Software Requirements Specification

Steve Wampler

Gemini Software Design Team
Gemini Project

Preliminary Design Review Document
Last Revision: 5/17/95

Document Number: SPE-C-G0014
Issue Number: 2
Revision Number: 0
Date: June 6, 1995
Approved By: R.J. McGonegal
Systems: J. Oschmann
Science: M. Mountain
Project: D. Kurz
Proprietary: NO
Procurement Sensitive: NO
Disclosable: YES

Action Items

The following items remain unresolved at this time:

1. Definition of a G8MT standard for acquisition and storage of detector data. See "Data flow" on page 3-2.
2. Link chosen to transfer data. See "Data flow" on page 3-2.
3. Desirable hardware specification. See "Hardware constraints" on page 3-12.
4. G8MT standards for online software and development environment.See "Software constraints" on page 3-13.
5. Supportability plan. See "Supportability" on page 3-15.
6. Descriptions for star catalogs. See "Software Interfaces" on page 4-10.

1 INTRODUCTION

1.1 PURPOSE

The document defines the operational requirements for Gemini Control System software. The goal of the SRS is to provide a functional tool for guiding the development of controls and data acquisition systems in a manner consistent with their operation within the Gemini System. It establishes both general criteria and specific functional requirements for software and controls design in the Gemini Project.

1.2 SCOPE

It is oriented toward the developers of control and data acquisition systems, not toward the science user of the Gemini telescopes. As such, it logically follows the Operational Concepts document, which addresses the control of the Gemini system from a more general perspective. The impacts of the science requirements, as well as requirements of control system design are covered in this document.

1.3 APPLICABLE DOCUMENTS

This document is based on information found in the following documents:

TABLE 1 - 1 **Related Documents**

Reference Number	Title	Date
SPE-PS-G001	Gemini Science Requirements, Version 1.1	November 11, 1992
SPE-C-G0026	Goals and Requirements for Software and Controls	August 23, 1993
SPE-C-G0015	Operational Concept Definition	April 26, 1993
PG-C-G0005	Software and Control Management Plan	April 5, 1993
SPE-C-G0037	Software Design Specification	April 20, 1994
SPE-C-G0011	Software Configuration Control Plan	August 23, 1993
SPE-ASA-G0008	Gemini Electronic Design Specification	June 21, 1994

1.4 GLOSSARY, ABBREVIATIONS, AND ACRONYMS

A&G	Acquisition and Guiding
Access Mode Allocation	A method of allocation resources that works using both access method and access privileges. For example, the Telescope Operator may be able to issue direct commands to the TCS, even while the TCS is under the control of a planned observing program. An on-site observer might have privileges that allow the monitoring of systems that are allocated to others.
ACK/NAK	Any communication protocol that mandates either a positive or negative acknowledgment to all messages.
(network) Bridging	The use of hardware to isolate sections of a network, so messages with both source and destination on the same section are not transmitted to other sections.
CASE	Computer-Aided Software Engineering
CLUI	Command Line User Interface
Data-loss compression	Any compression algorithm that results in the loss of original data (resolution) after uncompression. Typical compression savings from these algorithms depend upon the amount of data loss that is acceptable, but can exceed 99%. Also see ‘loss-less compression’.
DBMS	Database Management System
Eavesdropping	Allowing an observer to watch an observation from a secondary site
ESO	European Southern Observatory
FITS	Flexible Image Transport System
GUI	Graphical User Interface

(command) Handshaking	A software protocol where subsystems acknowledge acceptance of a command. Unlike ACK/NAK, handshaking acknowledges acceptance of the message contents.
HOS	High-level Operations Software
ICS	Instrument Control Software
IOC	Input/Output Control Module
IPC	Interprocess Communication
LAN	Local Area Network
Loss-less compression	Any compression algorithm that results in no data loss on uncompression. Typical compression savings with loss-less compression are 25-75%. Also see data-loss compression.
NFS	Network File System
Observing Modes	The ‘styles’ of observing supported by the Gemini Control System
OCS	Observatory Control Software
Planner	A software-environment for developing observing software
RDBMS	Relational DBMS
ROM	Read-Only memory
RPC	Remote Procedure Call
Scheduler	The portion of the Gemini Control System that is responsible for dispatching Science Programs to the Sequencer for execution.
Science Program	A formal description of an observers plan for using the Gemini 8-m Telescopes, suitable for near-automatic execution.
Sequencer	The command interpreter for Science Programs and telescope control
SRS	Software Requirements Specification (this document)
STARCAT	Space Telescope Archive and Catalog
TBD	To Be Determined
TCS	Telescope Control Software
VLT	ESO’s Very Large Telescope
VOS	Virtual Observatory System - a model for designing the Gemini Control System that seeks to minimize both intersystem dependencies as well as hide subsystem implementation details from system level operations.
VSI	Virtual System Interface - a layer of interface abstraction within the VOS that serves to isolate and hide implementation-specific interface requirements.
WAN	Wide-Area Network

1.5 OVERVIEW

Gemini software is all software directly involved in the control of Gemini systems, including commercial packages used for that purpose. However, only the non-commercial portions of this soft ware are addressed in detail here, along with the interfaces to those commercial packages.

If this document is viewed as presenting a set of standards for software development, then the software used with the Gemini telescopes can be viewed as falling into the following categories:

A. *Developed software* - non-commercial software developed for the control of the telescopes and instrumentation. This software falls under the specifications presented here and this document can be viewed as a guide to developing that software.

B. *Supported software* - commercial or “public-domain” software used for telescope control and instrumentation. One of the design goals for Gemini software is to use as much off-the-shelf software as possible, in an effort to decrease life-cycle costs and maintenance concerns. However, the nature of this software makes it difficult to force into the specifications here. Consequently, this document presents the interface specifications required of such software and encourages potential adopters of this software to evaluate it in terms of the specifications presented here.

C. *External software* - any software available to the Gemini system that is not integral to the system. User-supplied data reduction tools, operating systems, CASE tools, etc. are generally considered external software. Only the interface requirements to such software is considered part of the Gemini software.

Software that is embedded into hardware and that presents no software interface to the Gemini system is also considered external software and need not be considered further. Embedded software that does interface with the Gemini system via software is considered Gemini software and is subject to the requirements presented in this document.

All software will be maintained under a version control system, and must include facilities for easy (nearly automatic) inclusion into the Gemini software environment.

1.6 CREDITS

This document draws heavily in form and general content from ESO's VLT Software Requirements Specification, Issue 1.0 (VLT- SPE-ESO-10000-11).

2 GENERAL DESCRIPTION

2.1 SYSTEM USERS

The users of the Gemini system are classified into the following categories.

A. *Astronomer*. This person is using the Gemini system for the collection of science data. In a very real sense, the astronomer is the customer for the services provided by the Gemini telescopes. The astronomer has worked out, with the Gemini system, a science plan for the collection of the data. This science plan may include interactive observing.

B. *Science observer*. This is the on-site person responsible for monitoring the data acquisition and validating the data integrity being collected for the astronomer, as well as ensuring that the science plan is functioning to the needs of the astronomer.

C. *Telescope Operator*. The on-site controller of the telescope and instruments. This person is responsible for ensuring the integrity of the system and for keeping the system functioning accurately during observations. The Telescope Operator works with the Observer and the science plan to produce as good data as is possible.

D. *Support*. On-site (or near-site) support personnel are responsible for the maintenance of the system, hardware and software, as well as the installation of subsystems and configuration changes.

E. *Developer*. Developers are responsible for the designing, testing, configuring, and upgrading of subsystems.

F. *Administrator.* Administrators are responsible for high-level functional control of the Gemini system as an integrated system. They work with other users to determine appropriate observation scheduling changes, maintenance down-time, and system modifications. It is possible that a particular individual would fall into several of the categories. For example, it is not unusual for the astronomer to also be the science observer.

2.2 OPERATIONAL LEVELS

The Gemini system, when powered on, exists in one of several disjoint *operational levels*. Access to the system is restricted according to the current level of operation.

The operational levels are:

A. *Observing level*. The observing level is the "normal" operational mode of the system.

B. *Maintenance level*. Maintenance level permits access to all subsystems for routine maintenance and diagnostic work. The telescope is typically at maintenance level during daylight (non-observing) hours.

C. *Test level*. The most primitive operational level, test level operation is used for installation/deinstallation of subsystems, including full diagnostics and installation calibration activities.

2.3 ACCESS MODES

At any level, but particularly during observing level operation, the software imposes on the Gemini system a series of *access modes*. These access modes balance ease and convenience of use against flexibility, control, and the security of the system.

The access modes provided by the Gemini system are:

A. *Observing*. This mode provides simple and safe access to the system for the collection of science data, selection of instrument parameters, and quick-look data reduction. Access to the system is through the sequencer with no direct control of telescope and instruments

B. *Monitoring*. This is a special, read-only case of the observing mode that can be invoked on a subsystem basis. Under no circumstances should monitoring affect the performance of an ongoing observation.

C. *Operation*. This is the access used for direct control of the telescope and instruments, typically during observation level operation. It is normally available only to the Telescope Operator and the science program sequencer when at operation level.

D. *Planning*. Access to the Gemini system is provided during science planning. Actual access to the telescope is not permitted in this mode, but the virtual telescope capability of the Gemini system provides a telescope simulator that is useful for planning observations, as are on-line databases.

E. *Testing*. Test access allows full, direct control of any subsystem. All features of the subsystem are available in this mode. Under no circumstances should testing affect the performance of an ongoing observation.

F. *Administrative*. It is possible to inquire about system utilization, efficiency, etc. using administrative access. No control is available in this mode, only status and scheduling information is accessible. Under no circumstances should administrative access affect the performance of an ongoing observation.

It is entirely possible for a single user to be accessing the system through several modes simultaneously. In fact, this is the typical situation with the Telescope Operator, who is often concurrently accessing the system through the observer, monitor, and operator modes. There are times when an astronomer might be using both observer and monitor modes.

2.4 LOCATIONS OF USERS OF THE SYSTEM

Users of the system will be located at one or more facilities. These are referred to in general as Gemini 8m Telescopes operations facilities. The Gemini 8m Telescopes enclosure and Gemini 8m Telescopes control facility are referred to collectively as the Gemini 8m Telescopes site.

2.4.1 Gemini 8m Telescopes Enclosure

This is the enclosure housing the telescope. In general such use would be limited to maintenance and engineering.

2.4.2 Gemini 8m Telescopes Control Facility

This is the facility collocated with the enclosure. It is from here that nightly operations would take place.

2.4.3 Gemini 8m Telescopes Site Support Facility

This is the facility where off duty personnel reside while stationed on the summit. This would be the Hale Pohaku facility on Mauna Kea and Cerro Tololo in Chile (assuming no support facility is constructed on Cerro Pachon).

2.4.4 Gemini 8m Telescopes Base Facility

This is the facility where offices, shops, etc. are located. On Hawaii this would be with the JACH in Hilo and in Chile it would be either the La Serena or Cerro Tololo facilities.

2.4.5 Gemini 8m Telescopes Remote Operations Facility

These are super sites, in general located in partner countries, which have direct links to the telescopes via communications lines. They may or may not be collocated with any administrative organization associated with the Gemini telescopes. These sites have remote operations capability.

2.4.6 Satellite Site Facility

These are sites with high bandwidth connections to the super sites. They have remote operations capability but at a reduced level.

Will these high bandwidth connections be private or public, or doesn't it matter? I don't understand the different between a remote operations facility and a satellite site. Why does the satellite site not just connect directly to the telescope? — Steven.

2.4.7 Institutional Site Facility

These are sites, generally located at existing observatories, universities, etc. that have some form of connectivity, perhaps via satellite sites and/or super sites, to the telescopes. Although they have remote operations capability, the link capacity will determine at what level.

Does this statement mean that the Gemini system has to support remote operations at a level which depends on the remote site's capability? Perhaps a table showing what is achievable as the capabilities of the site get better would be useful. The sort of thing I have in mind is:

TABLE 2 - 1 Remote Operations Capability

Capability of site	Remote observing capability
High speed link direct to Gemini telescope — same speed as Gemini Control and Data LANs.	Full control and monitoring capability from remote workstation (but no access to event and synchro buses).
Slower communication link direct to Gemini telescope.	Remote control and monitoring by copying the operator's screen (with some data compression).
"	"
"	"
A PC, printer, ISDN[a] connection and a modem.	A voice communication link, compressed digitized video link, FAX, and the ability to log in and examine the status files and observing logs.
A telephone line only.	A voice communication link only, plus FAX if a machine is available.

a. Integrated Services Digital Network — A network which allows selectable voice, video or packet communication between two or more devices across a telephone line.

2.5 USER-LEVEL REQUIREMENTS

2.5.8 User Needs

Each class of users has different needs of the system. The general requirements of each user class are summarized below.

TABLE 2 - 2 User-level requirements

User Class	Operation Level	Access Mode	User Interface
Astronomer	Observing	Planning, observing, monitoring	Simple, safe, controls grouped for convenience of observation activity.
Science Observer	Observing	Observing, monitoring	Same as for astronomer.
Telescope Operator	Observing, maintenance, test	Observing, monitoring, operation, testing	Organized for ease of control and quick response to unsafe conditions, ability to monitor all subsystems as needed.
Support	Maintenance, test	Monitoring, operation, testing	Must allow full access to all subsystems.

TABLE 2 - 2 User-level requirements

Developer	Maintenance, test	Monitoring, testing	Must allow full access to each subsystem in test.
Administrator	Observing, maintenance, test	Monitoring, administrative	Simple, safe, status inquiries only.

Also, from a human engineering perspective, user interfaces for different access levels should be uniform across all subsystems, though different access levels should present different 'look-and-feels'.

2.5.9 Users of the System

Unless the contrary is explicitly stated, all requirements and modes presented are intended to be available to all users of the system.

It should also be noted that what the system has to guarantee in terms of operations must fulfill all possible requirements, but it does not mean that all features will be made available to all users at all locations at all times.

The available capabilities will have to be adapted to the operational needs both locally and remotely and these decisions can only be taken later in the life of the project.

2.5.9.1 ASTRONOMERS AND OBSERVERS

- Definition

  Observing astronomers are the end users of the system. They range greatly in experience from occasional users of the system to very experienced users such as staff astronomers at the Gemini 8m Telescopes site and service observers.

- Purpose of work

  Whatever their level of experience, observing astronomers will wish to concentrate on the efficient acquisition of astronomical data and on-line assessment of data quality, rather than on the details of controlling the telescope and instruments.

  To allow observing astronomers to achieve this objective, the system must offer to a user an interface which, while fulfilling the various operational requirements in the different modes and offering status information both automatically and on request at any required level, is still simple to learn and secure in its use.

- Privileges

Observing astronomers shall have no privileges as far as the direct control of the telescope is concerned. They shall not be able to send control commands directly but they must be able to enquire about the status of the telescope or any subsystem at any time. The intent is not to restrict the capabilities of the observing astronomer in any way but rather to establish a single point of control and responsibility. Programs, as opposed to observers, may have the capability of direct control of the telescope. This would allow the observer to create an observing program which requested a telescope control function but would not allow the observer to enter (for instance) a command to slew the mount. Astronomers are typically given control access to instruments, however.

- Operation mode

Traditional interactive operation shall normally be replaced by operation via an automatic sequencer. This is essential to support operational requirements such as service observing and flexible scheduling (as defined later in Section 2.7 on page 2-12).

