NAOMI−INGRiD−TCS Level 0 Observing System
Proposed Requirements

Nigel Dipper, Durham; Andy Longmore, ATC; Richard Myers, Durham; Andy Vick, ATC

Version: 1.0
Date: 20 May 2000

1 Purpose

This document is a suggestion for an interim observing tool that can coordinate Naomi, INGRiD and WHT TCS operations for effective observing. In particular it will support automated multi−point jittered (dithered) observations with Adaptive Optics.
This functionality is a small subset of that required in a general−purpose full−function "OCS" such as ORAC[1,2] and is not intended to take the place of such a system.
This document does not contain a tutorial introduction to Naomi, INGRiD or the WHT TCS or their software architectures (in the case of NAOMI, an introductory technical description is available via the ATC WWW projects page[3]).

2 Scope

The proposed tool provides a minimum subset of functions needed to enable observing during Naomi commissioning and the following science runs to be carried out with reasonable efficiency.
It supports only the operations of INGRiD and the TCS that are necessary for AO observing.

<u>It consists of a simple GUI and observation scripting.</u>

Most details of Naomi operation are supported by the Naomi TopGui and are not reproduced in these requirements.

The proposed system does NOT support:
1. Automated script preparation
2. Nested display of script progress (i.e. if the script language supports function calls then it is not a core requirement that stepping into a function would open a new script progress window displaying the function internal statements)
3. Manipulation of the current script whilst running
4. Links to the data reduction environment
5. Support for any other instrument

3 Document Overview

We proposed the environment under which the GUI and scripting system would need to execute for efficient observing with Naomi and also the requirements for a minimal effective development environment. We then propose the GUI and script requirements and the performance needed. Finally we propose the requirements for documentation. Tests are not described here.

4 Operational and Development Environments

4.1 Operational

The GUI and scripting system only need to execute on the NAOMI Solaris workstation and can be displayed elsewhere via X−windows. The GUI and scripting system are intended for use by the Observer (with template scripts provided). An OPTIONAL extension would be the ability to display a duplicate of all or part of the GUI for provision of information to, for example, the Telescope Operator.
Communications are required with INGRiD, the TCS and Naomi. Naomi communications should all go via the "Electra Sequencer" using Python[4] or C libraries supplied by Durham. The DRAMA control interface to NAOMI will not be available until after first light on current schedule predictions

and its use for the current purpose would not be consistent with the development of a commissioning and first−light observing tool.

4.2 Development

Development of Naomi communications aspects without Naomi hardware or software will be possible using Python stubs provided by Durham. The stubs can emulate the sequencer and required subset of the real−time system API. Therefore no sequencer, communications library or hardware will be required for development and unit tests, but only a standard Python distribution[5]. System testing may be performed either in Durham using the Naomi hardware with TCS and INGRiD emulation or at ING as part of commissioning with all hardware present.

4.2.1 Implementation

Use of the Python stubs does not limit the implementation language to Python any further than is already inherent to the requirements. As the sequencer is written in Python and communicates with Python syntax, that level of Python dependence is present in the nature of the target system.

5 General Requirements

L0−GENREQ−001:
Concurrency and status refreshment. As the controlled subsystems support concurrency, then external commands will typically return quickly before the action is complete. The GUI and scripts therefore automatically provide a high degree of concurrency. In general the GUI should update its status periodically whether commands are in progress or not.
The GUI does NOT need to wait for actual command actions (as opposed to command communication functions) to complete before allowing other actions to be initiated. Where a script actually requires command actions to run to completion it is the responsibility of the script (or module) author to ensure this. It is NOT the responsibility of the script processor described here to ensure completion after each command but it MUST support a script language statement (or statements) which can wait for a controlled−system (i.e. Naomi, INGRiD, TCS) status to be achieved and handle any resulting timeout.

