REMOT project

Definition of Control Systems and Remote Users. Interfaces and Adaptations

M. Callegari, S. Monai, P. Marcucci, M. Pucillo, C. Vuerli

Osservatorio Astronomico di Trieste

Rapporto Tecnico N. 34/97

Pubblicazione O.A.T. N. 1887/97

Presentato nell'ambito del progetto UE REMOT (RE 1008) come documento D6.3.

D6.3: DEFINITION OF CONTROL SYSTEMS AND REMOTE USERS. INTERFACES AND ADAPTATIONS

Massimo Callegari (e-Mail: *callegar@oat.ts.astro.it*, phone +39 - 40 - 3199232)
Sergio Monai (e-Mail: *monai@oat.ts.astro.it*, phone +39 - 40 - 3199232)
Paolo Marcucci (e-Mail: *marcucci@oat.ts.astro.it*, phone +39 - 40 - 3199214)
Mauro Pucillo (e-Mail: *pucillo@oat.ts.astro.it*, phone +39 - 40 - 3199243)
Claudio Vuerli (e-Mail: *vuerli@oat.ts.astro.it*, phone +39 - 40 - 3199213)

OAT: Osservatorio Astronomico di Trieste
Via G.B. Tiepolo 11
I - 34131 TRIESTE

1. INTRODUCTION

The National Galileo Telescope (TNG) is an Italian 3.5 m optical telescope, located at the Canary Islands. The main features of its Control System are: standard software design, portability and modularity (both at hardware and software levels), close telescope-instruments integration, fault tolerance, availability of remote control services.

Within the context of the REMOT project, a Teleoperation System will be implemented, used and interfaced to the TNG Control System (via a software linking layer), in order to show the feasibility of remote control of a scientific facility with a general approach (instead of having ad-hoc solutions for any remote control application).

The following sections will provide a brief description of the TNG Control System and introduce the design of the interfaces to be built between it and the Teleoperation System mentioned above. All this in order to be able to perform the integration of the two parts and look at the whole integrated system as a single scientific facility. It is expected that the remote user doesn't know how and where (in the TNG software) the integration is performed, nor does he/she know how the Teleoperation System works (i.e. the Teleoperation System integration and usage have to be transparent actions for the end user). The only things that the remote astronomer has to know are the TNG Control System features and the telescope characteristics.

2. GENERAL STRUCTURE OF THE TNG CONTROL SYSTEM

The TNG Control System design has been guided by three main considerations:

- the technology is subject to quick evolution; this implies the necessity of particular hardware and software solutions, chosen in order to guarantee an adequate support for the future;
- the lifetime of the Galileo project should be as longer as possible: this is why the TNG design will be aimed at taking into account the evolution of astronomical instruments and of astronomers' needs;
- the whole system should be managed in an easy way by all those people that will work on it to develop new instruments.

As a result, the TNG Control System has been designed, and is now being developed, using only stable standard systems, either hardware and software. A modular, flexible structure has been chosen for the overall architecture of the system, which has been maintained as homogeneous as possible, keeping the number of different components as low as allowed by current technology. Besides, a set of software interfaces has been built in order to hide all the low level details, allowing programmers to operate in an user-friendly environment.

Since the preliminary definition of the general specifications, a Control System essentially independent from the instruments, has been proposed. Hardware and software are based on recognized industry standard components, and the software is completely table driven.

The TNG Control System architecture can be described as based on a distributed network of VME crates and Workstations[1], with the software on the VMEs (*tasks*) managing device specific jobs, and the software running on the different Workstations (*processes*) monitoring/controlling the behaviour of the VMEs and providing the Remote User Interface[2] (see Figure 1).

[1] VMEs and Workstations can communicate on the telercope LAN through Ethernet using the TCP/IP protocol.
[2] This is the general situation foreseen in the context of the Galileo project. Within the REMOT project, the Remote User Interface will be provided as a part of the Teleoperation System by the Teleoperation System builders. The most desirable situation would be the case in which the Teleoperation System is designed in order to allow dynamic configuration of the Remote User Interface.

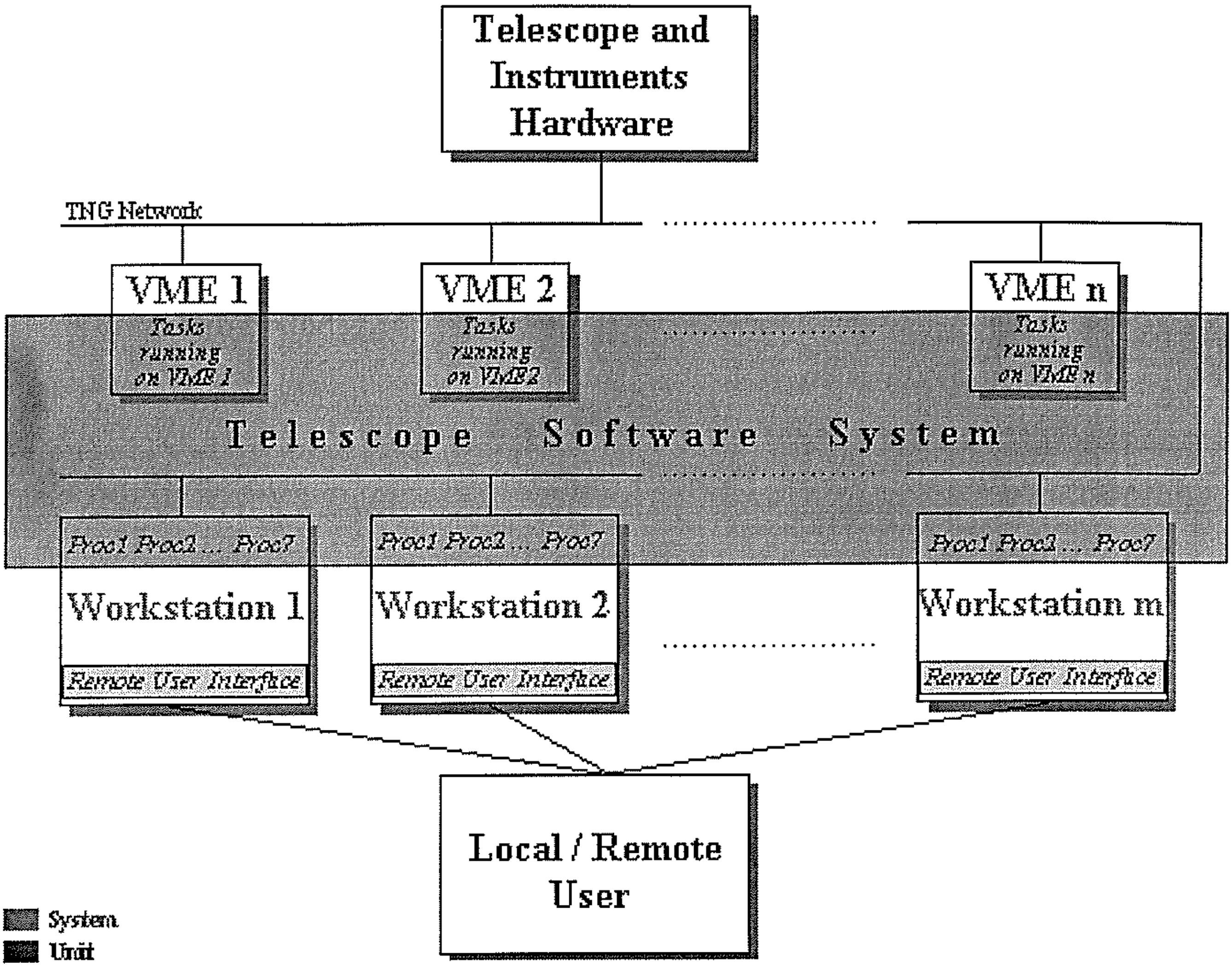

Figure 1: The TNG Control System

VMEs and Workstations are termed *systems*, tasks and processes are named *units*. Both systems and units need parameters (*items*), and are controlled through a set of commands (*items*). This is sketched in Table 1.

Every action on the telescope or on its instruments can be described in an abstract way (*naming convention*) by referencing a name of the type *<system>_<unit>_<item>*. This convention also applies to the names of the files containing the definitions needed by the software to operate correctly, and can be seen as a base point to manage remote access to the TNG Control System.

The conversion from symbolic names to physical addresses is made possible by means of a Telescope Data Base (TDB), which contains all the definitions concerning the components of the TNG Control

System, and is complemented by a library of routines which allow the processes running on the Workstations to access the data necessary to resolve the name-address translation.

AMBIENT NAME	DESCRIPTION IN THE VME AMBIENT	DESCRIPTION IN THE UNIX WORKSTATIONS AMBIENT
SYSTEM	Telescope Local Processors	Unix Workstations
UNIT	Task	Process
ITEM	Parameter/Command	Parameter/Command

Table 1: Naming Conventions Reference Table

2.1 Definition files

All the components of the software running on the Workstations are built as table driven modules: at startup each module will read its data from the on-line Telescope Data Base, from the on-disk definition files or directly from its own private files. These follow the naming convention just mentioned above. They are created by a set of off-line utilities and contain all the information (needed by the processes running on the Workstations) necessary to interact with the hardware of the Telescope and of the Instruments on one side and with the user on the other side.

The definition tables contained in these files are organized in a tree-like structure designed with the following characteristics (see Figure 2):

- at the **root level** the general structure of the TNG Control System is defined; here one file (*system.scf*, whose name is defined in the environment) is filled with all the definitions for the systems in the TNG network;
- at a **1-st level**, definitions of units (of each system) are provided; the names of the files (*<system>.ucf*) are derived from the names of the systems found in the previous file;
- at a **2-nd level**, definitions of the parameters used by each unit, and of the microcommands accepted by them, are given; the names of the files are built combining the name of each system with the name of its units found in ".ucf" files (this will involve having two kinds of names: *<system>_<unit>.pcf* for parameters and *<system>_<unit>.mccf* for microcommands).

Both at the 1-st and 2-nd levels, the definition files for the graphic and interaction panels may be included (*<system>_<unit>.pan* for interactive panels and *<system>_<unit>.drw* for interactive panels).

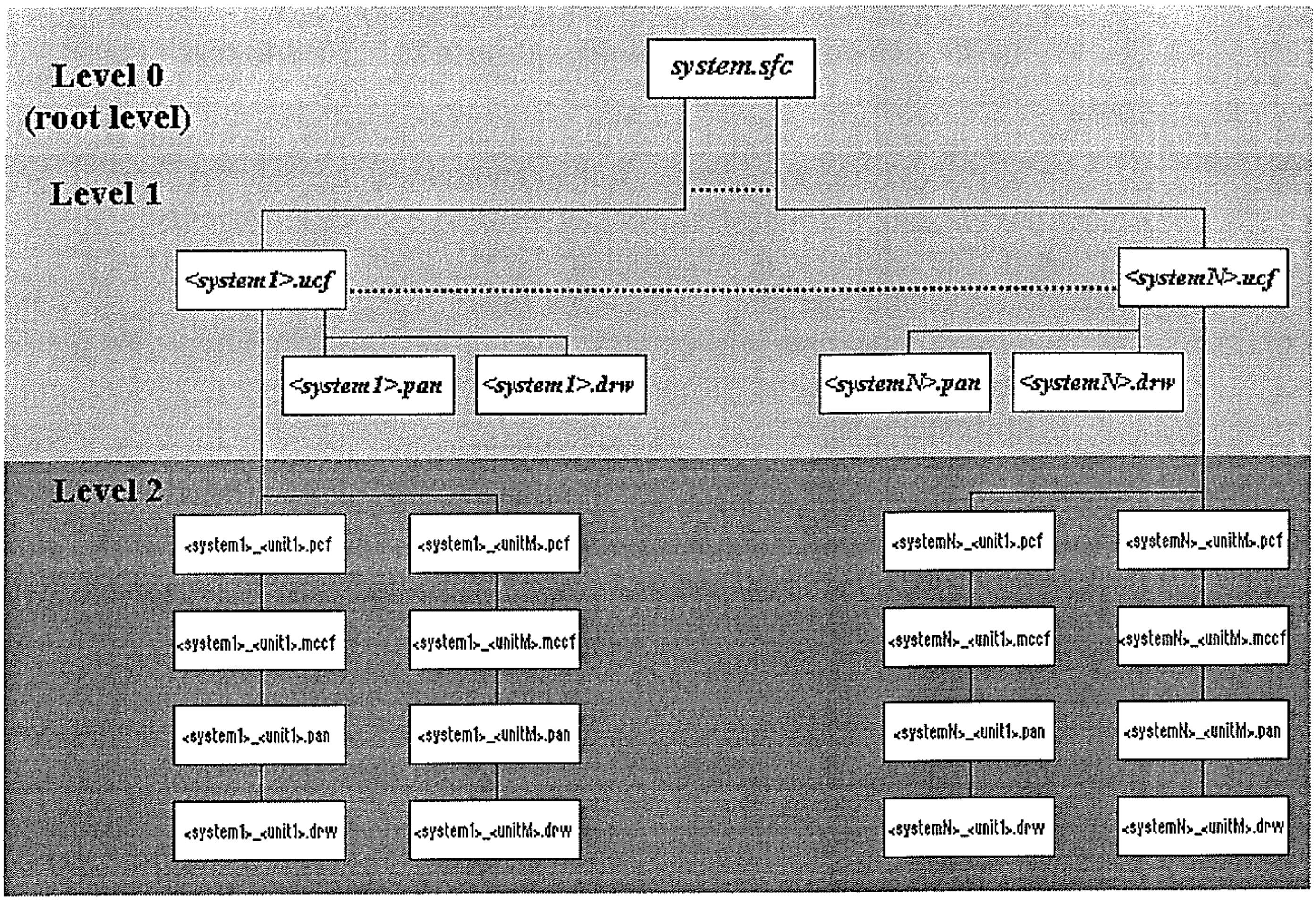

Figure 2:Organization of the configuration files and related tables

2.2 Communications

The mechanism for the communications is based on two UNIX tools:

- System V messages for communications between units residing on the same system;
- BSD sockets[3] for communications between units running on different systems.

[3] Sockets are realized using both the udp datagram protocol and the tcp internet protocol. In particular, the last one is able to manage communications in Wide Area Networks also, and allows remote control and observing as an extension of its usage.

