Software integration at TNG. Active Optics: a practical example

C. Vuerli[a], C. Bonoli[b], A. Balestra[a,e], A. Baruffolo[b], L. Corcione[c], D. Fantinel[b], D.Gardiol[d], P. Marcucci[a], F. Pasian[a], C. Pernechele[b], M. Pucillo[a], R. Smareglia[a]

[a]Osservatorio Astronomico di Trieste – via G.B. Tiepolo 11, 34131-Trieste, Italy
[b]Osservatorio Astronomico di Padova – vicolo Osservatorio 5, 35122-Padova, Italy
[c]Osservatorio Astronomico di Torino – strada Osservatorio 20, Pino Torinese, Italy
[d]Telescopio Nazionale Galileo – Roque de los Muchachos – La Palma, Canary Islands – Spain
[e]European Southern Observatory, Karl Schwarzschild Str. 2, 85748 Garching bei München, Germany

Keywords: Telescope - Commissioning - Subsystems integration

ABSTRACT

Since March 1997, the TNG Telescope is in its Commissioning phase. In this paper we describe the structure of the control software of TNG and the on going activity of the software integration team.
The Telescope Communication Network has been completely installed, the control software has been set up and the integration phase is currently in progress.
The TNG control software has been designed having in mind the needs of a modern telescope control system: it is based on stable and widespread industry standards; its architecture is fully modular and intrinsically open in order to allow future enhancements and/or modifications of its components. Moreover, the code was written paying a particular attention to its portability. All these characteristics make the TNG control system open to future technology evolutions, both hardware and software-wise.
The TNG control software provides a coherent environment where the information flow is constantly guided and controlled through its path across the system. Despite the multiplicity and non-homogeneity of the different subsystems, TNG provides the operator a common framework from the raw data gathering, to the real-time applications, up to the operator interface and archiving system.
This was made designing and building a set of layers of increasing abstraction that were mapped onto the various physical components. A brief description of the steps followed during the integration of a number of subsystems will be given.

1. INTRODUCTION

The aim of this paper is to present the current state of the art of the software control integration activities at the Galileo Telescope.
All the aspects of the software integration will be illustrated; in particular the two major integration activities will be presented:

- Integration between the GATE environment and the WSS environment
- Integration between the WSS environment and the HAT on-line archiving system
- Integration between the WSS environment and IDL procedures

Before going in detail in the integration activities, the first part of the paper is dedicated to illustrate the whole hardware and software environment; a brief discussion on the currently installed hardware, on the local area network characteristics and on the basic software environment (installed operating systems and developing tools) can be found in the fist chapters.

The major problems faced during the integration phase and the general guidelines in performing this task are then presented. Finally, concrete examples of software integration at TNG will be illustrated putting particular emphasis on the integration activities concerning the Active Optics, one of the most complex subsystems of the telescope and for this reason fully comprehensive of all aspects of the integration phase.

SPIE Vol. 3351 • 0277-786X/98/$10.00

2. HARDWARE AND SOFTWARE ENVIRONMENT

2.1 HARDWARE ARCHITECTURE

The Galileo Telescope Control System is based on four different hardware architectures:

1. Transputer's Networks. In the Telescope control environment they are used in the active optics subsystem and for the CCD cameras. Furthermore some instruments will use them in controlling little movements of internal components.
2. VME Crates. Six VMEs are currently connected to the Telescope LAN in order to control:
   - Active Optics and CCD cameras
   - Building and Telescope Services
   - Drive System
   - Hydrostatic
   - Derotators; each of the two derotators (at the two nasmyth foci) has a dedicated VME crate
3. Personal Computers. They are mainly used in software development activities and for applications that require a massive use of graphical tools. As an example we can consider the PC card inserted in the VME bus controlling the active optics subsystem; it is used to display images coming from guide cameras, to monitor possible displacements of the star in respect of the initial position and to notify these displacement's values to the VME dedicated to the pointing/tracking.
4. HP Workstations. Each workstation of the control system communicates (through standard IP channels) with one or more VMEs and, by means of them controls the correspondent subsystems. Such links between workstations and VMEs are not static because they may be re-configured dynamically in order to balance the CPU load among all the available machines. Other workstations connected to the Telescope LAN run the software handling the data archiving.