A certain degree of interaction shall be provided, meaning in this case that the user will interact with the scheduler program, rather than with the control programs directly.

Observing commands will normally be submitted via the User interface to a queue for later execution. It must also be possible to break and resequence this queue. e.g. as a result of the quality assessment of previous data.

In particular, Operations staff will be able to enable direct interactive operation, but this shall not be considered as the normal operation mode for the reasons explained in Section 2.7 on page 2-12. It is evident that, for some functions (such as adjustment of spectrograph slit width for seeing conditions) it must be necessary to include interactive capability. However each instance of such a function should be examined as a candidate for automation - such as focussing.

The existence of the scheduling queue shall be transparent to the on-site observer during the initial phases of telescope operation. Only after experience has been gained with the system will the existence of the queue become evident to the on site observer.

2.5.9.2 OPERATIONS STAFF

- Definition

Currently, operations staff includes 'night assistants' and staff of the 'operations group'. Depending on the operations implemented on the Gemini 8m Telescopes, they may become telescope operators, operations supervisors of the Gemini 8m Telescopes telescope, and instrument operators and have to overview the Gemini 8m Telescopes operation.

These users will be very experienced users of the system. They will be very familiar with the Gemini 8m Telescopes and/or its instruments.

- Purpose of work

Operations staff will control the Gemini 8m Telescopes indirectly via a scheduler program or directly via commands. They will supervise telescope operation and will be able to advise observing astronomers on what they have to do to use telescope and instruments efficiently. They shall also monitor general performance and system safety.

- Privileges

Operations staff shall have privileges to access all commands and maintenance procedures in case of problems. This includes direct control of physical units.

However, they shall not have access to subsystems while these are in normal operation. If they need to access other parts of the system appropriate reconfiguration procedures have to be run.

- Operation mode

Operations staff shall have access to operation tables in update mode, while observing astronomers will have access to them only in read mode (for example the list of filters mounted on an instrument). Again, the intent is not to restrict the capabilities of the observing astronomer in any way but rather to establish a single point of control and responsibility. At the present time it is not clear how to handle visitor instrument requirements in this area.

They shall be able to change the operational status of units according to the results of tests performed on such units (e.g. to see if a faulty unit can be declared as operational again and redefined as part of the environment in use).

2.5.9.3 SOFTWARE DEVELOPMENT AND MAINTENANCE STAFF

- Definition

Software development and maintenance are staff:

  - based either at the Gemini 8m Telescopes site or base facility, or
  - based at the Gemini 8m Telescopes remote operations facility, or
  - non Gemini staff from a contract company or from an associated Institute.

These users will have a deep understanding of the way the Gemini 8m Telescopes software works.

- Purpose of work;

Software development and maintenance staff intervene when there is a major problem to be solved or an upgrade to be installed. They also perform system generation and installation of new software packages or new releases, according to established test and validation procedures.

- Privileges

Software development and maintenance staff need the highest privilege in order to be able to modify everything in the system. Nevertheless strict configuration control guidelines must be followed to prevent interference with ongoing normal operation.

- Operation mode

Software development and maintenance staff usually work at the test level (see Section 3.2.1.3 on page 3-4) for the part of the software under test. Other parts of the Gemini 8m Telescopes may also need to run in test mode to support integration tests. This means that a number of commands are available which allow the hardware concerned to be tested (test commands) and are not accessible to observing astronomers.

2.6 OPERATIONAL CONTEXT

This section derives directly from Gemini 8m Telescopes program requirements and specifies the operational context, which is a direct consequence of the structure of the Gemini 8m Telescopes system.

The Gemini 8m Telescopes operation has to cope with a complex environment produced by:

- multi-telescope concept
- multi-instrument context
- visitor instrument context
- multi-user context.

2.6.1 Multi-telescope context

Although there are two telescopes as part of the Gemini system there are no plans to support concurrent use of them as if they were a single facility. It is, however, a long term goal of the Gemini project to be capable of being used in conjunction with nearby, perhaps non-Gemini, telescopes on the same site (for instance for interferometry).

2.6.2 Multi-instrument context

Normally three scientific instruments are mounted on the cassegrain focus of the telescope. The Gemini 8m Telescopes are thus characterized all the time as in a multi-instrument context.

Parallel access to all the mounted instruments shall be provided, though only one instrument has access to the telescope beam (active instrument).

Instruments which feed two detectors by using beam splitters or field splitters is considered as a single active instrument. Feeding two separate instruments via beam splitters or field splitters and providing separate control nodes is not a requirement.

Various situations are possible for the other (inactive) instruments:

- they shall be able to take calibration or flat field exposures in parallel
- they shall be able to prepare for an exposure to start as soon as the telescope beam is switched back to them (in this case, they are in a hot standby situation)
- they shall be able to work; at all foreseen operation levels (observing, maintenance, test) (see Section 3.2.1 on page 3-3).

Regardless of the status of an inactive instrument, it shall not be possible for any of its permitted actions to adversely impact the active instrument.

2.6.3 Visitor instrument context

The requirement to provide for support, installation, and operation of outside instruments brought by the observer has several implications for the system. Due to the specialized nature of visitor instrumentation it is unlikely that complete integration into the Gemini environment is either feasible or warranted. In this case a subset of the available functionality must be made available through a standardized interface. The Gemini Telescopes view all instruments as operating as *servers*, responding to commands from the upper levels of the system. Visitor instruments must be capable of operating in this mode to be adequately supported.

It should be the goal of this interface that it be a subset of the existing instrumentation interface (rather than a separate system). At a minimum this interface should support:

- acquisition of status information of instrument by other systems
- capability to enter preprogrammed observing sequences
- capability to offset the telescope position and focus

It should be a goal, but not a requirement, that the interface provide support for

- coordinate transforms between visitor instrument and the Gemini system
- archiving (or archivibility) of visitor instrument data
- maintenance of operations tables (such as filter lists) which can become part of an observation's permanent record

Due to the nature of visitor instrumentation it is unlikely that more complicated functionality can be supported. In particular coordinated motions of the Gemini system components with those of the visitor instrument (other than simple raster scans) will not be supported.

This does not mean that more complicated functionality will not be possible for visitor instruments that require it. Such functionality will not be offered as a standard service but will require a joint effort on the part of Gemini and the visitor instrument team. As more complicated functionality will be supported via standardized interfaces for the Gemini instruments, such as coordinated motions, it should be possible to adapt visitor instruments to this standard. The difficulty with offering such services as an externally supported standard, as opposed to an internally supported standard, is that decisions to change internal standards do not impact external users.

It is important that the visitor instrument interface be stable and long-lived, as the time between successive uses of the same visitor instrument can be as long as one or two years.

The support of visitor instrumentation is made simpler if the visitor equipment adheres to the Gemini standards. However it must be recognized that:

- the Gemini standards may not prove cost effective for some visitor instruments
- the Gemini standards will evolve over time as requirements and technology change
- the evolution of Gemini standards might require existing visitor instruments to rewrite or rework large areas of their control system
- visitor instruments built later in time than the initial complement of Gemini instruments may want to take advantage of different and/or less expensive technology if it better suits the goals of their project

For these reasons it is probably not possible to establish a combined standard to which all instruments, both Gemini and visitor, adhere to completely. It is much better to establish a subset of Gemini facilities which will be made available to visitor instruments via long lived, stable interfaces. Visitor requirements outside of these would be handled on an as needed basis.

The support of both Gemini and visitor instruments would benefit by the provision of a Gemini observatory simulator. This simulator, appearing to the instrument as a standard set of hardware and software interfaces, would present a functional definition of the observatory.

2.6.4 Multi-user context

The requirements to have instruments operated as single units imply that several user stations will be active at the same time on the Gemini 8m Telescopes. On these user stations, different kinds of users may be working at the same time with the Gemini 8m Telescopes software.

Independently of the location of users at the telescope site, they shall be able to access (according to their privileges) any part of the whole setup with a simple logon and configuration operation. In other words. any subsection of the whole Gemini 8m Telescopes system should be accessible and controllable from any single point (but of course with protection ensuring security and safety).

A particularly simple case of multi-use of the system is multi-point monitoring. By this it is meant that, while some (active) user is in control of the Gemini 8m Telescopes, someone else can follow what they are doing by monitoring all the relevant data from telescope and instruments. This will typically be needed by the operation supervisor. All other users wishing to monitor Gemini 8m Telescopes operations have to go through the procedures set up by Operations and get permission to do so.

The multi-point monitoring mode might also be important when certain difficult or rare problems occur, when expert advice is needed and can only be obtained from colleagues situated remotely. Multi-point monitoring allows them to follow directly the results of tests performed and investigate how the system is working (e.g. by selecting different display pages with the up-to-date status information on different parts of the system).

Multi-point monitoring also allows a local observer to be monitored and advised by a remote supervisor.

Monitoring shall exist both in the form of automatic displays of status information at different locations, and in the form of explicit access to the required status information from any point.

Monitoring shall not affect the performance of ongoing observations.

2.7 OBSERVING MODE REQUIREMENTS

The different observing modes (*interactive, queue-based, remote-operations* and *service*) impose constraints on software design.

2.7.1 Interactive observing

Requirements:

A. Normally executed with the sequencer by providing a computer executable program, in order to make efficient use of the Gemini telescope.

B. Interactive operation is supported, but always through the Observatory Control System (OCS).

C. There is a visual user interface to the OCS to allow for changes to the viewing program.

D. It is possible to have more than one station participate in the observing.

Interactive observing with time allocation for full nights is a first basic requirement of the Gemini 8m Telescopes. It is therefore an essential requirement that telescope operation is supported by the software in a smooth and very friendly way in this mode.

Interaction will normally be via an automatic sequencer (see Section 2.5.9 on page 2-6). This is clearly a top priority requirement and one which will have to be realized before implementing any other mode. The initial implementation of the automatic sequencer will operate in a "pass through" mode where all commands are accepted and transmitted with minimal checking and delay.

2.7.2 Queue-based

Queue-based observing is the primary observation mode used with the Gemini telescopes, as it is the best means of providing sound science data while maximizing efficient use of the telescope. While its requirements include all those given below for service observing, it has the following special requirements.

A. The observing program must be fully automated, requiring very little human interaction during the observation. The means that the Gemini software must include a sufficiently rich programming environment to make this feasible. In addition, this should be a visually-oriented environment providing a simple, easy-to-use interface to the astronomer.

B. There should be a full telescope simulator to enable the astronomer to test observing programs for completeness, errors, and functionality. This simulator should function within the virtual telescope environment of the Gemini system.

C. All control software must provide support for simulated use within the virtual telescope.

D. There is a requirement for software to assist in object selection both within an observing program and across observing programs, in order to optimize observing efficiency. This software must consider target positions, weather conditions, and instrument configurations.

E. There must be software to support the flexible scheduling, both manually and via a scheduler, allowing for the interleaving of observing programs in a manner that is transparent to the individual observing programs. This includes managing the collection of science, environmental, engineering, reference, and calibration data.

F. To maximize the use of the available observing time it must be possible to queue all of the observing that is possible with the currently available instruments. These would be in the form of preprogrammed observing sequences. It should be possible to resort the queue so that the next observation to take place comes to the front of the queue. This sorting will be based on properties of the individual observing sequences, current site conditions, and other rules established by the observatory directorate.

While these rules have yet to be worked out one could imagine rules such as "no more than 15% of observing will be preempted" or "a 10% overhead to the prime observing project is acceptable in order to pick up observations in similar parts of the sky with identical filters".

Thus queue scheduling is a superset of preprogrammed observing, with similar requirements. As the costs of implementing such a scheduler are currently difficult to estimate it may prove necessary to implement it in a future phase of the project. The system design, if it does not include a scheduler, should specifically allow for its future implementation.

2.7.3 Remote operations

Remote operations includes both remote observing, with the science observer off-site, and remote telescope operation, with control of the telescope also off- site. It also covers remote eavesdropping, monitoring, configuration, and diagnosis. (There will always be support staff on-site during observations, however.) The requirements for supporting remote operations are:

A. All software should be developed to permit remote operations. There should be no conceptual difference between software working on-site and remotely.

B. All observing facilities should work both on-site and off-site. It should be possible to do full operations remotely.

C. Team observing, with multiple observers at different sites should be supported.

D. It must be possible to restrict specific operations to specific remote sites. For example, at the Mauna Kea site, remote telescope control might be restricted to Hale Pohaku. The method used to restrict such operations should be independent of the operations themselves, and dynamic.

E. It is assumed that the hardware involved in remote operations has been specified with sufficient bandwidth to support remote operations. However, we will take advantage of commercially available protocols such as ISDN, TCP/IP, Internet, etc.

The choice of the name remote operations is meant to suggest an entire category of operations, such as remote observing but also remote access for diagnostic support. In this sense it is more general than pure remote observing.

Despite the obvious limitations introduced by the link bandwidths available at the different locations, the system shall be totally transparent to local or remote use. It is only necessary that the functionality of the system be transparent, it is accepted that the speed of the link will determine the perceived transparency of the system. However the system design should minimize the impact of link bandwidth on transparency.

Security of operation shall be considered and might imply different operation levels and privileges at different sites.

It is required that the remote operations software be considered from the beginning in the Gemini 8m Telescopes software design, to avoid redesign later. This should reduce the amount of specific software needed for remote operations, as the common layers of software shall cope from the beginning with a distributed environment.

2.7.3.1 REMOTE CONTROL

Remote control means that the function normally associated with a local Telescope Operator, that of entering telescope control commands, would be available from a remote site. In practice remote control will be restricted to the Gemini 8m Telescopes Enclosure and Gemini 8m Telescopes Control Facility. The philosophy behind this is that, for safety considerations, telescope control commands cannot be issued without a staff member (not necessarily the same person) having direct access to:

- a hard wired “stop” button
- real time video and audio
- control of the telescope

2.7.3.2 REMOTE OBSERVING

Remote observing means that users shall be able to observe from a remote site such as Cerro Tololo, Hilo or even home Institutes. This possibility has to be provided on the Gemini 8m Telescopes.

The extent to which realistic observing conditions can be reproduced depends, of course, largely on the link bandwidth available. Experience with previous telescopes at other observatories shows, however, that, even with very limited bandwidths, remote observing can be implemented, provided the software is suitable for this.

Remote observing, as meant in this context, does not mean remote control. Remote users shall normally interact with the system via

- operators at the Gemini 8m Telescopes control facility
- operators at the Gemini 8m Telescopes site facility
- operators at the Gemini 8m Telescopes base facility
- operators at the Gemini 8m Telescopes remote operations facility
- via the scheduler program

and shall not control any part of the Gemini 8m Telescopes directly.

In the last case they will use a remote User interface to submit commands to the Gemini 8m Telescopes scheduler at the Gemini 8m Telescopes site.

2.7.3.3 MULTIPOINT MONITORING

In this mode the monitor's screen appears as a duplicate of that seen by the observer. The monitor's keyboard would not have any effect on the observer's environment.

2.7.3.4 REMOTE MONITORING

Remote monitoring is the simplest level of remote observing. It is sometimes called 'eavesdropping'. It is a requirement for the Gemini 8m Telescopes operation and will complement service observing, making it friendlier for users. Remote monitoring coincides to a large extent with multipoint monitoring, but allows the remote user to “pick and choose” the information that is displayed on the remote screen. There is no requirement that the remote screen be a duplicate of the local screen. The remote keyboard will have no effect on the local user's environment.

