GBT MEMO 88

GBT M&C Requirements Analysis Model

October 5, 1992

Mark H. Clark

1 Introduction

The goal of the Monitor and Control system is to provide a computer system for efficient monitoring, control, and data acquisition for the GBT. The GBT M&C group will use the Object Modeling Technique (OMT)[1] as its software development method. OMT consists of four major steps or iterations: analysis, system design, object design, and implementation. This memorandum describes the results of the analysis iteration. Figure 1 shows an overview of the analysis process. We have used a wide variety of informal sources to assemble a problem statement. The goal of such an analysis is a rigorous restatement of the requirements or problem statement in a form suitable for iteration into a design and implementation. It is – as much as possible – a description of **what** the system will do, not **how** it will be implemented and, therefore, needs to be reviewed and criticized by the end user. An analysis should answer three questions via three models: 1) What are the significant objects in the system, 2) How do they interact, and 3) What do they do? These three models are:

1. Object Model - this model takes the real-world objects and abstracts them into class objects and also their associations or relationships to each other. The class objects include both the data and the methods or operations that manipulate the data. This model includes object diagrams, a data dictionary and descriptions of the object diagrams. Classes are represented by boxes that may contain up to three sections: name, attributes, and methods.

2. Dynamic Model - this model shows the state transitions that the objects proceed through as external or internal events are applied to the objects. Using the Dynamic Model, one should be able to trace through

[1] **Object-Oriented Modeling and Design** by Rumbaugh *et al*, Prentice Hall, 1991

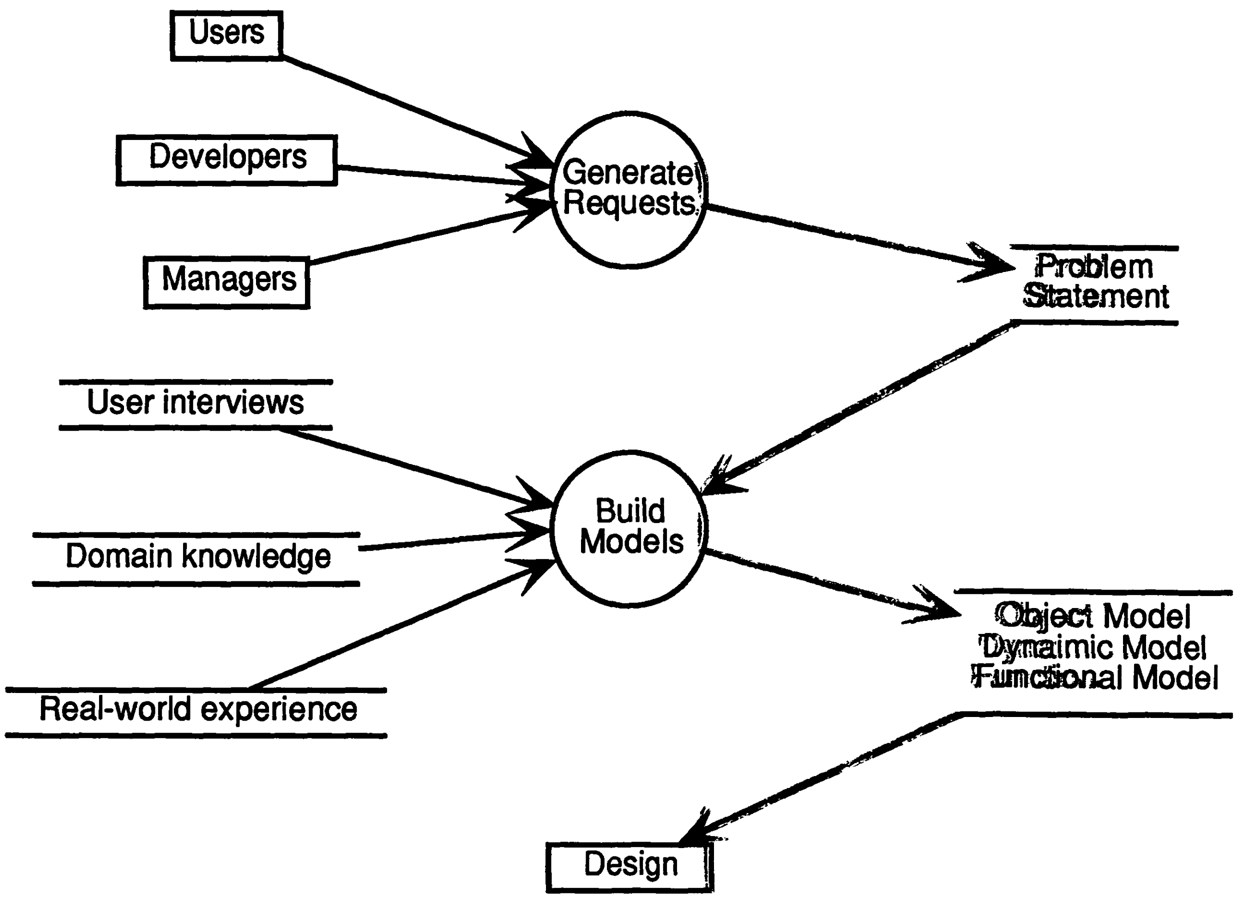

Figure 1: Requirements Analysis Process

possible scenarios in a working system. This model includes state diagrams and their descriptions.

3. Functional Model - this model shows the flow of data to and from both objects and data stores and also the processes or operations needed to drive the data through the system. This model includes data flow diagrams and their descriptions.

The goal of the models is to provide a clear understanding of what the M&C system will provide.

This document contains a number of specifications which one could argue belong more appropriately in a design rather than a requirements document. Since the OMT is almost completely seamless between the traditional stages of requirements, design, and implementation; it becomes somewhat arbitrary where one wishes to end analysis and start design. Moreover, the existence of the following design requirements necessitate the importance of being addressed early in the development cycle. The design requirements are:

1. The specification of what happens during an observation and when the observation begins may be made separately. This will allow for a more straightforward design for interactive use of the telescope.

2. The system is designed as modular as possible in both hardware and software.

3. The design should, as far as practical, attempt to implement a virtual telescope, i.e., a general framework that could, in principal, control any radio telescope. This implies that knowledge of astronomy and knowledge of the telescope be independent domains. This will allow specification systems and specific telescope control systems to be independent pieces of software. Given a stable design for the telescope, any future astronomical definitions, resulting from new scientific discoveries may be appended to the specifier systems without affecting the core control system. For example, with the design proposed here, the use of the cosmic microwave background reference frame for Doppler adjustments may be added to the specifier system or even driven by an observer's program without one line of code being changed in the telescope control system.

2 Object Model

2.1 Data Dictionary

- accesser - provides an index to all monitoring information in the telescope.
- actuator - a directional movement by a physical mechanism on a mirror either by a linear or rotational movement.
- administrator - a person responsible for controlling and monitoring telescope use, including up-time, allotment of time slots or "flex" observing time, maintenance scheduling, and repair assignments.
- antenna - an ordered set of all the mirrors, a subset of which are active at any one time. The antenna commands the active mirrors' movements in a coordinated manner.
- arm - supports the prime focus frontend and must be extended with the turret in a specific position in order to use the primary frontend and must be retracted in order to use or change any secondary frontend on the turret.
- backend - a device that is used to digitize the IF or to accept digitized IF, i.e., total power counts (TP), to produce backend data. Backends have high speed control of the receiver through synch signals. To allow more than one backend to operate concurrently, some backends must have the capability of being either a master or slave to synch signals.
- backend data - formatted data from a backend capturing all of the backend's information for describing the observation.
- boresight - a fiducial beam path from feed phase center to source with mirrors aligned for maximum gain.
- clock - provides UTC to managers, sequencers, and monitors.
- collator - concatenates the data associated parameters, data headers, specifier descriptions, and backend data into observational data in units corresponding to observations.

- command set - those attributes of a device whose values determine the observing parameters.

- compressor - equipment used to provide He for the receiver refrigeration units.

- console - a displayer for virtual control panels for telescope subsystems.

- continuum receiver - a backend used for measuring total power by accepting digital TP counts from frontends.

- converter - combination of mixers and LOs with optional routers for selecting LOs. Converters are used to shift and/or flip frequency ranges to obtain proper IF ranges for transport or processing.

- coordinator - a manager for coordinating other managers. It is the central synchronizer for all devices involved in an observation.

- data associated parameters - time-dependent measurements, as collected by the registries, corresponding to integration periods or observations.

- destination - any program, requiring on-line access to observational data including the storage, archiver, data monitor, data analysis, and observer applications.

- device - a well-defined physical sub-system of the telescope having a digital interface. A telescope may be thought of as an aggregate of devices requiring synchronization for accomplishing observations.

- dish - primary mirror for collecting RF signals. It has two movements: elevation and azimuth. Elevation is defined from 5 to 95 degrees and azimuth is defined from -270 to 270 degrees. The local reference frame is the horizon system and the sidereal reference frame is a TBD absolute equatorial system.

- displayer - any windowing program which provides a user frontend for the telescope including user-interfaces, near-realtime monitoring programs, analysis systems, or virtual control panels.

- engineer - responsible for the development, repair, and maintenance of telescope devices.

- executor - provides a transition between a mirror's asynchronous commands and the synchronous commands required by the servo system. Position and movement are specified in a local reference frame. Movements associated with secondary mirrors are specified relative to the boresight. Movement associated with the dish are in the local reference frame.

- feedhorn - final "mirror" to affect the RF signal before passing into the receiver. A frontend may have multiple feedhorns and, therefore, multiple beams. This mirror may have one movement for secondary receivers: rotation. The primary receiver feedhorn has two translations and a rotation movement. The feedhorns are adjusted by moving the frontend box. The local reference frame is the boresight and the sidereal reference frame is a relative TBD equatorial system, i.e., offsets from the boresight's position on the sky.

- friend of the observer - provides telescope-specific and/or general observing expertise to optimize the user's observing efficiency.

- friend of the telescope - on-duty responsibility for the safety of the telescope and its personnel. Arbiter of all control access to the telescope.

- frontend - a device consisting of one or more ordered sets of feedhorns, receivers, routers, and converters plus associated electronics physically packaged as one unit for producing IF signals and/or TP counts from the received RF.

- frontend select - places a frontend in the boresight path, by manipulating the arm, turret, and elevation axis.

- holography - a backend for performing differential phase surface measurements using satellite signals.

- HVAC - heating, ventilating, and air conditioning units and associated sensing equipment for maintaining specified temperature for specific areas and devices on the telescope.

- IEEE488 - parallel digital interface to commercial equipment via the IEEE-488 protocol, a parallel bus protocol used on many commercial instruments.

- LO - an oscillator, produces one or more CW signals used by converters. Some LOs are capable of varying their frequency during an observation. The frequency may be varied to compensate for Doppler shift or for frequency switching.

- LO monitor - provides independent measures of the LO's frequency and level.

- LO registry - accepts frequencies as generated by the LO and stores them in a short-term buffer to provide a time-tagged history of frequency information.

- magistrate - provides a single control point for controlling the telescope for an observation. A single control point is needed for security, interfacing to the specifier, and to provide full monitoring and control arbitration capabilities for the operator. The magistrate contains the control primitives of the telescope.

- manager - coordinates a device or other managers for purposes of observations.

- mirror - any device that affects by physical movement the antenna's beam. Position and movement of mirrors may be specified in either a local reference frame or a TBD sidereal reference frame. Movements of all secondary mirrors in the local reference frame are specified relative to the boresight, i.e., a canonical beam path between the dish and frontend. Movement of the dish (primary mirror) in the local reference frame is defined relative to the local gravity reference frame and geographic north (horizon system).

- mixer - a device for changing a signal from one frequency range to another by mixing the input signal with an LO signal.

- monitor - provides remote access to and a running history of internal parameters of an object.

- observation - a contiguous period of data collection under constant specified experimental conditions.

- observational data - combination of specifier descriptions, command sets, data associated parameters, and backend data with associated FITS descriptions. End product of an observation.