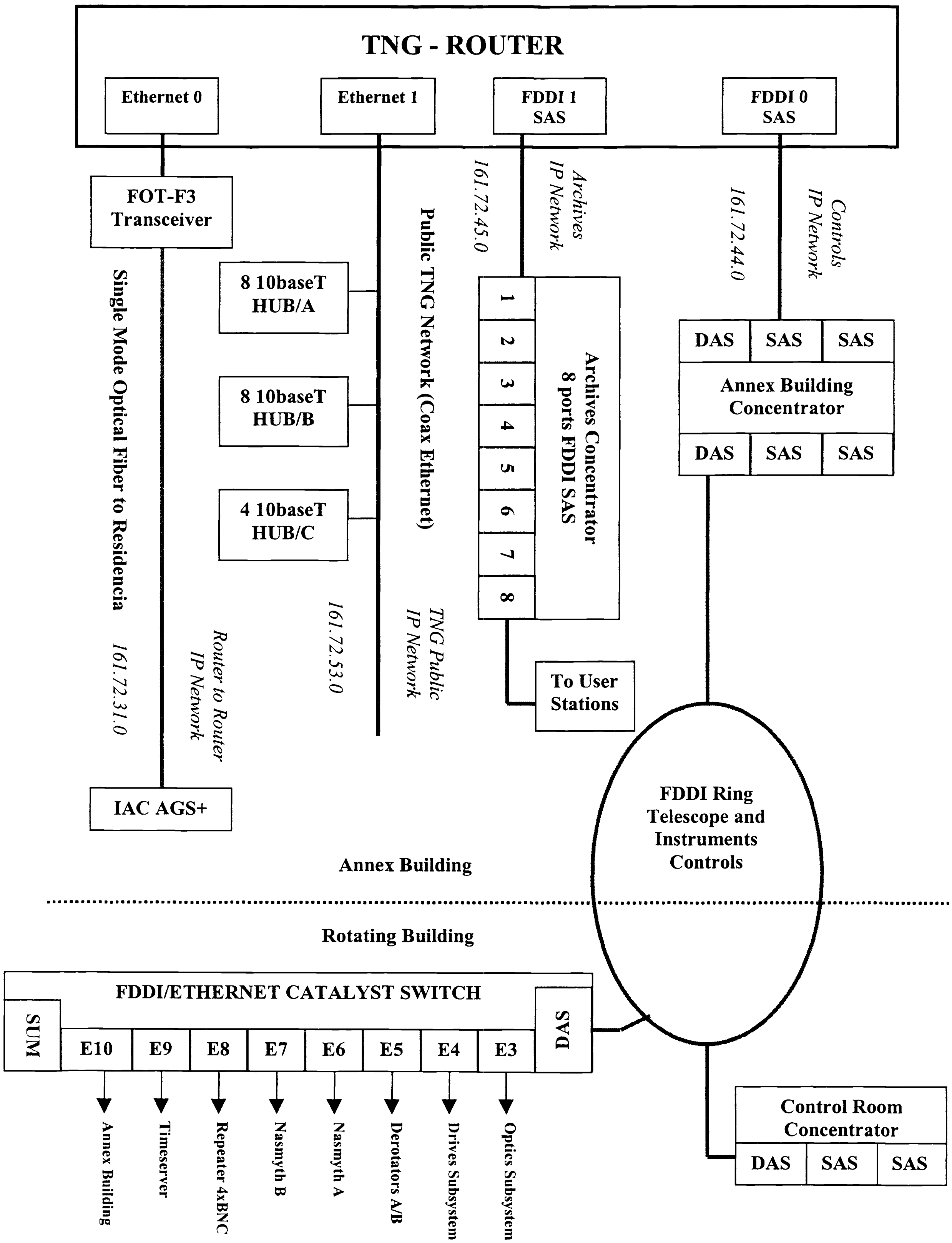
2.2 NETWORK ARCHITECTURE

The Network Architecture at TNG is based on two physical transport media: *Ethernet* and *FDDI*.
FDDI represents the backbone of the LAN; all the workstations (both the control workstations as well as those dedicated to the archiving system) communicate via FDDI. FDDI has been chosen in order to improve the transfer rate of large amounts of data and because it is now a well consolidated standard. The FDDI ring hosts two concentrators and a fast switch FDDI/Ethernet putting in communication the FDDI ring and eight separated Ethernet segments. All the workstation are equipped with SAS FDDI cards allowing to connect it to a concentrator.
All the connections going to the Telescope (Center Piece, Nasmyth Rooms, Derotators, Drive System Rack) use currently Ethernet as transport medium; in particular all VMEs are connected to the LAN through coaxial Ethernet cables. The first tests of acquisition/transmission of technical images obtained very good transmission rates. It is not foreseen then to extend FDDI to VME crates at this stage although this is not excluded in the future.

As shown in the Figure above, the TNG LAN is subdivided in three well separated physical networks. Different IP networks have been assigned to each physical sub-net. The IP networks used by TNG are class-C sub-networks of the class-B IP network assigned to IAC (Instituto de Astrofisica de Canarias).
The three sub-networks of the TNG LAN are listed here:

- Telescope and Instruments Controls Network.
- Data Archiving Network.
- TNG Public Network.

The three physical networks of the LAN are interconnected by means of the Telescope Router, a Cisco 4500 with two FDDI ports and two Ethernet ports.


TNG - ROUTER
Ethernet 0
Ethernet 1
FDDI 1 SAS
FDDI 0 SAS
FOT-F3 Transceiver
Single Mode Optical Fiber to Residencia
161.72.31.0
Router to Router IP Network
IAC AGS+
8 10baseT HUB/A
8 10baseT HUB/B
4 10baseT HUB/C
Public TNG Network (Coax Ethernet)
161.72.53.0
TNG Public IP Network
161.72.45.0
Archives IP Network
1
2
3
4
5
6
7
8
Archives Concentrator 8 ports FDDI SAS
To User Stations
161.72.44.0
Controls IP Network
DAS
SAS
SAS
Annex Building Concentrator
DAS
SAS
SAS
FDDI Ring Telescope and Instruments Controls
Annex Building
Rotating Building
FDDI/ETHERNET CATALYST SWITCH
SUM
E10
E9
E8
E7
E6
E5
E4
E3
DAS
Annex Building
Timeserver
Repeater 4xBNC
Nasmyth B
Nasmyth A
Derotators A/B
Drives Subsystem
Optics Subsystem
Control Room Concentrator
DAS
SAS
SAS

2.3 SOFTWARE ARCHITECTURE

With the term *Software Architecture* we intend all the *programming languages* used to create the control software and installed on the developing machines at Telescope and the *Operating Systems* under which the control software is running.

2.3.1 OPERATING SYSTEMS

The following operating systems are installed on the different hardware architectures:

- PDOS 4.2 Operating Systems running on all VME CPUs
- HP-UX 9.07 Operating System running on all HP workstations. However the WSS control software installed on HP workstations has been successfully compiled and tested under the release 10.20 of HP-UX. Before the end of April 1998 then the latest version of HP-UX will be installed on all workstations at Telescope.
- Windows95 Operating System on all PCs.

2.3.2 PROGRAMMING LANGUAGES

Under PDOS Operating Systems a C ANSI compiler is installed thorough which the code of GATE and of all related tasks has been developed.

On HP Workstation the HP C native compiler is installed and used for the WSS development. On all workstations at Telescope the release 4.0.1 of IDL is available; IDL has been utilized in creating applications for some Telescope subsystems. Such applications are mainly composed by a graphical User Interface and provide all the needed computations on the subsystem's telemetry and data. Together with the release 10.20 of HP-UX, the most recent version of IDL (5.0) will also be installed.

The transputer's Networks have been programmed using Occam 2.1.

Finally, all the applications running under Windows95 Operating System have been developed using Visual C++.