A remote observer might also need a real-time video and voice link with the operator in the control room, perhaps using a portable video camera which the operator can position as necessary. It is assumed this facility is provided entirely by hardware, and is beyond the scope of the Gemini software.

2.7.3.5 REMOTE ACCESS

Quite apart from remote observing, remote access to the Gemini 8m Telescopes and its instruments is required for monitoring and diagnostic purposes. This might be necessary to back up local users expertise and to help in case of problems. Remote access, in this case, must be possible from the Gemini 8m Telescopes base facility.

Distributed access to the Gemini 8m Telescopes software, once implemented. also allows, without extra requirements, local access (at the Gemini 8m Telescopes site) or remote access (Gemini 8m Telescopes support, base, and remote operations facilities).

2.7.4 Service

Service observing implies that the Gemini staff are responsible for performing the actual data collection for the astronomer. A special case of service observing, *queue-based*, is given earlier, and assumes that the observing program can be executed with minimal interaction with the client astronomer. The more general case, where a science observer is present on site and the client astronomer is remotely connected, is considered here. The requirements specific to service observing are:

A. The observing program must be automated, requiring little human interaction during the observation. The means that the Gemini software must include a sufficiently rich programming environment to make this feasible. In addition, this should be a *visually-oriented* environment providing a simple, easy-to-use interface to the astronomer.

B. This programming environment should be available both to the astronomer, for developing the program, and to the observer, for review and adjustment of the program. This access may or may not be done concurrently on a shared environment.

C. The programming environment should allow for the communication of special notes, instructions, and comments from the astronomer to the observer, possibly involving *multimedia* techniques. The use of this feature would, however, be discouraged for queue-based observing.

Observing, as such, is the purpose of the Gemini 8m Telescopes system. But interactive observing is perhaps, in some cases, not the most efficient way to achieve this, although it will be necessary for certain kinds of observations (such as adaptive optics) and in certain phases (such as commissioning and test periods).

Service observing, on the other hand, means that the observing program can be performed by someone other than the proposing astronomer. It will be done on his behalf at a convenient moment that is not exactly predefined and may be intermixed with programs from other users.

Service observing requires switching during the night among telescope modes and instruments. This might typically be a few times (2) per night, when sky conditions change.

It requires also that the observing program is so well specified that it can be executed by someone else. In other words, it requires a computer executable observing program.

Service observing requires a suitable organization to support it, and first of all, considerable experience with a smooth running system. For these reasons it can only be applied to the Gemini 8m Telescopes once the Classical observing requirement has been fulfilled and been in operation for a while. Nevertheless, the software and data structures to support classical observing which have to be present in a system that is only interactive at the start might require considerable redesign to be adapted later to the service observing requirement.

2.8 OBSERVING SUPPORT

2.8.1 Planned Observing

To achieve efficient service observing, queue observing, and flexible scheduling, it should be possible to carry out observations automatically, in accordance with predefined sequences of exposures, as is commonly done in space observatories.

This corresponds to what is meant by planned observing, which still requires competent monitoring at the Gemini 8m Telescopes site or remotely.

At the same time, one does not want to lose the advantages and the extra flexibility of ground-based astronomy. So whichever scheme is adopted to perform automatic sequences, interaction shall be allowed at the desired level (for example, only on error conditions, any time the user is allowed to break a sequence, etc.).

2.8.2 Flexible Scheduling

Flexible scheduling means the possibility of reacting to changes in weather and other conditions by allocating the current use of the telescope to the optimal observing program for those conditions.

It will require the use of service and/or remote observing, as users cannot stay around indefinitely waiting for the conditions required by their program. This can be complemented by remote monitoring, if users can be on standby at a suitably equipped remote site.

Flexible scheduling requires the possibility to change telescope scheduling quickly, exchanging observing programs. This can be achieved only with the help of appropriate scheduling software, which must advise the operations team on the best choice to make which is compatible with the relative priorities of the various observing programs. It should be recognized that such ‘expert scheduling’ software does not currently exist - it should be a requirement of such software that it have an operational mode which merely supports the decision making process by presenting appropriate information.

Flexible scheduling requires switching during the night among telescope modes and instruments (typically 2 times per night). Thus flexible scheduling is a superset of queue scheduling, with similar requirements.

2.9 GENERAL SOFTWARE REQUIREMENTS

2.9.1 Control information flow

These requirements apply to the commands and replies between sub-systems (including status information requests and replies), information flow to and from on-line databases, and commands and software for the IOCs.

- Requirements

A. The syntax of control flow commands is to be consistent across the system, whether accessing workstation software or IOC software. See Section 3.2.5 on page 3-6.

B. All subsystems must respond to a common set of commands to test operational status, inquiries as to version, perform self-tests, etc. These common commands are described in Section 5.3 on page 5-2.

All IOC subsystems must respond to additional common commands for such activities as *start, stop, initialize, reset parameters,* etc. These IOC common commands are described in Section 5.3 on page 5-2.

C. The support structure for communicating commands must be reliable, with a uniform ACK/NAK protocol adopted across all systems. Timeouts must be supported at approximately **500 msec**.

D. Handshaking of commands between IOCs must occur within **100-200 msec**, signaling acceptance of each command.

E. For commands allowing delayed replies, timeouts for that reply must also be supported.

F. Peak control information within the system is expected to be **100 TPS**. This assumes *bridging* between communication sections, to isolate traffic in relevant sections only.

2.9.2 Astronomical data flow

The following requirements apply to the control and transfer of astronomical data in all forms (science, engineering, reference, etc.)

2.9.2.1 REQUIREMENTS

A. *Data flow*. Data from detectors must be stored in the most effective method permitted by available technology. Astronomical data is often detector readout limited so that disk access and data transfer times are not significant. Maximum acceptable readout time for detector data is very dependent on detector type and size as well as the intended application:

1. For focusing and related activities, maximum acceptable detector readout time is about **0.1 sec**, though only a portion of the detector may be read during that time.
2. For mosaicked, large optical detectors, a full readout of the detector must be done in about **2 or 3 minutes**.

B. *Concurrent data access and display.* Since the Gemini system supports monitoring of operation, there must be the capability of providing multiple, simultaneous access to data. Data transfer between the virtual telescope system and attached workstations therefore imposes significant transfer requirements on the LAN. The LAN must support a transfer rate of 20-40 Mbits/second.

C. *Data acquisition format.* Data is normally acquired as uncompressed data, but may be compressed using a *loss-less* compression technique for transmission from the Gemini system or across the system LAN. The goal of compression is to minimize bandwidth impact on the LAN and WAN and to save space on removable media.

For data that requires preprocessing, such as infrared detector data, only the preprocessed data is stored.

D. *Storage of data*. Data from all instruments and detectors is stored as compressed data, using a standard format. There is a first level of storage within IOCs, to secure data in the event of link failures.

A second level of storage is on the Gemini system data disk(s), possibly also on removable media. Quick-look data quality assessment is done using this level.

Archiving of data is automatically done while in observing and maintenance level operation to the Gemini Archive subsystem. Shipping of data to a central archive follows later.

E. *Data transmission format.* Data is transmitted between Gemini and home Institutes using a FITS format and contains all header information provided with the data.

F. *System-wide data capacity.* The data capacity of the system is limited by transfer methods and technology, as well as archiving capacity on site. The system data capacity is capable of retaining **7 days of data produced by the largest instrument, the last 3 days of which must be available interactively from hard disk or similar medium.**

2.9.3 Video information flow

Video information originates from target acquisition, guiding, and site monitoring cameras. The requirements for transferring video data are:

A. The system must allow for fast transmission of rough images every **0.5 sec**. This may be assisted through the use of data-loss compression techniques (e.g. JPEG, MPEG, etc).

B. In addition, there is the need for transmission of images matching the original resolution. This high-quality transmission must require less than **20 sec**, and can only be assisted with loss-less compression.

2.10 OPERATION PRIVILEGES, PROTECTIONS, AND PROCEDURES

To preserve the integrity of the system, there must be a system of privileges established at each operating level of the system. These privileges should be determined in a simple manner during logging into the system.

Protection against accidental interference is to be implemented using an *Access Mode Allocation* system that dynamically identifies and assigns resources as needed. Critical resources (those that can support only a restricted number of simultaneous uses) are assigned solely through this allocation system. The allocation system must ensure that the system cannot remain deadlocked with respect to this resource allocation.

Finally, procedures must be implemented for convenience and system integrity, to simplify and codify common tasks. The tasks that require such procedures include:

- Telescope start-up and shutdown.
- Telescope system self-testing.
- Instrument start-up and shut-down. This is not permitted to interfere with telescope operation.
- Instrument self-testing and self-diagnosis This is not permitted to interfere with telescope operation.

- Configuration and reconfiguration.
- Dynamic reconfiguration of observing configuration (beam switching without restarting instruments and telescope).
- The control software should know what subsystems are installed and their status at all times.

2.11 GENERAL PERFORMANCE AND RELIABILITY REQUIREMENTS

2.11.1 Capacity

The Gemini software should have no hard restrictions on the number of simultaneous users, but should allow for policy decisions that do restrict the amount of simultaneous access.

2.11.2 Response time

The response time requirements vary with the function. The appropriate limits are:

A. Every command must be accepted/rejected within **2 sec** and before the corresponding action occurs. (This is different than the ACK/NAK response of the communications protocol - here, the target system must have examined the command and verified its validity.

B. Status display update must be within **4 sec** at the local stations (certain functions, such as telescope position, may have tighter constraints). Remote station update response is given in the Requirements for Remote Operations section.

C. Requests of subsystems for status information must be answered within **5 sec** and be possible in maintenance level operation.

D. Requirements for response times within the user interfaces are given in the User Interface requirements section.

2.11.3 Availability

All software bugs should be logged and then fixed as soon as possible after detection. The goal is to have restart conditions occur only on hardware failure.

Fault recovery, exception handling, fail-safe checks, etc. should be used to improve reliability.

2.12 TEST AND CHECKOUT REQUIREMENTS

The telescope and instrument software shall contain built-in test (BIT) facilities to verify Gemini 8m Telescopes system and Gemini 8m Telescopes software performances.

Every Gemini 8m Telescopes software module shall have corresponding test specifications to check normal operation of releases, to be used both for acceptance tests and as an on-line test procedure.

The Gemini 8m Telescopes control software shall also provide for execution of self-test sequences of the Gemini 8m Telescopes system and subsystems. These shall automatically exercise all subsystems present in a given operational configuration.

Regression tests should be a part of every Gemini 8m Telescopes software package.

2.13 CONTINGENCIES

2.13.1 Fault notification

Subsystems must notify the user when faults occur. This notification must be specific as to origin and problem. The notification must also be capable of being electronically logged. It may also prove useful to have multiple levels of fault notification such as detailed, verbose, short, etc. to aid in tracking down problems.

2.13.2 Fault tolerance

Should a subsystem fail (e.g. one detector, one instrument) predefined procedures must exist to redefine the environment in such a way that operation can restart with the remaining equipment.

In case of computer hardware failure concerning the user station equipment, it shall be possible to transfer control from one user station to another via a simple software reconfiguration procedure.

In the case of IOC failure, no transfer of control to another IOC will be possible, due to the local connections and interfaces to the control electronics. In this case there shall be a procedure to replace faulty cards and/or assemblies. If it is possible to observe with that particular IOC in a failed state (in general, this is limited to IOCs that are associated with individual scientific instruments) then it must be possible to reconfigure the system to do so.

2.13.3 Redundancy

Full redundancy is not a requirement of the Gemini 8m Telescopes and it will be acceptable to have to replace units in case of failure.

There are subsystems which are relatively inexpensive to support as redundant systems, such as telescope control computers. For each area where redundancy is decided to be cost effective, procedures for switching to the backup system will be established. There is no requirement for automatic switching to the backup system.

All communication shall be based on the use of standard communication protocols, where retry procedures are applied (a form of software redundancy) as part of the protocol.

Certain network concepts may be preferable as they offer intrinsic redundancy (e.g. double loops) and re-routing possibilities in case of node failures (single point failure protection).

2.14 CONSTRAINTS

The Gemini system software is designed under the following constraints.

2.14.1 User constraints

A. There should be no restrictions imposed by the software on users. Only policy decisions (permissions, access privileges, etc.) should prevent any user from accessing any part of the Gemini system from any local or remote station.

B. Similar functionality should be presented to the users using similar user interfaces. However, user interfaces should clearly reflect access modes and operating levels.

2.14.2 Hardware constraints

Hardware constraints are covered in the individual chapters the Software Design Description.

2.14.3 Software constraints

There are a number of general constraints placed on the Gemini Software. These are:

A. Commercial packages, off-the-shelf public domain software, and standards are to be used whenever feasible.

B. .Existing external software will be integrated with the Gemini software. The interfaces involved in this integration are considered part of the Gemini software system.

C. All Gemini software is to be developed using standard methodologies and development environments. One of the goals of Gemini software is that all components be easily (preferably automatically) combined into an integrated system.

D. Gemini software developers should maintain accurate *change logs* showing software modifications as they are applied to the system software.

E. Gemini software developers should adhere to a standard method for the reporting and recording of errors from both internal and external sources.

F. Gemini software should be developed in *evolutionary* fashion, using the CVS version control system.

G. All Gemini subsystem packages should include as part of the software both a *simulator* module for inclusion in the virtual telescope (see Design constraints, below), and *user interface* modules for the user interface environments that the subsystem will be operating in. The interfaces required of these user interface modules are described in Section 3.3 on page 3-6.

H. All Gemini software is to be fully documented, internally with appropriate comments, and external documentation. External documentation must include Unix-style man pages.

I. All Gemini subsystem packages must provide modules for the testing and diagnosis of the subsystem.

J. All instrumentation control software **must** provide *full access* to **all** instrument functionality. It is likely that different user interface modules (see above) would present different portions of this functionality to the user. The information required of each interface module is found in the Functional Requirements specifications for each instrument.

K. All Gemini software must be *version* labeled, both in source and binary form. The version information is to be retrievable from executing software via control commands.

2.14.4 Design constraints

A. There are different requirements for software running on different layers. For example, strict real-time control is restricted to the IOC layer.

B. Also, the use of a *virtual-telescope* model in the Gemini system means that the integrated system can be tested and developed independently of the target hardware. This is useful not only in the use of the *telescope simulator* during science planning, but in maintenance and testing as well. Therefore all hardware subsystems must provide a software simulation module (as stated earlier) that responds in reasonable fashion to commands directed at that hardware. This simulation may require a standard environment, such as VxWorks, EPICS, and VME crate/cpu, but it cannot require any hardware specific to the application.

C. Gemini subsystem should be as self-contained and autonomous as possible, thereby decreasing the functional width of the interface to the rest of the Gemini system.

D. No subsystem package should make *any* assumptions about the surrounding environment beyond that provided in the interface specifications.

3 General Requirements

3.1 Data Specifications

The final purpose of the Gemini 8m Telescopes software is the acquisition of astronomical data in digital form in the most efficient way.

To achieve this, many other data concerning the telescope and instruments (parameters) and control commands will have to be exchanged between different processing units in order to setup and control telescope and instruments. Additionally, video and voice data are also necessary (for example, field monitors).