- observer - the prime "customer" of the telescope, defines telescope observing and is the initial recipient of the observational data.

- panel - provides remote access to a device's software and digital interfaces, both control and monitor, for displayers.

- personality - contains and/or accepts mirror-specific characteristics in order to modify ideal movement commands.

- phase calibrator - marks the delay of the IF for VLBA observing by inserting "birdies" maser-synchronized into the receiver.

- phase monitor - measures the synchronization of the LO's standard by comparing the standard at the maser and at the LO.

- pointing - system which determines and corrects for the true pointing of the antenna's beam, by means other than those available to the servo system. These means include the laser ranging, pointing model, temperature sensors, weather station, and auto-collimation systems.

- position registry - accepts various measures of antenna pointing and stores them in a buffer for providing time-tagged position information, either directly or by interpolating.

- power meter - provides power measurements.

- prime focus box - HVAC for the prime focus frontend box.

- primitives - the total of all command sets for all devices on the telescope.

- receiver - initial electronics to handle the RF as collected by the feedhorn.

- receiver room - HVAC for area which houses all of the secondary fron-tends.

- refrigeration - operates and monitors cooling equipment power supplies for each receiver.

- registry - a buffer for storing time-tagged measures of dynamic aspects of an observation. The registry is needed so that real-time generated information can be accessed over an ethernet. The registry is able to provide its information in a variety of ways including interpolation, dumps, or timed intervals.

- role - a job or function carried out by telescope personnel. A specific person can be assigned more than one role, e.g., the operator may be responsible for aiding the observer as well as for telescope integrity, and so would be filling the roles of both friend of the observer and friend of the telescope.

- router - a device for controlling signal routes. Current plans include routers for the IF, LO, test, phase calibrators, and synch signals. A router may include patch panels and/or splitters as well as electronic switches.

- secondary mirror - any mirror other than the dish.

- sequencer - provides time-event driven operation and monitoring of devices.

- servo monitor - set of monitors having high speed access directly to PCD's az/el and feedarm hardware control systems.

- SIB - standard interface board, provides a serial digital interface to in-house equipment via a specific implementation of the VLBA monitor and control bus (MCB) protocol.

- specifier - defines all experimental conditions for running an observation and provides for iterating through a series of observations. It houses the observers' user-interface and is the primary repository for astronomical information and calculations.

- spectral processor - a high-speed spectrometer backend currently on the 140' and used extensively for pulsar work. This device is already built, but requires significant work to integrate it fully into the GBT.

- spectrometer - a backend used for spectroscopy. This device is yet to be defined and built.

- spectrum analyzer - used for displaying and capturing spectrum. Will be used on the telescope for remotely and locally monitoring IFs.

- subreflector - a mirror used between the dish and turret frontends or tertiary. The subreflector has five movements: three translations and two tilts. The local reference frame is the boresight and the sidereal reference frame is a TBD relative equatorial system, i.e., offsets from the boresight's position on the sky.

- synch ports - drivers and receptors for synch signals. The current list includes cal, sig/ref, next freq, repeat, adv sig/ref, blanking, and bad data. Because of the synch router, it is probably better to model this portion of the system as ports instead of signals, since any driver could be configured to drive any receptor; e.g., a sig/ref signal could be routed to drive a next freq port. Ports are located on receivers, backends, mirrors, and LOs.

- synch signals - binary signals written and read at synch ports.

- tertiary - a mirror on the turret used between the subreflector and frontends. The local reference frame is the boresight and the sidereal reference frame is a TBD relative equatorial system, i.e., offsets from the boresight's position on the sky.

- turret - a turntable used to place secondary receivers or tertiary mirrors into the boresight as reflected from the subreflector.

- vacuum pump - used to aid in the cooling of receivers.

- VLBA backend - a device normally used to act as both a specifier and a backend for VLBA antennas. Note: there may be other units which need to assume both these functions such as non-NRAO pulsar equipment.

- weather station - a registry associated with a set of instruments for providing weather status.

2.2 Object Model Diagrams

For clarity the object model is divided into six modules. As we progress through the OMT, these modules will be broken down into a number of lower level subsystems. The eventual goal is to develop a finite number of easily modifiable and maintainable objects. The modules are:

1. Control
2. Monitor
3. Electronics
4. Data
5. Antenna
6. Specifier

2.2.1 Control

Control, as shown in Figure 2 is initiated by telescope personnel in various roles (seen at the top of the diagram) and culminates with the various mechanical and electronic devices (seen in the bottom right corner of the diagram) that make up the telescope. A detailed list of devices is impossible due to continuing hardware design, but they may be arbitrarily defined as any instrument or set of instruments on the telescope which it is convenient to think of as a single functional unit. Examples include instruments as simple as a frequency counter and as complicated as the antenna itself. Personnel need access to devices for direct operation and for observing; both of which require intervening software which the other classes in the diagram represent. Remember, all the boxes represent classes or software modules, not pieces of hardware, though the class may be named for the piece of hardware they are associated.

Classification of human interaction is by function or role rather than by specific jobs since responsibilities assigned to a specific person is a managerial decision, not a design decision. Also, one person may at any time fill several roles. However, the design does specify access privileges for each role. The observer and friend of the observer have access to displayers interfacing to

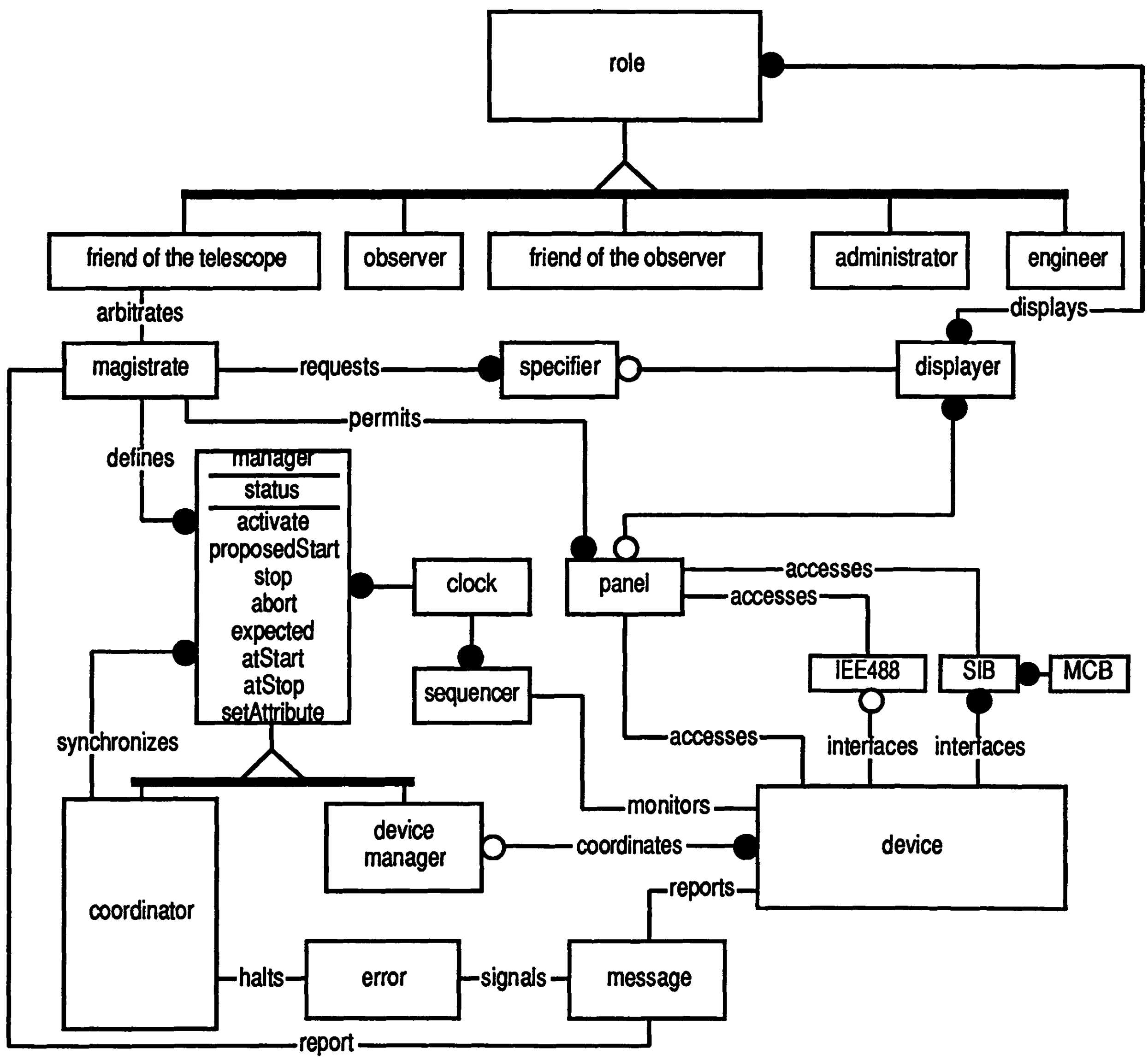

Figure 2: Control Object Diagram

the specifier and log, The engineer has access to displayers interfacing to the panel, log, and accesser. The administrator has access to displayers interfacing to the log. The friend of the telescope has access to all displayers plus the magistrate. Figure 3 shows these relationships without the intervening displayers.

All devices have complete digital interfaces for operation and observing. Note that some devices' fine grain control is important for observing, e.g. backends, while others are almost wholly for maintenance and operations, e.g., receiver room HVAC. Each device, in so far as possible, is treated as an independent unit, i.e., the system continues to work as well as possible when a device is removed or out of order, failure of one device does not cause system lock-up though appropriate alarms are triggered, safety is handled within each device, there are no software real-time dependencies between devices, and devices may be run and tested in a standalone mode.

A device's attributes define its virtual "control panel." Therefore, a device class declared in a stand-alone application or in a test program is sufficient to exercise all those settings and readings available for that device. In fact, the class panel is just such a program for providing remote access to a device for engineering purposes. Access via a panel is arbitrated by the friend of the telescope since the panel is available during observing. The second way a device may be accessed is from the device manager which uses a subset of the device class methods (its command set) to provide interaction for observational purposes. Finally, the sequencer periodically accesses a device's attributes to provide event-driven monitoring and calls a device's "sanity-checking" methods to confirm correct operation.

Before any sets of new attributes are activated, i.e., loaded into hardware or acted on by the system, the system confirms that they are legal and consistent both with each other and with the current set of active attributes. Since each device boots itself with a legal set of attributes and all attributes must have some legal value, the system should always be in a sane state. Undefined attributes cannot exist in this scheme since new states must always be generated from the previous state and no attribute may be "undefined." Combinations of attributes may cause some subset of attributes to be irrelevant for a given state, but those attributes still remain defined. No device or control class will send, accept, or act on any illegal or inconsistent commands or requests.