3. GATE

GATE (Galileo Telescope Environment) is the main package running on all VME CPUs controlling one of the Telescope subsystems or one of the instruments placed at Nasmyth foci of TNG.
The GATE software is composed by a pool of *System Tasks* that includes GATE itself (TASK0). These tasks must be present on all VMEs as they provide the basic functionality of the package. GATE cannot work properly if one of them is not present.

The base functions of GATE to which the pool of System Tasks provide are listed here:

- Local User Interface handling
- Local commands execution
- Tasks creation and monitoring
- WSS Communications handling
- Interface to heterogeneous systems (PCs, PLCs, transputer's networks) handling

Besides the System Tasks, other tasks may be present in the GATE environment; they are indicated with the term *Application Tasks*; each of them provides for a specific peculiar functionality related to the particular Telescope subsystem or Instrument. The Application Tasks then differ from VME to VME depending on which subsystem or Instrument a VME is controlling.

3.1 SYSTEM TASKS OF GATE

The system Tasks of GATE are discussed in this paragraph focussing for each of them the provided functionality and the main architectural characteristics. GATE cannot work properly if one of these tasks is not installed.

GATE:
This is the main task, known also as ***TASK0.*** When started, GATE creates and maintains the whole GATE environment handling a common area of global parameters and a set of semaphores and events. This common area is shared by all tasks of GATE. Through it data and control information flow one task to the others.

GATE may be activated in two different ways that we indicate with the term ***living mode.***

When the living mode is set to *SUPER,* no communication channels are open towards the connected workstation, so GATE puts the whole environment in a standalone state. In this case GATE can be controlled only by means of a local User Interface. The communication tasks are not activated in this case. This living mode is used in the development/test phase.

The normal living mode for GATE is *USER*. In this case all the communication channels towards the connected workstation are active. The user still may interact locally with GATE but only through a very restricted set of commands and only for monitoring purposes. When GATE is running in *USER* mode, it is activated by WSS.

TASK1:
TASK1 handles the channel on which all commands are received from the connected workstation. When started, TASK1 opens the command socket as server and waits for a connection from the partner workstation. The client is in this case the WSS User Interface. Each received command is inserted by TASK1 in the *IMMEDIATE* or *DELAYED* commands queue depending on the command queue destination flag.

TASK2:
This task picks up the commands from the two command queues (*IMMEDIATE* and *DELAYED)* dispatching them to the destination task;
The commands are processed following the FIFO policy. Furthermore the *IMMEDIATE* queue has higher priority.

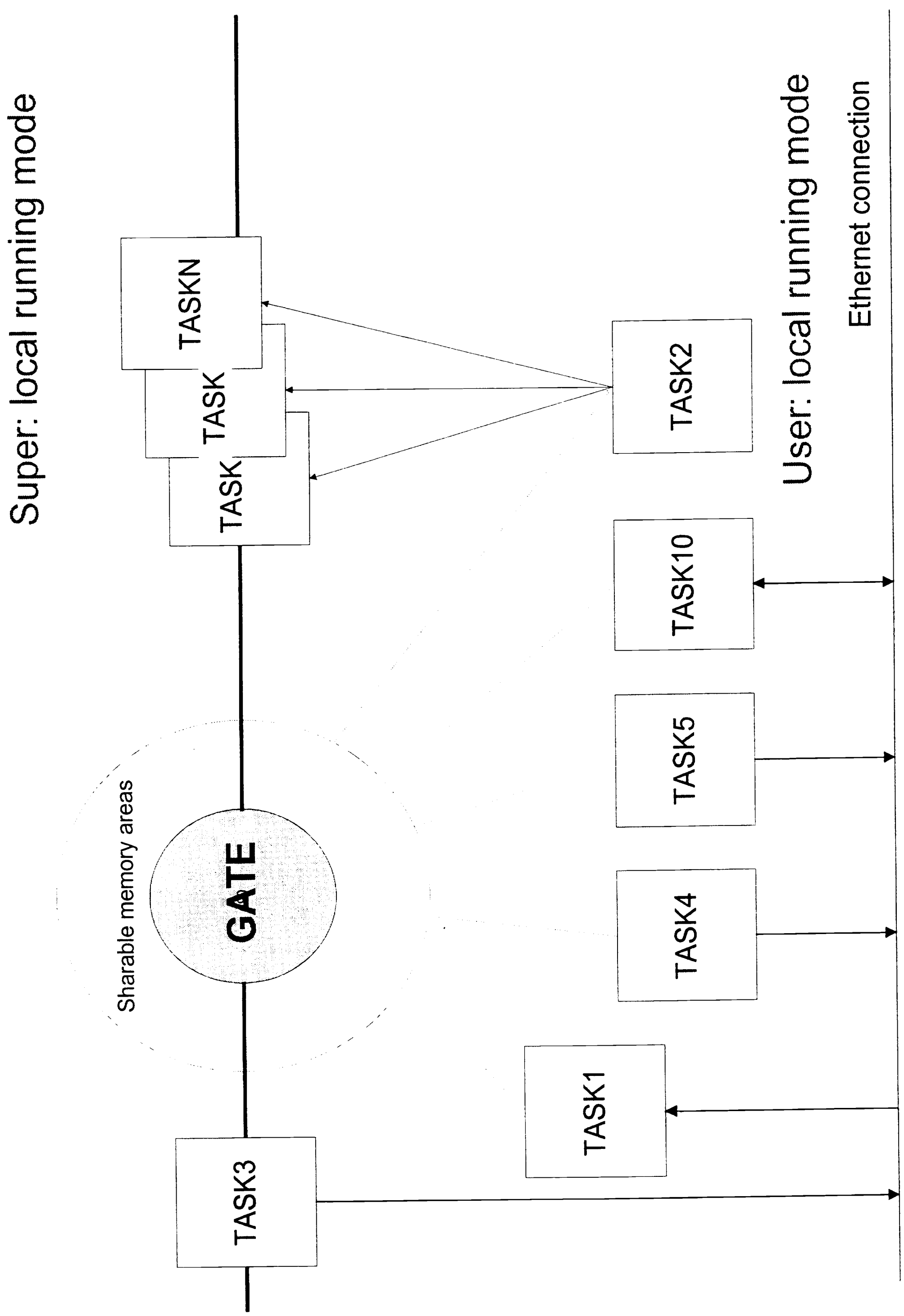
TASK3:
This task handles the alarm conditions notifying them both on the local interface and transmitting them over the alarms channel to the WSS. These alarms are generated when the value of a telemetry parameter is found out of range.