3.1.1 Control Information Flow

Control information must be transferred, typically in the form of commands and replies from users, to telescope and instruments. Replies might contain status information and, in general, data concerning instruments and telescopes, to be stored together with the astronomical data.

Control information on all controlled variables must be provided by all subsystems on request. No request for information shall produce a delay of control activities or locking, even if the corresponding equipment is not available or faulty.

Delay times for the exchange of control information must stay within precise time limits to be defined in “General Description” in Chapter 2. One can afford to retransmit commands in case of transmission error or collision, but the protocol has to be predictable in that commands cannot get lost and replies have to come back reliably.

In a number of cases, synchronization with the Time Reference System at the Gemini 8m Telescopes site is also necessary.

Access to control parameters, telescope and instrument information for monitoring or other use makes a significant contribution to the control flow, and may be logged at quite high rates for short periods (i.e. up to 200 Hz for some information).

It is explicitly required that all such information is available to the Gemini 8m Telescopes software and is capable of being available to all users of the Gemini 8m Telescopes, subject only to restrictions with respect to updating. It must also be possible to restrict user access to such information.

In particular also, the meteorological information coming from a weather station should be available centrally.

3.1.2 Astronomical Data Flow

3.1.2.1 Data flow

Detector data must be acquired and stored in the most effective way technology will allow; effectiveness should be evaluated in terms of cost, space requirements, longevity, and speed. This shall lead to the definition of a Gemini 8m Telescopes standard, used on all instruments. In general, operational overheads must be kept as low as possible, to maximize actual observing times.

Intermediate storage of raw data in memory on different nodes and in different formats should be kept to a minimum. However, there must be at least two copies - one to secure data as acquired and one to do assessment of data quality on-line (this last copy preferably on removable media).

The link chosen to transfer data should represent as small a bottleneck as possible for data acquisition.

3.1.2.2 Format of data acquisition

Normally, raw data will be acquired and stored as such for quick look evaluations. There might, however, be cases (for example, infrared detectors) where fast preprocessing is needed and where, therefore, raw data will not be stored as such, but in a preprocessed format.

3.1.2.3 TRANSPORT DATA FORMAT

Astronomical data will have to be transported between GEMINI and the home institutes of visiting astronomers in FITS format (as defined by NOST 100-1.0, "*Definition of the Flexible Image Transport System (FITS)"*, NASA Science Office of Standards and Technology).

3.1.3 Other Information Flow

TV data concerning site monitoring and voice need to be capable of being available at all operations facilities. It will be a question of interfacing and bandwidth costs whether such information is actually available at a specific location. It is not a requirement that point to point video be available between Gemini 8m Telescopes operations facilities. It is a requirement that voice connectivity, perhaps point to point, be available on a permanent connection.

Other astronomical information such as that coming from sky field monitors, autoguider cameras and sky monitoring devices such as cloud and seeing monitors shall also be capable of being available.

3.2 OPERATION

3.2.1 Operation levels

Operation of telescope, instruments and subsystems can be carried out at three different levels. There will be parameters associated with each level of operation which define the status of the system and / or control the system. The specific access by different types of users to particular parameters of the different levels are given in Section 2.5.8 on page 2-5.

3.2.1.1 OBSERVING LEVEL

This is the normal operational mode. It allows a certain number of embedded tests, normally at a fairly high level. Monitoring is also done at this level.

It is anticipated that all user categories have access to this level.

3.2.1.2 MAINTENANCE LEVEL

This allows maintenance tables (for example, with instrument parameters) to be updated.

It is anticipated that the majority of parameters at this level will be accessed by operations and development staff.

3.2.1.3 TEST LEVEL

This allows the installation and testing of new packages or new releases. Any low level test can be performed in this mode.

It shall be possible to update all non-protected parameter values, i.e. those not used by operations at observing level.

Access to this level will be extremely limited.

3.2.2 privilege and protection levels

Privileges and protections are also important parameters to define user operations.

3.2.2.1 PRIVILEGES

A further subdivision within the levels can be achieved by implementing privileges associated with categories of users or with the location of the user station (local or remote). For example some users might only be allowed monitoring, as a subsystem is in use by someone else.

3.2.2.2 PROTECTIONS

Protections must also be enforced (see also Security in Section 3.5.6 on page 3- 20) among users and the operational software should indicate clearly to users the current operation level and check the compatibility between subsystems in different modes.

3.2.3 Capacity

The capacity of the system can be expressed in terms of nodes, which is defined as the number of workstations, or in terms of users, which is defined as the sum total of users at all the nodes. The capacity requirements will be expressed in terms of nodes.

Each node will have the capability to run at all operation levels.

When the Gemini 8m Telescopes telescope is used in its normal observing mode, there will be a single operator node for the telescope and two data acquisition and instrument control nodes.

Some tests might be run in parallel on instruments that do not have the light beam at that moment, so in principle additional nodes might be working at the same time. The system will provide for one auxiliary data acquisition and instrument control nodes.

In addition, the system must support off-site observing modes. The system will provide for a single off-site data acquisition and instrument control node - to be located at either the Gemini 8m Telescopes Site Support or Base Facility.

One supervisor will monitor the system, and other users might need to monitor the running of observing programs, locally or remotely. The system will provide for a single local monitoring node and a single remote monitoring node.

As a conclusion, the Gemini 8m Telescopes control software shall allow simultaneous operation of up to six active control nodes and up to two more monitoring nodes (one local and one remote) without appreciable degradation of performance.

In practice the operation and facilities foreseen so far for the Gemini 8m Telescopes will limit this number to a maximum in the order of three active nodes, but the Gemini 8m Telescopes computers and software shall be capable of coping with the load of 10 active nodes, should the case arise.

3.2.4 Performance criteria

This section is intentionally kept at the level of performance criteria rather than response times (found in Section 2.11.2 on page 2- 22). Every command must be acknowledged in a positive or negative way before the occurrence of the corresponding action within given response times.

3.2.5 Procedures

There must be automatic procedures to implement startup and shutdown of the telescope and instruments. These must allow startup and shutdown of instruments independently of the telescope and without affecting the telescope operation.

Reconfiguration procedures must exist, to change the observing environment.

The definition of the observing environments must be dynamic, i.e. feasible during operations without the need to restart everything. The same applies to the related light path.

Operations staff have privileges to change the environment, meaning selecting a suitable combination of instruments.

The operational software should know which subsystems are installed and operational at any given time.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 User Interfaces (UIF)

The user interface defines the way users see the Gemini 8m Telescopes system. Given the large number of instruments, there can be many different stations which are active at the same time. It is essential for operational and maintenance reasons that, in spite of the obvious differences of the setups and commands available, the same philosophy is applied throughout. This calls for a homogeneous user interface, which can be achieved only by applying the same user interface tools to the whole project, providing the Gemini 8m Telescopes user interface's 'look and feel'.

The user interface should not be seen as a package linked to a specific computer. Given the requirement to be able to access the Gemini 8m Telescopes from several points, the user interface should rather be seen as a package to be callable from a large number of stations, depending on where a user is. It should also be network transparent so that it does not matter where it is being run.

The user interface tools shall be based on standards, defined in Section 4.2.5 on page 4- 7, which will be portable across different computer hardware platforms (Portable User Interface Toolkit). The intent of a portability requirement is to facilitate migrating existing and future Gemini systems to different hardware as the need arises. It is the current intent to limit the selection of computer hardware platforms to as few as is practical.

3.3.2 Hardware interfaces

Main processor computer hardware requirements will be defined in the Hardware constraints (see Section 3.4.2 on page 3- 12).

This section deals instead with external hardware interfaces, namely the ones from the microprocessors to the control electronics. The definition of these terms is as follows:

- main processors - these are the computers with which the user interacts
- control electronics - these are the computers controlling the instruments

- microprocessors - these are computers embedded in the instrument (for instance a DSP or transputer required for array control functions, or special purpose controllers for the primary mirror support actuators).

Standard interfaces to the control electronics shall be available, in the form of:

- standard bus systems
- a standard set of interface cards to be used on all the subsystems and instruments
- a standard software skeleton running on the control electronics

The existence of hardware standards is clearly essential for maintenance and repairs. It is also essential to avoid software duplication, and to simplify the Gemini 8m Telescopes software. Microprocessor software in particular tends to contain hardware specific software, though one should try to keep it as hardware independent as possible, isolating different software layers.

The standard software must be adequate for the real-time requirements and must offer drivers to the standard electronics to be used on all the Gemini 8m Telescopes software subsystems and instruments.

Links between electronics interfaces and main processors must meet the requirements imposed by the data specifications (see Section 3.1 on page 3- 1).

3.3.3 Software interfaces

The Gemini 8m Telescopes software covers all aspects of control and data acquisition related to the telescope, instruments, and auxiliary instrumentation.

It also covers all the operational aspects of the Gemini 8m Telescopes, including on-line scheduling and rescheduling.

There is also software which, although it will be interfaced to the Gemini 8m Telescopes, is referred to as external. The external software consists of:

- commercial software integrated into the Gemini 8m Telescopes software (e.g. DBMS)
- preexisting software used in the Gemini 8m Telescopes (e.g. image processing systems, star catalogues)
- software associated with visitor instrumentation
- embedded software dedicated to hardware control, but not communicating on-line with the other Gemini 8m Telescopes software (in general this would be microprocessor code)

The Gemini 8m Telescopes software must interface to the external software and clearly the interfaces are fully part of the Gemini 8m Telescopes software.

3.3.3.1 On-line image processing interfaces

In order to make efficient use of the telescope, to support different observing modes, and to support the versatility requirements, some form of on-line image (or pixel) quick-look analysis is required. The following statements are proposed:

"*It shall be possible to monitor the quality (image quality, spectral resolution, signal to noise, etc.) of the astronomical data as it comes in.*

Standard reduction procedures should be available for basic on-line calibrations of the observed data. Ultimately, one would like to have fully reduced and calibrated data at the end of the observations. Advanced pipeline procedures might make this feasible, at least for observations of a standard nature."

The above statements define the goal of quick-look analysis for the Gemini 8m Telescopes.

Quick-look data processing should be provided on the Gemini 8m Telescopes, with procedures suitable for fast on-line data preprocessing. A prerequisite for this is that acquired data are made available as directly as possible in a common format, and that all additional data related to an exposure and logging information are made available on-line at the same time.

Quick-look should be usable within exposure sequences to provide results and feedback parameters to the control software in a programmed way, without the need for manual intervention. This document does not try to be specific about the requirements for Quick-look other than that it should be synchronous.

Near-line processing should be available for simple data reductions required for data integrity validation (i.e. remove instrument and observatory effects so the observer can make decisions about further observing actions). This data reduction proceeds sequentially through requests, but asynchronously from data acquistion. In particular, data acquisition takes precedence over near-line data reduction.

Off-line pixel processing for full data reduction should also exist at the Gemini 8m Telescopes site, but does not have any interface to the Gemini 8m Telescopes software. The Astronomical communities have made considerable investments in image processing software, and therefore, compatibility with and adaptations to these packages must be sought.

It should also be noted that some Gemini 8m Telescopes subsystems, such as adaptive optics, may require their own special on-line pixel processing software, which is better defined in the requirements for those subsystems. This is largely due to the difficulty of applying on-line the same algorithms used for full off-line reductions — in general due to the time critical nature of the image processing needs.

The same situation might also occur with other instruments, where specific observer support software has to be foreseen for on-line use.

In all these cases the specific on-line (quick-look) software development shall be seen as a subset of the development for the off-line data reduction system, to avoid as far as possible duplication of development effort.

3.3.3.2 ON-LINE ACCESS TO CATALOGUES AND PREVIOUS DATA

The output format of the Gemini 8m Telescopes data must be compatible with the GEMINI archive requirements.

As comparisons with previous data might be of great value and affect the actual observing program, on-line interactive access to the data archiving system should exist, so that access to this database is possible for Gemini 8m Telescopes users.

The specific types of data available; flat fields, calibrations, science exposures, etc.; the amount of a specific exposure available; header only, averaged exposure, complete raw data set; and the time frame within which such data will be made available; same night, weekly, after proprietary period; will be established by the Gemini Archiving Requirements.

Computer access to star catalogues is also required, so that an automatic selection of candidate guide and standard stars can be made.

3.3.3.3 ACCESS TO OTHER PACKAGES

The Gemini 8m Telescopes software must be able to interface with all commercial software packages available on the Gemini 8m Telescopes and integrated into the Gemini 8m Telescopes operation.

A relevant example of such a package is a general database management system (DBMS), where operational information such as schedules, logs, problem reports and maintenance information related to various pieces of equipment should be kept.

3.3.4 Communication interfaces

Being in itself a distributed telescope system and having a large number of instruments, the Gemini 8m Telescopes system has internal communication needs.

3.3.4.1 LOCAL AREA NETWORK (LAN)

The LAN shall support the majority of the Gemini 8m Telescopes system internal communication needs. This LAN must be capable of dealing both with the data bandwidths required (at peak and on average) and with the required response times and synchronization needs. This LAN shall be supplemented with a Local Time Bus, for distribution of absolute and relative time signals, and both a digital reflective memory bus and an analog event-based bus, for distribution of signals with requirements not satisfied by a LAN.

Independently of the physical layout of such a network, its functional requirements can be split into several categories:

- Local coordination and synchronization needs within a subsystem (for example, coordination of active support system for primary mirror, bi-directional traffic of commands/replies). This layer could be considered as internal to a given subsystem or instrument, but as more than one subsystem might need it, one should aim for a unique solution at the hardware and software level.
- Global synchronization needs, such as the universal time synchronization requirement. The required accuracy will be defined in Section 2.9 on page 2- 19.
- Exchange of control information, in the form of commands and status information. (Bi-directional traffic) (see also Section 3.1.1 on page 3- 1).
- Collection and transfer (for archiving or remote access) of raw astronomical data. The capacity will be defined in Section 2.9 on page 2- 19(see also Section 3.1.2 on page 3- 2).
- Access from Wide Area Network (WAN) for remote diagnostics and monitoring from operations facilities (Bi-directional traffic).

No distinction is made here between WAN and point to point links as there shall be no difference in the software between the two cases. However the system architecture will be designed so as to minimize the communication load placed on peer and higher level networks.

Control data and astronomical data have already been defined in the Data specifications (see Section 3.1 on page 3- 1). The reason for repeating them here is to have a complete view of the required network functionality.

To eliminate conceptual access problems, while coping with different bandwidths, LAN and WAN interfaces shall be homogeneous and shall be based on standards which allow migration on different media, should they become required during the Gemini 8m Telescopes project life.

For maintenance reasons and hardware independence, a clear hierarchical model must be implemented, supporting separation of logical and physical layers, e.g. ISO/OSI model. It is recognized that this hierarchy may need to be violated for (in general) performance reasons. This results in point-to-point connections between peer systems or direct connections bypassing the hierarchy.

Network redundancy should also be considered in the design phase as a way to increase reliability and security, in particular for control information.

Due to the uncertain future of the Internet, only non-essential tasks may employ it. All essential tasks, not including remote observing, must take place on resources controlled by the project (such as leased lines).

Violation of the hierarchical nature of the system can lead to testing and maintenance problems. The use of these features must be limited and constrained by the following guidelines:

1. Peer-to-peer connectivity should *only* be used to overcome a demonstrated performance problem.
2. Bypassing the hierarchy (connected between grandmother and granddaughter with no path through the mother) should *only* be used for transmission of status information or bulk data, *not* control flow.