2.2.1 Communication System on the Workstations

The basic concept here is that each unit can reach every other unit, depending only on the system configuration:

- the sender process running on a Workstation fills a message header providing it with all the information needed to reach the target unit;
- the header is analyzed by a library function which determines if the target of that message is either a unit located inside the same Workstation of the sender or if it resides on an outer system;
- if the address is recognized to be local to the sending Workstation, System V messages will be used;
- else the message will be forwarded to the WSCOMM process (see below), which takes care of TCP/IP connections between different systems.

2.2.2 Communication System on the VMEs

The scheme of the communications between VMEs and Workstations is slightly different from the inter-Workstations one: the protocol used is TCP/IP and the VMEs are always servers, while the Workstations are always clients.

2.3 The VME Environment

The VME environment directly governs and controls the telescope and all the devices attached to it. It is dedicated to telescope movements, optics, building driving and handling of instrumentation.

The hardware of the VME net is based on the following structure:

- communication bus based on VME standard;
- local CPU of advanced model (MC68030/68040);
- real time, multi-tasking operating system (PDOS);
- system memory with double port (VSB);
- data memory expandable in 4-8 Mbytes steps;
- two serial service ports (for debugging, diagnostics, ...);
- intelligent Ethernet interface with TCP/IP protocol support.

A monitoring process, named GATE (GAlileo Telescope Environment), performs the initialization of the VME ambient and the coordination of tasks running on the VMEs.

2.4 The UNIX Workstations' Environment (WSS)

The Workstation Software System (WSS) is built up by a set of seven processes, each dedicated to a unique task:

- WSINIT is responsible for the inizialization of the whole WSS;

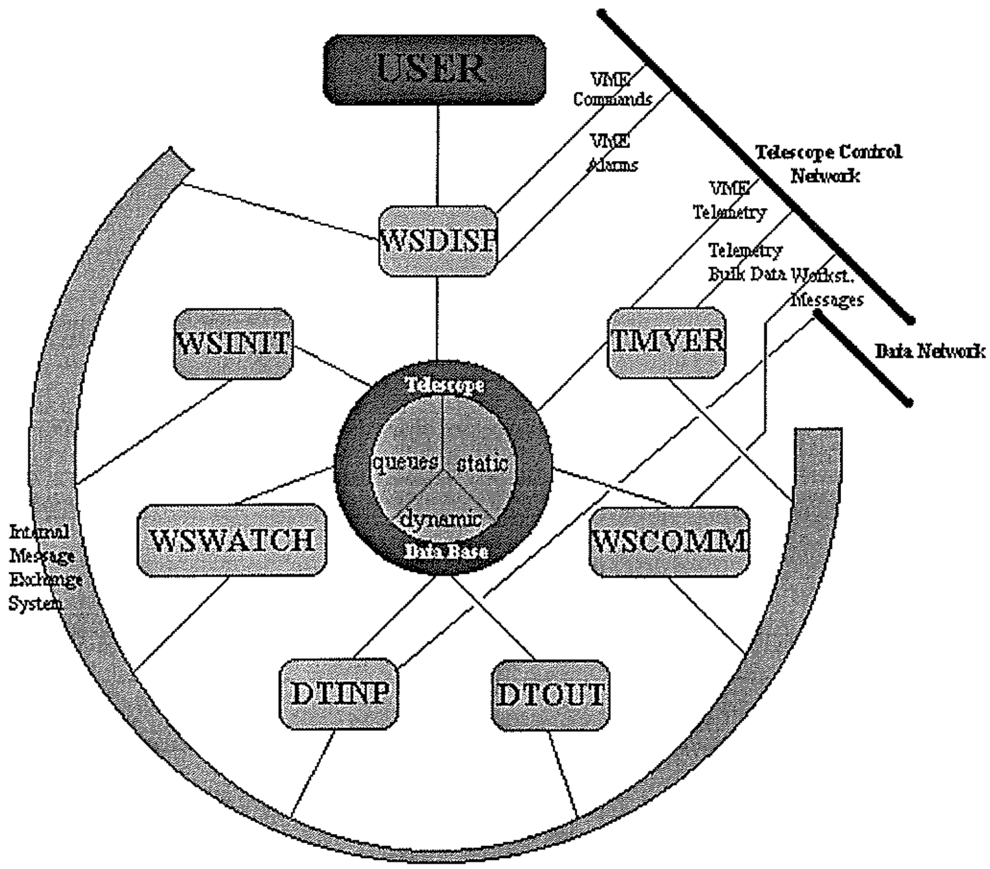

Figure 3: Workstation Software System Layout

- WSDISP has the responsibility of supervising all the interactions between the user and the TNG Control System;
- TMVER carries on the job of receiving, verifying and storing in the TDB all the telemetry data arriving from the VMEs;
- WSWATCH performs controls on submission and execution of commands sent to VME systems;
- WSCOMM manages the connections among the Workstations;
- DTINP, DTOUT handle the data from acquisition to storage on a temporary magnetic disk area on the instrument workstation.

The software structure of the WSS is exactly repeated in each Workstation of the TNG Control System. This includes the Telescope Data Base and the definition files replication. The different role of each Workstation is defined in a configuration file.

A special class of processes (Ancillary Processes, AP), which have no direct interaction with the external environment, is also provided, and is used to support the main processes.

The layout of this structure and the mutual interactions among processes are reported in Figure 3.

2.5 The Telescope Data Base (TDB)

The Telescope Data Base contains all the definitions concerning the components of the TNG Control System, and is complemented by a library of routines which allow WSS processes to access the data in it and to resolve the name-address translation. It is based on the UNIX shared memory capabilities, so that it can be accessed by all the processes of the WSS.

The Telescope Data Base can be seen as an instant snapshot of the complete TNG Control System. It is created by the WSINIT process during the TNG Control System startup phase and reads the definition files.

It is actually composed by three segments (see Figure 4 also):

- the *static segment* contains some housekeeping information followed by the static section of the Data Base: here all the access control structures are located, followed by the static contents of the definition files read from the disk;

- the *dynamic segment* constitutes the dynamic counterpart of the static segment, and contains changing status information, related to the items contained in it;

- the *internal data structures segment* is allocated for queues of commands, time schedule table of WSS processes, and telemetry log buffers.

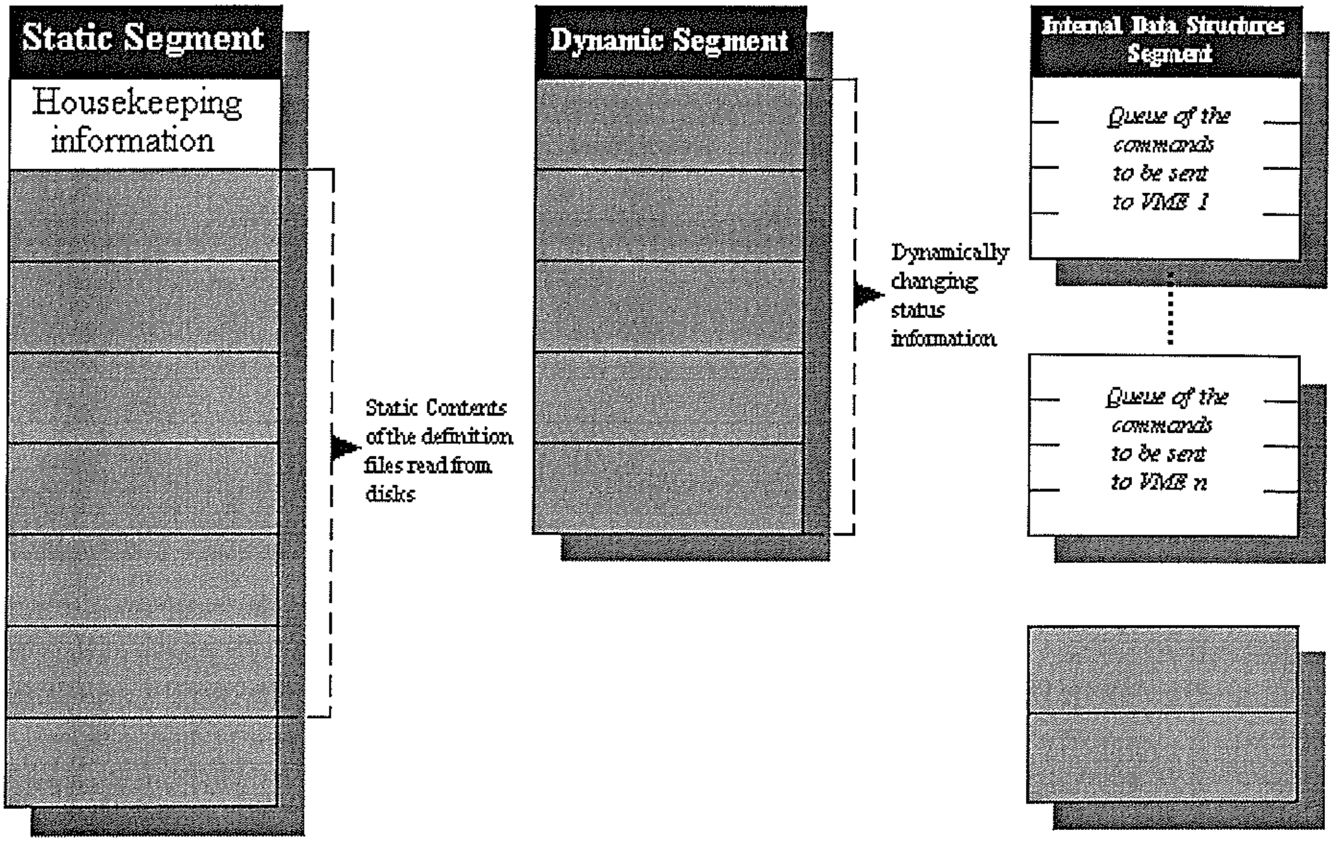

Figure 4: Telescope Data Base internal structure

2.6 Command Handling

Commands will be usually entered from the Workstation using the Remote User Interface. They originate both from an operator action or from a predefined table, are then submitted to the encoding and formatting chain and are finally sent to the instrument controller (VME). Here they undergo the same chain, but in reverse order, and are then transmitted to the direct instrument interface.

Two kinds of commands are foreseen: *immediate*, handled independently from the arrival time and performed as soon as they are received, and *delayed*, executed following the arrival order, if there is no immediate command queued for execution.

It is possible to store in memory a sequence of microcommands (called macrocommand), which can be retrieved and executed by a single microcommand.

All commands are parametrized into a table located in a memory area which can be accessed by all tasks; this table is sent to a VME from the Workstation at the beginning of the operations and contains all the parameters needed to completely describe each command.

2.7 Telemetry and Alarms

The telemetry transmitted by each VME should allow three basic functions:

- real time monitoring of the status of the telescope and of the instruments;
- command verification;
- alarms in case of failure or threshold limit overflow.

There are two basic kinds of parameters to be sent:

- status parameters (shutter on/off, dome open/close, ...);
- analogue parameters (temperatures, voltages, ...).

Telemetry is generated by an independent task allowing the receiving Workstation to read and use it in a completely asynchronous way with respect to command transmission, thus avoiding close loops between VMEs and Workstations.

Command verification is based on the following concepts:

- the actual command verification will be performed at the Workstations' level;
- VMEs will provide the Workstations with all the information needed to correctly perform the verification (time and possible encountered errors both for dispatched and executed commands);
- a dedicated task has the duty of cyclically monitor the status of all parameters controlled by each VME; as soon as it finds out that a parameter exceeds its threshold, it sets an alarm flag and notifies the abnormal situation to the Workstation.

2.8 TNG Definition Tables and Files Mainteinance

Inserting systems into the WSS can be considered a quite simple task, provided that characteristics and behaviour of systems, as well as interactions of the WSS with them, have carefully been planned and defined. When a new system has been inserted, it is visible at the first time that the TNG Control System will be rebooted.

3. PROVIDING A SOFTWARE INTERFACE BETWEEN THE TNG CONTROL SYSTEM AND THE TELEOPERATION SYSTEM ARCHITECTURE

This section will describe the integration needed to interface the Control System of the TNG with the software provided by the TCP Systemas e Ingenieria, which will realize the Teleoperation System for the generalized remote control of scientific instrumentation.

As a result, the TNG should be operated remotely.

Two purposes will be made, in order to reflect the complete structure of a remote observing experiment:

- a first, static and limited configuration will be defined, in order to describe the interfaces needed to implement the Demonstrator required in the REMOT project;
- a further elaborated, dynamic configuration will be presented, with the aim of illustrating the real case situation.

The general configuration from which both the two purposes take their structure can be described as follows (see Figure 5 also):

- the TNG environment is made up of the Telescope Control System [1], and the configuration tables describing Commands [2] and Parameters [3] which will be used in order to have the whole system work;
- moreover, there are Panels [4], used to show to the user the status and the results of his/her Observing Session; these are the tools used to build the Local User Interface;
- this is all controlled by means of a set of Rules [5], which, depending on the Commands/Parameters provided to the system, will allow to perform the related actions or display the corresponding panels;
- the TNG environment provides the possibility to interact with the telescope via the User Interface, which is used to send commands (chosen by the user) to the Control System;

- the whole Local Control System of the TNG should thus be thought as composed by parts [1] to [5];
- the integration of the TNG software with the Teleoperation System one, will be planned by providing an Ancillary Process [6] which will take care of "translating" the Commands/Parameters provided by the Teleoperation System into Commands/Parameters of the TNG environment: when a command of the Teleoperation System will reach the Ancillary Process, this one will translate it to the corresponding set of commands for the TNG and then pass it to the rules controller [5]; on the other hand, the results of this control phase will be messages or commands (sent to the Ancillary Process, which, in turn, will provide them to the Teleoperation System in a suitable format) whose destination will be the Remote User Interface [10]; this one will show to the user the results of the commands he/she has submitted;
- with this kind of design, the whole REMOT system should be seen, by the TNG Control System, as if it was one Local Process (shown at the left hand of part [5] in Figure 5) allowed to use the Telescope;
- parts [7],[8],[9] will be part and responsibility of the Teleoperation System builders, so they won't be described from the OAT point of view;
- important to note is part [11], where all the Commands, Parameters and Rules will be described; it will assume different meanings (and will be designed in a different way) depending on the type of user interface to be built: in the case of the REMOT project, being the Remote User Interface static, all the information will be cabled into the interface itself; if the interface should be built in a dynamic way, then [11] should be provided as a set of tables, (as is the case of [2],[3],[5]), describing all the modules that the Remote User Interface should be able to manage; in this situation, part [11] should exactly correspond to parts [2],[3],[4],[5], in order to completely reflect the structure of the TNG Remote User Interface and its behaviour within the context of telescope remote control.