Observing requires coordinated control and synchronization of all active

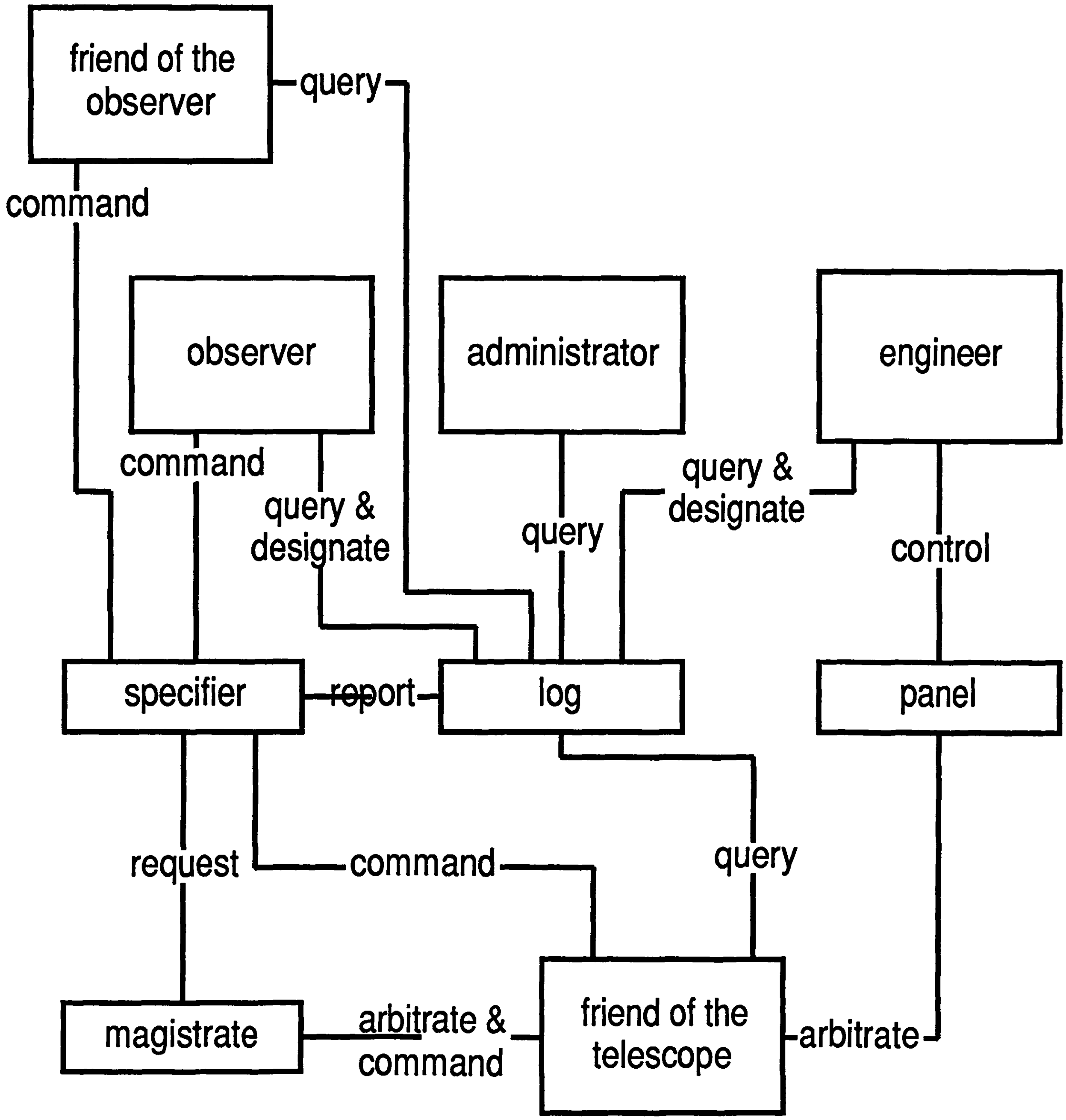

Figure 3: Role Object Diagram

devices. To satisfy the design requirement of modularity, synchronization is achieved by the clock (IRIG-B implementation) providing common time throughout the system rather than real-time software interrupts between devices. So, the set of active devices are synchronized with each other both via time for predefined software events, and by binary synch signals for fast hardware-arbitrated events. The coordinator synchronizes the sequential steps to accomplish an observation through each device's associated manager. Each manager is designed so that it is able to respond to the same synchronization commands. Thus, the coordinator is a special kind of manager which controls other managers. A manager also provides access to a device's command set. This *a priori* defining of synchronization and the fact that most devices' attributes are static during an observation results in control generally iterating between two mutually exclusive stages of definition and use, i.e., experimental conditions are not changed, except in predetermined ways, during observing. A device is configured so that its actions during an observation are unambiguously specified prior to the observation, and after the observation the specification is included as part of the observational data.

Mirrors and LOs pose special problems for specifying all attributes prior to the observation since they include dynamic actions. The actions of these devices depend on sidereal references, i.e., points on the sky and changes in Doppler compensation as a result of relative movements of the earth. These have traditionally been major concerns of control systems as reflected by the fact that they are among the first devices to be put under computer control. Consider three implications of the design requirements listed in the introduction on page 3, i.e., synchronizing devices using a clock and pre-observation setups, removal of astronomical aspects from the telescope internals, and acceptable interactive control for observations. This third implication is somewhat at odds to the first two when applied to mirrors and the LO because these parameters may change rapidly with time in the telescope reference frame. Interactive control is less constrained if one can separate the observation's start time from the other attributes in the command set, i.e., allow the observer or specifier to define all of the experimental conditions and then independently decide when to begin observing. Otherwise, dependencies between setup latencies for various devices exist. However, one cannot specify positions on the sky or velocities in a rest frame using non-astronomical parameters without including a time as part of the setup. We have compromised by permitting the mirrors and LOs to have only two frames of reference: one terrestrial and one sidereal. This allows us to specify start time independently of the command set and to limit the number of time dependent parameters in the telescope primitives.

The magistrate provides a single control point to the telescope for observing requests. Of course, observing requests are a subset of all commands that are implemented to allow operation of the telescope. The friend of the telescope, by having full command of the magistrate, is able to monitor and arbitrate all computer access to the telescope. The friend of the telescope may set up the system to automatically remove observation control from the observer by specifying trigger conditions, such as errors of a given severity. Computer security measures focus at the magistrate (and the gateway machine it resides on) and embrace standard techniques, e.g., passwords, logging access, and physical security of key machines, In addition, each module is responsible for maintaining a safe and sane state irregardless of its input. Any device's command set, needed for observing, is available at the magistrate. Therefore, the magistrate's attributes define the telescope primitives which define the atomic commands that any specifier or control system is built on.

The specifier houses observers' user-interfaces. The specifier is designed separately to permit the user-interface an unencumbered and varied growth path. This division is also critical for providing the possibility of defining a virtual telescope, i.e., general frameworks for porting control software or user-interfaces from telescope to telescope.

Though the system provides for extensive intervention by the friend of the telescope in all aspects of control, there is no inherent reason in this scheme that permissions cannot be directed to a specified state and left constant, thus providing full control to the observer.

As described below, an extensive monitoring system provides access to internal variables via various application programs for purposes of engineering, observation monitoring, logging, generation of data associated parameters, user-interface construction, and administration.

The message class provides a mechanism for any control program to send character strings to telescope personnel and to set error states. The various displayers may be specified as destinations. The programmer may define an addendum which consists of additional static text to explain the message and/or a set of graphical "snapshots" of monitoring information related to a specific message. Each message is assigned a level which allows the user to suspend the display of messages by level as well as by subsystem. The levels are: no suppression, censored, operational (default), verbose, and debug. Each message includes an error level which is passed immediately to the error class.

The error class provides the means for internal events to affect system behavior. In many cases the detection of an error condition by some subsystem will simply prevent it from signaling its readiness to pass into the next observing state. However, for those errors which require that the observing state of the entire system be changed, the error class provides a mechanism for any program to make an error level request. Error requests must be made through the message class. The levels and the resulting actions follow:

1. No error.
2. Warning
3. Warning - Save observation state description.
4. Error - Save observation state description. Abort the current scan and go to the idle state.
5. Error - Save observation state description. Stop all computing, processing and control and go to manual mode or as close as possible. Sound physical alarms if possible.

2.2.2 Monitor

Monitoring provides direct access to a device's internal parameters as shown in Figure 4. The problem is tapping information within devices throughout the system, represented by the device class in the top right corner of the diagram, and providing it in an understandable format via the various options for viewers or storing it for later reference, as seen at the left of the diagram. Note that this mechanism can also provide information for data associated parameters through registries as requested by the collator.

Of course, the goal is to monitor the hardware itself. It is the responsibility of the digital interface and device software to capture the device's parameters in variables, but then the parameters are accessible through the monitor in two fashions. Parameters may be captured as determined by the device's internal processes and specified *a priori*, or they may be captured at

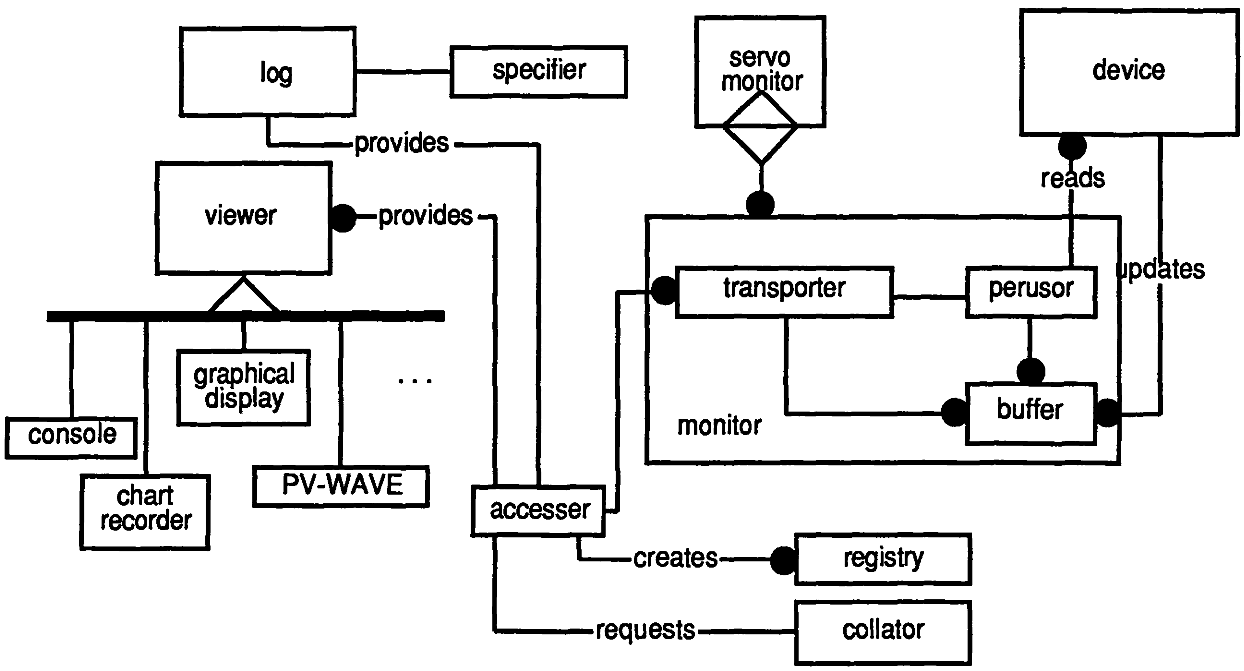

Figure 4: Monitor Object Diagram

specified intervals as specified dynamically by an external request. When parameter capture is determined by the device, the buffering of the information provides a short term history of the parameter, even when the parameter is not being accessed by a displayer or log. This capability will be used extensively with the error reporting facilities in the snapshot feature mentioned above to provide parameter status both before and after an error is triggered. The mechanism that allows dynamic access to parameters also allows access to any program variable, so essentially the engineer has a debugger running at all times and can "on the fly" select variables for monitoring. The accesser provides means for reading the various tables holding monitored and monitorable parameters. The accesser is an index of values available to any program wishing to use the monitors. Besides being accessible by any displayer, the accesser also provides information to the log for archival purposes. The log will be implemented using a commercial data base management system (DBMS). A DBMS permits more sophisticated searching of archived monitor information.

It is the working assumption of this analysis that the key to providing an understandable system for observers and telescope personnel alike is in the quality of monitoring (or feedback) provided. Computers permit automation of both tedious and high-speed activities, but sometimes, because of limitations of hardware and software, the control programs have limited feedback, often to a level below the manual interfaces they replaced. [2] Though computer control by M&C is necessary, the quality of monitoring will determine the system's success.

2.2.3 Electronics

A block diagram for electronics as generally envisioned early in the project is shown in Figure 5. The path of the received signal from an RF beam to backend is shown by the large arrows. The RF signal at the frontend may have added to it a noise calibration signal, a tone generated by the LO, or phase calibrator ("birdies") used for VLBA. The noise calibrator is generated in the receiver. A test router selects between the phase calibrator, tone, or neither. And the LO generating the tone is selected by the LO router which also selects the LO input to the frontend's converter and to the LO monitor.