3.4 GENERAL CONSTRAINTS

3.4.1 User constraints

It is envisaged that observing astronomers who have travelled to the Gemini 8m Telescopes site will make use of the Gemini 8m Telescopes control room facilities. This will allow centralized support and coordination of their operations, providing both operations support for individual instruments and supervision for all of them.

However, even at the Gemini 8m Telescopes site there will be users (e.g. software development and maintenance staff) installing or enhancing other parts of the system, possibly working directly at the telescope.

Finally, in the case of remote operations, other users (e.g. support staff) might be on Cerro Tololo or in Hilo.

To allow coordination both locally at the Gemini 8m Telescopes site between the various users and with remote users, the software shall support access to the system from any user station. It will then be an operational decision, implying privileges and priorities for the various categories of users, and definition of what a given user can actually do.

Access from any user station will make user stations in principle identical and software configurable as the user station of this or that subsystem. This should greatly simplify the coordination problem posed by the large number of simultaneous users (as already described in the Capacity requirements, Section 3.2.2).

3.4.2 Hardware constraints

There are a number of constraints for the Gemini 8m Telescopes computer hardware. Although some of these constraints may appear redundant the project recognizes that:

- during the next 5 to 10 years new hardware will be available with features and costs that could benefit the project
- the hardware chosen as the standard at this point of time may be no longer available at some point in the future
- the cost of maintaining an existing hardware standard, even if available, may exceed the costs of adopting a new hardware standard

These constraints recognize that the majority of the expense in changing hardware standards is the cost of the software. Particular constraints are:

A. Computers used at the Gemini 8m Telescopes site, particularly in the test phase when they are outside the control room and near the subsystem under test, shall be checked against altitude and humidity specifications for the Gemini 8m Telescopes site (see also [EDS] in Section 1.3).

B. Computer hardware must be able to run the Gemini 8m Telescopes software environment (operating systems, Gemini 8m Telescopes software) and provide compatibility in data format (identical internal data representation).

C. Common development and implementation tools must be both available and supported

D. Identical network access must be supported

E. Local processing power must be such that telescope and instrument control does not represent a significant overhead in the whole process of executing an observing program (the overhead, if any, should be limited by the time it takes for physical devices to act)

F. The choice of a scalable hardware architecture with computers at various performance levels should solve the problem of adequate on-line data assessment, as the amount of this activity is very variable and dependent upon the kind of detector and method used.

G. Due to the limited bandwidths which may be available remotely, there will be constraints on the functionality of remote operations and access.

Depending on funding, it is probably reasonable here to make a minimum and a desirable specification. The minimum situation could also coincide with what is needed in the test period before coming to full remote operation.

In any case, remote operation must include remote monitoring from the Gemini 8m Telescopes base facilities, together with access tools for diagnostic and test use. It should be a goal to support remote observing from the Gemini 8m Telescopes base facilities.

3.4.3 Software constraints

Some of the general requirements which have a direct effect on the software are here explicitly transformed into software constraints.

These are:

A. Individual instruments must be able to run fully independently.

B. Telescope software at the two telescopes must be maintained to be identical in the upper layers (even if hardware should differ).

C. Additions of new instruments should aim, as a goal, at introducing no modification to already operational parts. Modifications should be confined to the operational procedures and should not affect the bulk of the existing software.

D. Switching to different configurations must be possible at any time with appropriate procedures.

E. There must be easy procedures to reconfigure the system when subsystems are modified or removed.

F. The number of main packages of software must be kept to a minimum to facilitate maintenance, but compatibly with the need to have the right degree of modularity.

G. Commercial and public domain packages should be used whenever possible.

H. Existing software packages should be reused wherever possible.

I. Existing software expertise should be consulted whenever possible.

J. All software which does not directly control specific hardware must be written as machine independent, portable code. Even for microprocessor software, the software should be hardware independent, to allow a later choice of the target microprocessors.

K. To allow for expansion and maintenance, Gemini 8m Telescopes standards must be defined for the on-line software and the development environment.

L. On-line version control must be implemented. That is, the version control system must be available to recover/restore versions at all times.

M. At boot time, the Gemini 8m Telescopes software shall check the consistency of versions of all the various software components.

N. Table-driven software should be used whenever possible, to avoid unnecessary compilations.

Whether the software is table driven, message driven, or a combination of both is a function of the individual work packages and defined in the appropriate work package description.

Changing system constants, such as arcseconds/bit for an encoder, shall not require recompiling but will be updated as part of system startup, and, for some constants, will be modifiable during operation. System status parameters will be maintained to an extent that will allow restarting the system and regaining the previous state. The extent of duplication of the previous state will be dictated by safety and practical considerations.

Strict checking should be applied on this to preserve maintainability and reconfiguration of the system.

3.5 ATTRIBUTES

Regardless of application, the software and control systems produced as part of the Gemini Project Work Packages exhibit numerous common attributes. The specific attributes described in this document are:

- Simplicity
- Supportability
- Reliability and Availability
- Maintainability
- Human Engineering
- Security and Safety
- Environmental Compatibility
- Testability

- Expandability
- Modularity
- Concurrency

3.5.1 Simplicity

I think you should include Rick's information about complexity criteria here, so show that we are aiming to achieve as simple as system as we can which meets the requirements. — Steven.

3.5.2 Supportability

A supportability plan will be part of the Gemini Control System. The goals and issues to be addressed by this plan and the elements of the plan are detailed below. The plan is TBD.

3.5.2.1 Goals

- have supportability influence design
- translate availability and readiness requirements into supportability requirements
- identify and plan for necessary support
- provide support at minimum cost

3.5.2.2 Issues

- maintenance personnel availability and work hour constraints
- personnel skill level constraints
- operating and support cost constraints
- target failures correctable at each maintenance level
- allowable downtime at site
- turnaround time to fix and maintain system
- standardization requirements

3.5.2.3 Elements

- maintenance planning
- manpower and personnel

- supply support
- support equipment
- technical data
- training and training support
- computer resources support
- facilities
- packaging, handling, storage and transportation
- design interface

3.5.3 Reliability and Availability

A reliability program is part of the Gemini Control System. The requirements for this program and some measures of reliability are detailed below. The plan is as provided by Glen Heriot of the Canadian Gemini Project Office.

3.5.3.1 Reliability and Availability Requirements

Reliability is defined as *"The duration or probability of failure-free performance under state conditions"* or *"The probability that an item can perform its intended function for a specified interval under stated conditions"*. The [GSR] sets as a requirement 2% and a goal 1% for total system (mechanical, electrical, and software) downtime due to failures - this translates to a maximum of 15 minutes per night or 1 night per month of downtime. This in turn sets quite stringent requirements on both MTBF and MTTR for the software and controls.

To guarantee maximum availability of the control system, retry procedures must be embodied in the software in case of error or failure (e.g. time-out, hardware failure) to achieve recovery on-line whenever possible.

Should recovery also fail, the error or failure has to be reported in a clear form (to identify the cause of the problem) and the system shall put itself into a safe state, whenever a safety aspect might be involved.

To avoid unnecessary downtime, it must be possible for the system to reconfigure itself in order to continue observing, in a different mode if required, given the failure of a single non-critical subsystem.

To increase software robustness, range checking and validity checking shall be supported before execution of any input command. This must be possible ahead of time, preparing observing sequences for automatic observations and simulating observations to estimate results.

On-line pre-checking of the operational status of equipment should be done prior to sending critical or time consuming commands. It must be possible to apply continuous monitoring to all subsystems on request, both when in operation and when idle, to check their operational status.

A measure of fault rates should be done during commissioning to establish baseline rates for system reliability monitoring.

There are to be recovery procedures to restart after error failure.

The system should be constantly monitoring active subsystems to be sure they are operating correctly before sending command to each subsystem. This monitoring should continue on inactive subsystems.

The goal for recover and/or reconfiguration is 5 minutes from onset of the error condition to observing again.

Specific requirements are:

A. measurable, realistic reliability needs
B. performance criteria for the system
C. definition of failure
D. conditions of use and environments
E. means of verification
F. period of time during system life

3.5.4 Maintainability

A detailed plan for maintaining and periodically upgrading the Control System over its lifetime will be part of the Gemini Control System. The plan will consider:

A. Maintenance requirements including an estimate of required resources.
B. The method of upgrading the system to add capabilities and performance. Areas where upgrades are anticipated should be identified with an estimate of the required effort and resources.

These issues will be addressed in the context of the mountain environment where the system will be operating. The plan will be delivered with the control system.

3.5.4.1 MAINTENANCE REQUIREMENTS

Maintenance of commercial software (Solaris, VxWorks) used by this work package is the responsibility of the WPR and the maintenance costs are not covered by the work package budget.

Community software support (EPICS) is available nominally free-of-charge through the normal release and bug-fix procedures used in the community.

All subsystem software is to include modules to aid in the maintenance and testing of the subsystem. For example, each subsystem is to include a simulator that provides a reference behavior for that subsystem. Simple mechanisms should exist for replacing a subsystem with its simulation.

The following self-check levels are to be supplied with sub-system software:

A. *Monitor level.* Each subsystem should have a background task running whenever that subsystem is operational, performing such tasks as checking power supply levels, temperatures, performance, correct responses to commands. The OCS is to be notified of any detected problems.

B. *Self-test level*. Each subsystem should provide a module for fully exercising all subsystem components, both hardware and software. This module is executed automatically during start-up and on demand through the defined interface. Problems are to be automatically reported to the OCS via the defined interface.

C. *System level.* There are also software modules for testing the subsystem as an integrated portion of the entire system. This software would be executed on demand during maintenance operation level.

3.5.4.2 LEVELS OF MAINTENANCE

The levels at which maintenance may be performed are:

3.5.4.2.1 Organizational Level

These should be done in situ during an observing session.

- repair by unit replacement (for instance, an extra computer system)
- repair units by module replacement (e.g. replace a VME card in an IOC)

3.5.4.2.2 Intermediate Level

This could be done at an Gemini 8m Telescopes base facility during the day.

- repair by module replacement

3.5.4.2.3 Depot Level

This could be done at an Gemini 8m Telescopes base facility during the day.

- module repair

3.5.4.2.4 Contractor

Done at the contractor/vendor's site.

- repair or replace

3.5.4.3 MAINTENANCE GOALS

The prime objectives are to minimize:

A. downtime due to maintenance
B. cost of maintenance
C. numbers and skill levels of personnel
D. efforts to perform maintenance
E. errors in maintaining systems
F. failures induced in maintenance

3.5.4.4 MAINTENANCE ISSUES

A. maintenance worker constraints
B. levels of maintenance
C. sparing plans
D. periodic testing
E. scheduled or preventative maintenance
F. planned support equipment
G. turnaround time
H. repair versus discard

3.5.4.5 MAINTENANCE CONSTRAINTS

A. operating hours
B. downtime or availability

C. attended / unattended operation

D. environment

3.5.4.6 Quantitative Maintenance Requirements

Quantitative maintenance requirements will be allocated to the system, subsystems, and each component. The requirements must be achievable and stated in such a way that verification is permitted. The requirements will be expressed as:

A. Mean Time to Repair for each maintenance level

B. Maximum Time to Repair for each maintenance level

C. Preventative Maintenance hours per year

3.5.4.6.1 Maintenance Interval

All equipment shall support a programmed adjustment and maintenance interval of 30 days or longer.

3.5.5 Human Engineering

All Gemini Software must be designed with human engineering requirements under consideration. The human engineering requirements for Gemini Software include:

A. provisions for minimizing stress effects and fatigue

B. feedback on operation on specific tasks

C. people and machine interfaces

D. procedures

E. training and experience

F. interaction with team members

G. management and organizational behavior

3.5.6 Security and Safety

The Gemini Control System development effort will obey and abide by both the letter and the spirit of all applicable engineering practices, laws, regulations, and policies. All necessary safety approvals will be obtained before devices will be accepted. The safety precedents and requirements are detailed below.

3.5.6.1 DEFINITION OF SAFETY AND RISK

MIL-STD-822B, *System Safety Program Requirements*, defines safety and risk as:

Safety: Freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property

Risk: An expression of the possibility of a mishap in terms of hazard severity and hazard probability

In addition the Gemini project defines a hazard as:

Hazard: Something that could cause death, injury, illness, or damage/loss.

3.5.6.2 SECURITY AND SAFETY REQUIREMENTS

The Gemini system must be self-monitoring to invoke safety monitoring to prevent risk to people or damage to equipment. The software should be able to quickly bring the Gemini system to a safe state upon notification of such danger. Subsystems must be able to detect such danger and report it appropriately. In the event that the risk persists, subsystems must be able to move themselves into safe states to protect people and equipment (i.e. if there is a failure in the higher-level systems).

Safety protection must be applied whenever there is the risk that the actions of the control software could endanger people or cause damage to any Gemini 8m Telescopes subsystem, for example, by driving beyond limits or by overexposing detectors. This protection, where implemented, must be independent of the software. In general this will require mechanical hard stops, electrical interlocks, electrical hard limit switches, soft limit switches, software limits, and watch dogs,.

The order in which these systems will work is as follows:

1. software limits - the software will not allow unsafe actions; the command will be rejected
2. soft limit switches - the software will detect unsafe areas and halt; the software will allow movement off of soft limits
3. software watch dogs - the software will halt if its watch dog has not been reset
4. hardware watch dogs - the system will halt if its watch dog has not been reset
5. hard limit switches - these switches will remove power from actuator when beyond soft limits; the software/electrical systems will allow movement off of hard limits
6. hardware interlocks - these will prevent both software and hardware from action - there will be no bypass of these systems
7. hard stops - the mechanism cannot move beyond this point due to mechanical limit(s). In general hard stops will use dampers to avoid damage to equiment.

Security and Safety

The Gemini 8m Telescopes software shall be able to bring the Gemini 8m Telescopes system quickly to a safe state upon detection of danger. Safety aspects shall be analyzed during the functional specification phase of the software.

Security must be provided in order to both prevent accidental mix-up of commands from different users on different parts of the system and to prevent intrusion from the wide area network into the Gemini 8m Telescopes. In particular, the astronomical database must be protected from intrusion, whether the purpose is to access private data or to be destructive. It is acceptable, and may well prove to be the best solution, to provide intrusion security by a well designed network gateway acting as a firewall.

A system that is operating in Engineering/Maintenance mode must ignore directives from other systems, though status information should still be provided for use by other systems.

There should be security preventing the intrusion into the system by unauthorized users, or users at unauthorized access levels.

All systems are appropriately interlocked. This interlock must not depend on any software for reliable operation. Details of the interlock system are found in the *Mount Control System Work Package Definition.*

The interlock philosophy is as follows:

- All hazards capable of causing death and/or loss of irreplacable equipment shall be passively interlocked.
- All hazards capable of causing injury and/or severe damage to equipment shall be actively interlocked (severe damage implies that repairs are not repairable at the depot level.
- All other hazards may be interlocked via software.

The precedence for conforming to safety requirements will be:

1. design for minimum risk
2. incorporate safety devices
3. provide warning devices
4. develop procedures and training

The general safety requirements are:

A. eliminate hazards through design, including material selection
B. isolate hazardous substances from people

C. minimize hazard to people during operation and maintenance from high voltage, electromagnetic radiation, sharp edges, hot surfaces, chemicals, etc.