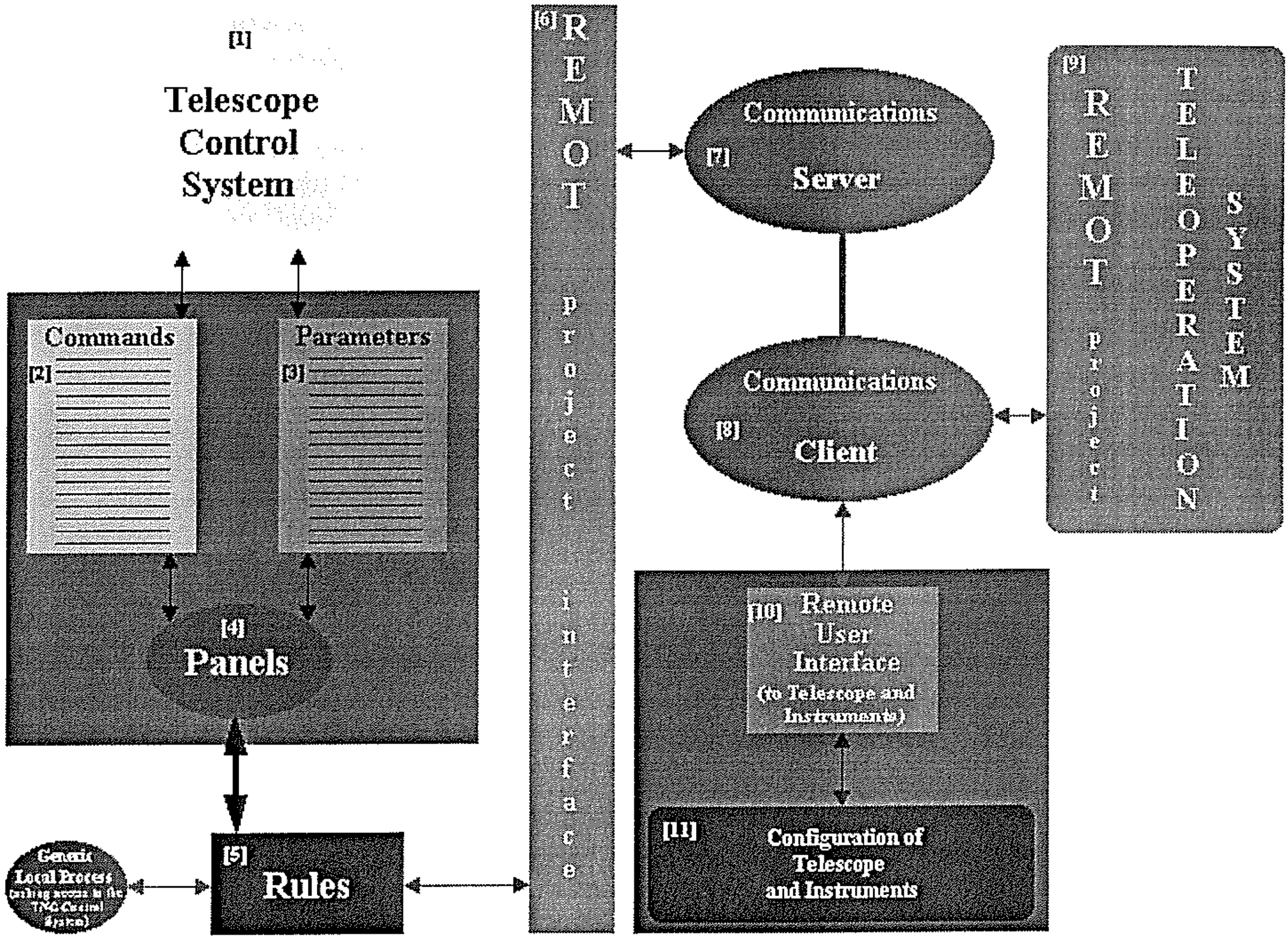

Figure 5: TNG to Teleoperation System software interface design

3.1 Interface of the TNG Control System Software Towards the Generic Teleoperation System (AP): Commands and Parameters

As previously stated, if the Remote User Interface to be built is considered to be static (as it will be the case in the REMOT project), fixed, with a few panels (see reference in document D6.2: Experimental Setting for the Demonstration of the Teleoperation System), and is used to perform a limited set of actions on the telescope (just like, moving the Dome and the Telescope, pointing the Telescope, ...), it is acceptable to think at it as a cabled software, which, by exactly knowing the configuration of the Instruments to be controlled, will provide to the user a sufficient set of windows, necessary to let him/her know what is happening to the system while performing the experiment and what are the results of the

"Observation" (and, obviously, letting him/her choose the command to be submitted and to be sent to the Telescope Control System).

Being the acquisition of an image a very complex action, even if desirable for the demonstration phase, it is not assured that this test will be covered in the REMOT project: this will mostly depend on the TNG status at the moment of the Demo.

The way of working in the REMOT project will be to implement an Ancillary Process which will translate Teleoperation System commands into the corresponding TNG Control System commands, and will check if these commands are allowed to be submitted to the TNG Control System at some particular conditions. If the control phase has a positive response, the request will be sent to the Telescope Control System; otherwise a message will be sent back to the Teleoperation System (in a suitable format), and displayed to the user via the Remote User Interface.

3.2 Dynamic Configuration of the TNG Control System (Via Configuration Tables) and Rules to be Coded

If the system to be remotely controlled has to be considered as a "complete" teleoperation experiment, then the Remote User Interface should be "the same" as the Local User Interface for any single experimental set: every owner of a complex system will like to access to it in the same way he/she is used to do when he/she logs-in in his/her particular local system.

This means that, in order to have a general purpose Teleoperation System, this one should be provided with the skill of dynamic configuration of the Remote User Interface.

From the TNG point of view this means that the Teleoperation System should be able to load the TNG configuration files describing the instruments and the telescope and to dynamically configure the Remote User Interface in the same way as it happens at the local side.

This implies the ability of performing the control of the "set of rules" depending on the sets of Commands and Parameters being submitted to the Local Control System via the Teleoperation System.

Obviously this way of working is outside the scope of REMOT, and would imply an additional effort to be made, but could be useful if the Teleoperation of scientific instruments should be made in a complete way.

4. CONCLUSIONS

In this document the Local Control System of the TNG has been presented, and the purpose of interfacing it to the Teleoperation System, being built within the context of the REMOT project, has been analyzed.

Two points con be considered of interest, once it has been agreed that the software interface design is to be considered as an additional process to be added to the TNG Control System software (it seems no reasonable to break the TNG Control System down to add new software inside its complex structure):

- the Remote User Interface has to be provided by the Teleoperation System builders:
    - for a first-test phase (REMOT demostrator) a "static" Remote User Interface can be considered sufficient;
    - for a complete teleoperation experiment, a dynamic, table driven Remote User Interface, has to be provided;
- the software interface between the REMOT system and the TNG Control System will be a "translator of commands" which the TNG Control System will see as an additional process of its set of software tools.

5. BIBLIOGRAPHY

[1] *The Galileo Project. Workstation Software System and User Interface.* A. Balestra, P. Marcucci, M. Pucillo, C. Vuerli.

[2] *GALILEO Project. Workstation Software System Architecture Design Document.* TNG Technical Report n.9, 1991 (Publ. Osservatorio Astronomico di Trieste n.1428). A. Balestra, P. Marcucci, F. Pasian, M. Pucillo, R. Smareglia, C. Vuerli.

[3] *A distributed VME telescope control system for remote operations.* TNG Technical Report n.14, june 1992. C. Bonoli, D. Fantinel, A. Baruffolo, F. Bortoletto.

[4] *The Galileo Table Editor.* TNG Technical Report n.15, october 1992 (Publ. Osservatorio Astronomico di Trieste n.1450). P. Marcucci, M. Pucillo.

[5] *The Galileo Message Exchange System.*TNG Technical Report n.17, december 1992 (Publ. Osservatorio Astronomico di Trieste n.1457). A. Balestra.

APPENDIX A: DEFINITION TABLES AT THE TNG

Definition table for TNG systems

nodenum	long	main code assigned by Table Editor
nodename	char[40]	full name
acronym	char[24]	acronym; used by the system to compose the name of an item in the Tdb - MANDATORY
dbcode	long	Tdb internal code, assigned by software
arpa_node	char[16]	full Internet address-MANDATORY
byte_sex	long	TRUE if the system supports the little endian byte ordering
send_data	long	TRUE if the system sends scientific data besides telemetry (e.g. a TLP connected to an instrument)
tm_period	long	base period for the telemetry; must be defined following the kind of data sent, must be a multiple of one second (TLP only)
ncode	long	number of code files to be sent to the system at boot-strap (TLP only)
type	char[8]	acronym of the WS to which the TLP is connected; for the WS it is their own acronym
protectionl	long	protection level, not implemented yet
havepanel	long	TRUE if the system has a dedicated control panel
firstqueuelong		pointer to the command queue in the Tdb, assigned by the software
nscreens	long	number of monitor screens connected to the system

Definition table for TNG units

unitnum	long	main code, assigned by Table Editor
unitname	char[40]	description
acronym	char[24]	acronym; used by system to compose the name of Tdb items; must be used together to the name of the system to which the unit belongs to access to it - MANDATORY
name	char[40]	full pathname and possible options for the process associated to the unit in the workstation; for workstations only, must be empty for TLP
dbcode	long	Tdb internal code, assigned by software
protectionlong		protection level; not implemented yet
havepanel	long	TRUE if the unit has a dedicated interactive panel
ancillary	long	TRUE if the process defined in name is an ancillary process
waitsignallong		TRUE if the process defined in name must wait for a start signal from INIT

Definition table for TNG parameters

tag	long	main code, assigned by Table Editor
name	char[24]	full name
acronym	char[6]	acronym; used by system to compose the complete acronym of an element; is used together with the names of the system

		and the unit to which the parameter belongs to access the parameter itself in the Tdb - MANDATORY
descr	char[40]	description
vmecode	long	TLP internal code - MANDATORY if the parameter is referred to a TLP
dbcode	long	Tdb internal code, assigned by the software
type	long	type; not implemented yet
format	char[4]	format, can assume the following values [f|d|s[nn]] ■ f floating point ■ d integer ■ s byte ■ nn gives the number of elements if the parameter is an array
access	char[4]	access mode; can assume the following values: RO=read-only, WR=write-only, RW=read-write
decpoints	long	number of decimal places; used for the display
class	long	class, assigned by the software; reserved for internal use
vme_only	long	TRUE if the parameter is internal to the TLP; parameters with this flag set to TRUE are reserved to the internal functioning of the TLP and are not involved in the telemetry operations
tm_flag	long	TRUE if the parameter must be sent with the telemetry; parameters with this flag set to TRUE at boot-strap time are considered stable components of telemetry operations and cannot be removed from the telemetry; it is anyway possible to add and remove other parameters to/from the telemetry list at run-time
convert	long	TRUE if the value of the parameter must be converted to physical units
coeff	double[5]	polynomial coefficients for the conversion from engineering units to physical units; The polynome used is of the type: ax4+ bx3 + cx2 + dx + e
def_value	double	default value; used by WSS to assign a default value to parameters of variable type; used by WS only
check_limits	long	TRUE if the value of the parameter must be verified against the limits of attention and alarm
intr_low_limit	long	lower limit in engineering units
intr_high_limit	long	upper limit in engineering units
low-alarm_thr	double	lower ALARM limit, in physical units
high-alarm_thr	double	upper ALARM limit, in physical units
low-attn_thr	double	lower ATTENTION limit, in physical units
high-attn_thr	double	upper ATTENTION limit, in physical units
phy_unit	char[12]	physical units
tm_rate	long	period for telemetry, computed in units of tm_period as defined for the system to which the parameter belongs;

Definition table for TNG microcommands

opcode	long	main code, assigned by Table Editor
name	char[24]	full name
vme	char[24]	acronym of the destination system (TLP or WS) - MANDATORY
descr	char[40]	description

acronym	char[24]	acronym - MANDATORY
vmecode	long	private TLP code, as defined in the destination system
dbcode	long	Tdb internal code, assigned by software
queue	long	TRUE if the destination queue is the immediate one
waitflag	long	TRUE if the microcommand must be completed before successive microcommands are executed
task	char[24]	acronym of the destination unit
exec_verify	long	TRUE if the microcommand submission for execution must be reported through telemetry
compl_verify	long	TRUE if the microcommand execution must be reported through telemetry
min_exec_time	long	minimum execution time
max_exec_time	long	maximum execution time (must be referred to the value of tm_period)
counter	long	number of operands (max. 5)
convert	long	TRUE if operands must be converted to engineering units
coeff	double[5,5]	table containing the polynomial conversion coefficients
opdescr	char[5,40]	operands description
optype	char[5,4]	operands type: f=float, d=long, sn=char[n]
min_value	double[5]	minimum value accepted for operands
max_value	double[5]	maximum value accepted for operands
def_value	double[5]	default value for operands
verify_flag	long	TRUE if the microcommand execution must be verified through a telemetry parameter
tm	char[24]	acronym of the parameter to be used for the microcommand verification (current value versus preset value)
tolerance	long	tolerance in thousandths allowed in the verification

Definition table for TNG messages

id	long	main code, assigned by Table Editor
acronym	char[24]	message acronym - MANDATORY
dbcode	long	Tdb internal code, assigned by software
typ	long	message type, allowed values are: 0=system, 1=info, 2=warning, 3=alarm
descr	char[80]	message description
txt	char[80]	message text, can contain any kind of data transferred with a casting operation; sender and receiver tasks must obviously agree on the content and meaning of the message

Definition table for TNG color palettes

id	long	main code, assigned by Table Editor
acronym	char[24]	acronym of the palette - MANDATORY
dbcode	long	Tdb internal code, assigned by software
descr	char[80]	palette description
value	char[16,40]	table containing palette color names (max. 16)

Definition table for TNG interactive panels

id	long	main code, assigned by Table Editor
acronym	char[24]	panel acronym - MANDATORY
dbcode	long	Tdb internal code, assigned by software
screen	long	default screen (0-2), if the system has less screens, the panel is opened on screen 0
descr	char[80]	panel description
fname	char[80]	name of the file containing the panel definition (see the appropriate table)