[2] "The GBT Human-Computer Interface" by J. R. Fisher, talk given at **Operating the Green Bank Telescope** symposium in Green Bank, March 20, 1992.

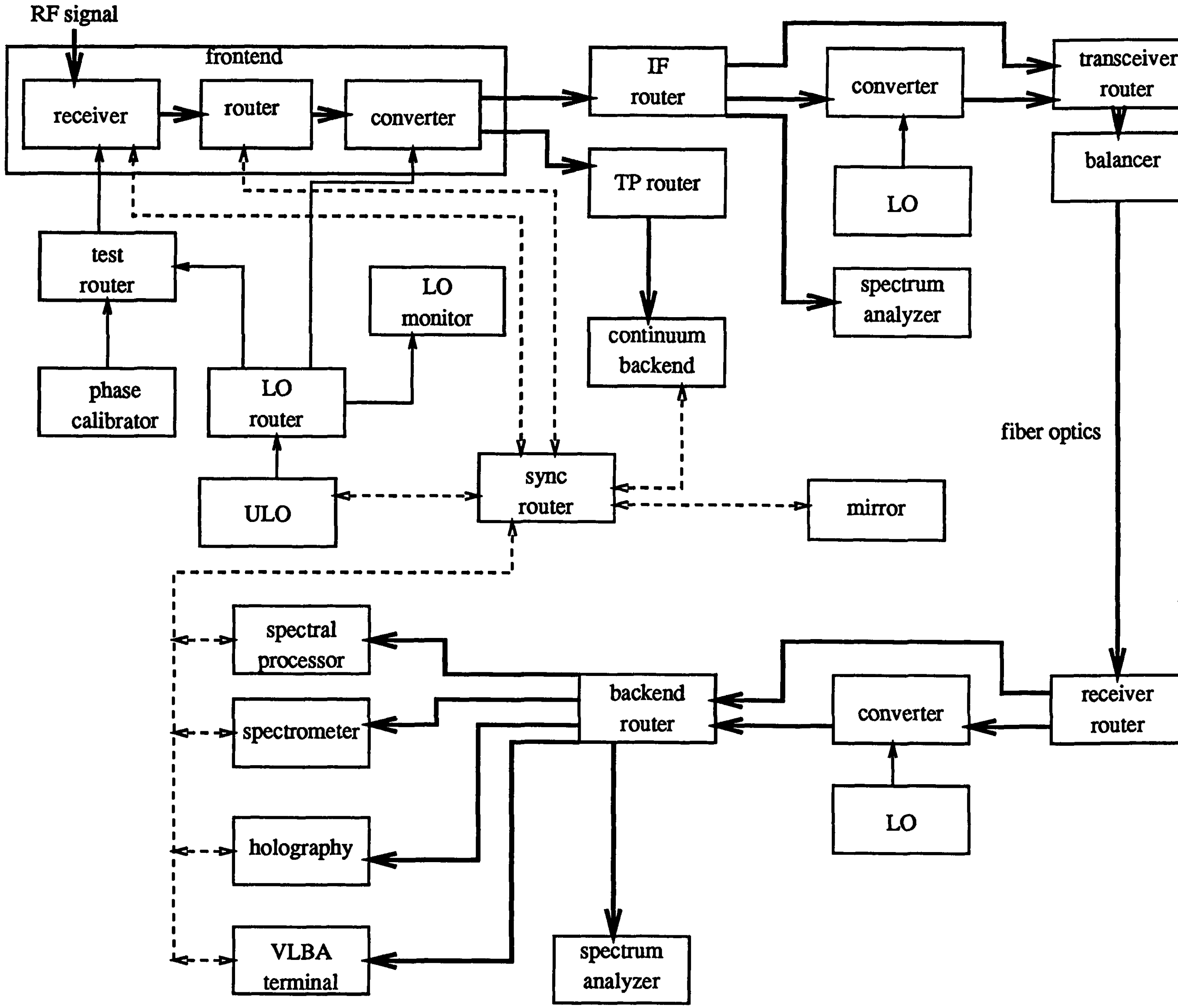

Figure 5: Electronics Block Diagram

Both the receiver and the LO accept synch signals from the synch router for real-time control. A frontend may have one or more receiver/converter pairs. The receiver determines polarization and feeds the signal to its associated converter where the signal is mixed with the LO's tone as selected by the LO router. The result of this mixing is passed to the total power (TP) router as a digitized signal and to the IF router as an analog signal. The TP router selects inputs to be passed to the continuum backend. The IF router selects whether or not the signals pass through another converter before being selected by the transceiver router for balancing. The balancing is necessary to achieve the appropriate level for transfer over optical fiber to the operations control center. The receiver router selects the signals to be converted before being fed to the backend router. The backend router distributes the final IF to the various ports on the backends. Spectrum analyzer may be accessible from IF routers at both the telescope and the operations control center. The backends pass the processed and packaged data, i.e. backend data, on to the collator.

The block diagram is useful, but it needs to be abstracted to classes as in Figure 6 for analysis and design. The details of the hardware design are yet to be specified and much is tentative, but the basic classes and their associations should not change.

The dominant relation between classes is the passing of signals between objects. Signals are RF (input and output), IF (input and output), LO, test (tones, wideband noise, or phase markers), TP (signal representing total power of an IF), and binary signals between sync ports (sig/ref, cal, advanced sig/ref, blanking, next freq, bad data, and repeat). The control and tracking of signal paths is provided by routers which are implemented by electronic switches, splitters, and/or patch panels. Notice that except for converters' internal LOs, all other signal paths are mediated by routers.

The implementation of the various routers may vary widely, especially for those in the operations control center since they need to distribute as well as select signals unlike the routers on the telescope. The same IF signal could be the input to a number of ports on a backend or backends. The actual implementation is not defined, but it is likely to include patch panels as well as electronic switches and splitters. The manual configuration of a patch panel must be somehow fed into the M&C system either by manual entry into tables or by having the patch panel digitally detect the connections being made. The relation of the alidade control room – if any – to the operations

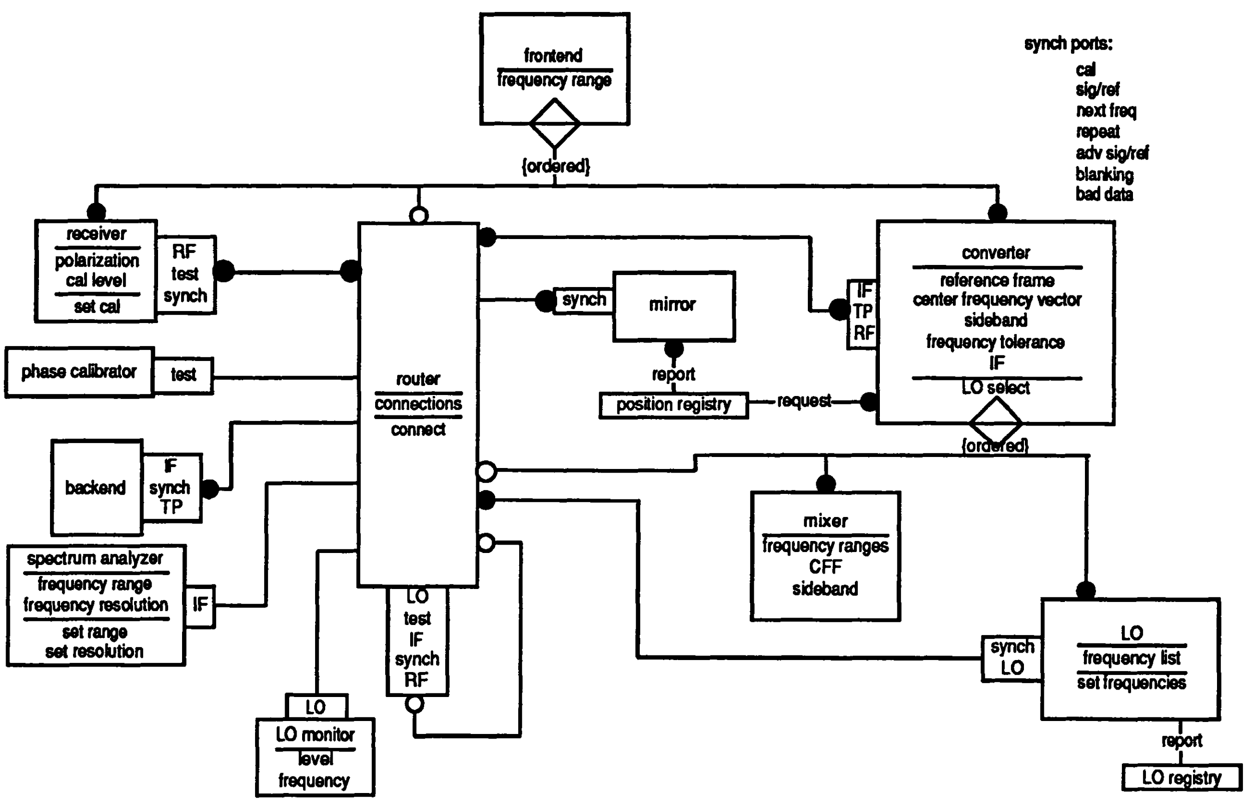

Figure 6: Electronics Object Diagram

control center has yet to be defined.

The existence of routers significantly increases the responsibilities of M&C because the observer can modify from observation to observation the interdependences between devices handling signals. For example, previously a specific connection was manually set up between a frontend and a backend and the implications of that connection, such as center IF frequency, were then manually entered into the control system (normally just once for an entire observation run); now, there is no reason the system cannot compute the parameters of a signal as seen at any device. To compute these parameters the M&C system must keep track of the signal's path through all devices. This can become especially convoluted when one considers that an IF generated at the receiver may pass through one to three converters before entering almost any number of ports on any subset of the backends. The importance of clear feedback to the user cannot be overestimated in such a system.

A few comments on the object diagram representation. Note that IF, test, and LO routers interact directly with each other. Since the only essential difference between the various routers is the kind of signal they control, they all can be represented by one class. Associations with a router are qualified by the type of signal being channeled. For example, a backend can have associations with three types of routers: the TP router for the continuum backend, the backend router for the other backends to access the IF signal, and the sync router for all backends using the sync signals. Therefore, the one-to-many relationship of a backend to router is qualified by the type of signal: sync, TP, or IF.

A mixer always has one or more LOs associated with it. Each mixer has a frequency range and sideband associated with it. Likewise, the frequencies of some LOs are not constant, but can be set either discretely or continuously.

An LO may have dynamic capabilities for varying frequency during an observation. The LO accepts frequency vectors in two reference frames: the telescope and the solar system barycenter. Conversions from other astronomical velocity frames of reference take place in the specifier.