TASK4:
This is the task handling all the telemetry transmission and commands verification. The task acts on the basis of a polling sequence looking for the values of all telemetry parameters and for command verifications

TASK5:
TASK5 is the task having in charge the DATA channel. Through this channel different types of structured data are sent to the connected workstation. The structured data currently using this channel are:

- *RING BUFFERS*
- *TECHNICAL IMAGES*

TASK10:
TASK10 is the task devoted to the communication with the partner workstation. When created by GATE, it performs the handshake procedure sending to WSS a packet containing the acronym of the local VME system. After the handshake packet delivery, this task exchanges *HEARTBEAT* packets with the partner WSS.



GATE general layout

4. WSS

The WSS (Workstation Software System) is the high level software running on HP workstations hardware platforms under the HP-UX operating system. Through WSS different categories of users (technical staff, resident astronomers, observers) may interact with the various parts of the Telescope (Telescope subsystems) and with all Instruments. This interaction always takes place by means of GATE running on VME platforms. The WSS processes may be subdivided in:

- *Main WSS processes* providing for the basic functionality of WSS.
- *Ancillary WSS processes* written in order to interact with a particular Telescope subsystem or with an instrument

The base functions of WSS to which the pool of main WSS processes provide are listed here:

- Creation/handling of a User Interface allowing controlled and secure access to the WSS/GATE environment
- Delivery of commands to the connected VME systems or to other WSS processes (typically to ancillary processes)
- Monitoring of command queues. Notification of any fault in the commands (both external and internal) execution
- Collection of telemetry coming from connected VMEs and related to subsystems or instruments controlled by these VMEs
- Notification of alarm conditions coming from one the connected VMEs.
- Local telemetry dissemination (on demand) to other WSS systems.

4.1 MAIN PROCESSES OF WSS

The main processes of WSS are discussed in this paragraph focussing for each of them the provided functionality and the main architectural characteristics. WSS cannot work properly if one of these processes is not installed.

INIT:

INIT is the main process of WSS. It provides for the following tasks:

- create the shared memory segments on which the on-line database is stored.
- Handles the handshake procedure with all VME systems connected to the local workstation (delivery of the configuration files and remote activation of GATE).
- Creation of all other WSS processes (main processes as well as ancillary processes)

INIT handles also periodic activities like the creation and maintenance of the Telemetry Log file

DISPLAY:

The main WSS process DISPLAY (acronym *UIF*) provides for two main tasks.

- Creates and maintains the *WSS User Interface*
- delivers commands to other local processes (internal commands) as well as commands to be sent to a connected GATE (external commands).

VERIFY:

This process (acronym *TMV*) handles both telemetry channels and data channels towards all connected GATEs. The architecture of VERIFY is basically constituted by a polling procedure looking periodically if one of the channels is active (this happens when there are pending data in the channel waiting to be processed).

Over the telemetry channel VERIFY receives both telemetry packets and command verification packets.

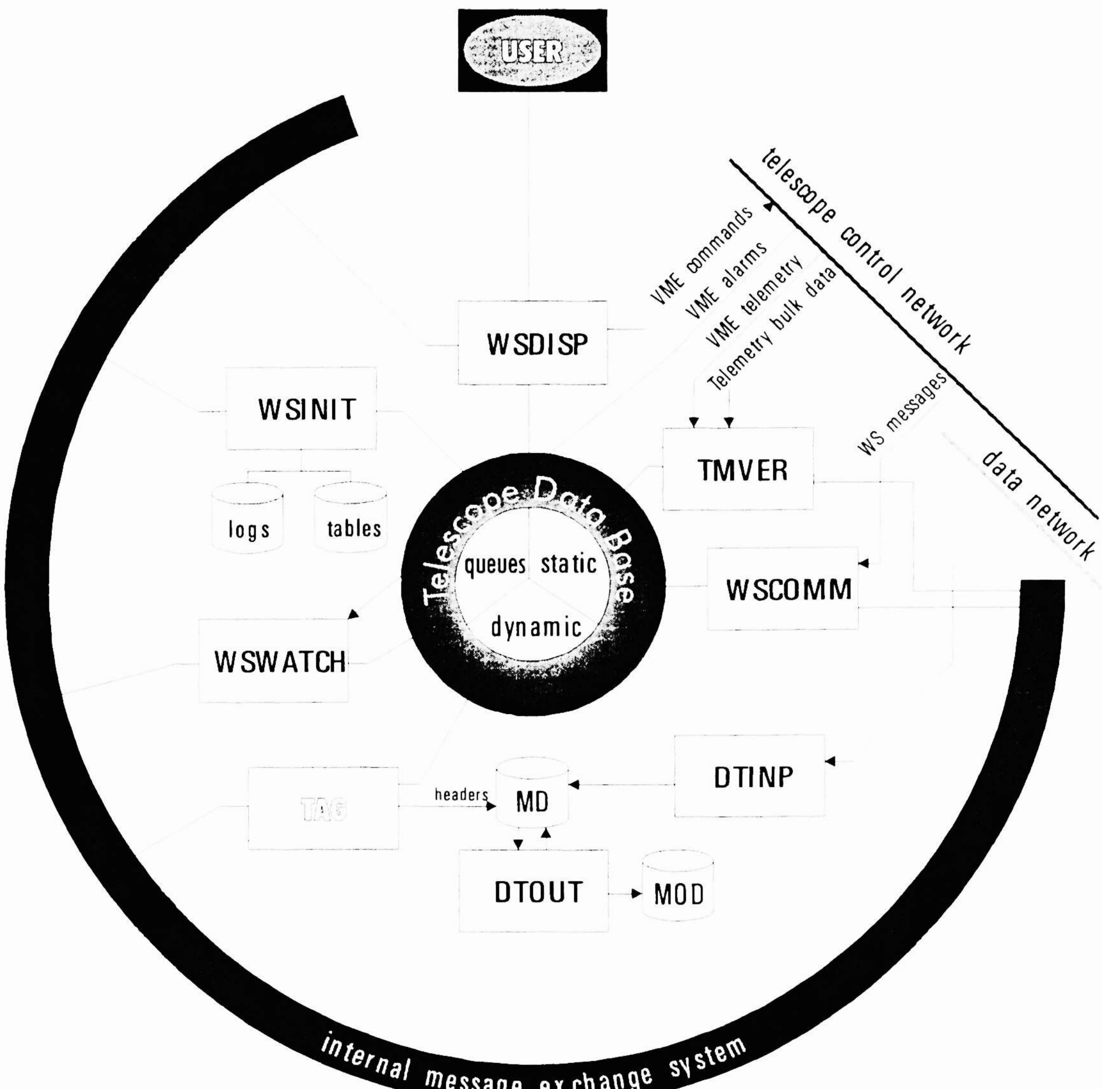
On data channels VERIFY receives both Ring Buffers and Technical Images.