D. minimize risks due to environmental conditions, such as temperature, noise, vibration, etc.

E. minimize risks created by human error

F. use interlocks and other protective devices when hazards cannot be eliminated

G. provide distinctive markings and warnings to protect people

3.5.7 Testability

A Controls Test Plan is part of the Gemini Control System. This plan will address all areas of testing from design, acceptance, commissioning through to hand-over. The objectives and requirements of this plan are detailed below.

MIL-STD-1309C, *Definitions of terms for Test, Measurement and Diagnostic Equipment*, has these definitions:

Testability: A design characteristic which allows the status of a unit to be confidently determined in a timely fashion.

Built-in-test: An integral capability of the mission equipment which provides an onboard, automated test capability to detect, diagnose, or isolate system failures. The fault detection and, possibly, isolation capability is used for periodic or continuous monitoring of a system's operational health, and for observation and, possibly, diagnosis as a prelude to maintenance action.

3.5.7.1 TESTABILITY OBJECTIVES

The major objectives are:

- test effectively with minimum effort and cost
- reduce maintenance induced problems
- reduce the cost of test equipment and programming
- reduce cost of documentation

- testability requirements

TABLE 1. Goals for Testability Values

Item	Organizational % Capacity	Intermediate % Capability	Depot % Capability
Fault Detection (all means)	90	100	100
Fault Detection (BIT)	90	95	95
Fault Isolation			
Eight or less modules	95	95	95
Three or less modules	90	90	90
One module	80	80	80

The interpretation of this table is that we should discover the failure of a subsystem 90% of the time *before* the failure impacts observing.

3.5.8 Expandability

Since the Gemini software will be developed in stages over a period of years, and since computer technology is expected to evolve rapidly over this same period, the software is to be designed to be easily extended and upgraded with modifications to non-changing components. The software itself, its installation process, and its documentation must be developed with this expandability in mind, using general industry standards.

3.5.9 Modularity

All software is to be developed using typical modularization and standardization techniques. In particular, each module's environment is strictly defined by its interface to other components. No module can rely upon information outside of this interface. Module selection should be done in logical fashion to minimize the size of the interfaces between modules.

The on-line databases can be considered part of this interface, but are only accessible through their defined interfaces.

The software must be strictly modular, i.e. the functionality of a subsystem should correspond to that which belongs to that subsystem and only to that, so that software for different subsystems can be installed and maintained independently of all the rest.

This is needed in particular for multi-instrument operation, for example, as instruments share the same subsystems on the telescope.

At the same time, the possibility must exist to acquire information about other parts of the system (for example, telescope coordinates) (as mentioned in the control flow requirements in Section 3.1.1).

It also important that there are no undesired interactions between subsystems. This may be enforced either at the client/server interface or at the message system level.

3.5.10 Contingencies

A. *Fault tolerance.* The security and safety of the system should be guaranteed even in the event of failure of any component, including the higher-level software.

B. *Redundancy.* Hardware redundancy is a not requirement of the Gemini systems. However, the ability to reconfigure the software if one actuator fails is desirable. Data redundancy is also a requirement, to prevent a single failure from causing the loss of collected data. The goal is to minimize the effects of single-point errors throughout the system.

3.5.11 Concurrency

As much as possible, the system is to take advantage of parallel operation to improve efficiency. The Telescope Control System should be capable of detecting and invoking parallel operation as it is responsible for control of all of the telescope and enclosure subsystems.

3.6 LIFE CYCLE ASPECTS

The Gemini 8m Telescopes software has to be developed according to a structured and formally defined development model. The purpose of this is to:

- make the development process visible (for quality and maintenance reasons)
- avoid incomplete specifications
- discover errors early (via review and audit procedures)
- secure structured, maintainable software
- guarantee software quality
- improve and ease cooperation.

This has to be defined in [SMP] and [SCP] (see Applicable Documents, Section 1.3), which will cover:

- Development methods
- Formal approval, review and audit procedures
- Documentation
- Coding and debugging
- Simulation
- Component and integration-verification and testing
- Configuration and version management.

3.7 DEVELOPMENT ENVIRONMENT

The development environment for the Gemini 8m Telescopes software consists of the computer hardware and system software (operating system, languages and tools) chosen optimally to support the development model presented in the previous section on life-cycle aspects.

The choice of such an environment has to be defined, together with the choice of all the Gemini 8m Telescopes standards, but excluding the target computer hardware, in Section 4.2.5 on page 4- 7.

The following criteria for later selection shall be kept in mind:

- Emphasis on development and productivity requirements, including cross support tools
- Portability of software (target hardware independence)
- Hardware and vendor independence
- Use of industrial and defacto standards.

3.8 INSTALLATION ASPECTS

Test procedure methods have to be defined in the Software Test Plan (STP), while test plans shall be written for all individual software packages and modules comprising the Gemini 8m Telescopes software.

Apart from the component and integration test procedures, a formal release system should exist at package and module level, which should be checkable on-line by the operational procedures for consistency. Every system must thus be able to supply its current version upon request.

The test operation level has already been discussed in Section 3.2.1 on page 3- 3.

The phasing of the various parts of the project is given in the [SMP].

4 SPECIFIC REQUIREMENTS

This section presents the specific attributes and requirements for Gemini software.

Only the high-level requirements for Gemini software are presented here. Detailed specifications for the subsystems are found in the individual chapters of the Software Design Description.

4.1 ATTRIBUTES

Regardless of application, the Gemini software exhibits numerous common attributes. These attributes are described in this section.

4.1.1 Maintenance

The Gemini system maintenance philosophy is described in the *Software Management Plan* (SMP).

Preventative maintenance is scheduled as specified in the *Gemini Design Requirements Specification*.

4.1.2 Modularity

All software is to be developed using typical modularization and standardization techniques. In particular, each module's environment is strictly defined by its interface to other components. No module can rely upon information outside of this interface. Module selection should be done in logical fashion to minimize the size of the interfaces between modules.

The on-line databases can be considered part of this interface, but are only accessible Reliability and availability

A measure of fault rates should be done during commissioning to establish baseline rates for system reliability monitoring.

There are to be recovery procedures to restart after error failure. See section Section 3.2.5 on page 3 - 5.

During science planning, there should be validity and feasibility checks to help ensure effective and efficient use of the telescope. Where appropriate, these checks should also be performed during operation.

The system should be constantly monitoring active subsystems to be sure they are operating correctly before sending commands to each subsystem. This monitoring should continue on inactive subsystems.

4.2 OTHER CONTROLS AND SOFTWARE REQUIREMENTS

4.2.1 On-line database subsystems

All telescope and instrument parameters are kept in an on line database to permit easy implementation of table-driven applications. The interface between software control packages is normally done via interface calls to the on-line database.

The requirements for this database package are:

A. All telescope, instrument, and detector control information is to be available at any operation level.

B. Access times to the database are to be in the range of **2-3 msec** per access.

C. Asynchronous writes are to be supported, allowing for concurrent operation.

D. Time-access critical information is available in memory.

E. There is to be a consistent and logical (i.e. name based) access method.

F. The database must support both remote access and distributed data.

The internal (within the IOC) implementation of this database is to be based on EPICS. The implementation within the host workstation is TBD.

4.2.2 Communication subsystems

4.2.2.1 Remote Operations

A fundamental criteria of Gemini telescope operation is that it support a full implementation of remote operations. This includes *remote observing, remote control of telescope, enclosure, and instruments, multipoint monitoring, remote monitoring, remote access for testing, development, diagnostics, and maintenance,*

It is expected that all operational capability found in on site operations is extended to remote operations, with some degradation in performance resulting from WAN bandwidth considerations. This means the video data signals must be encoded digitally and transferred via the WAN to remote sites.

There must be some form of security to control access to system features, possibly restricting some operations to specific remote sites (e.g. Hale Pohaku or Hilo in Hawaii, Cerro Tololo in Chile, etc.)

4.2.3 Data specification

The various types of data were presented earlier. This section examines the input, output, and interrelationships of the various data types.

4.2.3.1 Data framework

Final storage locations for the data types is presented here, along with a description of the different databases that are available.

A. The on-line data store holds astronomical data for the current observation.

B. Astronomical data are automatically stored onto the Archive medium (external software).

C. Star catalogs are available in Astronomical object catalogs (external software).

D. Telescope and instrument parameters are distributed in databases across the IOCs for those systems. There is also a central repository maintained by the OCS that holds these databases for down-loading to the IOCs. (developed software)

E. All additional data that is not required on line (configuration information, detailed documentation, operation logs, etc.) are stored in a relational DBMS. (supported software).

4.2.3.2 DATA INPUTS

Input data are all data that are predefined at operation start. This includes catalogs, calibrations and flat fields available in archives, etc.

Observing commands (whether entered interactively or via the Sequencer) are input data that provide information on the course of operation and trigger events.

4.2.3.3 DATA OUTPUT

Replies to commands, including status information, updates to parameters, video and astronomical information are considered data outputs.

These data end up in the different databases.

Operational information, such as logging messages, alarms, and errors are special forms of output data, since they are made available for later inspection and debugging.

4.2.3.4 INTER-FUNCTION DATA DEFINITIONS

One criteria is that sufficient information be recorded during an observation to recreate the sequence of events that occurred during the observation. This requires that all input and output data be logged appropriately.

Given an initial set of configuration parameters, the Gemini system operates via a sequence of commands. This operation is complemented by using previously stored data, calibrations, and star catalogs

4.2.4 System operational requirements

4.2.4.1 NORMAL OPERATION

A. *Automatic operation.* This is the normal mode of operation. The observation is performed through a preplanned program requiring little or no interaction with the observer.

B. *Interactive operation.* Science planning and program changes are accomplished through interactive operation. Is is possible to enter interactive operation from automatic operation to handle exceptional conditions.

This is the normal mode of operation at maintenance and test levels.

C. *Modes and control.* Normal operation is possible at all operation levels (observing, maintenance, and test) and applies to the following conditions (where appropriate) on any subsystem:

1. Observing
2. Stand-by
3. Maintenance and testing

D. *Performance*. Overall performance of the Gemini telescope is defined as the percentage of viewable time during which exposures have been taken (i.e. sum of all exposure times over available time for exposures).

To improve this performance, all possible concurrencies in system operation should be used. Best use of concurrency occurs when using the Sequencer.

E. *Start-up and shut-down*. There are start-up and shut-down procedures that must exist at many different levels:

1. Cold start-up, starting the system from scratch (including time to download all software) should take about 5 minutes. This does not include time to start-up the telescope or instruments.
2. Warm start-up, starting from scratch but also excluding software download time should take about 1 minute.
3. Telescope start-up, measured from end of cold or warm start-up should be about 4 minutes.
4. Instrument start-up, measured from end of telescope start-up, should take 2 minutes or less.

There must be a way to shut down all subsystems (hardware and software).

All start-ups and shut-downs are to be automatically logged with time stamps, to allow for statistics on system availability.

F. Logging.

1. System logging information should include all important events, properly time-stamped and indexed. The goal is to be able to recreate the steps in a observation from the system logs.

G. Engineering Logging

1. It must be possible to log engineering data at up to 200 Hz rates for short periods of time. This data must be available to external software packages for analysis.
2. Long-term logging of engineering data must be possible at slower (1 Hz or less) rates, into a common format (baselined as SYBASE).

4.2.4.2 OPERATION IN FAILURE MODE

A. *Fault-tolerance and recovery.* See Attributes, Section 4.1 on page 4 - 1.

B. *Error logging*. All errors should be reported using a common, system-wide procedure.

There are three types of errors:

1. *Fatal*. Fatal errors occur if there is no acceptable recovery procedure that will allow operation to proceed. Under fatal error conditions, the system falls back to a safe "backup" state requiring human intervention for restart.
2. *Serious*. Under serious errors, the system does not need to move off-line, but the current operation cannot be completed. Serious errors require human intervention to restart full operation.
3. *Warning*. All other unexpected conditions result in warnings that are properly logged. The system continues operation, though perhaps with reduced efficiency.

All subsystems must group errors into these categories. In addition, errors that result in an "alarm" should be described, along with the proper action required to acknowledge and eliminate the alarm condition.

Besides the time-stamp, error logging should provide enough information to trace the condition back to its apparent source, both in equipment and in event sequence.

There should be tools available to extract error (and other) logging information by subsystem component, time- sequence, previous events, and so on.

The Gemini Control System formally distinguishes *alarms* from *errors*. Errors result from failures to successfully complete commands, while alarms represent asynchronous failures. Note that an error may result in an alarm.

There are two types of alarm conditions. The first are automatically monitored alarms, which exist as long as the errors persist and are then automatically cleared. The second type of alarm require human acknowledgment to clear.

C. *Recovery.* In addition to start-up procedures, there must be well-defined recovery procedures for any subsystem that has become inoperative.

Command retries must be included in the system for most common timeouts or no-response conditions. These retries should occur automatically in the command handling to avoid unnecessary error conditions.

D. *Performance*. The performance of error-mode recovery is specific to the subsystem and is defined in the Functional specification for that subsystem.

- Normally, there is no recovery possible from a fatal error except to shut-down and then restart the subsystem. However, in the case of an instrument failure, it may be possible to continue operation by rescheduling to use observations that do not require that particular instrument.
- For serious errors, it may be possible to continue operation with degraded performance. For example, failure of automatic tracking may require manual tracking; other errors may result in operation with a different instrument.
- Under normal conditions, the number of warnings should be small. The system should monitor the rate of warning messages since an increase might indicate that some tuning or maintenance is appropriate. Ideally, such conditions should be noted by the subsystems before reaching the OCS level.
- Failure conditions should not *cascade*. That is, failure of one subsystem should not affect other, working, subsystems, including communication links.

4.2.5 Standards

Given the size of the Gemini system and its long expected lifetime, it is important that standards are provided for system design and development.

4.2.5.1 SOFTWARE STANDARDS

4.2.5.1.1 Software methods

All Gemini supported software is to be developed using a formally defined model. The Ward/Mellor approach to developing real-time systems is used and covers:

A. Analysis, design and development methods

B. Review procedures

C. Documentation

D. Coding and debugging

E. Simulation

F. Testing and integration

Functional analysis is done using a CASE tool suitable for use with Ward/Mellor techniques, such as **TSEE**, by Westmount Technologies.

The design is to use the techniques and diagrams introduced by Ward and Mellor and reviewed by the Gemini staff.

Detailed design and development standards are not specified, but expected to conform to system goals and established practice.

An object-oriented approach is encouraged but not required.

4.2.5.1.2 Operating systems

The upper levels (User-interface and OCS) are assumed to not require a real-time operating environment. However, the operating environment at these levels is expected to provide sufficient performance for both human interaction and communications.

Real-time support is required at the IOC level.

A. Development system software. The choice for a development environment is based on the following criteria:

- Productivity and development tools
- Software portability and hardware independence
- Vendor independence
- Industry and de-facto standards
- Support for state-of-the-art user interfaces
- Support of a distributed environment

Given these criteria, the following recommendations exist.