The LO for the frontends' converters will be implemented using two Hewlett-Packard synthesizers. These synthesizers may be operated in several ways. For example, a list of frequencies may be preloaded and triggered by an external signal, and the blanking time between frequency transitions is different between the first and following iterations through the list. The device attributes are defined to allow several modes of operation for selecting

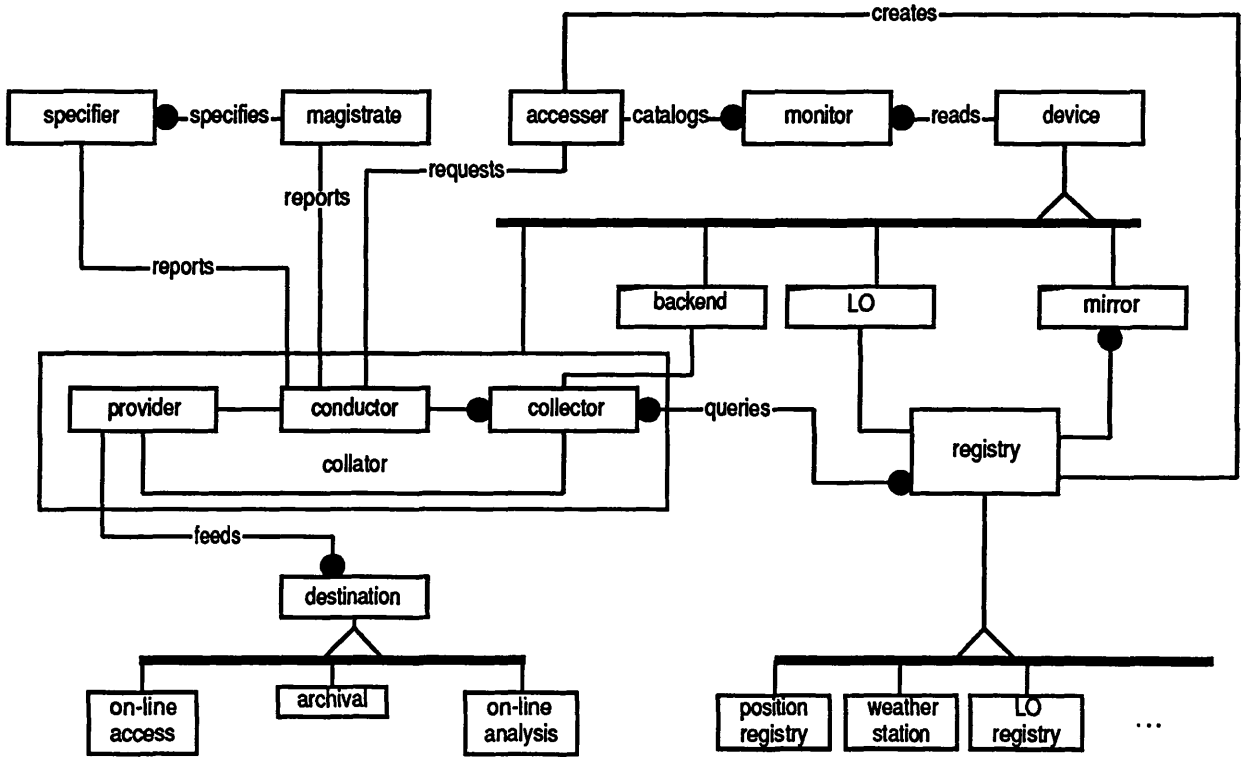

Figure 7: Data Object Diagram

tradeoffs between blanking time, total observation time, integration time, and setup time.

2.2.4 Data

Figure 7 displays the four sources of observational data: the specifier for observational descriptions, the magistrate for resulting command sets, registries for data associated parameters, and backends for backend data which contains the actual astronomical data. The data from these sources are combined and packaged (including FITS headers) in the collator. The collator is able to concurrently handle multiple backends and feed multiple destinations.

An implication of placing the astronomical specifications outside the core system is that data associated parameters are specified in telescope parameters not the observer's specifications, e.g., position is given by UT and horizon coordinates for each mirror. In general, data associated parameters are given as close as practical to the base device attributes. However, the experimental conditions of the observation as defined by the command sets for all active devices are stored in the observational data.

Backends define integration periods while the position registry tracks antenna positions. Data associated parameters need to contain positions representing integration times (either beginning, center, and/or end of the integration period as determined by the backend). These integration times are part of the backend data which are read by the collator which, in turn, queries the position registry for mirror positions to add to the observational data.

Destinations for the final packaged data include storage mechanisms, observer's on-line data analysis, and a resident analysis program which acts as a displayer for providing data monitoring.

2.2.5 Antenna

As seen in Figure 8, the antenna is built around the mirror class seen in the center of the diagram. Depending on the configuration, two to four mirrors may be active. For prime focus observing, the dish and the feedhorns define the beampath; and for secondary focus observing, the dish, subreflector, perhaps the tertiary, and the feedhorns determine the beampath. Each mirror has a virtual coordinate system that defines its movement. These coordinate systems are defined by the canonical beampath or boresight of the antenna, not the physical mechanism used for moving the mirror (axes). For example, the subreflector could have one coordinate system for translations in x, y, and z, and one coordinate system for tilt in x and z; though all of these movements are accomplished by commanding six axes or actuators. The dish's coordinate system is relative to the local gravity reference frame (horizon), and the remaining mirrors are relative to the antenna boresight. Conversions from other astronomical coordinate systems take place in the specifier. Each coordinate system is controlled asynchronously by a position vector. The antenna is a device for coordinating the movements of the beam via the active mirrors. Because movement of all mirrors is set up before the observation, mirror adjustments such as focus or position angle are based on commanded position rather than indicated position.

The position registry holds a running history of the commanded, indi-

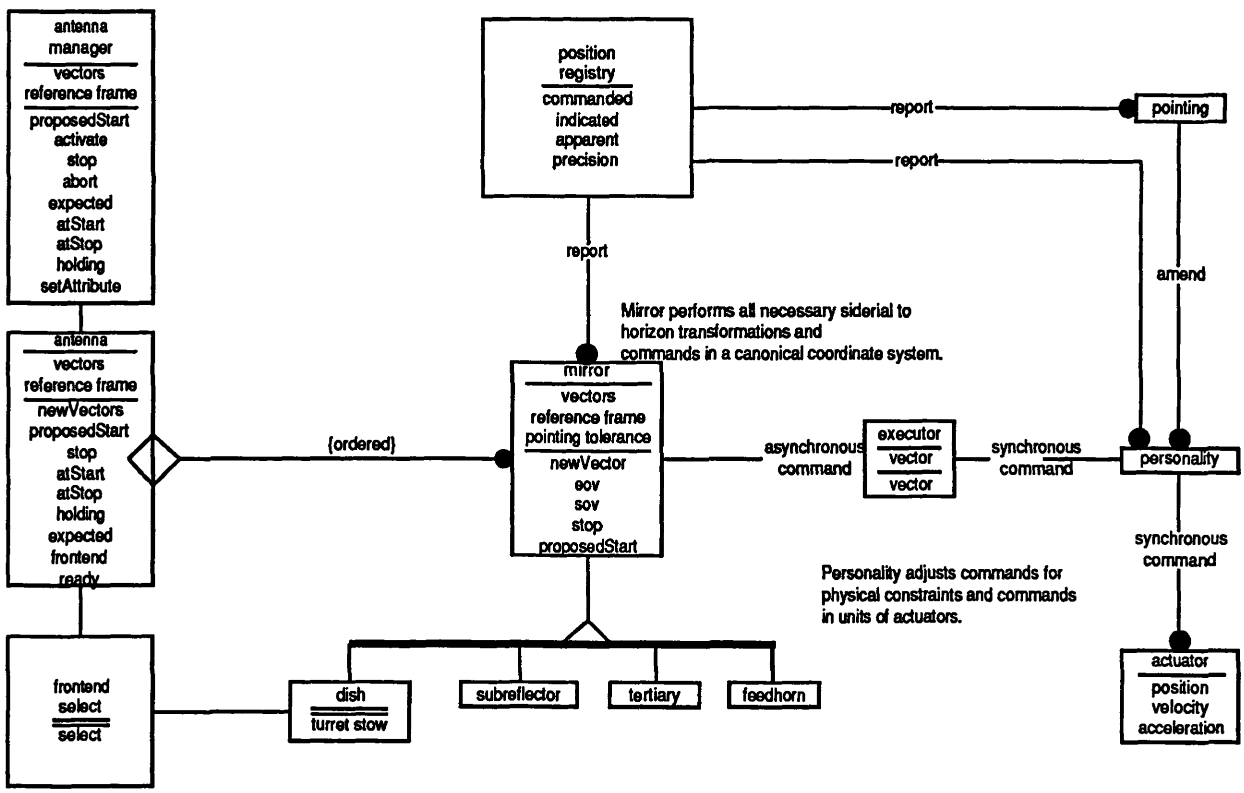

Figure 8: Antenna Object Diagram

cated, apparent, and precision positions of the mirrors for interrogation by any object needing the information, e.g., the pointing or collator classes.

The executor accepts position vectors asynchronously and interpolates them to produce synchronous streams of position commands. The commands are, in turn, modified by the personality module to compensate for realities of the antenna and the atmosphere as determined by pointing.

Pointing combines the information from the model, weather station, autocollimator, laser ranging, pointing model, and position registry to provide corrections to the interpolated position vectors of the mirrors. The interface between the M&C system and precision pointing system is defined by the difference of an ideal telescope and the physical telescope as currently measured. The M&C system assumes it is dealing with a perfectly rigid structure in a vacuum so that its position is accurately reflected in the actuator position readings. Precision pointing provides corrections to all mirrors (dish, subreflector, tertiary, and feedhorns) near the currently indicated position, so that the executor's synchronous commands (position, velocity, and acceleration every 100 msecs.) may be modified to resolve the measured errors. The decision as to which mirror to use to correct the pointing is the responsibility of the personality module under guidelines from the antenna. Considerations of receiver gain as well as bandwidth response of the various mirrors must be taken into account.

2.2.6 Specifier

The specifier translates observing requests into telescope primitives and provides several layers of abstractions for interfacing user-interfaces. The specifier's functionality is determined not only by the capabilities of the telescope, but by the definition and design of its control mechanisms. The specifier, to a great extent, follows a layered model analogous to the OSI communication protocol or the MACH operating system kernel, i.e., telescope, observing, and user-interface layers. Many telescope control systems follow this scheme, at least informally, e.g., JCMT's ICL/SMS, IRAM's OBSINP/OBSE, and POP's primitives/procedures. The GBT's design differs somewhat by attempting to fully separate the telescope from the observing layers.

Remote observing has implications for the implementation of this layering scheme. The most direct way to provide for remote observing is to use one of the interfaces between the specifier's layers as the boundary between the telescope and the observer's location, whether local or remote. If the telescope primitives are selected as this boundary, then it becomes difficult to provide monitoring of observing procedures and sequences by either the friend of the observer or the friend of the telescope since these processes would likely be remote. Therefore, the observing layer must be accessible at the telescope and must provide "shadowing" capabilities for possible monitoring of the observer's actions and observing program.

Most of the observing primitives are determined directly by the nature of specific devices on the telescope. Examples include the specific capabilities of a frontend or backend. These telescope primitives are not profitably specified by more generic abstractions without obscuring essential experimental characteristics and flexibility of the specific devices. However, there are three ways the telescope primitives are enhanced to provide basic functionality for observing.

First, astronomers use a number of frames of reference for describing position and relative motion depending on the domain of their investigation. It is obvious that the observing layer needs to accept antenna positions and movement in all major astronomical coordinate system; LO control in all rest frames and Doppler definitions, and time in UT or LST. In general, the observing primitives need to track astronomical definitions that affect observation specifications.

Second, means of specifying series of related antenna movements beyond what can be described by a single position vector need to be provided. Specifically, series of antenna movements may be described parametrically such as pointing scans, point map, and scanned map.

Finally, the telescope primitives only specify the system state for one observation; series of related observations or entire observation runs must be handled by the observing and/or user-interface level. How observations are grouped together by commands or header information is a major characteristic of different control systems.