Definition table for TNG interactive panels items

id	long	main code, assigned by Graphic Editor
acronym	char[24]	item acronym - MANDATORY
group	char[8]	codes (up to 7) identifying the groups to which the item belongs; group codes are single alphabetic characters, and can be used for simultaneous operations on more items
type	long	item type, assigned by Graphic Editor; ranges between 1 and 15 (see Table 1)
color	long	item color, ranges between 0 and 23 (see Table 2)
style	long	item style; allowed values are: 0=solid, 1=dashed
thickness	long	item thickness; ranges between 0 and 7
x1	long	x coordinate of item
y1	long	y coordinate of item
x2	long	width/radius
y2	long	heigth/radius
font	long	font for labels, ranges between 0 and 9 (see Table 3)
mode	long	operating mode for STATUS type items; values range from 0 to 4 (see Table 4)
dbcode	long	Tdb internal code, assigned by software
sensible	long	TRUE if the item is sensible to operator's actions; can be used at run-time to enable or disable items or groups of items (see group)
threshold	double	threshold level at which items of type STATUS change
text	char[40]	text string to be shown in the items of type TEXT
pcf	char[28]	acronym of the Tdb parameter to be shown or to be used to compute the trasformation of a dynamic item; the acronym may end with an optional character string with the following format: [/[S|E|C][nn]] ■ /S means that the operator defined value must be shown ■ /E means that the engineering value must be shown ■ /C means that the current value must be shown ■ no option means that the current value of element 0 must be shown ■ nn is the index of the element to be shown in case of items of type array
mccf	char[40]	acronym, followed by operands, of the command to be activated following an action on the item; applies to input items only; operands can be acronyms of other Tdb parameters, which will be substituted with their current value
stat	char[2,40]	name of two color elements, or two

		bitmaps, or two character strings to be assigned to the two stati of an item of type STATUS; threshold is the value at which the item switches from one status to the other
dynamic	long	TRUE if the item is a dynamic one; a dynamic iem can undergo a transformation of its position, rotation angle or color following the value assumed by the parameter specified in pcf; the algorithm makes use of the minimum and maximum values allowed for the parameter, as stored in Tdb, and of the range defined below in this table to scale the output values correctly; if range=0 (or ctname=0) then no transformation takes place
xr	long	x coordinate variation range
yr	long	y coordinate variation range
angle	double	initial reference angle
angler	double	rotation angle variation range
xrot	long	x coordinate of rotation center
yrot	long	y coordinate of rotation center
ctname	char[24]	acronym of the color palette to be used for the transformation
ctmin	long	first color element to be used in the color palette
ctmax	long	last color element to be used in the color palette
radiogrouplong		index of the group for mutually exclusive buttons; it is equivalent to the concept of radiobutton

Table 1 : interactive panel graphical elements

N.	Name	I/O	Text	Dyn	Grph	Description
1	LINE	O	n	y	y	line segment
2	RECTANGLE	O	n	y	y	rectangle
3	BOX	O	n	y	y	filled rectangle
4	CIRCLE	O	n	y	y	circle
5	FILLCIRCLE	O	n	y	y	filled circle
6	LABEL	-	y	n	n	label
7	OUTPUT	O	y	y	n	output text field
8	SLIDER	I	n	n	n	slider with apply button
9	LEDBAR	O	n	n	n	color led bar
10	BUTTON	I	n	n	n	press button
11	STATUS	O	y	n	n	two status output field (color, text or bitmap)
12	ANALOG	O	n	n	n	analogue gauge
13	SETPAR	I	y	n	n	editable text field
14	CLOCK	O	n	n	n	bar indicator with number (scalable)
15	CHECKBUTT	I	n	n	n	check button (on/off)

I/O – shows Input and Output items;
Text – y means that the element contains text;
Dyn – y means that the element is a dynamic one;
Grph – y means that the element supports graphic transformations, besides color transformations

Table 2 : standard colors

Code	Color
0	black
1	blue
2	green
3	cyan
4	red
5	magenta
6	yellow
7	white
8	aquamarine
9	light steel blue
10	sky blue
11	coral
12	salmon
13	tan
14	sea green
15	spring green
16	yellow green
17	light gray
18	gray
19	khaki
20	medium goldenrod
21	gold
22	turquoise
23	violet red

Table 3 : character fonts

Code	Font
0	tng_standard
1	tng_fixed
2	tng_fixed_bold
3	tng_fixed_large
4	tng_large
5	tng_large_bold
6	tng_large_italic
7	tng_large_extra
8	tng_symbol
9	tng_symbol_large

Table 4 : operating modes for elements of type STATUS

Code	Mode	Description
0	lamp	bicolor led, off=color[0], on=color[1]
1	text	off=tex[0] on dark background, on=text[1] on green background
2	fault	off=tex[0] on dark background, on=text[1] on red background
3	bitmap	off=bitmap[0], on=bitmap[1]

APPENDIX B: GRAPHICAL INTERFACES, COMMANDS AND PARAMETERS FOR THE TNG IN THE REMOT PROJECT

Within the context of this appendix we'll briefly describe the Graphical Interfaces that the remote user expects to see when connected to the TNG Control System via the remote connection between the Teleoperation System (to be built in the REMOT project) and the Ancillary Process (which will be provided as a communication tool in the Galileo System).

In order to fully specify any action to be done, the behaviour of any button of the various windows of the Remote User Interface will be described and the characteristic commands to be sent to the TNG Control System and the parameters to be captured by the Remote User Interface will be defined.

1. GRAPHICAL WINDOWS FOR REMOTE USERS

In this section all the graphical windows needed to carry on the Demo Pilot demonstration, will be described, with their functionalities.

One thing to note is that all the windows designed to report information to the user (such as out of range values, login error information, file names requests, action confirmation requests, or system problems information reporting), must be blocking, in the sense that no other window can be superimposed or used, until the OK button of the former ones will be pressed.

Moreover, the terms "hidden" and "visible" will refer the fact that a button on a window is respectively shown as faded or in a normal form.

The BYE button of all the graphical windows will be used to close them.

1.1 System Access Window

The first kind of window that a user expects to see when asking remote access to the TNG Control System must be the System Access Window, as described in document D6.2.

Within the context of the Demo Pilot for the REMOT project, we think that it is sufficient to have the possibility of displaying the information related to the current observation to the Primary and Secondary Users: the "System Administration" and "Other" menu voices can be considered as an additional check that, for the moment, are outside the scope of the project.

We think that the authorisation of the user login request can only come from the TNG Access Control System, so the only data to be exchanged between the System Access Window and the Ancillary Process (in this context) will be the strings that the Remote User will write on the corresponding fields of the System Access Window itself, and the eventual authorisation to use the Galileo system (or a failure message, which can be displayed in a separate window that the user has to read - see figure 2 below -).

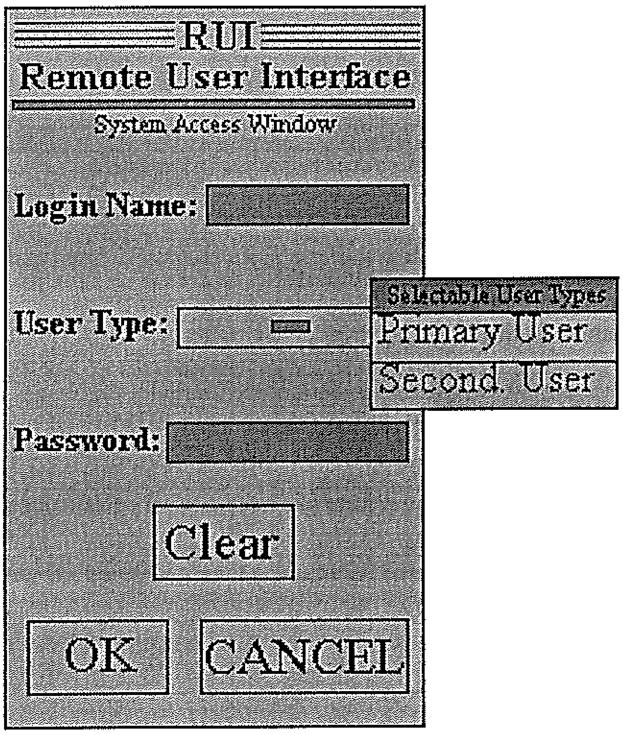

Figure 1: System Access Window

The main fields of the System Access Window can be described with the following:

- <u>Login Name</u> is, obviously, a string that is used to recognise the name with which the user is known to the system; after this string has been written in the corresponding System Access

Window field, it will be responsibility of the Galileo Access Control System to verify the rights of the user (once it has received this kind of information from the Teleoperation System);

- *User Type*: this has to be a selectable list of possible users; as previously stated, we are satisfied enough if the "Primary" and "Secondary" Users are allowed to operate;

- *Password*: this has to be written by the user and hidden to everyone; it has to be sent to the Galileo Access Control System, which will verify the rights of the user.

If a user is not authorised to access the TNG System, a graphical window with the related error message will appear (see figure 2).

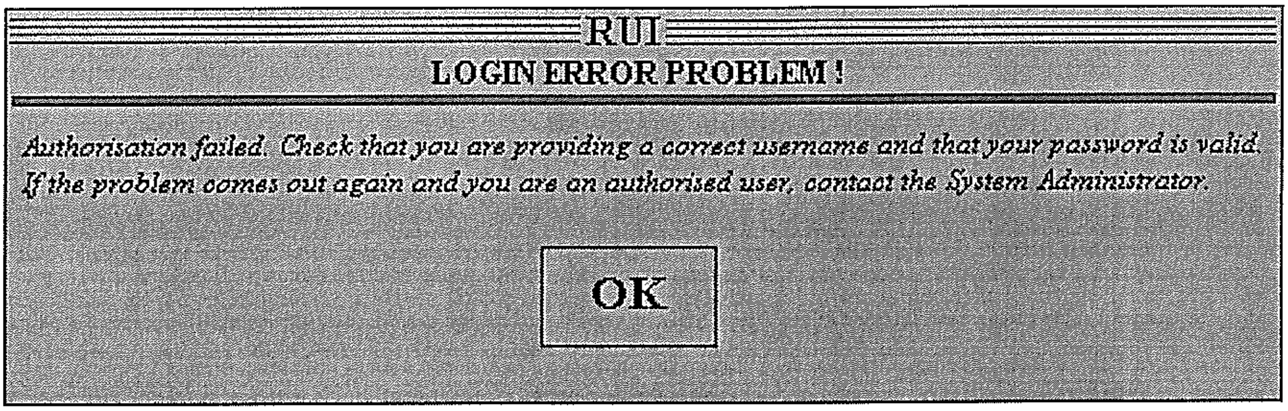

Figure 2: Login Error Window

One thing has to be said about the "User Type" functionality: depending on the kind of user (when authorised), the other windows of the Remote User Interface that will appear when the user is allowed to use the Galileo System, will behave in two different ways:

- if the user is recognised as a Primary User, all the buttons and functionalities will be made available to him/her, depending on the particular status of the Galileo System;
- if the user is a Secondary User, he/she will only be allowed to monitor the "observation" (performed by a Primary User), but all the buttons and functionalities of the various Windows of the Remote User Interface will be disabled to him/her; some exceptions to this behaviour will be the following:

* it will be possible to display the image resulting from an observation (when available);
* the user will be allowed to push the buttons designed to let him/her close the windows he/she is using, or to log out from the Galileo System;
* it will be possible to choose the subsystem to monitor, immediately after having been logged in, or during any moment of the observation if that subsystem is not actually being monitored.

The "OK" and "CLEAR" buttons will be used to confirm-and-send the data and clear any choice on the System Access Window respectively. The CANCEL button is used to exit the Window, in case the user does not intend to log into the System.

After the user has been logged in, the System Access Window must disappear.

1.2 Available Systems Window

When the user has been authorised to access the TNG System, the Available Systems Window will be displayed.

This window can be used by both the Primary and Secondary Users and will be closed only by pressing the LOGOUT button, which, in addition, will log the user out of the TNG System.

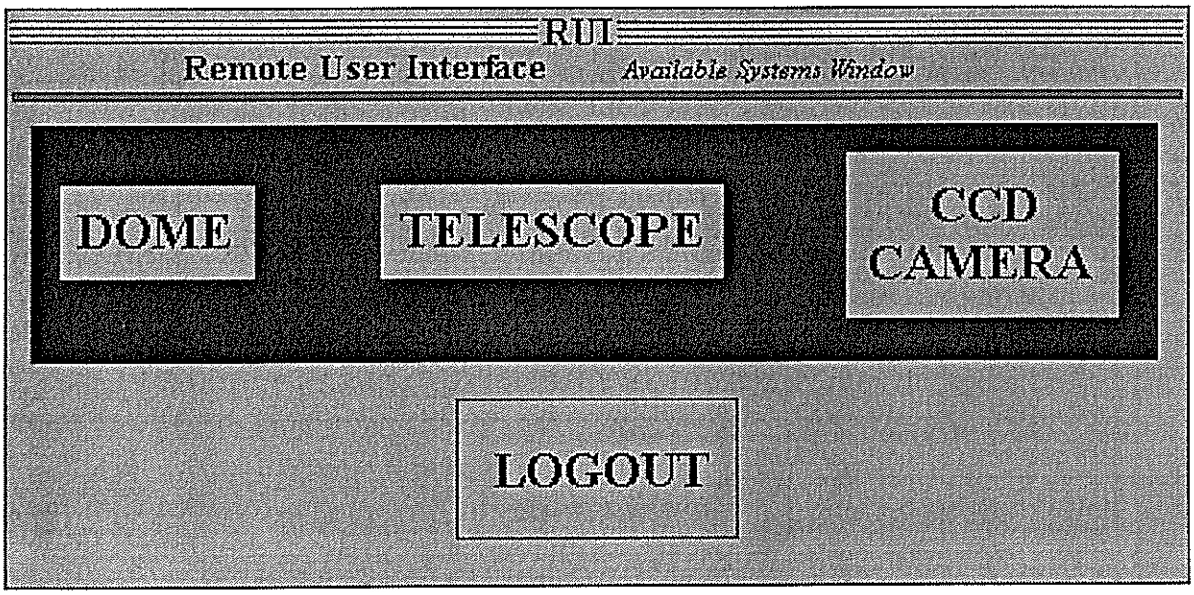

Figure 3: Available Systems Window

Any subsystem will be "accessed" by pressing the button marked with its name. As a consequence of this action (pressing the button), the window related to the requested subsystem will be displayed.

All the buttons for which the corresponding subsystem has an open window must be hidden and disabled (this implies that only one window per subsystem can be activated in a user session).