1. The development operating system is **UNIX System V, Release 4**, using only **POSIX** compliant calls to the operating system. In particular, vendor specific extensions are **not** permitted. Compliance to POSIX is measured against the book: *POSIX Programmer's Guide - Writing Portable UNIX Programs,* by Donald Lewine, O'Reilly & Associates, Inc. 1991.
2. Any special needs that cannot be acceptably met by the available operating system calls are to be discussed with the GPO on a case-by-case basis.
3. Some packages (such as GUI interfaces and database interfaces) have higher portability requirements that may require the specification of higher level interfaces to the OS.
4. The GUI systems are to use **X-windows, version 11, release 5 (X11R5) or release 6 (X11R6)**. The principle language for developing GUIs is **Tcl/Tk**.
5. The **GNU software tool suite** should be used where possible.
6. **Imake** is recommended, with GNU's autoconfigure utility an acceptable alternative.
7. **CVS** is the recommended version control system.

B. *IOC and cross-development system software.* A number of real-time operating systems were examined for compatibility with the goals for the Gemini Project. At the same time, the availability of cross-support development environments was considered.

The recommendations are:

1. The IOCs will use the real-time operating system **VxWorks** (from Wind River Systems).
2. The communication interface between the IOCs and Unix workstations is to be **IMP.**
3. It is assumed that all IOC software is to be developed on Unix workstations in a windowed development environment.
4. The cross-compiler to be used for IOC software development is one specified by the Gemini Project. It is the **GNU cc** compiler as furnished with VxWorks.
5. IOC software development is to be done using a common development environment, as specified in the *Software Programming Standards* (SPS)

C. *Software below the IOC level.* There is likely to be software below the IOC level, but it should not require downloading, except possibly for upgrades, typically being placed into ROM or FLASH memory.

Any access to this embedded software while it is connected to the system is strictly through software in the IOC. From outside the IOC this software appears as part of the electronics.

Although this software is not considered part of the Gemini software, it is obviously an advantage (for maintenance purposes) to have it conform to the requirements of Gemini software.

D. *Installation system software..* The host workstation operating system is Unix, currently baselined as **Solaris 2.3 or above**.

The installed target IOC operating system is **VxWorks**, with **EPICS** as the interface.

4.2.5.1.3 Languages and coding standards

Details of these standards can be found in the Software Programming Standards document.

4.2.5.1.4 Software configuration and tools

These are defined in the Gemini Configuration Control Plan.

4.2.5.1.5 Software communication standards

The following recommendations are made:

A. Communication hardware independence is accomplished by using the DARPA **TCP/IP** communication protocol over LAN's and the WAN.

B. The communication software must support the standard **ARPA** services (telnet, FTP, SMTP, etc) as well as **NFS, RPC, IPC,** and the Unix **socket interface library**.

4.2.5.1.6 Software Interfaces

The recommendations are:

A. The software interface between workstation and IOC is to use **DRAMA's IMP protocol** for all control communication, using SDS as the command structure..

B. Interprocess communication on the host workstations is through the same **IMP protocol**.

C. Communications between real-time components is to based on the **EPICS Channel Access protocol**.

D. Data communications are typically through **IMP/SDS**.

E. Further details of the software interfaces are found in the Software Design Description.

4.2.5.1.7 External software packages

A. *Off line Data reduction*. The off-line data reduction facility is not considered part of the Gemini software. However, it will be possible to connect such software into the Gemini system if interface routines are developed for connecting to the Gemini system. These interface routines are part of the Gemini software and must conform to the standards given in this document. Packages that might be used for off-line data reduction include **ADAM, IRAF, Midas, IDL** and **Khoros**.

B. Quick-look analysis will be through **PV-Wave/IDL**.

C. *Archiving*. All astronomical data are saved into the Archiving system. Data storage and transport for astronomical data will be in FITS format. On-line access to the archiver will be through **STARCAT**.

D. *Star catalogs*. The catalogs available on the Gemini system are described in TBD, as are software access requirements. Both on-line and off-line access is done according to **STARCAT**.

4.2.5.1.8 Databases

Astronomical data are stored both in the Archiving system and in the data storage subsystem.

IOC control databases are distributed across the IOC subsystems, with down-loadable initialization copies available on the Gemini disks.

Configuration, logging, maintenance schedules, and subsystem documentation are to be kept in the **commercial relational database**.

4.2.5.2 HARDWARE STANDARDS

Only the development systems are considered here. Installed systems are to be chosen later, based on available technology and experiences with the development systems.

4.2.5.2.1 Hardware requirements

The computer hardware used for development must conform to the following:

A. Workstations must match specified software standards and present a well-designed development environment, including cross-support for VxWorks development, if needed.

B. Workstations are expected to be state-of-the-art systems (CPU, communications support) in a scaleable family. This allows the migration of development systems to advancing technology.

C. It is assumed that workstations support Ethernet IEEE-802.3 and FDDI interfaces for communications.

D. Internal data formats must be compatible across workstations used for development.

E. SCSI-interface peripherals are to be available.

F. Reliability, manufacturer support, and upgrade capability of the development systems will be considered when selecting the final target systems.

4.2.5.2.2 Development computers

The following recommendations are made:

A. Workstations.

1. Where Gemini is providing the computer equipment as part of a contractual agreement, the developmental workstations are to be **SUN SPARC-10 (model 41) or SPARC Classic** with **64MB** of dynamic memory.
2. Internal Gemini software development is done in a multi-platform environment, enabling final selection of target machines to be made based on experiences with development equipment.

B. Peripherals

1. Workstations are to have at least **2GB** of disk.
2. **DDS DAT cartridges** are to be available for data acquisition on development systems, backups, and possibly for software transport (**Internet** is the preferred means of transport).
3. A **CD-ROM** is needed for SUN OS and software upgrades.
4. Any compression on a transport medium is to be done using a standard commonly available algorithm and archives are to be in **tar** format

C. IOCs.

1. Where Gemini is providing the IOC CPU as part of a contractual agreement, the development systems are to be Motorola MVME-167B (68040@33MHz) with at least 16MB of dynamic memory.
2. The IOC computer cards for development work are defined in the ICS work package description.
3. The target IOC computer cards are not to be established until as late as possible, to take advantage of technology advances while staying compatible with the development systems. A baseline system is included as part of the ICS work package.

4.2.5.2.3 Electronics hardware interfaces

All accesses to the control electronics hardware will be through VME IOC's, using the EPICS software. No other control interfaces are permitted.

4.2.5.3 SUB-NETWORK

There are places in the Gemini system where software exists below the IOC level that is interfaced to IOC software. The standards for this interface is not part of this specification. The use of this software is to be explained and discussed on a case-by-case basis.

In cases where there is a distributed network of target microprocessors involved at this level, it is appropriate to provide a standard sub-network. This type of Fieldbus is to be implemented using an **ALAN/BRADLEY** bus, **ProfiBu**s, or **other approved Fieldbus**.

In cases where there is a distributed set of commercial equipment to be controlled the recommended method is to use an **ALAN/BRADLEY** system to control the equipment manually and interface to it using **RS-422** or similar connection to an IOC.

4.2.5.4 COMMUNICATION HARDWARE

All communication hardware must meet the data flow requirements and software protocols defined in this document. The following recommendations are based on currently available technology. It is entirely appropriate that changes will be made to take advantages of changes to that technology.

A. The control LAN is based on Ethernet IEEE 802.3.

B. The Time distribution systems are described in the Software Design Description..

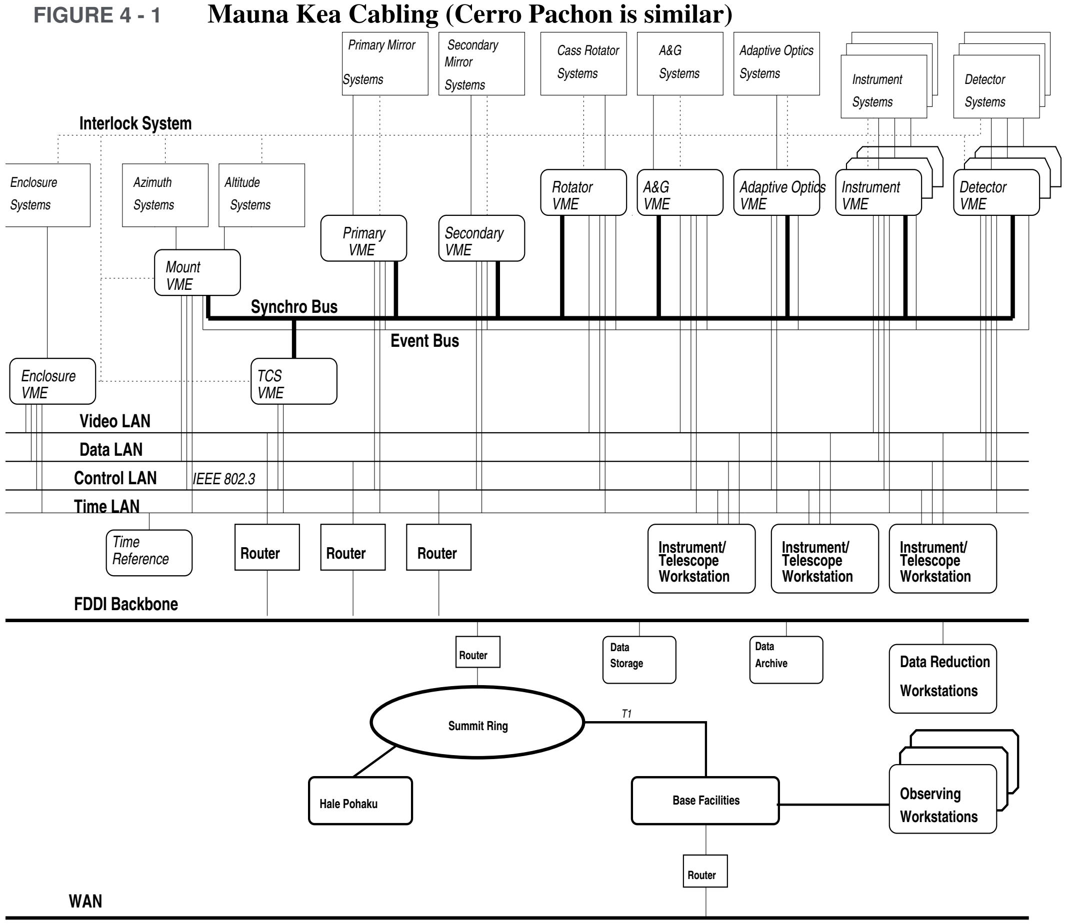
C. The detector LAN is also based on Ethernet, at least for the development systems. A solution meeting the performance requirements of the final system is to be decided later.

D. The backbone LAN cannot be Ethernet because of bandwidth requirements. A FDDI system is recommended for the backbone LAN.

E. Cabling of LANs is shown in Figure 4 - 1. Also shown are cabling requirements to the base facility. Requirements for cabling are given in the Electronic Design Specification.

F. Interlock connections must be provided for all critical subsystems. The software interface to the interlock system is defined in the *Instrument Control System Infrastructure* Work Package Definition.

FIGURE 4 - 1 **Mauna Kea Cabling (Cerro Pachon is similar)**



G. Routers, gateways, etc. are defined in the Electronic Design Specification. Their logical placements are shown in Figure 4 - 1, above.

H. Links to the WAN are defined in the Electronic Design Specification and shown in Figure 4 - 1, above. There is a requirement of at least one dedicated link of a **T1** bandwidth for remote support, development and testing.

4.2.5.5 Re-use of existing software

Wherever possible, Gemini software is to take advantage of existing software. However, all existing software is to be evaluated in terms of the specifications given here. This helps reduce life-cycle costs and maintenance efforts.

4.2.6 Life cycle constraints

Life cycle constraints are discussed in the Gemini Software Management Plan.

4.3 User Requirements for EPICS Developers

The Experimental Physics and Industrial Control System (EPICS) toolkit is the foundation of the Gemini control system. EPICS was originally developed at the Los Alamos and Argonne National Laboratories for use in large accelerator control and diagnostics systems. It is now an established standard within the international High Energy Physics community and is gaining acceptance amongst astronomical sites.

Within the EPICS community there is understood to be an informal distinction between two classes of developers, labeled 'Internals' and 'Applications'. Although these definitions are nowhere written down, this understanding serves as the basis for many discussions with the community and for the nature of training classes.

4.3.1 EPICS Components

Note: The bulk of this section is taken from the EPICS document 'EPICS Overview'.

EPICS consists of a set of hardware and software components from which a control system can be created.

4.3.1.1 Hardware Components

- OPI Operator Interface.

  This is a UNIX based workstation which can run various EPICS tools.
- IOC Input Output Controller.

  This is VME/VXI based chassis containing a Motorola 68xxx processor, various I/O modules, and VME modules that provide access to other I/O buses such as GPIB.
- LAN Local area network.

This is the communication network which allows the IOCs and OPIs to communicate. EPICS provides a software component, Channel Access, which provides network transparent communication between a Channel Access client and an arbitrary number of Channel Access servers.

4.3.1.2 SOFTWARE COMPONENTS

4.3.1.2.1 IOC Software

Database

The heart of an IOC is a memory resident database together with various memory resident structures describing the contents of the database. EPICS supports a large and extensible set of record types, e.g. ai (Analog Input), ao (Analog Output), etc.

Each record type has a fixed set of fields. Some fields are common to all record types and others are specific to particular record types. Every record has a record name and every field has a field name. The first field of every database record holds the record name, which must be unique across all IOCs attached to the same TCP/IP subnet.

A number of data structures are provided so that the database can be accessed efficiently. Most software components, because they access the database via database access routines, do not need to be aware of these structures.

Database Access

With the exception of record and device support, all access to the database is via the channel or database access routines.

Database Scanning

Database scanning is the mechanism for deciding when to process a record. Four types of scanning are possible: Periodic, Event, I/O Event, and Passive.

Periodic: A request can be made to process a record periodically. A number of time intervals are supported.

Event: Event scanning is based on the posting of an event by any IOC software component. The actual subroutine call is: post_event(event_num)

I/O Event: The I/O event scanning system processes records based on external interrupts. An IOC device driver interrupt routine must be available to accept the external interrupts.

Passive: Passive records are processed as a result of linked records being processed or as a result of external changes such as channel access puts.

Record Support, Device Support, and Device Drivers

In order to remove record specific knowledge from database access, each record type has an associated record support module. Similarly, in order to remove device specific knowledge from record support, each record type can have a set of device support modules. If the method of accessing hardware is complicated, a device driver can be provided to shield the device support modules. Many record types, in particular all types not associated with hardware, do not have device support or drivers.

The IOC software is designed so that the database access layer knows nothing about the record support layer other than how to call it. The record support layer in turn knows nothing about it's device support layer other than how to call it. Similarly the only thing a device support layer knows about it's associated driver is how to call it. This design allows a particular installation and even a particular IOC within an installation to choose the set of record types, device types, and drivers it wishes to use. The remainder of the IOC system software is unaffected.

Every record support module must provide a record processing routine. It is this routine that is called by the database scanners. Record processing consists of some combination of a standard set of functions.

Database Monitors

The routines described in this section provide a callback mechanism for database value changes. This allows the caller to be notified when database values change without constantly polling the database. A mask can be set to specify value changes, alarm state changes, and/or archive changes.

At the present time only channel access uses database monitors. No other software should use the database monitors. Because they are of interest only to channel access, the monitor routines will not be described.

4.3.1.2.2 Channel Access

Channel access provides network transparent access to IOC databases. It is based on a client-server model. Each IOC provides a channel access server which is willing to establish communication with an arbitrary number of clients. Channel access client services are available on both OPIs and IOCs. A client can communicate with an arbitrary number of servers.