6 GUI Requirements

L0−GUIREQ−001:
Style
A standard GUI widget set such as Tk should be used

L0−GUIREQ−002:
Panels
The main GUI shall contain the following panels:
a) Field of View (FOV) display panel
b) Naomi/Script control panel
c) Naomi Status panel
d) INGRiD Status panel
e) TCS Status panel

L0−GUIREQ−003:
Error handling
The GUI should handle error returns from commands or status requests and reflect them to the user. Error information may be OPTIONALLY logged and/or sent to syslogd and the Talker.

L0−GUIREQ−004:
Communication timeout handling
The GUI should handle communications timeouts on external command/status requests

L0−GUIREQ−005:
Modal Error Box Dialog
A modal dialogue error box should report communications errors or timeouts which would compromise further system operation. (A modal dialogue is one which must be completed before further interaction with other parts of the application is allowed)

L0−GUIREQ−006:

Field of View Display Panel
The FOV panel should display:

a) The Naomi Wavefront Sensor (WFS) probe patrol limit (2.9 arcmin)
b) The INGRiD field of view
c) The PEC field of view
d) The NAOMI probe field of view (5 arcsec) and current position (retrievable from the NAOMI process monitor)
e) The centre of the WFS probe patrol field
f) OPTIONALLY: the RA/DEC of the field centre, the PA. These are available in the DRAMA 1 second variables database.
g) OPTIONALLY: star IDs and positions retrieved from a file or database (requires TCS information as in item f).
h) OPTIONALLY: A grabbed, overlayed INGRiD image
i) OPTIONALLY: A grabbed, overlayed PEC (NAOMI Pre−correction camera) image
j) OPTIONALLY: an image of the field taken from the DSS or equivalent
k) OPTIONALLY: switch controls to display/hide the above

L0−EXTGUIREQ−001:
OPTIONAL acquisition control capability extension to FOV panel functionality:
The FOV panel functionality required in *L0−GUIREQ−006* remains purely that of a display even if all options are implemented. This OPTIONAL extension requirement is for probe motion CONTROL by the following command to NAOMI:

```
L0GuiSupport.SetProbe(x,y)
L0GuiSupport.SetOffset(x,y)
```

for WFS loop open and closed respectively. x,y is the demand probe position relative to probe field centre in arcsec on sky. If this is implemented by something resembling a drag−and−drop metaphor involving the probe being dragged to a guide star (f) then OPTIONALLY this could be naturally extended (for the open loop case) to dragging a guide star (f) to the probe being implemented by TCS Motion followed by sending the following command to NAOMI:

```
L0GuiSupport.AutoCentreProbe()
```

L0−GUIREQ−007:
Naomi/Script control panel
The Naomi/Script control panel has 4 control modes and 1 modal dialog:

a) Open loop control mode
b) Close loop control mode
c) Script file selection dialogue
d) Script Control mode
e) Script/breakpoint control mode

L0−GUIREQ−008:
Naomi/script control panel: exclusive modes
The controls in each control mode of the Naomi Status/control panel shall be mutually unavailable, e.g. no open loop mode control is available in closed loop mode. The EXCEPTION is Script/breakpoint mode which shall allow either OpenLoop or ControlLoop mode controls according to the current loop state (in addition to script continue/abort controls). Availability of comtrols is summarised as follows:

Mode	Open Loop	Close Loop	Manual Probe Control	Manual Offset Control	Script Load Button	Auto Centre Probe	Run, Step & Abort	Pause & Abort	OK & Cancel
Open Loop	No	Yes	Yes	No	Yes	No	No	No	No
Closed Loop	Yes	No	No	Yes	Yes	Yes	No	No	No

File Select	No	No	No	No	No	No	No	No	Yes
Script Paused	No	No	No	No	No	No	Yes	No	No
Script Running	No	No	No	No	No	No	No	Yes	No
Script Breakpoint (loop open)	No	Yes	Yes	No	No	No	Yes	No	No
Script Breakpoint (loop closed)	Yes	No	No	Yes	No	Yes	Yes	No	No