Furthermore, the LOGOUT button has to be active only after all the windows related to the various subsystems (Dome, Telescope, CCD Camera) have been closed. During the rest of the time it must be hidden and disabled.

This window must be iconifiable.

1.3 Dome Control Window

This window will be displayed by pressing the DOME button of the Available Systems Window.

The only actions that a user can perform on it are:

- *open the telescope doors*: the (Primary) user presses the OPEN DOORS button and the related command will be sent to the TNG Control System;

- *close the telescope doors*: the (Primary) user presses the CLOSE DOORS button and the related command will be sent to the TNG Control System;

- *exit the window*: this action is obtained by pressing the BYE button (both in the cases of Primary and Secondary Users).

The OPEN DOORS and CLOSE DOORS buttons must be mutually exclusive for the Primary User, in the sense that when one of them is visible and enabled the other one must be hidden and disabled.

In the case of the Secondary User both these buttons must be hidden and disabled.

In both cases (Primary and Secondary Users) the information on the Doors Opening and Closing Status will be displayed on the "Doors opening/closing Display" and on the "Doors Current Status" display. On the second one a green light will be switched on in agreement with the fulfilling of a user request (i.e. when a Primary User asks to open the doors, the green light, in correspondence to the "open" voice, will be shown when the opening of the doors has been completed, and the red light will be shown over the "closed" voice; vice-versa, the two lights will be inverted when a Primary User asks to close the doors and the operation has been completed).

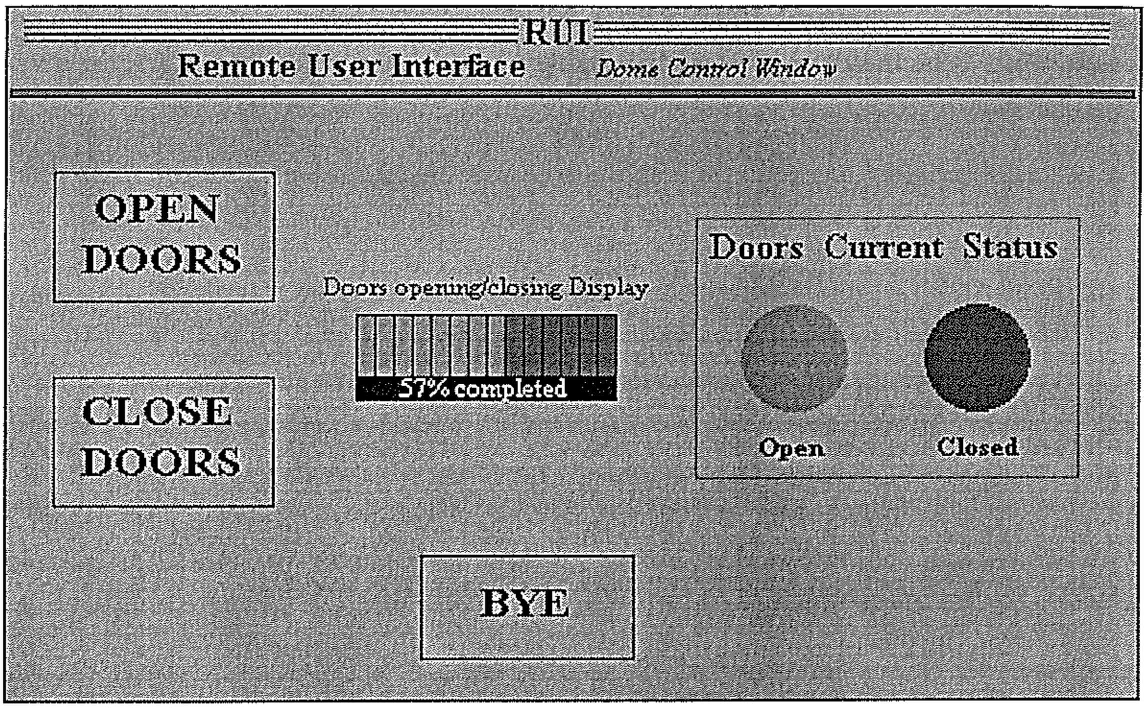

Figure 4: Dome Control Window

The BYE button must be hidden and disabled (for the Primary User) when the doors are open (this information is shown on the "Doors Current Status" display), and will be made accessible only if the doors are closed (in the case of the Secondary User, this button will be the only button always visible and enabled, and the user can push on it in any moment of his/her monitoring session).

The Dome Control Window must be iconifiable.

1.4 Telescope Pointing Window

This window will be opened as a consequence of pushing the TELESCOPE button on the Available Systems Window.

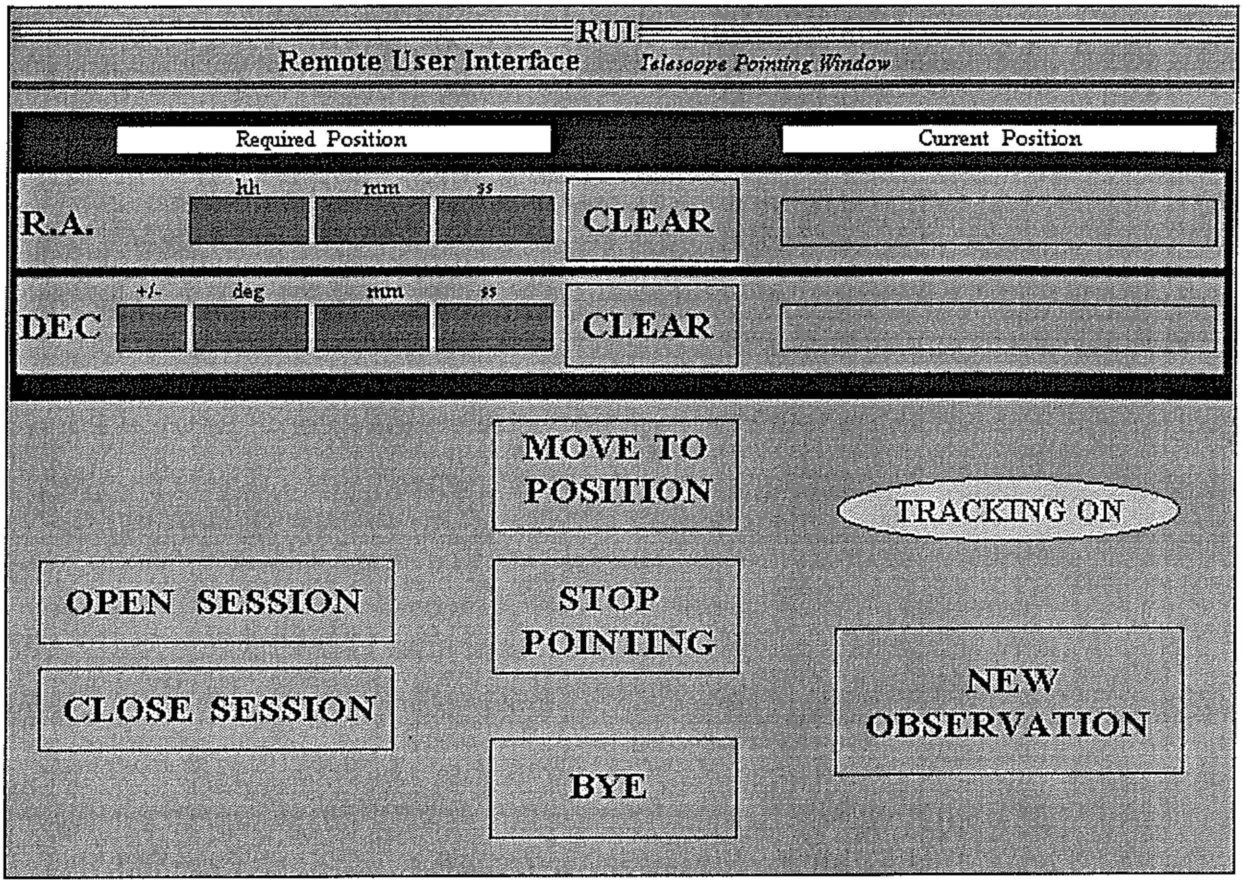

Figure 5: Telescope Pointing Window

It can be thought as logically split in three parts (from a Primary User point of view):

a) *writable fields*: here the (Primary) user can insert the data[1] related to the parameters which will characterise his/her observation (after the user has inserted these values, a control must be made that they are in the correct ranges, and, in case they are out of range, the related error window must be shown - see figure 7 below -):
   - R.A. refers to the Right Ascension data information. This field is divided in three subfields, which respectively refer to values of hh, mm, ss and must be in the following ranges:
     - * hh: [00, 23]
     - * mm: [00, 59]
     - * ss: [00, 59]

[1] Default values for these fields must be provided as 00:00:00 in both R.A. and DEC cases.

- DEC refers to the Declination. This field is divided in four subfields, which respectively refer to values of sign, deg, mm, ss and must be in the following ranges (the satisfaction of the ranges here must include the control that angles are no greater than 90:00:00):
  - * +/-: [+, -]
  - * deg: [00, 90]
  - * mm: [00, 59]
  - * ss: [00, 59]

b) *read-only fields*: reading these fields the (Primary and Secondary) user can understand the current status of the Telescope (its position and the fact that it is tracking or not). In fact under the "Current Position" voice, the Telescope Pointing Window will show, every 2 seconds, the current values of the R.A. and DEC. When the telescope is tracking the TRACKING ON light will be displayed;

c) *buttons*: all the buttons (except the BYE one, which is visible and enabled to the Secondary User too) will be made visible and enabled only to the Primary User. The CLEAR buttons will be used to clean the R.A. and DEC values that the (Primary) user is introducing. The other buttons have the following functionalities:

- OPEN SESSION: when the (Primary) user is accessing a new observing session this must be the only visible and enabled button and the writable fields must be disabled. When the Primary User pushes the button, all the Dome Control Window buttons will be hidden and disabled, and the mirror covers of the Telescope will be opened. After the session has been opened by pushing the button, the button itself must be hidden and disabled and the MOVE TO POSITION and CLOSE SESSION buttons will be displayed and enabled together with the writable (R.A. and DEC) fields;
- CLOSE SESSION: when the Primary User pushes this button the mirror covers of the Telescope will be closed and the buttons of the Dome Control Window will be made visible and enabled. This button is mutually exclusive with the OPEN SESSION button,

but is always visible and enabled in all the other cases. When the user pushes this button, a confirmation window (see figure 6) will be displayed and the closing of the session will be operated only when the user pushes its YES button;

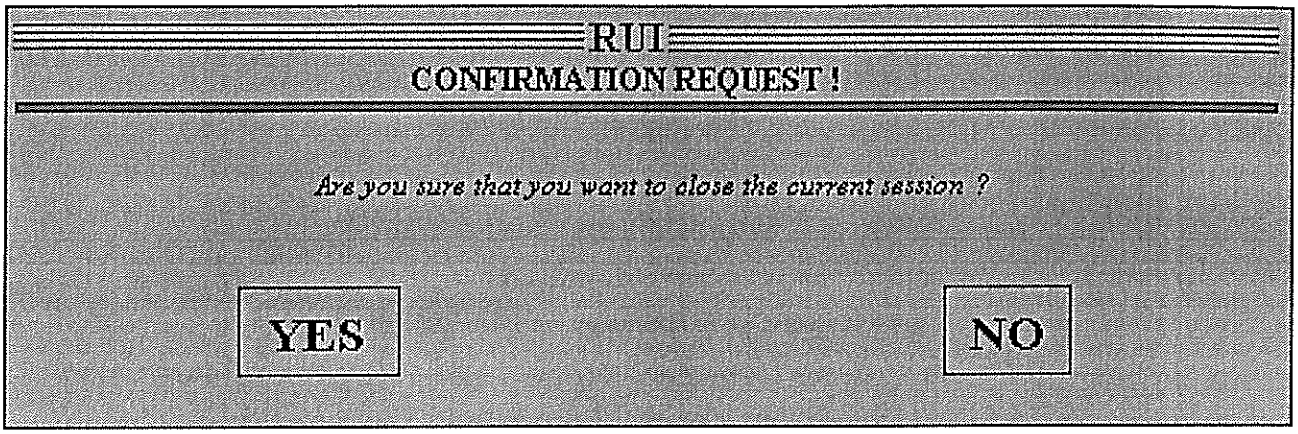

Figure 6: Confirmation of Session Closing Window

- MOVE TO POSITION: after the R.A. and DEC writable fields have been filled, the pointing of the Telescope will be activated by pressing this button. Before doing this action, a check will be made that the values are in their correct ranges. If the values are out of range, an error window will be displayed, to ask the user to introduce the right values[2] (see figure 7 below).
- STOP POINTING: this button is always disabled and hidden, except when the Telescope is pointing. In this case, it will be the MOVE TO POSITION button that will be hidden and disabled (so the two buttons are mutually exclusive). By pressing this button, the (Primary) user stops the Telescope pointing action;
- NEW OBSERVATION: this button is used to stop the tracking of the Telescope and to clear the writable fields of the Telescope Pointing Window. It is hidden and disabled only when the pointing of the telescope is active;

[2] When a value is inserted from an error window like the one shown in figure 7, it must be copied into the related writable field when the OK button is pushed. CLEAR should clear the related writable field.

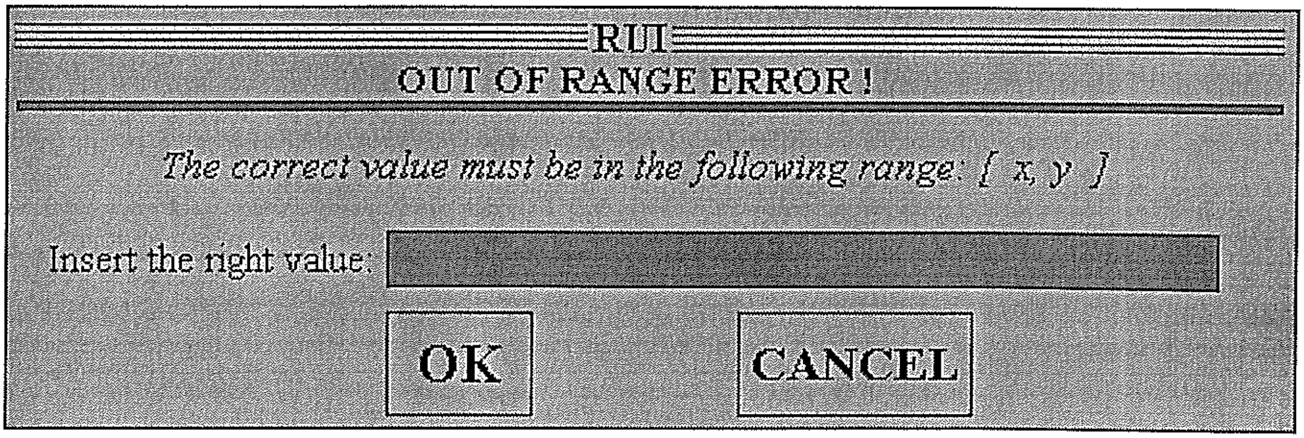

Figure 7: Out of Range Error Window[3]

- BYE: for a Primary User this button can be made visible and enabled only after the observing session has been closed (by pressing the CLOSE SESSION button). In the case of the Secondary User this button is the only visible and enabled button of the whole session (and the writable fields are always hidden and disabled);

It must be assured that the Telescope pointing window is iconifiable.