WSWATCH:

Two tasks are carried out by this process:

- Periodically checks the command queues (four command queues for each connected VME)
- Handles the alarms channels (one for each connected VME).

WSCOM:
WSCOM is the process devoted to inter-workstation communications maintenance. That is, each active WSCOM present on the network maintains a communication channel open for each other WSCOM found on network.
In particular WSCOM provides for heartbeat packets exchange with all other WSCOMs and with the TASK10 of each connected GATE, and reroute properly all messages and requests destined to another workstation.



WSREC:
This process has an open channel towards each other WSS present on the network. Through this channel WSREC receives data coming from on-line databases of other workstation writing them in the local on-line database.

TAG:
TAG (Telemetry Archive Gateway) is the process representing the contact point between the whole control system (WSS/GATE) and the archiving system. TAG is a process being part of WSS but communicates also with DTINP, a process of the archiving system acting as server and collecting from all TAGs (one for each WSS) all the relevant information for the subsequent data calibration and archiving off-line activity.

4.2 OFF-LINE TOOLS

It is worth to present briefly here two off-line tools being them integral part of WSS.

The *Table Editor* is a graphical tool allowing to maintain easily the WSS configuration files; the contents of such configuration files are copied in the static database segment at the WSS startup. By means of the Table Editor the basic components of the WSS database (systems, units, items) can be maintained.
The *Graphic Editor* is a graphical tool allowing to create/modify widgets to be loaded by the WSS User Interface during a WSS session. The Graphic Editor makes available a set of base component widgets that may be grouped to build complex panels.
These tools as well as the WSS User Interface have been implemented over *X11R5* and *Motif 1.2* shared libraries.

5. WSS-GATE INTEGRATION

In this chapter the steps to be followed in order to integrate a GATE system and a WSS system are presented. As may be seen below this is a fairly simple task. Summarizing what stated in the previous chapters the following aspects may be outlined:

- The Link between a particular GATE and a particular WSS is not static in the sense that the same GATE system may be linked (or assigned) to a different WSS system before the WSS/GATE system startup. The only condition is obviously that GATE must be reachable via IP from the partner WSS. During different WSS sessions then the same VME can be connected to different workstations. Furthermore if a workstation dies during a WSS session the VMEs connected to it will be *inherited* by another workstation.
- A pool of system tasks of GATE and a pool of main processes of WSS guarantee the correct integration between the two systems. They interact by means of a set of IP channels being them:
  - The *handshake channel.* It is used at the system startup. It is closed at the handshake procedure completion.
  - The *command channel.*
  - The *telemetry channel.*
  - The *data channel.*
  - The *alarms channel*

Using the Table Editor off-line tool, these steps have to be followed in integrating a WSS system and a GATE system:
1. Insertion of the WSS system in the systems configuration file if not already present.
2. Creation of the dynamic link between the two systems editing the record of GATE in the systems configuration file
3. Insertion of the GATE system in the systems configuration file if not already present.
4. Insertion of all units of the two systems in the correspondent units configuration files (if not already present)
5. Insertion of the set of commands for each unit of the GATE system in the correspondent command configuration files
6. Insertion of the set of parameters for each unit of the GATE system in the correspondent parameters configuration files

When all the above conditions are satisfied, the two system are ready to interact.

5.1 CURRENT WSS-GATE INTEGRATION AT TNG

There are four WSS systems and five GATE systems currently installed at Galileo Telescope. The WSS-GATE integration has been successfully carried out for the following GATE systems:

- *VHBS*. This is the GATE controlling the hydrostatic telescope subsystem
- *VTRK*. This GATE controls the Pointing/Tracking telescope subsystem
- *VBLD*. The GATE system controlling the Dome and all the Telescope services
- *VAOP*. GATE controlling the active Optics Telescope subsystem (M1, M2, M3) plus the CCD cameras.

For each of the GATE systems listed above all the integration steps (1-6) have been performed; it has been then verified the correct functionality for each of the following aspects: handshake procedure; commands delivery and verification; telemetry transmission; Ring Buffers collection and transmission; alarm conditions notification; for technical CCD cameras, the production and delivery of technical images too has proved working properly.

6. WSS-HAT INTEGRATION

As already mentioned in section 3, a gateway exists between the control system and the software actually archiving on-line the telemetry, the science frames and, if relevant, the technical frames. This is accomplished by TAG, a process belonging to both the WSS (thus sharing the Telescope Data Base and the message exchange mechanism) and the HAT (Handling and Archiving Tool), the system taking care of performing the archiving of telescope and instruments data and activity.

TAG has three basic tasks:

- saving periodically the telemetry logs produced by the WSS, by informing a HAT process (DTINP) that updates to the log are available on one or more of the workstations running WSS.
- receiving a message from the WSS any time an exposure (either technical or scientific) is started: at this stage, a "snapshot" of the Telescope Data Base is taken to record the status of the overall system; such information is fed to DTINP.
- receiving a message from the WSS any time an exposure is finished: the information on the location of the file containing the acquired frame (computer, path, filename) is given to DTINP together with the information on time.

