Corporation, Hampton, Virginia

16. Abstract

To report the findings of an investigation to assess the current and future graphics requirements of the LaRC researchers with respect to both hardware and software, and to propose a graphics system designed to meet these requirements.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Graphics software Graphics hardware	Unclassified - Unlimited Subject Category - 60

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	44	A03

N-305

For sale by the National Technical Information Service, Springfield, Virginia 22161