L0−GUIREQ−009:

Naomi/script control Panel, Open Loop Mode controls :

a) "Manually move probe" sub−panel which itself contains controls for WFS probe motions: Up, Down, Left, Right, UpFine, DownFine, LeftFine, RightFine, Centre.
Corresponding control callbacks send Naomi sequencer commands:

L0GuiSupport.ProbeUp()
L0GuiSupport.ProbeDown()
L0GuiSupport.ProbeLeft()
L0GuiSupport.ProbeRight()
L0GuiSupport.ProbeUpFine()
L0GuiSupport.ProbeDownFine()
L0GuiSupport.ProbeLeftFine()
L0GuiSupport.ProbeRightFine()
L0GuiSupport.ProbeCentre()

b) "Auto Centre Probe on Star" control button , sends to Naomi Sequencer:

L0GuiSupport.AutoCentreProbe()

(Captures probe on to Guide Star).
ACTIVATING THIS CONTROL PRODUCES A TRANSITION TO CLOSED LOOP MODE

c) "Close Loop" control button sends Naomi Sequencer commands:

L0GuiSupport.CloseWFSloop()

ACTIVATING THIS CONTROL PRODUCES A TRANSITION TO CLOSED LOOP MODE

d) "Script Load" control button
ACTIVATING THIS CONTROL PRODUCES A TRANSITION TO THE SCRIPT FILE SELECTION DIALOGUE

L0−GUIREQ−010:

Naomi/script control Panel, Closed Loop Mode controls :

a) "Adjust offset" sub−panel which itself contains controls for adjust offset motions: Up, Down, Left, Right, UpFine, DownFine, LeftFine, RightFine, Centre.
Corresponding control callbacks send Naomi sequencer commands:

L0GuiSupport.OffsetUp()
L0GuiSupport.OffsetDown()
L0GuiSupport.OffsetLeft()
L0GuiSupport.OffsetRight()
L0GuiSupport.OffsetUpFine()

L0GuiSupport.OffsetDownFine()
L0GuiSupport.OffsetLeftFine()
L0GuiSupport.OffsetRightFine()
L0GuiSupport.OffsetCentre()

b) "Open Loop" control button sends Naomi Sequencer commands:

L0GuiSupport.OpenWFSloop()

ACTIVATING THIS CONTROL PRODUCES A TRANSITION TO OPEN LOOP MODE

c) "Script Load" control button
ACTIVATING THIS CONTROL PRODUCES A TRANSITION TO THE SCRIPT FILE SELECTION DIALOGUE

L0−GUIREQ−011:
Naomi/script control Panel, Script File Selection Dialogue controls :
File selection Dialogue allows selection of script file which is loaded as current script file into Script Control mode. The following controls are needed (Above the normal ones for file selections)

a) "OK" Control: PRODUCES A TRANSITION TO SCRIPT CONTROL MODE
b) "Cancel" Control: PRODUCES A TRANSITION TO CLOSED LOOP MODE

L0−GUIREQ−012:
Naomi/script control Panel, Script Control Mode controls :
Script control mode features a scrolling display of the currently executing script with a highlight of the next line to be executed. The script control panel can be in "run" sub−mode (script free−running) or "paused" sub−mode (script stopped). The startup sub−mode is "paused" − with the first statement highlighted. The script control panel also includes a text label containing the filename of the current script, a script status sub−panel (*L0−GUIREQ−016*) and the following script execution controls:

a) "Run" − equivalent to a video play button. Resume/Execute the script to completion or a breakpoint statement (see script requirements) or a press of the pause control. Available in paused sub−mode (the initial sub−mode).
b) "Pause". Pause at the next top−level statement. Available in run sub−mode.
c) "Step". Execute the next (highlighted) statement Available in paused sub−mode.
d) "Abort". TRANSITION TO OPEN LOOP MODE. Available in both run and paused sub−modes.

