United Kingdom Infra Red Telescope

Joint Astronomy Centre

Royal Observatory Edinburgh

The Orac OCS: Overview Design

Alan Bridger

orac012-ocsd Version: 02 **Original: 22 April 1998 Modified: 22 February 1999**

This document sets out an overall design for the ORAC Observatory Control System. This design describes how important data flows through the system and how the components relate to each other.

1.0 Introduction

1.1 Purpose

This document presents the overall design of the ORAC Observatory Control System (OCS). It is intended to present a design of the system that will, as far as is possible, meet the general requirements set out in [O-1], and in doing so explain to the reader how the overall system will work. It will also therefore guide the detailed design and implementation of the separate components. The interfaces between the OCS and the other UKIRT systems will be described as part of the design of the separate systems in the OCS.

1.2 Scope

This document presents a design of the OCS that does not specify the likely language or environment to be used, but does attempt to specify how the various parts of the system interact with each other. It is the responsibility of each component to “fill in the details”.

1.3 Definitions, acronyms, abbreviations and references

These may be found in “*ORAC: Glossary and Reference List*” (orac000-gar). In addition the document uses the term “Sequencer” commonly in place of the full “Observing Sequencer” and there are occasional contractions (e.g. “OD”) which I hope are obvious in context. These saved my typing wrist a little.

1.4 Overview

Section 2.0 presents an overview of the system and describes the context of the Orac OCS.

Section 2.3 presents an overall description of the Orac OCS, concentrating on how important, *i.e.* scientific, information flows in the ORAC system. Some of this was previously presented as an appendix in [O-7].

Section 4.0 gives a short description of the engineering functionality that will be required of the OCS.

Section 5.0 describes the way in which the system, and its components, will be expected to handle errors.

Section 6.0 provides examples of how it is expected that observations will really be taken in the system, as an aid to understanding how it will work.

Finally, Section 7.0 lists the remaining unresolved issues.

2.0 System Overview

2.1 Context

The Observatory Control System is the system in the UKIRT Control System that will interface between the Observer wishing to acquire data and the systems responsible for taking, storing and reducing that data. Thus it must respond to a request from the Observer to carry out a certain, defined, observation (stored an *Observation Definition* in the *Observation Database)* and then pass on the relevant information in it to the other observatory control systems - the Observation Sequencer, the Instrument Control System, the Telescope Control System, the Data Handling System and Reduction System and also any facility systems like the FP and IRPOL systems. It is also responsible for sequencing the configuration of these systems and the data taking according to the *Observing Sequence* part of the *Observation Definition*.

The Observatory Control System consists of the Observation Sequencer and the Chooser.

A Scheduler could be added as a subsequent phase of the project.

2.2 Background

At present UKIRT does not have an Observatory Control System of the functionality described here. The AIM task in the current CGS4 and IRCAM3 systems performs some of the functions, but in reality the current OCS is the verbal communication between the Observer and the Telescope Operator.

2.3 System Context

Figure 1 shows the context diagram for the OCS.