1.5 CCD Camera Window

This window will be displayed by pressing the CCD Camera button on the Available Systems Window.

It can be seen as logically split into four different parts (from a Primary User point of view):

a) *writable field*: here the (Primary) user can insert the data related to the exposure time expressed in seconds. The inserted valued has to be checked to see that it satisfies the range [0.01, 36000.00]. In case the values are out of this range, an error message like the one in figure 7, will appear;

[3] The x, y values are the characteristic boundaries of the range and will depend on the field they refer to. A separate check must be made for the sign in the DEC field. It could be acceptable if this check is made immediately after a character is put in the related writable subfield. The "Out of Range Window" must report the fields to be corrected in the same format as they appear in their original Window.
In case the user pushes the CANCEL button, the corresponding field to be corrected in the original window will be cleared and the Out of Range Window will be closed.

b) *selectable menu*: the (Primary) user can choose one of the proposed filters. If no filters are chosen, the window will propose the #1 filter by default. When the (Primary) user releases the selectable menu and the chosen value has filled the corresponding field, the filter position request has to be immediately sent to the TNG system, which will send back an OK message when the action has been completed;

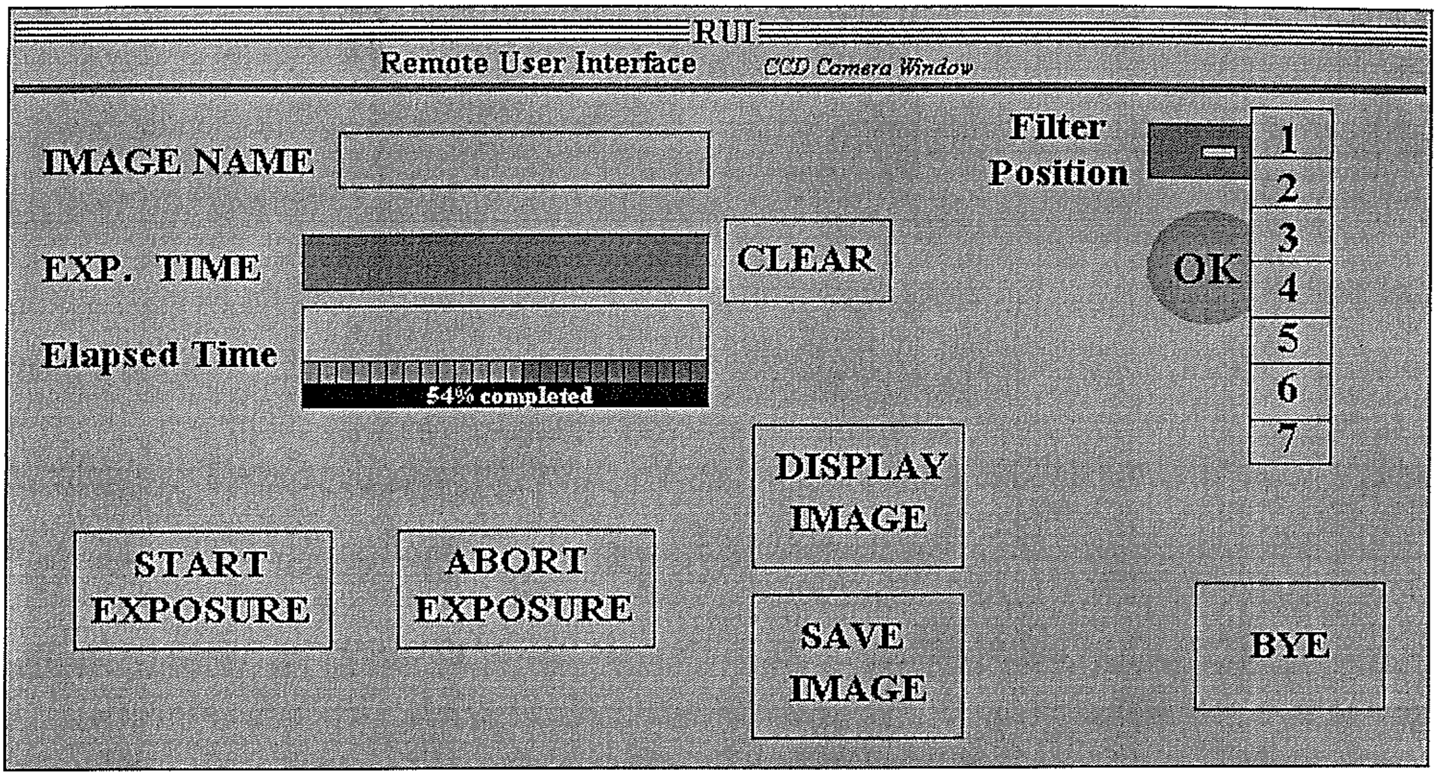

Figure 8: CCD Camera Window

c) *read-only fields*: while reading these fields the (Primary and Secondary) user can understand the time that has elapsed from the start of the exposure ("Elapsed Time" field) and the name of the image (defined by the TNG system and displayed on the "IMAGE NAME" field). Furthermore, the OK light will be shown when the filter wheel of the camera has reached the right filter position;

d) *buttons*: all the buttons (except the BYE one, which is visible and enabled to the Secondary User too) will be made visible and enabled only to the Primary User. The CLEAR button will be used to clean the writable field. The other buttons have the following functionalities:

- START EXPOSURE: by pressing this button the (Primary) user starts the exposure. This button is made visible only after the OK light of the filter position has been shown and the exposure time field has been filled correctly;
- ABORT EXPOSURE: by pressing this button the (Primary) user stops the exposure. This button is mutually exclusive with the START EXPOSURE button (when one of them is enabled and visible, the other one is disabled and hidden);
- SAVE IMAGE: this button is enabled and made visible to the (Primary) user only when the exposure has been completed. As a consequence of pressing this button, an information request window will be displayed, which asks to the (Primary) user the name with which the image has to be saved. The default name will be the one defined by the TNG environment on the "IMAGE NAME" field (see figure 9);

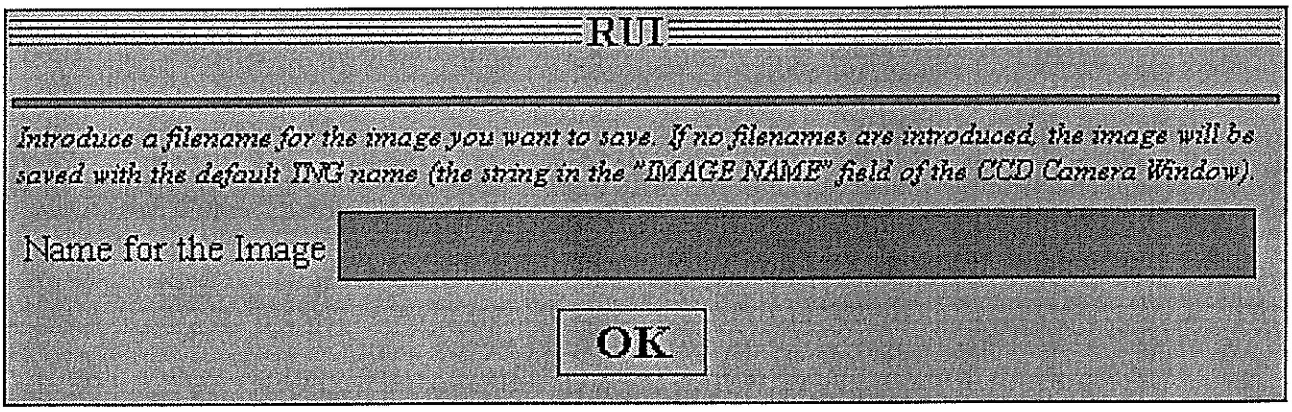

Figure 9: Information Request Window

- DISPLAY IMAGE: this button is enabled and made visible (to both the Primary and Secondary Users) only when the exposure has been completed. As a consequence of pressing this button a Graphic Window like the one shown in figure 10 of document D6.2 will be displayed to make possible the viewing of the acquired image (see figure 10);

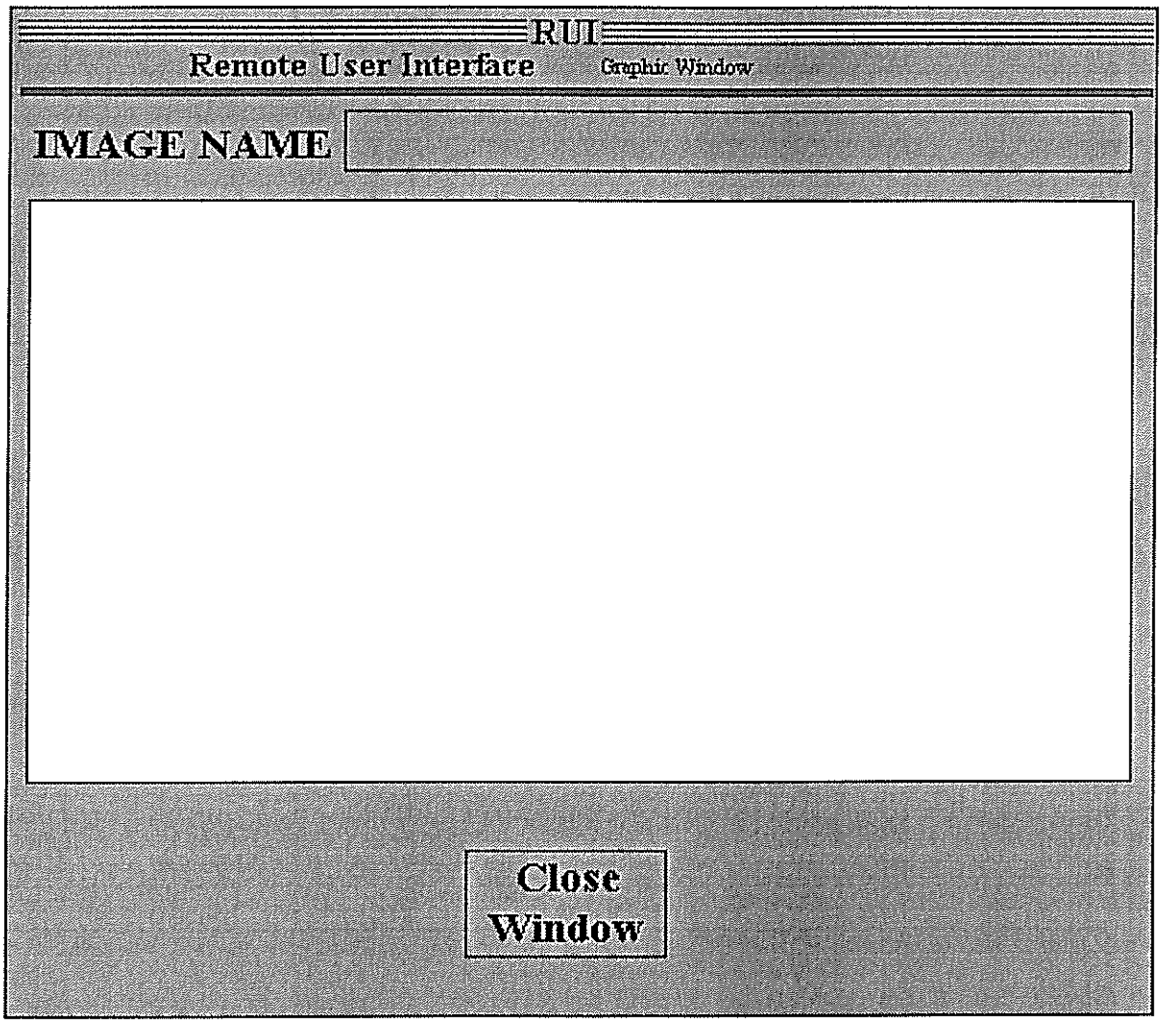

Figure 10: Graphic Window

- CLOSE WINDOW: this button is disabled and hidden while the exposure is running in the case of the Primary User. The (Primary and Secondary) user can access this button, unless he/she is saving an image, in which case he/she has to wait that the operation completes, before closing the window. The Graphic window is only used to display an image, and can be only read. The only information it holds is the IMAGE NAME[4] and the image itself. The (Primary and Secondary) user can close this window by pressing its CLOSE WINDOW button, even when the display operation is not acquired yet.

It is required that the CCD Camera window is iconifiable.

[4] The IMAGE NAME is the default name assigned to the image by the TNG environment and shown in the IMAGE NAME field of the CCD Camera Window.

1.6 System Problems Information Reporting

In order to let the user know that some problem has occurred at the Telescope side, two further windows are needed, into which a message related to the problem will be reported.