L0−GUIREQ−013:
Naomi/script control Panel, Script/breakpoint Control Mode controls :
This is a special case of script control mode in which the controls for either Open Loop control mode (*L0−GUIREQ−009*) or Closed Loop control mode (*L0−GUIREQ−010*) are available (according to the WFS control loop state) AS WELL AS the "Run" and "Abort" controls of the normal (user−initiated) paused sub−mode. This mode is entered on execution of a breakpoint script statement. The breakpoint message is displayed and the user may adjust Naomi control parameters freely before resuming (or aborting) the script. If the script is resumed using the "Run" control then the breakpoint message is removed and script execution continues. It is the responsibility of the script author to ensure that any changes made to the system configuration during the breakpoint are consistent with subsequent script operations. The minimal script statements support this.

NOTE: the philosophy here is that during conventional "pause" sub−mode as opposed to Script/breakpoint control mode, the user is discouraged from adjusting parameters by the non−availability of the Naomi Open−loop or Closed−loop controls. The reasoning behind this restriction is to avoid placing an onerous set of CORE requirements on the scripting language, which would otherwise need facilities to cope with asynchronous changes in configuration which the user could make between any two top level statements. This is not the case for programmed breakpoints where the possibility of changes may be anticipated by inserting status assertion check statements after each breakpoint. IF the chosen language can actually cope with asynchronous changes then the restriction can be removed and the full control can be made available during a user−initiated pause. Of course: the user always has the option of making changes during a pause by using TopGui to override parameters and locks if they have the knowledge and permissions. It is assumed such a user would understand the effect of such changes on a resumed script.

L0−GUIREQ−014: (NONE−GOALS)
Naomi/script control Panel, Script Control Mode scrolling display:

If the scripting language supports looping or conditional structures then it is NOT a CORE requirement that the script control panel scrolling display highlight supports the tracking of these (i.e. the ability to jump backwards or forwards over statements) and they may be forbidden from top−level scripts.

It is NOT a CORE requirement that any nested procedure calls which the scripting language may support should be made visible in additional scrolling display panels (i.e., only the top level of the script need be displayed).

L0−GUIREQ−015:
Naomi/script control Panel, Script Control Mode error handling :
The script statement executing system must handle timeouts and error returns at the top level (i.e., where they have not or cannot be handled explicitly by a relatively unsophisticated scripting language). The script must be halted and the user alerted if an un−handled error or timeout occurs. In general the Error Message should direct the user to the appropriate instrument/system for further investigations. In the case of Naomi this would be TopGui (a "nice to have" would be an invitation to launch or uncover/shift focus to TopGui)

L0−GUIREQ−016:
Naomi/script control Panel, Script Control Mode, Script Status sub−panel :
The Script control mode status sub−panel contains a number of labels and entries whose names and entries can be updated or deleted by scripts. A "nice to have" would be a hierarchical stack such as, for example:

Observing **P−BULBO** with INGRiD and NAOMI

Script: **My_script**

Script Status:

On source

9−point dither

dither point: **3**

Integrating

In this case the script updates the fields as it progresses and has some means of pushing and popping the display "stack"

L0−GUIREQ−017:
Naomi Status Panel
The Naomi status panel shall display the following Naomi process monitor variables:
a) WFS control loop state (i.e. "Open Loop" or "Closed Loop")
b) Naomi "Clear for Astronomy" (true/false)
c) Naomi "Idle" (true/false)
d) Naomi "OK" (true/false)
e) Naomi "Acquiring" (true/false)
f) Naomi "Configuring" (true/false)

L0−GUIREQ−018:
INGRiD Status Panel
The INGRiD status display shall INCLUDE the following information:
a) Exposing/seconds to run
b) Exposure time
c) Filter
d) Stop

L0−GUIREQ−019:
INGRiD Header Insertion Control
Adjacent to the INGRiD status panel should be a control for inserting a user−specified header into the
INGRiD science data headers. A typical use of this would be to annotate a science data file with qualitative information about the AO correction.