All HAT processes (except for TAG) run on the Archive Server and are basically off-line with respect to the WSS not to interfere with control operations.

- DTINP reads the updates to the log files from the control workstations to the Archive Server, where a unified log is created; it updates the Technical Archive is updated, by unscrambling the telemetry information and storing a number of table-selected parameters onto a Telemetry Database; it has furthermore the task of updating the Catalogue of Exposures on the Scientific Archive with the appropriate parameters.
- DTOUT builds FITS files out of the raw pixel frames read from the disk of the relevant control workstation, by gathering from DTINP the parameters needed to build the header; in the case of a parameter subject to important variations during the exposure (e.g. seeing or detector temperature), the changes are read from the telemetry log and the variance is inserted as a FITS keyword.
- DTORG at the end of the observing session creates an Observations Organizing Table by associating science exposures with the relevant "standard" calibration files.
- DTFLUSH is finally started when the observing session is ended, in order to prepare three copies on CD-Rs of the data acquired during the session itself (one each for the Observatory, the General Archive, and the observer).

7. WSS-IDL INTEGRATION

For some telescope subsystems like the Active Optics and the Tracking it has been decided to build not simple ancillary WSS processes but special ancillaries written in IDL. We will shortly refer these ancillaries with the term *IDL processes*. Such a decision has been taken because this kind of complex subsystems need to join hard graphical performances and great computational power; both these characteristics are fully satisfied by IDL.

7.1 WSS AND IDL: WHAT TO DO ?

The major problems to be solved in putting together WSS processes and IDL procedures on the same workstation is the need to guarantee a communication mechanism between such IDL procedures and all VME systems connected to the local workstation. Like WSS processes, IDL procedures need to send commands to the tasks of GATE and to receive telemetry on which to apply computational algorithms. Even if IDL procedures are able to handle TCP/IP communications it has been decided to avoid duplication of typical WSS tasks inside IDL procedures. This solution furthermore is not convenient from the point of view of GATE because it forces a duplication of the communication channels, one reserved for communications with WSS and the others for data exchange with one or more IDL procedures.

7.2 SOLUTION

The above problem has been solved introducing a special shared library named *apidlib* used in communications among the traditional WSS processes and IDL procedures. Because an IDL procedure is a Unix process like all other WSS processes, the basic idea is to treat an IDL procedure as a WSS process fully capable to communicate with all other WSS processes and to retrieve information from the WSS on-line database.

The *apidlib* shared library offers to IDL a set of functions making extremely easy to communicate with the rest of the WSS environment. Through the *CALL_EXTERNAL* mechanism, IDL procedures use the WSS User Interface as gateway in sending commands to VMEs and, by means of direct accesses to the WSS on-line database they are able to retrieve telemetry coming from the connected GATEs; furthermore, through the shared memory, IDL procedures exchange information with other WSS processes.

7.3 WSS-IDL INTEGRATION AT TNG

The IDL procedures currently developed at Galileo Telescope and fully integrated in the WSS environment are:

- *CCD:* This procedure controls the CCD technical cameras by means of the *VAOP_CCD* unit
- *AOP:*. This procedure controls the Active Optics Subsystem of the Telescope by means of the *VAOP_AOP* unit
- *TRK:* This procedure controls the Pointing/Tracking Subsystem of the Telescope through the *VTRK_TRK* unit

Note that each procedure is referred by means of its acronym. Because an IDL procedure is actually a process running on a WSS system, it is a unit of that system like any other WSS process.

8. TELESCOPE SUBSYSTEMS

8.1 POINTING AND TRACKING SUBSYSTEM.

The pointing and tracking subsystem is regarded at the same level as all the other subsystems and, like them, communicates with the WorkStation via GATE. It is unique amongst the subsystems, however, in running on a multiprocessor bus.

The hardware configuration for this subsystem has been described in previous papers [1]. To briefly summarize, the VME hosts 4 Eltec Eurocom-7 CPUs, sharing the various operations; communication among them and with the other boards (digital I/O, UT synchronization) happens by means of VME and VSB bus. System functionalities are rather complex and were subdivided as follows:

CPU#1 is responsible for the astrometric loop calculation. The pointing loop is based closely on that developed by Wallace for Keck and Gemini telescopes [2]. User coordinates (α, δ) are transformed to mount coordinates and corrected for instrumental and ambient effetcs. New position values (azimuth, altitude and parallactic angle) are passed to CPU#3 via VSB bus; encoder readouts are retrieved and their value is compared with the desired postition. The entire operation cycle happens at the frequency of 20 Hz.

CPU#2 is responsible for the communication with WSS: command reception, telemetry sending, heart-beat exchange and alarm notification.

CPU#3 is responsible for (Az, El, Rot) axis pointing, tracking capabilities and servo control. Space and speed loops, running at 500 Hz, provide a linear interpolation between two following positions, as received by CPU#1 at a 20 Hz frequency.

CPU#4 is responsible for diagnostic.

UT synchronization is performed through a bc635VME board, from DATUM Inc. BANCOMM Division. Each of the four processors on the bus cyclically programs the board to receive the content of the time registers. This mechanism allows all the processors to maintain the same absolute time.

An IDL procedure running on the HP WorkStation handles the user interface allowing coordinate input, command sending to the telescope and acts as an interface with on-line catalogs.

8.2 DOME AND TELESCOPE SERVICES SUBSYSTEM.

The hardware configuration of the VME controlling the dome and telescope service subsystem include an ELTEC Eurocom-6 CPU, mounting a Motorola MC68030 on a standard VME bus. Devices controlled by this VME are: Mirror cover, baffles, Nasmyth A and B shutters, wind screen, slit shutters, flaps, locking devices, cranes and dome tracking.

Field interface is done through three industry provided PLCs, directly controlling the different devices; a master PC collect and dispatch information to/from the PLCs and talks direcly with the VME via a serial link.

8.3 HYDROSTATIC SUBSYSTEM

The hardware configuration of the VME controlling the hydrostatic subsystem include an ELTEC Eurocom-6 CPU, mounting a Motorola MC68030 on a standard VME bus. The interface with the Hydrostatic bearing System is done through an PLC connected with the VME bus via parallel I/O boards.