Because the differences of telescope and observing primitives are limited to these well-defined areas, the specifier does not fully satisfy the structure of a true layered model because there are too few areas where one layer completely encapsulates the layers below it. However, this type model is useful in that it allows a selection of interfaces at varying levels of abstraction for the construction of user-interfaces. Figure 9 shows a possible scheme. Staffing and time precludes the porting of many of the widely-used interfaces,

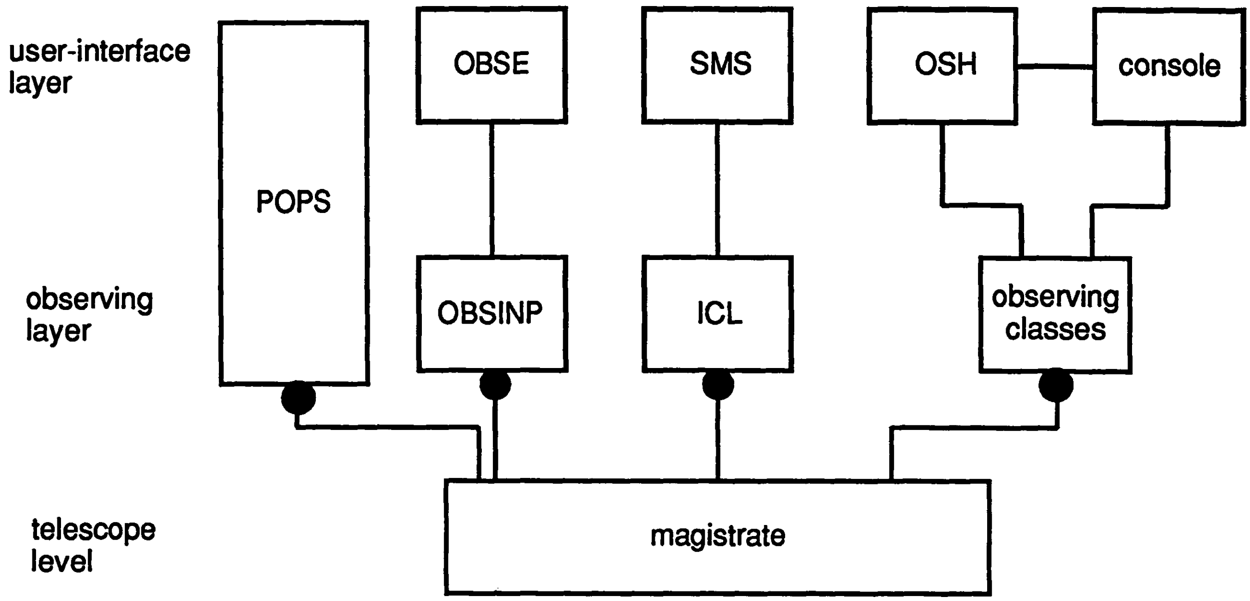

Figure 9: Specifier Object Diagram

such as OBSE, OBSINP, POPS, SMS, and ICL, but we do hope to eventually have a selection of programs for controlling the telescope. These three aspects of the user-interface may vary between user-interfaces while the remainder controls are determined by the specific devices to be controlled.

It is important that the interface to the magistrate provides sufficient feedback to allow automatic scheduling algorithms, if implemented, to make reasonable choices according to their specified criteria.

Besides various specification systems that we may have an opportunity to port to the GBT, the observer's requirements [3] provide specific needs for the GBT user-interface. OSH (observing shell), as shown in diagram, is designed to satisfy these requirements. OSH is an enhancement of GNU's BASH specifically for observing control. Characteristics of BASH include:

- control structures, e.g., loops, conditionals, and case statements
- job control

[3] **Observer Monitor and Control Requirements** by J. R. Fisher and F. J. Lockman.

- history expansion
- redirection
- interactive configuration parameters
- interactive editing of commands from the command history.

BASH itself as a command language satisfies many of the observing requirements:

- control programming language interface
- identical interactive and batch files commands
- smooth transition from interactive to programmed observing
- procedures or macros
- aliasing
- end-of-line or semicolon statement delimiters
- default and customized setups.

A number of enhancements should be straightforward to implement:

- logging facilities (beyond a straight command history)
- hardware control panels (see below)
- division of control parameters (keywords) into subsystems
- labeled setup files
- observing tables
- help facilities
- keywords
- key constants and unit sensitive parameters

- database of predefined procedures and setups
- interruptible observing tables
- command and setup file archival and indexing
- telescope simulator
- near real-time monitoring access
- shadowing.
- "min-match" variable and keyword recognition

Other enhancements are less straightforward:

- restarting an observing program at any statement
- full numerical library and functionality directly available in the command language
- character case (upper/lower) insensitivity.

The major drawback for any command language using parameters or keywords to specify device settings is that the number of parameters grows to unmanageable numbers very quickly. Modifying one or two parameters between observations is simple and easy using a textual format, but defining an entire setup with lists of assignment statements can be tedious and devoid of useful feedback. Therefore, as part of OSH, there is a GUI setup program, console, which allows the user to set parameters for subsystems via a virtual control panel with intuitive controls and monitoring in lieu of text. This program is used for interactive control of the telescope's subsystem; or for reading, writing, and modifying subsystem setup files. The implementation of the observing layer on which OSH and console depend borrows heavily from astronomical classes as generated by AIPS++.

If the AIPS++ settles on some standard user-interface that could lend itself to telescope control then it will be given serious consideration as an additional, or even alternate, user-interface for the GBT.

3 Dynamic Model

For purposes of analysis, the telescope dynamics have been separated into two types: system dynamics for stepping through observations and device dynamics necessary during an observation. The system dynamics are handled by the various managers, each of which synchronizes one or more devices, and by the "super" manager, or coordinator, which synchronizes all of the active managers for any given observation. As stated earlier there are two devices requiring dynamic control during an observation: the antenna and LOs. There exist other dynamic actions during an observation, but they are either internal to a device, e.g., backends, or are arbitrated by hardware, e.g., synch signals.

3.1 Observation Dynamics

Figures 2 and 10 represent the system coordination mechanisms for synchronizing devices for observations. The methods listed under the manager in Figure 2 allow the coordinator and, in turn, the magistrate to step the hierarchy of managers through the various observation states shown in Figure 10.

The specifier defines an observation by setting the various attributes of the magistrate that are immediately passed to the appropriate device managers which buffer these attributes. Assuming all managers are idling, the magistrate sends activate to the coordinator which sends activate to all of the active managers. For most managers and their associated devices, this results in the appropriate values being loaded into hardware. All devices, as soon as possible after receiving activate, compute the earliest start time possible, a cancel time when that start time can no longer be achieved, and the resulting delay necessary to start after the cancel time has been reached. The antenna's mirrors require physical movement and therefore have an inherent latency. When all device managers are pending, then the coordinator is pending. From the start, cancel, and delay times which make up the expected event, the coordinator is able to select a proposed start time. At this point, all devices are prepared or preparing to observe except for a start time. The proposedStart event launches all devices which terminates at the proposedStart time and initiates observation. Completion of the operate state is accomplished either by internally generated events or by an external

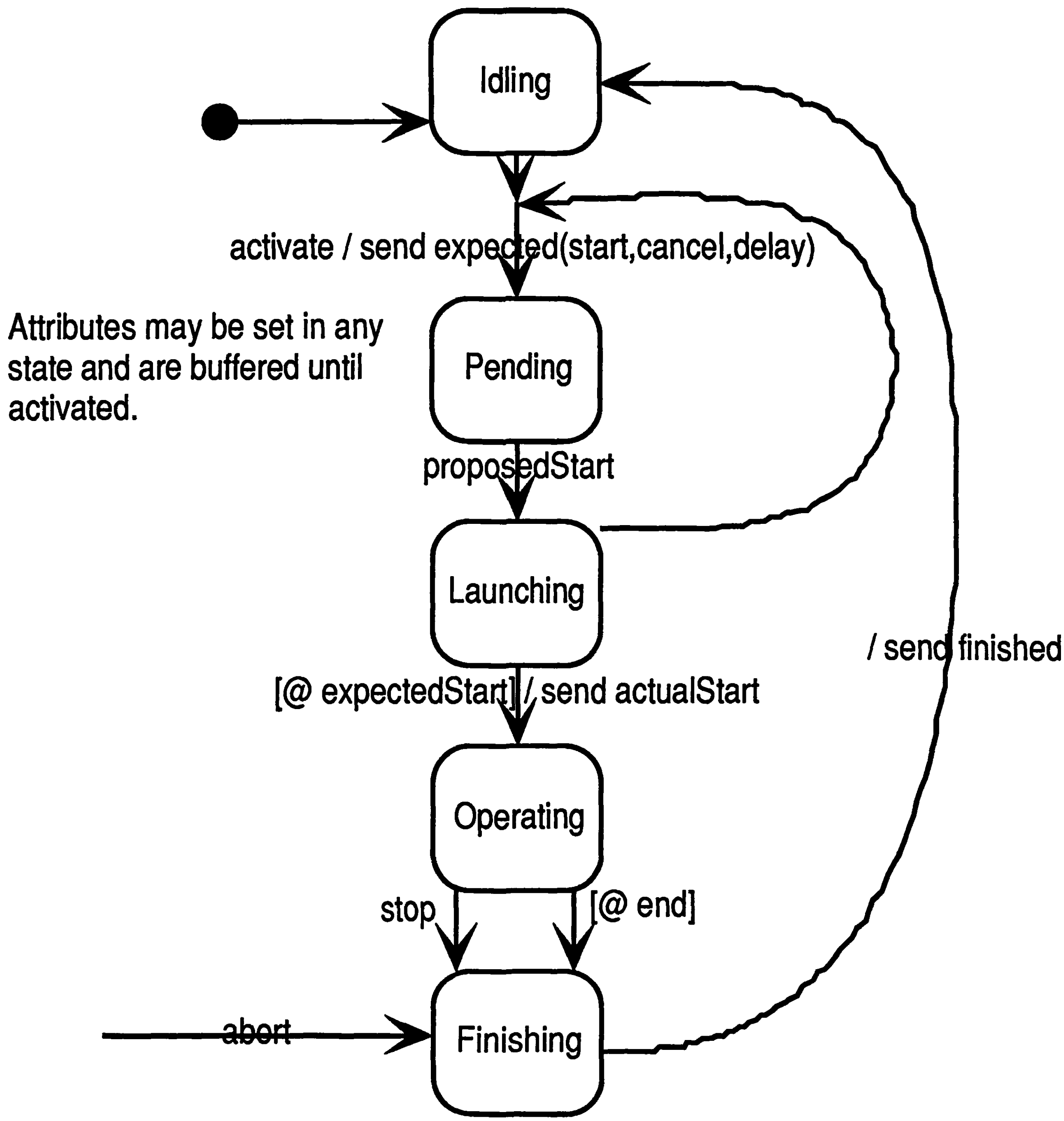

Figure 10: Manager Dynamic Diagram

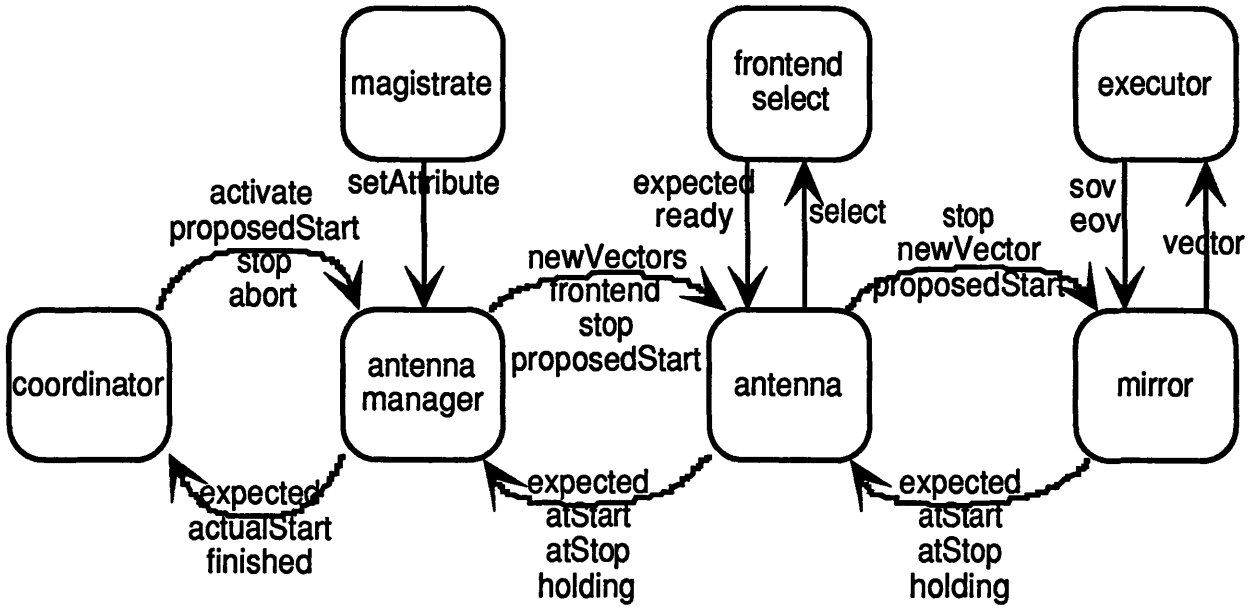

Figure 11: Antenna Flow Diagram

generated stop event.