It should be noted that channel access does not provide access to database records as records. This is a deliberate design decision. This allows new record types to be added without impacting any software that accesses the database via channel access. A channel access client can communicate with multiple IOCs having differing sets of record types.

4.3.1.2.3 OPI Tools

EPICS provides a number of OPI based tools. These can be divided into two groups based on whether or not they use channel access. Channel access tools are real time tools, i.e. they are used to monitor and control IOCs.

Channel Access Tools

- MEDM

  Motif version of combined display manager and display editor.
- DM   Display Manager.

  This tool reads one or more display list files created by EDD, establishes communication with all necessary IOCs, establishes monitors on process variables, accepts operator control requests, and updates the display to reflect all changes.
- ALH   Alarm Handler.

  This is a general purpose alarm handler driven by an alarm configuration file.
- AR   Archiver.

  This is a general purpose tool to acquire and save data from IOCs.
- Sequencer

  A tool which runs in an IOC or OPI and emulates a finite state machine.
- Other OPI CA clients

  It is possible to interface preexisting software systems to the channel access library. This has been done for several commercial packages including IDL/PvWave, Mathematica, and WINGZ to allow access and manipulation of EPICS process variables by these applications.

Other OPI Tools

- DCT   Database Configuration Tool.

  This tool is used to create a run time database for an IOC
- CAPFAST and GDCT   Graphical Database Configuration Tools

  These tools are used to create a run time database for an IOC
- EDD   Display Editor.

This tool is used to create a display list file for the Display Manager. A display list file contains a list of static, monitor, and control elements. Each monitor and control element has an associated process variable.

- SNC State Notation Compiler.

  It generates a C program that represents the states for the IOC or OPI Sequencer tool.

- Build Tools

  Tools are available to create the various database components from ASCII definition files.

- Source/Release

  EPICS provides a Source/Release mechanism for managing EPICS.

4.3.1.2.4 EPICS Core Software

EPICS consists of a set of core software and a set of optional components. The core software, i.e. the components of EPICS without which EPICS would not function, are:

- Channel Access - Client and Server software
- Database
- Scanners
- Monitors
- DCT
- Build Tools
- Source/Release

All other software components are optional. Of course any application developer would be crazy to ignore tools such as MEDM (or EDD/DM). Likewise an application developer would not start from scratch developing record and device support. Most OPI tools do not, however, have to be used. Likewise any given record support module, device support module, or driver could be deleted from a particular IOC and EPICS will still function.

4.3.2 GEMINI Breakdown of Tasks

Each of the Gemini Controls work packages will be involved with some aspect of EPICS development ranging from creation of new device drivers to database creation and the design of new CA clients.

The details of the EPICS work required can be broken down into several broad categories which reflect the nature of the various work packages. This breakdown is given in the following subsections:

4.3.2.1 NOT COVERED BY ANY WORK PACKAGES

- Channel Access - Client and Server Software, Scanners, Monitors

  Gemini will not modify the heart of EPICS. We leave this to Los Alamos.
- Host-based Development Tools (CAPFAST, EDD, etc.)

  Neither will Gemini plan on creating new development tools.

4.3.2.2 SIC WORK PACKAGE ONLY

- Record Support, Device Support, and Device Drivers

  For Gemini this entails support for new devices:

  PMAC, VMIC5578, BANCOMM

4.3.2.3 OCS AND DHS WORK PACKAGES ONLY

- Other OPI CA Clients

  The OCS may need a CA client that implements the OCS Attribute/Value protocol.

  The DHS may need a CA client that gathers logging information. This could be an EPICS AR (or AR_cmd) task.

4.3.2.4 REAL-TIME (IOC-BASED) WORK PACKAGES

- IOC Database
- IOC SNC programs
- Files associated with OPI Channel Access Tools

  These would include MEDM .adl display definition files, PvWave/IDL scripts, AR request files, ALH alarm configuration files, and the like.

4.3.3 GEMINI Definitions

For the purposes of work performed for the Gemini 8-m Telescopes Project the following definitions will be used to differentiate between the two flavors of EPICS development work:

Internals work would include any of the following:

- Channel Access - Client and Server Software, Scanners, Monitors
- Record Support, Device Support, and Device Drivers
- Host-based Development Tools (CAPFAST, EDD, etc.)
- Other OPI CA Clients

The first two items would be suitable topics for an advanced EPICS course on **EPICS System Development**. The latter two items would be covered under **Advanced Application Development** and a case can probably be made that they constitute a category unto themselves which should be labeled something like 'Internals - Tools'

Applications work would then include any of:

- IOC Database
- IOC or OPI SNC programs
- Files associated with OPI Channel Access Tools

These are all topics that would be covered in an **EPICS Basics, Building and Using Applications** course.

4.3.4 Required Developer Skills

For each class of developer a specific set of skills is required. The following sections list those that would be commonly used.

4.3.4.1 APPLICATIONS DEVELOPER

- Knowledge of VME and other I/O electronics
- EPICS Database design techniques
- SNC programming
- Use of standard EPICS OPI tools
- Basic knowledge of UNIX and VxWorks as development environments

4.3.4.2 INTERNALS DEVELOPER

- Advanced C/C++ programming techniques
- UNIX internals
- VxWorks internals
- Real-time programming techniques
- Motif application development skills
- Networking skills: TCP/P and UDP/IP sockets

4.3.4.3 EXPECTED DISTRIBUTION OF DEVELOPER CLASSES

4.3.4.3.1 Current Distribution Within The High Energy Physics Community

The experience of the High Energy Physics community is that only a core group of Internals developers is required for maintenance and enhancement of EPICS IOC software and of the standard development and CA client tool set. Specifically, the IOC core software is almost exclusively in the domain of the Los Alamos AOT-8 office.

Two major flavors of development tools are in use with support coming from key groups within Los Alamos and Argonne.

Many sites contribute custom CA client tools as this generally involves C or C++ programming under a UNIX O/S to interface to the standard CA library.

In addition, many sites create their own database record types and new device support and driver layers.

Although there is presently a large amount of activity in the enhancement of EPICS Internals software especially in the fields of non-IOC core software, this is an order of magnitude less than the amount of Applications work under way at over 20 distinct programs.

4.3.4.3.2 The APS CATS Experience

Another large group of predominately Applications developers that are starting work are the Collaborative Access Teams (CATS) at Argonne’s Advanced Photon Source (APS).

The APS is a facility to produce high brilliance X-rays for a variety of experiments and includes 36 experimental halls with 2 lines each. As of March 1994, there were 15 CATS at the APS with most CATS responsible for a single sector (2 beamlines). Each CAT has its own funding for designing/building/operating the physical beamline, instrumentation, controls, and data acquisition systems.

Because the experimental areas must be coordinated with the main facility control system and since 25% of the beamtime must be made available to independent investigators it was strongly recommended that EPICS be used as the toolkit for all CATS development work. Not only would this simplify the integration with the central control system but would also provide a common environment for facility visitors.

Typically, each CAT has 1 or 2 developers, and the major focus is not on the IOC software although some new records, like the Multichannel Analyser, have been created, but rather on the OPI CA client applications. Heavy use is being made of both Tcl/TK and IDL to support high-level sequencing and data reduction. This is similar to the Gemini architecture as the bulk of the control system intelligence is contained within OPI CA client tasks.

Other items of concern are:

- Different science programs in use at each CAT which have complex associated data acquisition and reduction routines.
- The creation of large data sets.
- Simultaneous local and remote monitoring and intervention.
- Ease of use for one-time users.
- Data export to home institutions.

4.3.4.3.3 Planned Distribution Within The Gemini Project

The Standard Instrument Controller work package is tasked with laying the Internals ground work for all other Gemini control system work packages. In this project, the only required IOC-based Internals work is the creation of record, device, and driver support for three VME I/O cards. It is anticipated that all other IOC-related work packages will be performing exclusively Applications work.

In the advent of future work packages requiring Internals work it is planned the RGO SIC development staff will be made available. In this way the number of Internals staff is kept at a minimum.

5 COMMANDS

5.1 COMMAND ROUTING

There are several means of transmitting commands and data across the Gemini system:

A. *Direct*: A command source may directly specify the target subsystem. This is the most common means of command transfer from the Observatory Control System to the subsystems.

B. *Channel*: Commands and data may be routed through a *channel*. This mechanism requires that some (unknown to the source) target subsystem has been attached to the channel as the target. It is possible that the channel is 'intelligent', in the sense that data may be converted from one form to another during transmission. This might be implemented by connecting two channels with a conversion module. An example of a channel connection might be the data stream between the A&G system and the Primary Mirror Control System. The A&G might be transmitting Zernickes to the PMCS, which is receiving Shack-Hartmann data. The intelligent channel would be responsible for the conversion. It is possible that a channel have multiple sources, multiple targets, or both! Finally, a channel might use a *separate* physical route, to prevent large data transmissions (say) from interfering with the 'normal' command streams.

C. *Broadcast*: Some commands and data signals are simply broadcast across the control system. Any subsystem may examine these communications and respond or ignore as appropriate. For example, the OCS might broadcast a **RunTest** command, causing all subsystems to perform self-testing. As another example, a critical system failure might cause a **ShutDown** command to be broadcast.

5.2 Command Structure

All commands contain the same general structure:

Identification:
Command_ID
Source
Target

The Command_ID is an identification that is unique to that specific instance of each command. A portion of the ID is monotonically increasing and functions as a timestamp for journalling and tracing. This is to be implemented using a technique analogous to the CFHT odometer.

The Source and Target fields indicate the originator and the recipient for the command, as expected. The Target may be a specific recipient, a Channel, or simply a Broadcast, but the Source is always the origination point of the command.

Instruction:
Opcode
Parameter set

The remainder of the command provides the body of the command. The Parameter list structure is dependent on the Opcode.

5.3 General Purpose Commands

This section presents the "generic" commands that all Gemini systems must respond to. There are also "Control Commands" common to all Gemini systems to provide specific functionality during observing. These control commands are presented in the Gemini Software Design Description. Specific commands are presented in the Software Design Descriptions for the individual subsystems.

5.3.1 Status Commands

The following commands are "status" queries used to obtain information about the state of the subsystem.

GetVersion The subsystem returns its version identification as a string value.

GetStatus The subsystem reports its current status as one of:

- DOWN - the subsystem is not operational
- BOOTED - the subsystem has been booted, but not yet configured
- CONFIGURING - the system is in the process of configuring (Steven).
- CONFIGURED - the subsystem is configured, but not yet initialized
- INITIALIZING - the subsystem is doing initialization actions
- RUNNING - the subsystem is running at observing level
- MAINTENANCE - the subsystem is running at maintenance level
- SIMULATION - the subsystem is running in simulation mode
- DISABLED - the subsystem is functional, but has been commanded to ignore control commands
- SHUTDOWN - the subsystem is in the process of shutting down
- LOCKED - an interlock exists on this subsystem.

GetState The internal state of the subsystem is returned as one of:

- READY - the subsystem is fully operational, but currently idle.
- BUSY_ON command - the subsystem is working, and currently processing the indicated command.
- NOT_READY - the subsystem is not responding to commands at this time. A GetStatus command can be used to determine the current status.

GetID The subsystem reports its unique identification tag.

GetConfiguration Detailed configuration information is provided. This configuration information includes details of any settings internal configurations (filters, positions, etc.) and, if requested, the detailed results of any self-tests.

5.3.2 Generic Commands

The following commands affect the behavior of a subsystem independently of the specific task of that subsystem.

SetStatus The target subsystem is to enter the state associated with the specified status argument. For example, SetStatus SIMULATION causes the targeted subsystem(s) to enter simulation mode. Setting the status to SHUTDOWN disables a subsystem. Not all status levels are reachable through SetStatus.

SetConfiguration This command directs the downloading of a configuration from the host to the target subsystem. It is important that care be taken to avoid configuration values that would result in motion, etc. This command can operate with the GetConfiguration command given above to provide 'sequencing points' for subsystem restarts or roll backs.

RunTest The subsystem runs a self-test and reports the result as one of:

- OK - the subsystem has detected no problems, it is running within specifications
- BAD - the subsystem has detected problems that prevent its successful operation
- WARNING - the subsystem has detected a problem that may prevent it from operating to full specification, but does not prevent it from functioning at this time

There may be an argument describing a particular self-test to run. The individual self-tests are specific to the subsystem and are given in the appropriate Work Package Description. Detailed results of self-tests are available through the GetConfiguration command.

SetLogging Logging by the indicated subsystem is set to occur at the specified level.

5.4 DATA COMMUNICATIONS

Communication of information is similar across all systems in the Gemini controls network. Instances of this communication is via routes, which may be direct channels, intelligent channels, or broadcasts. Communication via a route is typically asynchronous, with commands available to establish synchronization as needed.

OpenRoute A route of the indicated class is being established for some item. The command includes sufficient information on the route to permit the IOC to communicate through the route. The 'route' may be a specific subsystem, channel, or broadcast.

CloseRoute A previously-opened route is being closed. Modules at either end of the channel are notified and take appropriate action.

ResetRoute The route is to be reinitialized. Modules receiving this command are to flush buffers associated with this route and reset operating parameters to initial conditions.

TestRoute A test message is transmitted across the route. All receivers on the route are to respond with an identification.

AwaitItem The module sending this command is blocked until information is received via the indicated route, or a timeout occurs.

SetCallback A callback is established to process data across the indicated route.

ClearCallback A previously established callback is removed.

5.5 IOC COMMANDS

The following commands are common to all IOC-based subsystems. (Most control subsystems are IOC-based.) These commands are processed by the IOC subsystem.

5.5.1 Local Database Access

IOC crates in the Gemini system have local databases. Under the Gemini control system, it is expected that the majority of subsystem control is accomplished through these databases. The following commands provide access to those databases.

LocateItem If the IOC local database contains the requested item, it responds with an identification sufficient for establishing a route to that item. The LocateItem may be broadcast through the IOC network.

GetValue The value of the specified item is obtained.

PutValue A value is transmitted for the specified item.

StartStream A stream of data is transmitted through the indicated route. The IOC system continues to transmit data values until a StopStream command is received.

StopStream The indicated data stream is terminated.

SetEvent An event is established as a monitor of some database entry. Appropriate changes to that entry are to result in the indicated event being transmitted.

ClearEvent A previously set event is terminated.

5.5.2 Time Handling

Time synchronization is commonly required among control subsystems. These commands are to permit appropriate time synchronizations. Times will be provided in IRIG-B format as UTC.

TBD

5.5.3 Remote DB Access

Some subsystems may need to obtain information from other systems within the Gemini control system. This access is accomplished using the same communication and database access commands as given above.

5.6 COMMAND IMPLEMENTATION

Within the OCS, all commands are represented as ASCII strings. Fields within the command are labelled and nesting is permitted using braces ({ , }). For example, a typical command might be:

```
    Ident={ID=314159 Source=Operator_Console
Target=BROADCAST} Instruction=ShutDown
```

Details of the implementation are the province of the OCS work package developers, who are responsible for providing procedures for constructing/deconstructing/routing command strings. The mappings of specific IOC commands between ASCII strings and EPICS channel access calls are determined by the individual IOC subsystem developers. Work Package Descriptions include *baseline* command descriptions that need to be implemented for that specific work package.