8.4 THE ACTIVE OPTICS SUBSYSTEM

The active optics subsystem is a very good example for the software integration, being it a complex system, where a VME-based controller, 3 transputer-links, a Windows-like application on a PC board and an IDL User Interface on Workstation have to interact in controlling the mirrors and all the dedicated services (see [3]). Due to its complexity, this subsystem offers a complete and fully comprehensive overview on all aspects of the software integration at TNG.

8.4.1 THE ACTIVE OPTICS TRANSPUTER NETWORK

The low level Active Optics control system is based on a transputer network. The interface with the VME system is assured by a transputer custom adapter card ATX-260 Atenix. Three of these cards are dedicated to control the following systems:

1. The mirrors control network (91 transputers)
2. The Shack-Hartmann analysis cameras (3 transputers)
3. The tracking Cameras (3 transputers)

The network controlling the mirrors is itself composed by two sub-nets (one sub-net for M1 and the other for M2+M3), each one with a tree structure. Two links (of the four available) of the root transputer are connected to these sub-nets; the third one is connected with the VME-CPU and is used in loading bootable software only; finally the fourth one is connected to a PC used in developing software and for test purposes. The communication with the VME-CPU is made by means of a 64 MB dual port (VME-VSB) memory card.

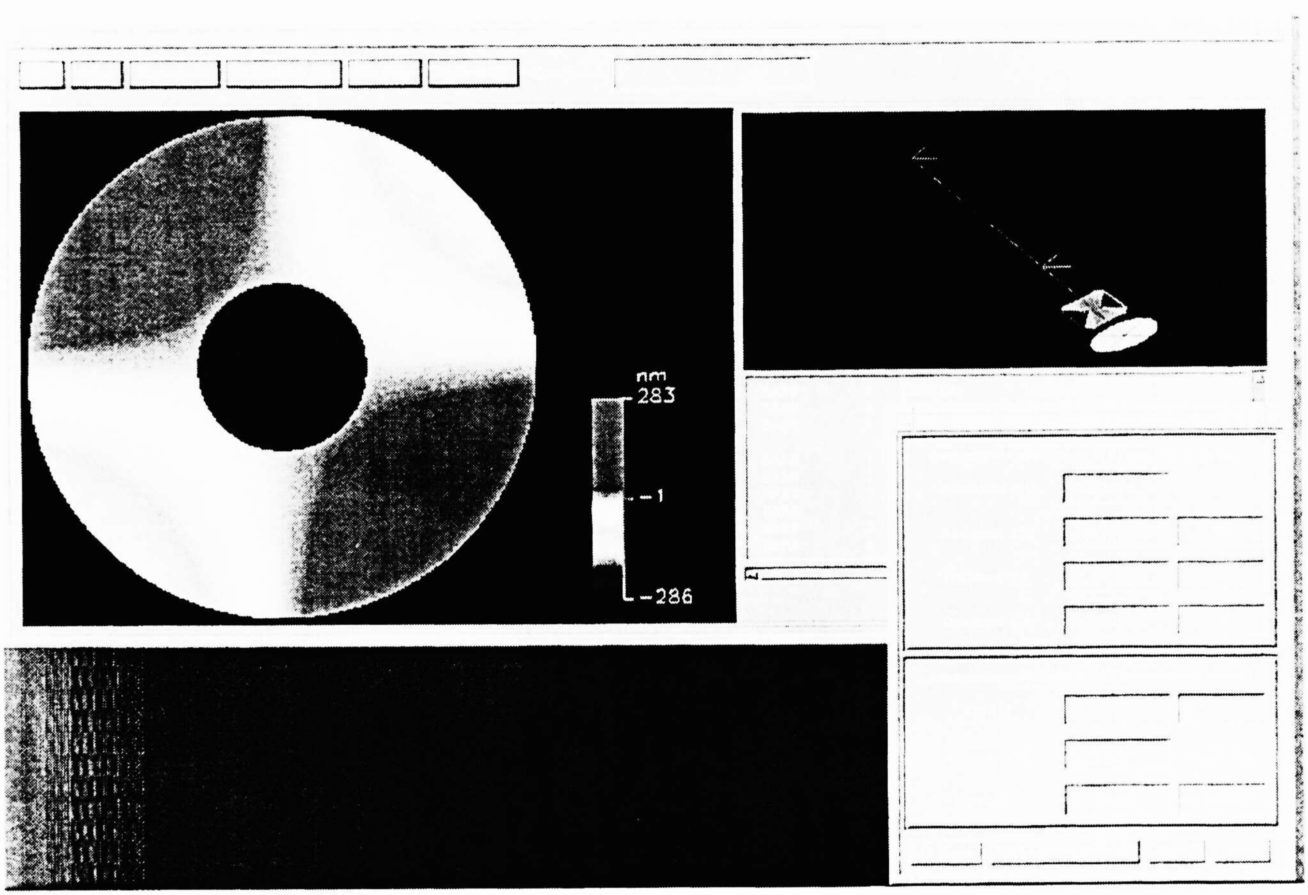
Each node of the transputer network controls a section of the system: 75 nodes are dedicated to the control of the M1 astatic actuators; 6 nodes control the six M2 exapod bars; 1 transputer is dedicated to control M3; the remaining 8 nodes are used in monitoring the four M1 fixed actuators (three reference and one dummy), the four M1 ring actuators, the M2 bar limit switches and temperatures.

8.4.2 THE ACTIVE OPTICS VME CONFIGURATION

The AOPT system is controlled by a VME crate, hosting an ELTEC-7 Eurocom CPU, an ATX-260 board, dedicated to the interface with the transputer links and a dual-port (VME-VSB) memory card, where all information and data are collected. Commands coming from the User Interface are decoded, interpreted and dispatched to the transputer links. Telemetry is collected from the transputer links and temporarily stored into the dual-port memory to be sent to the connected Workstation on the Telemetry channel. Image data are read from the dual-port memory and sent to the Workstation over the data channel.