3.2 Antenna Dynamics

The antenna consists of an ordered set of mirrors guiding the beam through the structure. Figure 11 lists the various events being passed between the classes shown earlier in the *antenna object diagram.* The relations between the antenna manager, coordinator, and magistrate were described in the previous section. Figures 12 and 13 outline the temporal aspects of these relations.

The general idea is that the antenna manager (as for all device managers) exists to provide a common coordination interface for all instrumentation making up the telescope. The antenna coordinates all the mirrors of which it consists. The mirrors provide transition from sidereal to horizon coordinate systems for tracks as specified by position vectors and computes starting, slewing, and stopping tracks. The executor generates real-time synchronous commands as interpolated from the asynchronously received position vectors.

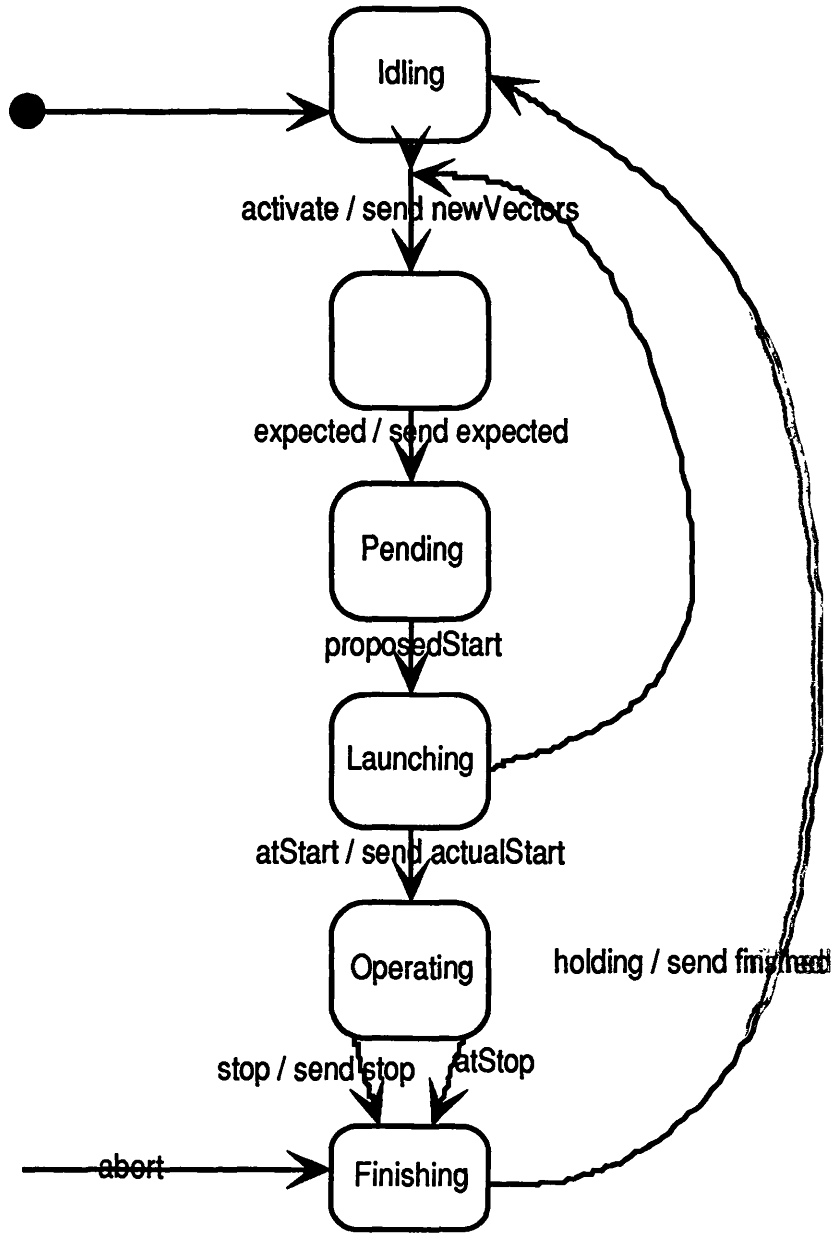

Figure 12: Antenna Manager Dynamic Diagram

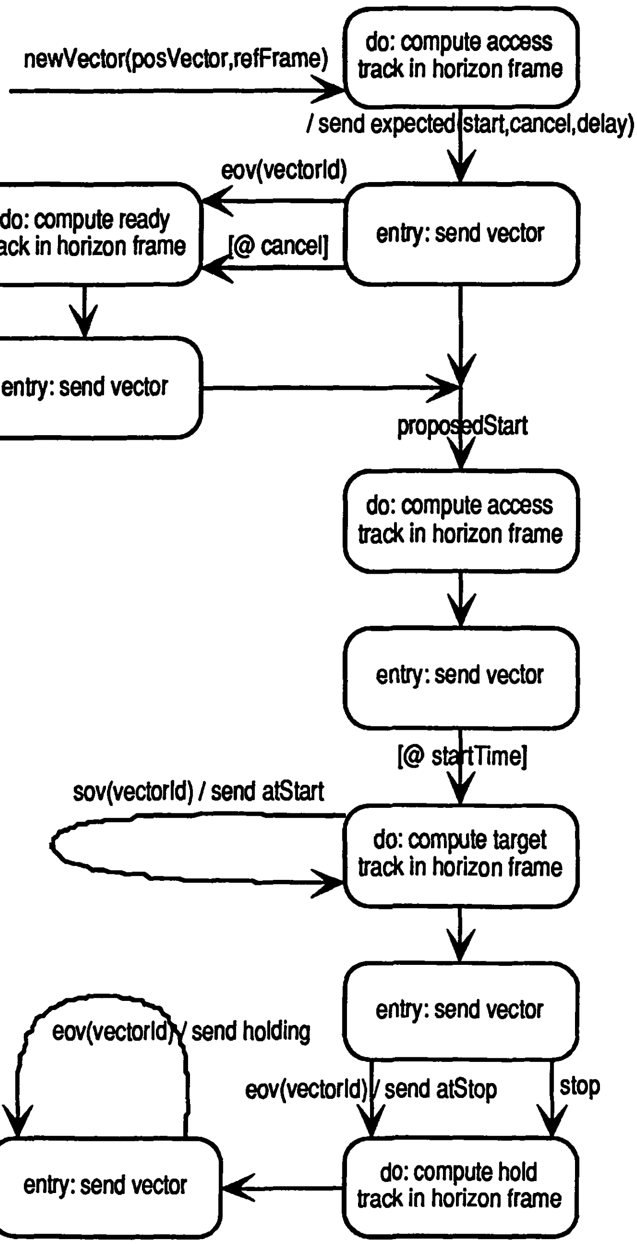

Figure 13: Mirror Dynamic Diagram

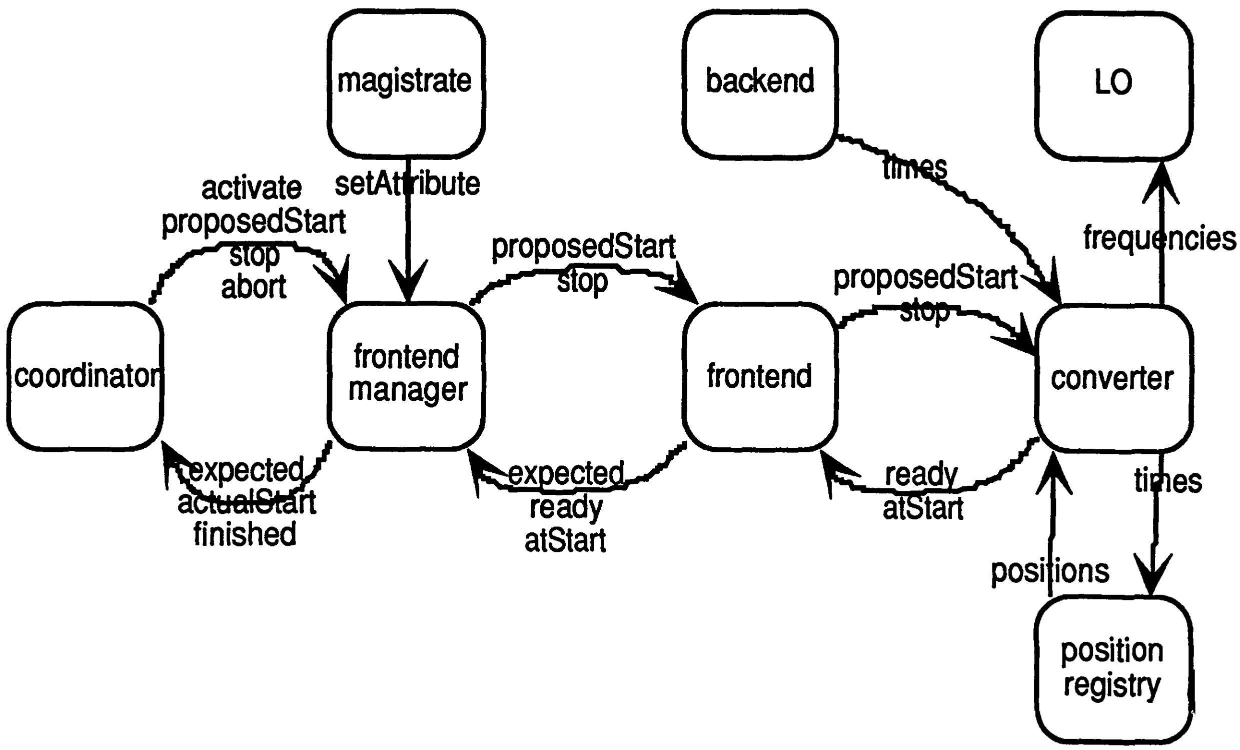

Figure 14: Frontend Flow Diagram

3.3 Frontend Dynamics

Figure 14 shows the flow of events affecting the dynamics of the frontend's converter and LO. Since Doppler shift attributes are specified only in a barycentric reference frame within the core system, all that needs to be calculated during launch, i.e., after observation start time is known, is the movement of the earth relative to this reference frame and the times (usually at integration time boundaries) the LO may be shifted. If the backend is set up to drive the LO frequency changes for the Doppler shift via synch signals, then at launch it provides possible frequency change times to the frontend converter. These times along with the associated commanded positions and the converter's attributes allow the converter to compute the specific LO frequencies and the times that a frequency change is required to satisfy the frequency tolerance. All of these information transfers and computations must occur during launch and will therefore incur a short latency for the frontend.

3.4 Collator Dynamics

Figure 15 shows the flow of events that take place during the collection and collation of data during an observation. The conductor manager provides the interface for coordination of all of the collectors and the provider that make up data collation. The conductor, as seen in Figure 16, creates and deletes all of the data buffers and is responsible for notifying each of the collectors where its associated data buffers reside. In addition, the conductor notifies the provider of where the collated data will reside and of its destinations. The collectors receive data from the backends, registries and magistrate, notifying the provider when a data buffer is full. The provider, in turn, delivers the data to the specified destinations.