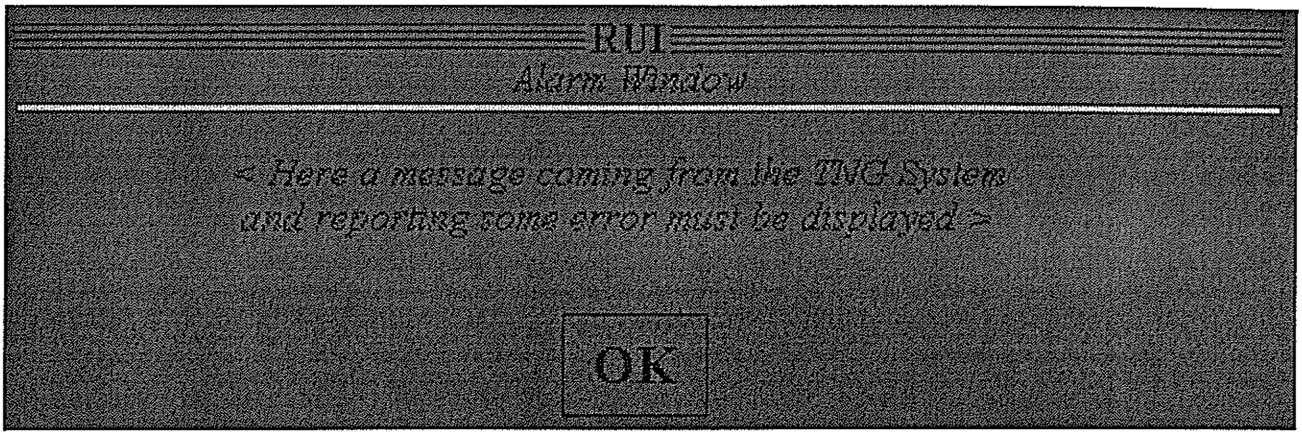

Figure 11: Alarm Window

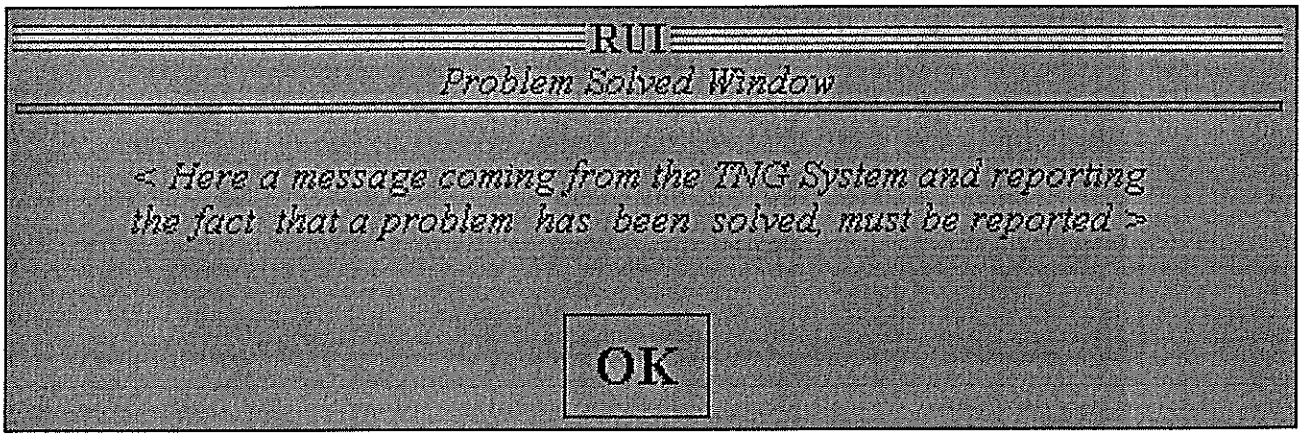

Figure 12: Problem Solved Window

The first one, which we call an Alarm Window, is like the one shown in figure 11 and will be displayed when an error occurs.

The second one, called a Problem Solved Window, is like the one shown in figure 12 and will be displayed after an error has been solved.

2. COMMANDS TO BE SENT TO THE TNG CONTROL SYSTEM VIA THE ANCILLARY PROCESS

In this section the macro-commands that the Remote User Interface has to send to the TNG Control System, via the Ancillary Process which will interface the Remote User to the Telescope, will be described.

They have to be exchanged between the Ancillary Process and the Teleoperation System in a string text format.

It will be responsibility of both the Ancillary Process and the Teleoperation System to interpret them with their correct meaning (once their description has been provided, as we are going to do in the following tables).

In the following the commands will be described in the same order as they are needed in the corresponding fields of the Remote User Interface Windows.

The protocol[5] related to the commands will be the following:

What to send (command)	*Length of the message*	*Text of the message*	*Checksum*
(1 Byte)	*(2 Bytes)*	*(Length Bytes)*	*(1 Byte)*

Figure 13: Protocol for the commands

The meaning of the various fields of this protocol must be interpreted as follows:

a) the *What to send* field is used to inform the Ancillary Process that a command has been sent to it. This information has to be specified in 1 byte that will represent this knowledge (the "c" alphabetic letter);

b) in the *Length of the message* field, the length of the string which will be written in the subsequent field is represented as a short integer value[6] written in string text format;

[5] Protocol formats might still be subject to some changes.

[6] The first byte will represent the most significant value (as in the big endians representation).

c) in the *Text of the message* field, the command with its operands will be written, representing them as a string in text format. The command acronym and its operands will be separated by commas (first the command, then its operands will be provided);

d) the *Checksum*[7] field will be used to verify that the information has been sent without corruption. The Checksum algorithm will simply be the sum of all the byte values of the command protocol, module 256.

> EXAMPLE: Some typical commands which the Communications Ancillary Process is waiting for (in the REMOT format) can be something like the following sequences of characters:
>
> | | | |
> |---|---|---|
> | c | 22 | LOGIN,UserName,PRIMARY |
> | c | 6 | SHODOM |
> | c | 26 | SHOTLS,22,15,37,-,45,11,10 |
> | c | 6 | LOGOUT |

The macro-commands will be organized into tables in which their name, their operands, the type they refer to, the related code and their description will be shown. Tables for operands will be introduced also.

2.1 System Access Window

The LOGIN command will be sent to the TNG Ancillary Process by the remote user when he/she clicks the OK button of the System Access Window.

Command Name	*Command Operands*	*Command Type*	*Command Code*	*Command Description*
LOGIN	LGNAME LGTYPE LGPASS	integer	1	This command is used to inform the Ancillary Process that a user is asking access to the TNG Control System

Table 1: Commands related to the System Access Window

Operand Name	*Operand Type*	*Operand Value*	*Operand Description*
LGNAME	text string		Login Name with which the User is known to the TNG System

[7] Within the context of the REMOT project we won't use this last structure, so the protocol for commands will only be composed by the 1st, the 2nd and the 3rd fields of information defined with points a), b) and c).

LGTYPE	text string	PRIMARY if Primary User is selected SECONDARY if Secondary User is selected	Type of access which the user is allowed to ask to the System
LGPASS[8]	text string		Encrypted Password of the User in the TNG System

Sub-Table 1: Operands related to the LOGIN command

2.2 Available Systems Window

Every command described in the following table will be sent to the TNG Ancillary Process by clicking the related button on the Available Systems Window.

Command Name	*Command Operands*	*Command Type*	*Command Code*	*Command Description*
SHODOM	no operands	integer	2	Tells to the Ancillary Process that the Dome Control Window is being opened
SHOTLS	no operands	integer	3	Tells to the Ancillary Process that the Telescope Pointing Window is being opened
SHOCCD	no operands	integer	4	Tells to the Ancillary Process that the CCD Camera Window is being opened
LOGOUT	no operands	integer	0	Asks to the Ancillary Process to logout the current user

Table 2: Commands related to the Available Systems Window

2.3 Dome Control Window

Every command described in the following table will be sent to the TNG Ancillary Process by clicking the related button on Dome Control Window.

Command Name	*Command Operands*	*Command Type*	*Command Code*	*Command Description*
DOMOPN	no operands	integer	5	This command is sent to the Ancillary Process to ask the TNG Control System to open the doors of the building.

[8] Because we assume that it's responsibility of the Teleoperation System software to provide and verify the User's Authorization check (Login Name vs Password verification), it won't be necessary to send this last field of information to the TNG system in the context of the Demo Pilot: we mean that at the TNG site we only need to be able to handle and control the list of users and check them vs the corresponding user types.

DOMCLO	no operands	integer	6	This command is sent to the Ancillary Process to ask the TNG Control System to close the doors of the building.
DOMBYE	no operands	integer	7	This command is sent to inform the Ancillary Process that the Remote User is closing the Dome Control Window.

Table 3: Commands related to the Dome Control Window

2.4 Telescope Pointing Window

Every command described in the following table will be sent to the TNG Ancillary Process by clicking the related button on the Telescope Pointing Window.

Command Name	*Command Operands*	*Command Type*	*Command Code*	*Command Description*
SESOPN	no operands	integer	8	This command is used to send to the Ancillary Process a request of opening the mirror covers of the Telescope.
SESCLO	no operands	integer	9	This command is related to the request, made to the Ancillary Process, of closing the mirror covers, and moving the Telescope to its standard position.
GOTOPS	RGTASC,DECLIN	integer	10	This command is sent to the Ancillary Process in order to ask to move the Telescope to a position required by the Remote User
STOPPT	no operands	integer	11	This command is used to inform the Ancillary Process that the Remote User is asking to stop the action of pointing the Telescope
NEWOBS	no operands	integer	12	This command is used to inform the Ancillary Process that the user is asking to perform a new observation. This involves stopping the tracking, related to the previous observation.
TELBYE	no operands	integer	13	With this command the user informs the Ancillary Process that he/she is closing the Telescope Pointing Window.

Table 4: Commands related to the Telescope Pointing Window

Operand Name	*Operand Type*	*Operand Value*	*Operand Description*
RGTASC	string		String used to represent the right ascension value for the Telescope pointing position.. The values must be expressed in hh, mm, ss
DECLIN	string		String used to represent the declination value for the Telescope pointing position. The values must be expressed in sign, deg, mm, ss[9].

Sub-Table 2: Operands related to the GOTOPS command

2.5 CCD Camera Window

Every command described in the following table will be sent to the TNG Ancillary Process by clicking the related button on CCD Camera Window.

Command Name	*Command Operands*	*Command Type*	*Command Code*	*Command Description*
FILTER	POSITN	integer	14	This command asks to the Ancillary Process to pass the positioning request of the selected filter to the TNG Control System. It is important that it is delivered as the user releases the mouse button used to choose the required position.
GOEXP	EXTIME	integer	15	This command asks to the Ancillary Process to start the exposure and maintain it for the specified time.
ABEXP	no operands	integer	16	This command is used to ask to the TNG system to abort the actual exposure.
SHOIMG	IMNAME	integer	17	This command is used to ask to the TNG System to display the image that the user has just acquired and the system has saved with the default name in the IMNAME operand.

[9] The values for the Right Ascension and Declination must be inserted as a list of strings (representing the corresponding values) separated by commas. In view of this, the operands of the GOTOPS command are composed by a list of strings represented as the following: hh, mm, ss, sign, deg, mm, ss. It is assumed that they correspond to valid values, because they come from the range membership check made by the Remote User Interface during the data fill in phase.

CCDBYE	no operands	integer	18	This command is used to notify to the Ancillary Process that the Remote user is closing the CCD Camera Window.

Table 5: Commands related to the CCD Camera Window

Operand Name	*Operand Type*	*Operand Value*	*Operand Description*
POSITN	integer	an integer from 1 to 7	The integer value corresponding to the filter position chosen by the Remote User

Sub-Table 3: Operand related to the FILTER command

Operand Name	*Operand Type*	*Operand Value*	*Operand Description*
EXTIME	float	a floating point value from 0.01 to 36000.00	The floating point value corresponding to the exposure time required by the Remote User.

Sub-Table 4: Operand related to the GOEXP command

Operand Name	*Operand Type*	*Operand Value*	*Operand Description*
IMNAME	string		The default name chosen by the TNG System for the image acquired during the last exposure.

Sub-Table 5: Operand related to the SHOIMG command

2.6 Miscellaneous commands

In this section a few commands not yet covered by the previous descriptions will be described.

In addition to all these commands, one more thing has to be said: referring to the Confirmation of Session Closing Window of figure 6, it should be noted that the YES button must be associated with the same command of the SESSION CLOSE button of the Telescope Pointing Window: SESCLO (see table 4).

Command Name	*Command Operands*	*Command Type*	*Command Code*	*Command Description*
SAVIMG	IMGNAM	string	19	This command is used to save the image specified on its operand (the default name is the string contained in the IMNAME field of the CCD Camera Window). It is referred to the Information Request Window of figure 9.
IMGBYE	no operands	integer	20 .	This command is sent to inform the Ancillary Process that the Remote User is closing the Graphic Window.

Table 6: Commands related to the Information Request Window and to the Graphic Window

Operand Name	*Operand Type*	*Operand Value*	*Operand Description*
IMGNAM	string		The name chosen by the Remote User to save the image acquired during the last exposure.

Sub-Table 6: Operand related to the SAVIMG command

3. PARAMETERS TO BE CAPTURED BY THE REMOTE USER INTERFACE

In this section the parameters, that the Remote User Interface has to receive from the TNG Control System, via the Ancillary Process which will interface the Remote User to the Telescope, will be described.

They have to be exchanged between the Ancillary Process and the Teleoperation System in a string text format.

It will be responsibility of both the Ancillary Process and the Teleoperation System to interpret them with their correct meaning (once their description has been provided, as we are going to do in the following tables).

In the following the parameters will be described in the same order as they are needed in the corresponding fields of the Remote User Interface Windows.

The protocol[10] related to the parameters will be the following:

What to send (parameter)	*Length of the message*	*Text of the message*	*Checksum*
(1 Byte)	*(2 Bytes)*	*(Length Bytes)*	*(1 Byte)*

Figure 14: Protocol for the parameters

The meaning of the various fields of this protocol must be interpreted as follows:

a) the *What to send* field is used to inform the Remote User Interface that a parameter has been sent to it. This information has to be specified in 1 byte that will represent this knowledge (the "p" alphabetic letter);

b) in the *Length of the message* field, the length of the string which will be written in the subsequent field is represented as a short integer value[11] represented in string text format;

c) in the *Text of the message* field, the parameter name with its telemetry values will be written, representing them as a string in text format. The parameter acronym and its values will be separated by commas (first the parameter, then its values will be provided);

d) the *Checksum*[12] field will be used to verify that the information has been sent without "corruption". The Checksum algorithm will simply be the sum of all the byte values of the command protocol, module 256.

The parameters will be shown into tables in which we'll show their name, their codes, the type they refer to, the related value and their description. Tables for variables will be introduced also.

[10] Protocol formats might still be subject to some changes.
[11] The most significant byte will be filled first (as in the big endians convention).
[12] Within the context of the REMOT project we won't use this last structure, so the protocol for commands will only be composed by the 1st, the 2nd and the 3rd fields of information defined with points a), b) and c).

EXAMPLE: Some typical parameters which the Communications Ancillary Process is passing to the Teleoperation System (in the REMOT format) can be something like the following sequences of characters:

p	5	OKLOG
p	6	SHOPEN,27
p	26	TELPOS,2,15, 7,-,40,1,19
p	6	EXIT

3.1 Dome Control Window

It is the first window of the Remote User Interface for which it is necessary to be able to get parameters from the Ancillary Process .