8.4.3 THE ACTIVE OPTICS USER INTERFACE

As stated above an IDL User Interface has been developed to allow the users to control/monitor this subsystem. The reasons of such a choice match with what already stated in chapter 5. In the same chapter the integration problem between WSS and IDL has been deeply discussed and a general solution (applied to this subsystem as well as to all other Telescope subsystems) was described. The figure below shows the general layout of this specific window.

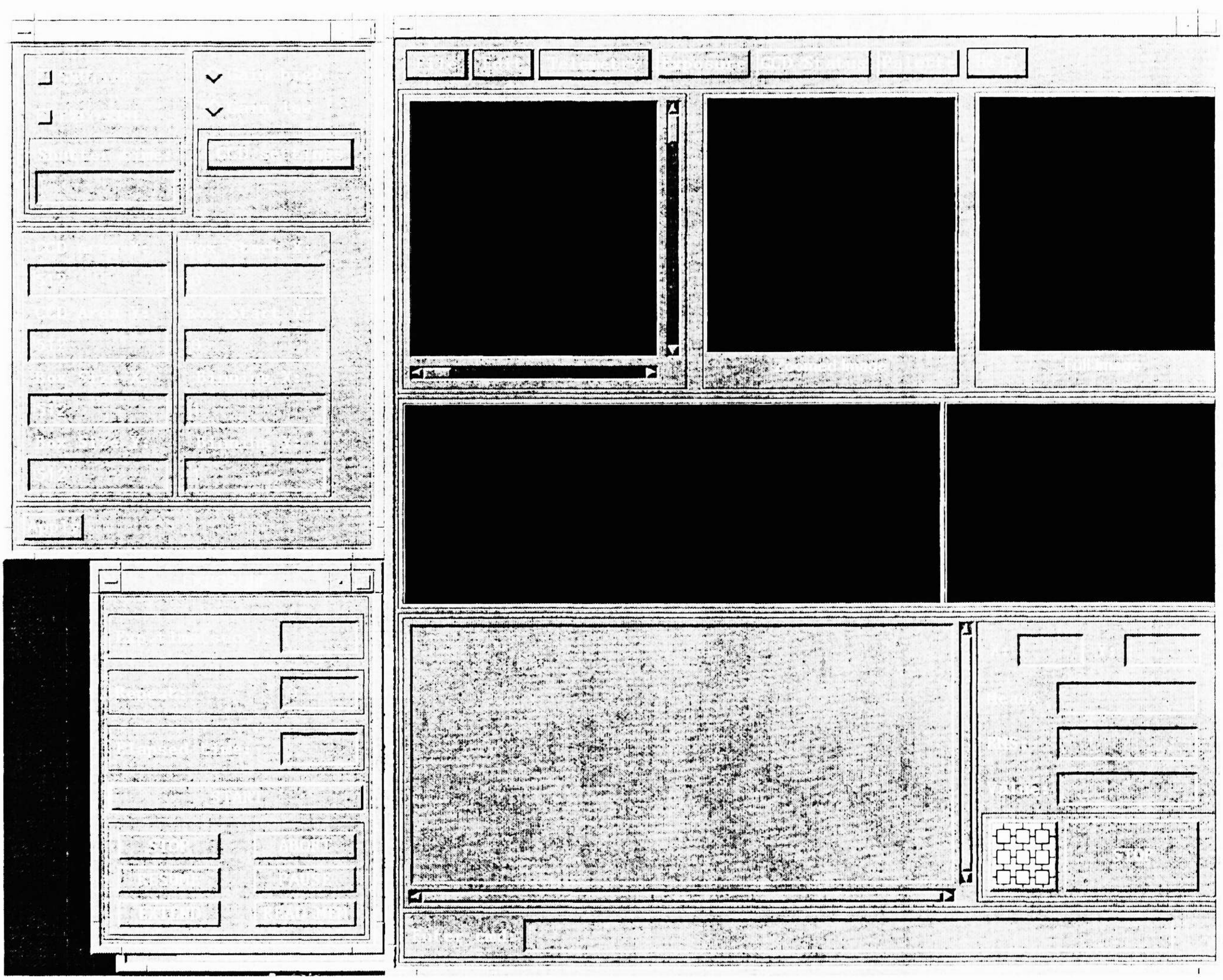


8.5 CCD CAMERAS

CCD cameras are controlled by the VME devoted to the active optics system. Its hardware and software configuration have been described in the above section.
Here we describe what was implemented specifically to handle exposures and images.

On the VME bus a PC board is installed where a Windows application runs, as an interface with the cameras. The application retrieves images from the dual-port memory board and displays them on a monitor. When used as a guider, the application also computes barycenter displacements with respect to the original location. The displacement values are then sent, via serial link, to the drive VME to be used inside the astrometric loop.

On the WorkStation, an IDL widget based application, shown in the figure below, allows the user to enter all the data to configure cameras and observation and works also as a quick look facility.



REFERENCES

1. C. Bonoli, D. Mancini, F. Bortoletto, M. D'Alessandro, L. Corcione, P. Schipani, R. Stefani, "TNG control system. Computer architecture, interfacing and synchronization", SPIE Proceedings: "Telescope control systems", 1995, 2479, 160
2. P.T. Wallace, "Gemini Pointing Algoritms",Gemini TCS Note 3.9, 1996
3. F. Bortoletto, "The Galileo Telescope Active Optics System", in: "The 3 Galileos: the man, the spacecraft, the telescope", C.Barbieri, J.H. Rahe, T.G. Johson and A.M.Sohus eds., 1997
4. A. Balestra, P. Marcucci, F. Pasian, M. Pucillo, R. Smareglia, C. Vuerli, "Galileo Project: Workstation Software System", TNG Technical Report N. 9, 1991
5. P. Marcucci, M. Pucillo, "The Galileo User Interface User's Guide", Publication of the Astronomical Observatory of Trieste N. 1454, May 1994
6. P. Marcucci, M. Pucillo, "The Galileo Table Editor", Publication of the Astronomical Observatory of Trieste N. 1450, September 1992
7. F. Pasian, "Archives at the TNG - Architectural Design Document ", TNG Technical Report N. 51, 1995