4 Functional Model

For a control system such as the one described here, it turns out that most computations are either trivial or well-known such as in Figure 17. Of more interest are the attributes, or parameter derivations. For example, the prime focus frontend can be configured for four different frequency ranges by changing settings (and feedhorns). Naturally, we want to define a method, setRange(range), which sets all the attributes needed to achieve the requested range. However, we also want to provide the user with methods for independent access to the specific attributes that were set by the "macro" method setRange such as filters, switches, and attenuators. Contradictory settings, circular inter-dependencies, and more complicated hierarchies are handled by a general mechanism called a parameter class [4] which allows the relations and inter-dependencies to be implemented in a straightforward consistent manner which can be made explicit to the user.

[4] **Parameter control in C++** by J. R. Fisher.

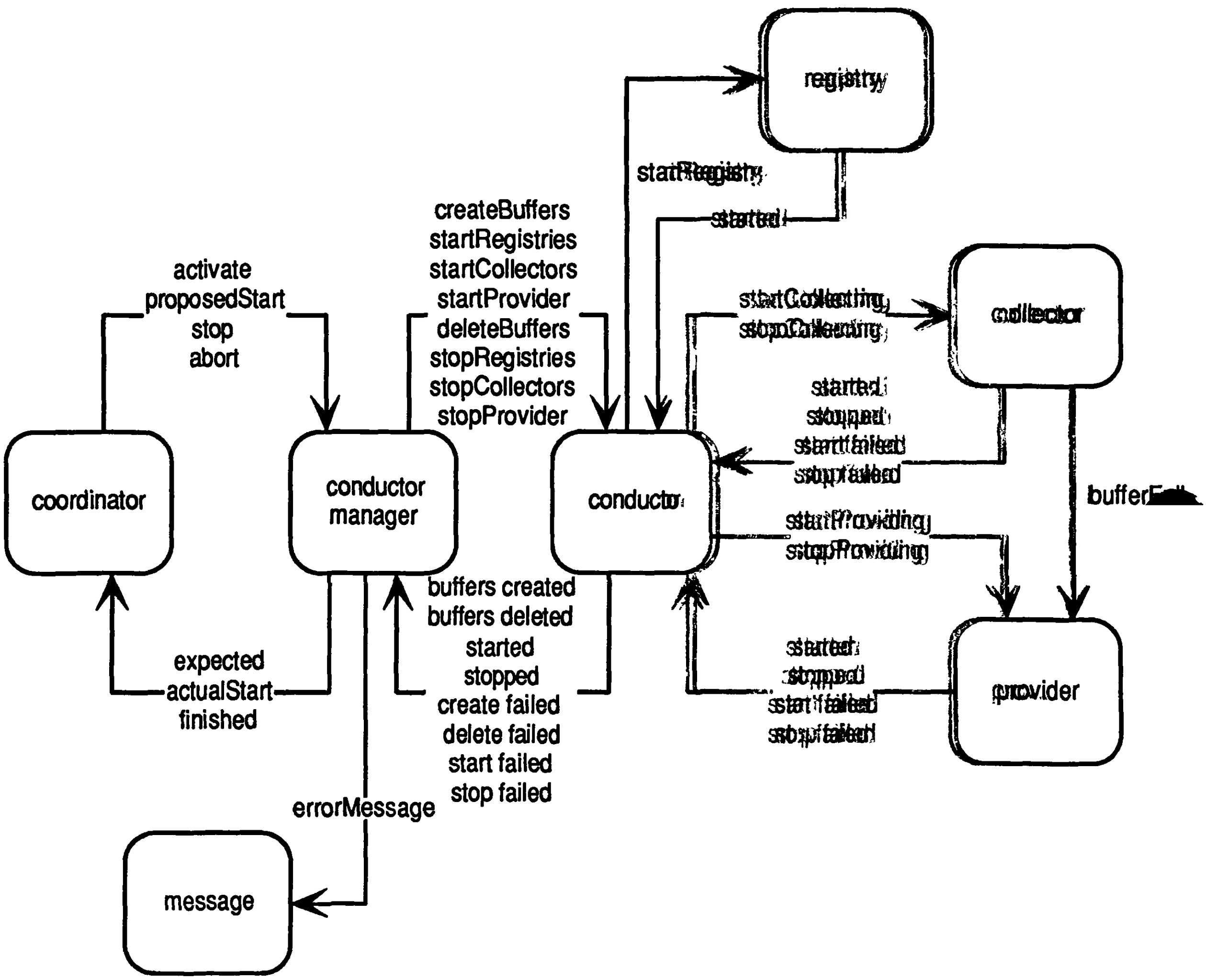

Figure 15: Collator Flow Diagram

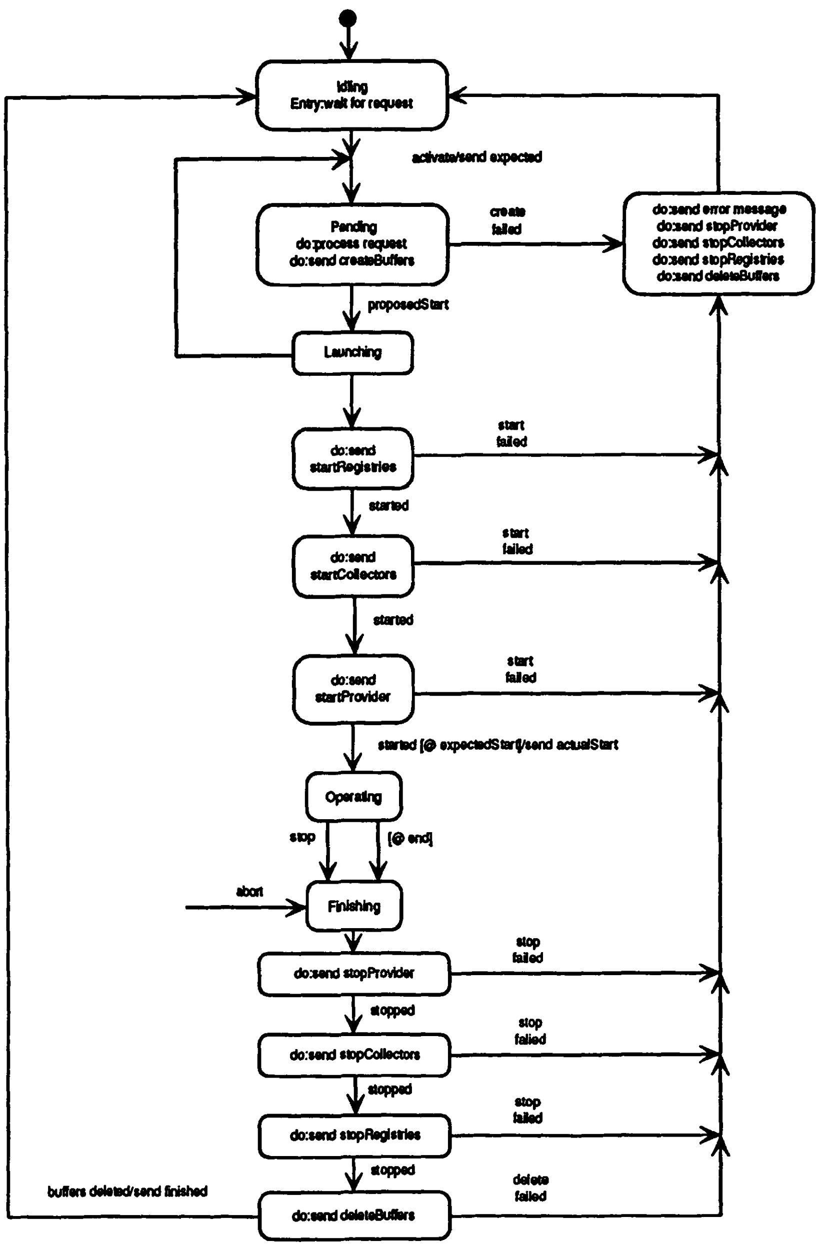

Figure 16: Conductor Manager Dynamic Diagram

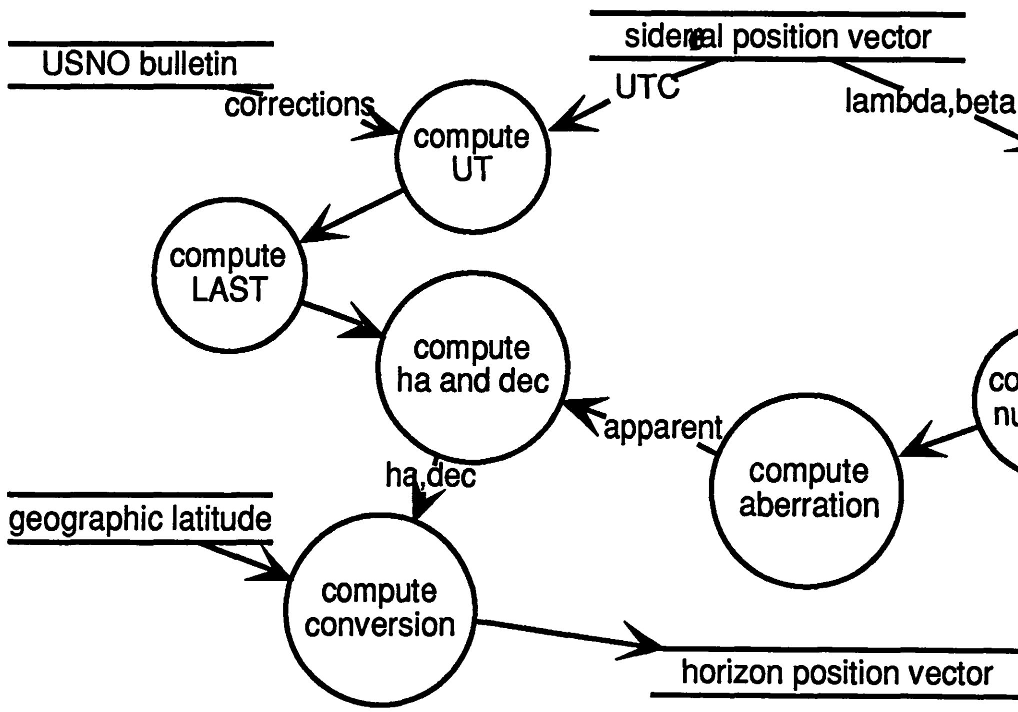

Figure 17: Coordinate Transformation Functional Diagram

4.1 Telescope Primitives

The telescope primitives are accessible through the magistrate and consist of the sum of the command sets of all of the telescope's devices.

- Control

  These are the managers' methods.

  - activate - loads the current set of attributes and initiates the all action necessary to start an observation.
  - proposedStart - arms the system to start observing at the specified time.
  - stop - terminates the current observation and returns it to an idle state in an orderly manner.
  - abort - terminates the current observation and returns it to an idle state immediately.

- Frontend

  - frontend select - positions the prime and secondary frontends.
  - cal level - sets the calibration signal level at the receiver.
  - synch controls - sets the operational state of the synch signal interfaces.
  - frequency range - sets the filters and other hardware switches for a given frequency range.
  - RF router controls - see router below.
  - converter controls - see converter below.

- LO

  - frequency list - series of frequencies through which the LO iterates when driven by a synch signal.
  - synch controls - sets the operational state of the synch signal interfaces.

- Converter
  - reference frame - sets either barycentric or telescope reference frame.
  - center frequency vector - array of frequencies describing the barycentric reference frame.
  - frequency tolerance - criterion for frequency shift due to Doppler shift allowed before the LO is changed to compensate.
  - IF - sets target IF.
  - sideband - selects upper or lower sideband.
- Router
  - connect - specifies the flow of a signal through the router.
- Antenna
  - reference frame - for each mirror selects the local or sidereal reference frame.
  - position vectors - for each mirror describes a locus of movement in the reference frame.
  - pointing tolerance - criteria for indicating pointing errors.
- Spectral processor

  As currently described in the spectral processor's keywords.
- Pointing

  TBD
- Continuum receiver

  TBD
- Spectrometer

  TBD

5 Issues

Work to be done:

- VLBA - the interface to the VLBA, especially for control of the antenna
- error analysis - actions and results as errors are encountered for all of the dynamic diagrams