These parameters are related to the visualisation of the Doors opening/closing status, and are sent by the Ancillary Process to the Teleoperation System every 2 seconds.

They are described in the following table.

Parameter Name	*Parameter Variables*	*Parameter Type*	*Parameter Code*	*Parameter Description*
SHOPEN	OCVAL	integer	51	This parameter is sent to the Teleoperation System in order to let it know how much of the doors opening has been completed. When the doors are fully open, the OCVAL has the maximum value that it can have (250), and in this case the green light of the "open" field in the Doors Current Status of the Dome Control Win-dow has to be switched on.
SHCLOS	OCVAL	integer	52	This parameter is sent to the Teleoperation System in order to let it know how much of the doors closing has been completed. When the doors are fully closed, the OCVAL has the minimum value that it can have (0), and in this case the green light of the "closed" field in the Doors Current Status of the Dome Control Window has to be switched on.

Table 7: Parameters related to the Dome Control Window

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
OCVAL	integer	a value in the range [0,250]	This Variable is used to report the opening status of the doors. The value 0 means that the doors are closed, the value 250 means that the doors are fully open. Any intermediate value is allowed to say that the doors are partially open or partially closed.

Sub-Table 7: Variable related to the SHOPEN and SHCLOS parameters

The display 

in figure 4 is used to show what is the part of doors already open/closed. This information can be shown by receiving the OCVAL variable from the SHOPEN or SHCLOS parameters and calculating the proportion of the total that it represents.

3.2 Telescope Pointing Window

In this case, the relevant information to be shown to the user is related to the Right Ascension and Declination values current status, plus the eventual knowledge that the Tracking utility of the Telescope is on.

The related parameters can be shown in the following table.

Parameter Name	*Parameter Variables*	*Parameter Type*	*Parameter Code*	*Parameter Description*
RASTAT[13]	RAVAL	integer	53	This parameter is used to display to the user (every 2 seconds) the current value of the Right Ascension.
DECSTA	DECVAL	integer	54	This parameter is used to display to the user (every 2 seconds) the current value of the Declination.
TRCKON	TACTIVE	integer	55	This parameter is used to inform the Remote User Interface that the Tracking on light must be switched on.

[13] Instead of using this parameter and the DECSTA one, the implementation will provide a unique parameter, called TELPOS, which will represent both RASTAT and DECSTA and their values (following the convention of note 7, that the telescope position is a list of hh, mm, sec, sign, deg, mm, sec).

TELPOS	RAVAL,DECVAL	integer	59	This parameter is used to display to the user (every 2 seconds) the current value of the Right Ascension and Declination values of the Telescope (which represent its current position).

Table 8: Parameters related to the Telescope Pointing Window

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
RAVAL	string		This variable is in the same format of the RTASC operand of the GOTOPS command

Sub-Table 8: Variable related to the RASTAT parameter

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
DECVAL	string		This variable has the same format of the DECLIN operand of the GOTOPS command

Sub-Table 9: Variable related to the DECSTA parameter

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
TACTIVE	integer	0 if tracking is not active 1 if tracking is active	This variable is used to tell to the Remote User Interface that the tracking utility of the telescope is active

Sub-Table 10: Variable related to the TRCKON parameter

3.3 CCD Camera Window

In this Window there are two parameters that the Remote User Interface has to ask to the Ancillary Process: the default name with which the image to be acquired with the CCD will be stored in the TNG environment and the time that has elapsed from the start of the exposure.

They are shown in the following table.

Parameter Name	*Parameter Values*	*Parameter Type*	*Parameter Code*	*Parameter Description*
NMSEND	IMNAME	integer	56	This parameter is used to send the default TNG name for the next image to be acquired to the IMAGE NAME field of the CCD Camera Window in the Remote User Interface.
ELTIME	SPTIME	integer	57	This parameter is used to send (every 2 seconds) the elapsed exposure time related to the acquisition of the image to the CCD Camera Window of the Remote User Interface.
FILPOS	PACTIVE	integer	58	This parameter is sent to the Remote User Interface to let it know that the required filter has been positioned and that the OK light can be switched on.

Table 9: Parameters related to the CCD Camera Window

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
IMNAME	string		This variable carries the default name proposed by the TNG environment for the image to be acquired during the next CCD exposure.

Sub-Table 11: Variable related to the NMSEND parameter

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
SPTIME	float		This variable carries the time that has elapsed from the start of the exposure.

Sub-Table 12: Variable related to the ELTIME parameter

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
PACTIVE	integer	0 if the filter has not been correctly positioned 1 if the filter has been correctly positioned	This variable is used to tell to the Remote User Interface that filter wheel has been correctly positioned as required by the Remote User.

Sub-Table 13: Variable related to the FILPOS parameter

The display 

in figure 8 is used to show what is the elapsed time in a per cent fashion. This information can be shown by receiving the SPTIME variable from the ELTIME parameter and calculating the proportion of the total that it represents.

3.4 Miscellaneous parameters

In this section a few more parameters, not yet covered by the previous descriptions will be described.

Parameter Name	*Parameter Values*	*Parameter Type*	*Parameter Code*	*Parameter Description*
ERROR	MSG	integer	70	This parameter is used to report the presence of a serious error (in the Telescope Environment) to the Remote User Interface, which has to display it to the Remote User.
WARN	MSG	integer	73	This parameter is used to report the presence of a recoverable error (in the Telescope Environment) to the Remote User Interface, which has to display it to the Remote User.
INFORM	MSG	integer	74	This parameter is used to report an information message to the Remote User Interface, which has to display it to the Remote User.
OKLOG	no variables	integer	71	This parameter is sent to the Remote User Interface when the TNG System recognises[14] the user as a valid one and allows him/her to have access to its environment.
NOLOG	no variables	integer	72	This parameter is sent to the Remote User Interface when the TNG System doesn't recognise[15] the user as a valid one and does not allow him/her to have access to its environment.

[14] In this case the Remote User Interface is allowed to display the Available Systems Window to the Remote User.

[15] In this case the login error message shown in figure 2 must be displayed to the remote user and, after it has clicked the OK button of this window, the System Access Window has to be shown again.

EXIT	no variables	integer	73	This parameter is sent to the Teleoperation System in order to inform the client connected to it that the TNG environment is closing its connection with the Teleoperation System.

Table 10: Parameters related to the error and information displaying

Variable Name	*Variable Type*	*Variable Value*	*Variable Description*
MSG	string		This variable carries the message that the TNG environment has to send to the user to report the fact that an error has occurred.

Sub-Table 14: Variable related to the ERROR, to the WARN and to the INFORM parameters

Being it the case that this parameter refers to messages that the TNG System asks to display to the user, a further explanation of the protocol usage will be made here.

We call a protocol for the reporting of information and errors the same configuration used in the case of commands and parameters, but with the "e" letter in the *What to send* field and a simple text in the *Text of the message* field (see figure 15 below).

What to send (information or errors)	*Length of the message*	*Text of the message*	*Checksum*
(1 Byte)	*(2 Bytes)*	*(Length Bytes)*	*(1 Byte)*

Figure 15: Protocol for the reporting of information and errors

As shown in Figure 14, three kinds of messages will be provided by the TNG Control System:

a) ERROR messages, which correspond to critical problems occurred in the TNG environment;

b) WARN messages, which report the existing of some errors that do not currently block the execution carried on by the TNG environment, but could cause serious problems to come out in the future computations;

c) INFORM messages, which are sent to the Remote User Interface, to let the user know that a particular action has been carried on at the telescope side.

This all when we refer to messages sent by the TNG Control System to the Teleoperation System.

Another kind of message which has to be taken into account are the error messages sent to the Teleoperation System by the Communications Ancillary process: these ones will be formed following the protocol for the errors, by writing a string in the *Text of the Message* field (no parameters are defined for this kind of error messages). When we follow this convention, we send the error message (i.e. the protocol is of the form < '*e*' *length* *Error_Message* >) followed by a NULL parameter (which has the following format: <'*p*' *4* *NULL* >).

APPENDIX C: CONTROLLING THE TELESCOPE BUILDING VIA THE TELEOPERATION SYSTEM

Within the context of this appendix we'll briefly describe the panel of the Remote User Interface needed to control the movements/settings of the TNG building, as well as the commands and parameters related to the various buttons of the Remote User Interface itself.

This has been maintained separated from the rest of the interfaces, because the latter ones (described in appendix B of this same document), are related to the telescope simulation case, while the former has to be logically linked to the use of the only real part op the telescope that, at this moment, we are able to act on: the building where the telescope is located.

1. Graphical Window Devoted to the control of the Movements of the Building

The graphical panel that we need, to be able to control the movements and to set the various parts related to the building of the TNG, can be described referring to the following image:

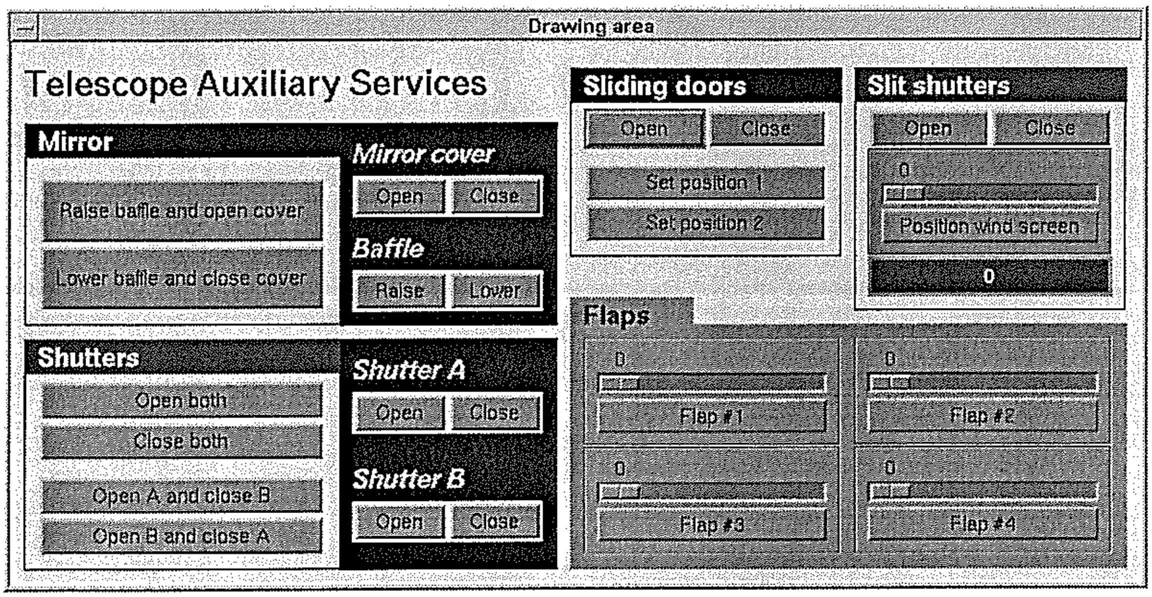

Figure 1: Graphical Window for the Building

All the buttons are related to the corresponding commands that we are going to describe in the next section, and have no return parameters, except the "Position wind screen" button, that is related to the WSPS parameter (that we'll describe in section 3).

2. Commands Related to the Graphical Window for the Building

In this section we'll present a table where all the commands related to the Graphical Window for the Building are listed and described referring them to the various buttons of the interface itself.

ACRONYM	DESCRIPTION	OPERANDS				
COVCLO	Close Mirror Cover	None				
COVOPE	Open Mirror Cover	None				
BAFFDO	Lower Baffle	None				
BAFFUP	Rise Baffle	None				
BADCOC	Lower Baffle & Close Cover	None				
BAUCOO	Rise Baffle & Open Cover	None				
SHABCL	Close Shutters A & B	None				
SHABOP	Open Shutters A & B	None				
SHACLO	Close Shutter A	None				
SHAOPE	Open Shutter A	None				
SHBCLO	Close Shutter B	None				
SHBOPE	Open Shutter B	None				
SACSBO	Close Shutter A & Open Shutter B	None				
SAOSBC	Open Shutter A & Close Shutter B	None				
SLSCLO	Close Slit Shutters	None				
SLSOPE	Open Slit Shutters	None				
SLDCLO	Close Sliding Door	None				
SLDOPE	Open Sliding Door	None				
SLDP1	Set Sliding Door Position 1	None				
SLDP2	Set Sliding Door Position 2	None				
FL1POS	Position Flap #1	**Number of operands**	**Operand(s) Type**	**Default Value**	**Min Val.**	**Max Val.**
		1	Float	1.000	1.000	90.000
FL2POS	Position Flap #2	**Number of operands**	**Operand(s) Type**	**Default Value**	**Min Val.**	**Max Val.**
		1	Float	1.000	1.000	90.000

FL3POS	Position Flap #3	**Number of operands**	**Operand(s) Type**	**Default Value**	**Min Val.**	**Max Val.**
		1	Float	1.000	1.000	90.000
FL4POS	Position Flap #4	**Number of operands**	**Operand(s) Type**	**Default Value**	**Min Val.**	**Max Val.**
		1	Float	1.000	1.000	90.000
WSPOS	Position Wind Screen	**Number of operands**	**Operand(s) Type**	**Default Value**	**Min Val.**	**Max Val.**
		1	Float	0.000	0.000	900.000

Table 1: Commands related to the Telescope Building Panel

The operands related to the commands described in the above table need to be set (as default) and range restricted, where foreseen, as defined by the corresponding values into the table.

This implies that we need a default set up to be made by the Teleoperation System when the user opens the graphic panel related to the building of the TNG. Moreover, the sliding buttons should permit the user to choose and see the desired value, by incrementing/decrementing it while moving the buttons themselves. The increment/decrement units should be the minimum possible as related to the precision of the data to be set.

3. Parameter Received from the Graphical Window for the Building

The only parameter that the Telescope Building Panel expects to receive is the WSPS one, which is related to the WSPOS command, that can be invoked by pressing the "Position wind screen" button.

It is described by the following table:

ACRONYM	NAME	DESCRIPTION	PARAMETER TYPE	LIMITS		WARNING		ALARM	
				Low	High	Low	High	Low	High
WSPS	Wind_scr_pos	Wind screen position	Float	0.000	0.000	0.000	0.000	0.000	0.000

Table 2: Parameter related to the Telescope Building Panel