L0−GUIREQ−020:

TCS Status Panel
The TCS display is OPTIONAL but could include the object name, coordinates, PA, and time to set. These data are available from the DRAMA variable pool.

7 Scripting Requirements

This section describes the required features which the scripting language must have and the way in which the script control panel must interface to them. The language may be simply a collection of custom−designed keyword,value commands which can fulfil these requirements or it could be a more powerful existing language such as Python or Tcl in which functions/procedures can be used to provide the required functionality. In the latter case many additional language features would be available automatically. These could be very useful although it may be necessary to restrict their use in the CORE implementation (see *L0−GUIREQ−014*). It is also possible that the scripting language may be the language of GUI implementation in which case the SCRIPT−GUI interface functions may be achieved by interpreting the script statements in the Gui context.

See the *general requirements* for notes on concurrency and script execution.

L0−SCRPTREQ−001:
Language statements
The scripting language will require line−oriented statements to fulfil the requirements listed below. These statements could be implemented using function calls in the scripting language.

a) Comment
b) Breakpoint − puts the script control system in Script/breakpoint control mode (see *L0−GUIREQ−013*). A message argument is displayed.
c) Command to Naomi (String argument)
d) Wait for NaomiStatus to achieve string Arg1 before timeout Arg2 − see *L0−SCRPTREQ−003*. Timeouts must be handled by the script control panel.
Note that a WaitForStatus with timeout of ~0.0s amounts to a status assertion or check which will raise a handled error if it fails
e) Command to TCS [+ WAIT MODE as above − see *L0−SCRPTREQ−003*]
f) Command to INGRiD [+ WAIT MODE as above − see *L0−SCRPTREQ−003*]
g) Update named status field (Arg1) to string (Arg2). The syntax must support deletion of a label. (see *L0−GUIREQ−016*).

L0−SCRPTREQ−002:
Language features: functions (procedures) and variables
It is NOT a CORE requirement that the scripting language support variables or function calls but these would clearly add to the power of the system.

L0−SCRPTREQ−003:
Language features: loop and conditional control structures
It is NOT a CORE requirement that the scripting language support conditional structures or loops. These would add to the power of the system. Note the restrictions imposed on script display by *L0−GUIREQ−014*. NOTE THAT WITHOUT CONDITIONALS OR LOOPS THEN "WAIT FOR CORRECT STATUS" statements are required (see *L0−SCRPTREQ−001*).

L0−SCRPTREQ−004:
Language features: libraries (modules and packages)
It is not a CORE requirement that the scripting language support libraries but that would add to the power of the system.

7.1 Example scripts

Example scripts will be provided by the NAOMI team based on ORAC[1,2] scripts once ING specify the scripting language. If scripts are required earlier in order to help the language selection process then they can be provided in pseudo−code which will probably be python− or perl−like.

8 Performance Requirements

L0−PERFREQ−001:
Updating

Gui controls shall update after a maximum of 1.5 seconds from status changes being reported (to the Naomi Process monitor in the case of Naomi status information).

9 Documentation Requirements

L0−DOCREQ−001:
Online Help

L0−DOCREQ−002:
Users Guide

L0−DOCREQ−003:
Script Development Guide

10 Acknowledgements

Many thanks to Dr. Gillian Wright, UK ATC, for kind provision of a fully annotated ORAC observing sequence and UFTI configuration file at very short notice.
Thanks to Robin Clark, ING, for reading this document and providing numerous helpful suggestions, which have been incorporated.

11 References

1. http://www.roe.ac.uk/atc/projects/orac
2. Bridger, A. et al. 2000, SPIE Munich
3. http://www.roe.ac.uk/atc/projects/naomi/index.html
4. http://www.python.org
5. http://www.python.org/downloads