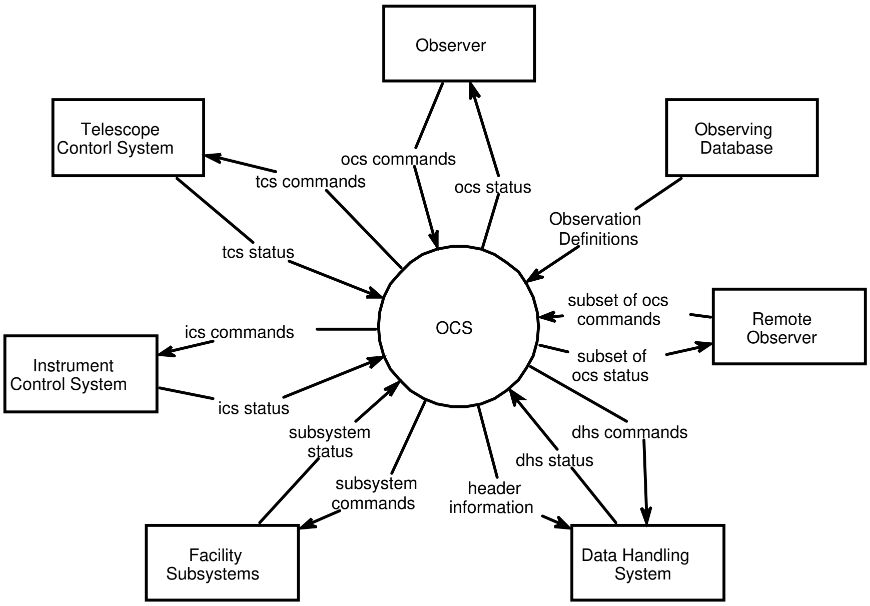

FIGURE 1. The Observatory Control System Context Diagram

In this diagram the external entities are:

Observer. This represents the Observer, or Engineer, issuing commands to the sequencer.

Telescope Control System. This is the existing UKIRT telescope control system.

Remote Observer. This represents a potential remote observer allowed to send commands to the sequencer. The commands sent and the status received may or may not be a subset of the full range.

Instrument Control System. This is the system that has overall control of the instrument - it can configure it and instruct it to acquire data. The instrument includes any array controller and data acquisition system.

Facility Subsystems. This represents other facility subsystems such as the FP or IRPOL systems.

Observing Database. This is a database of Observation Definitions as prepared by the new Observation Preparation System.

Data Handling System. This represents the system responsible for storing data and also for performing other services for the instrument control system (quick-look, provision of bad pixel masks etc.). The latter services probably have no relevance to the sequencer.

The major data flows on are:

ocs commands. This represents the commands to observe that come from the Observer or Engineer.

ocs status. This is a flow of status information - responses to **ocs commands** or displays of the observing status.

subset of ocs commands. This is a subset of **ocs commands** suitable for a remote observer.

subset of ocs status. This is a subset of **ocs status** suitable for a remote observer.

instrument configuration. This is how the component of an Observation Definition that represents an instrument configuration gets to the instrument system. This flow may be one of three things: A name of such a configuration; a data structure ("blob" of information) containing the configuration; or a set of atomic parameters designed to achieve the configuration.

ics commands. This represents the commands sent to the instrument control system instructing it to configure itself or to acquire data or to perform some other function.

ics status. This is a flow of status information - usually responses to **ics commands**.

subsystem commands. These are the commands sent to facility subsystems.

subsystem status. This is status information sent back by facility subsystems.

observation definitions. This is the flow of **Observation Definitions** from the database, as created by the preparation system.

dhs commands. This is the flow of commands sent to the data handling system by the sequencer informing it of changes to the data storage situation, possibly including telling it to expect certain data frames.

dhs responses. A flow of responses to the **dhs commands**.

header information. This is header information known or acquired by the Sequencer being made available to the Data Handling System for storage along with the associated data set.

tcs commands. This represents the commands sent to the telescope control system instructing it to configure itself or to acquire a target or to perform some other function.

tcs status. This is a flow of status information - usually responses to **telescope commands**.

target description. This represents the *target description* component of an *Observation Definition*. As with an instrument configuration it may be one of three things: A name of such a description; a data structure ("blob" of information) containing the description; or a set of atomic parameters designed to achieve the description.

3.0 System Design

An initial conceptual design for the OCS was outlined in [O-1]. Since then reviews of the project, discussions with instrument development staff and support staff and finally working on the designs for the components have cleared up a number of the unresolved details in the concept and thus it is now possible to present a more complete overall design. However, details of the design should still be found in the documents describing the individual components.

Although the context diagram presented above may be useful in understanding the relationship of the OCS part of Orac to other systems the overall system design presented in this section should give a much better idea of how things are intended to work in Orac. As such much more attention should be paid to this section.

The key parts of the overall design are understanding how the components relate to each other and how the important data (that necessary to take science data and the science data itself) flows through the system. Figure 2 outlines this, highlighting the data flow through the system.

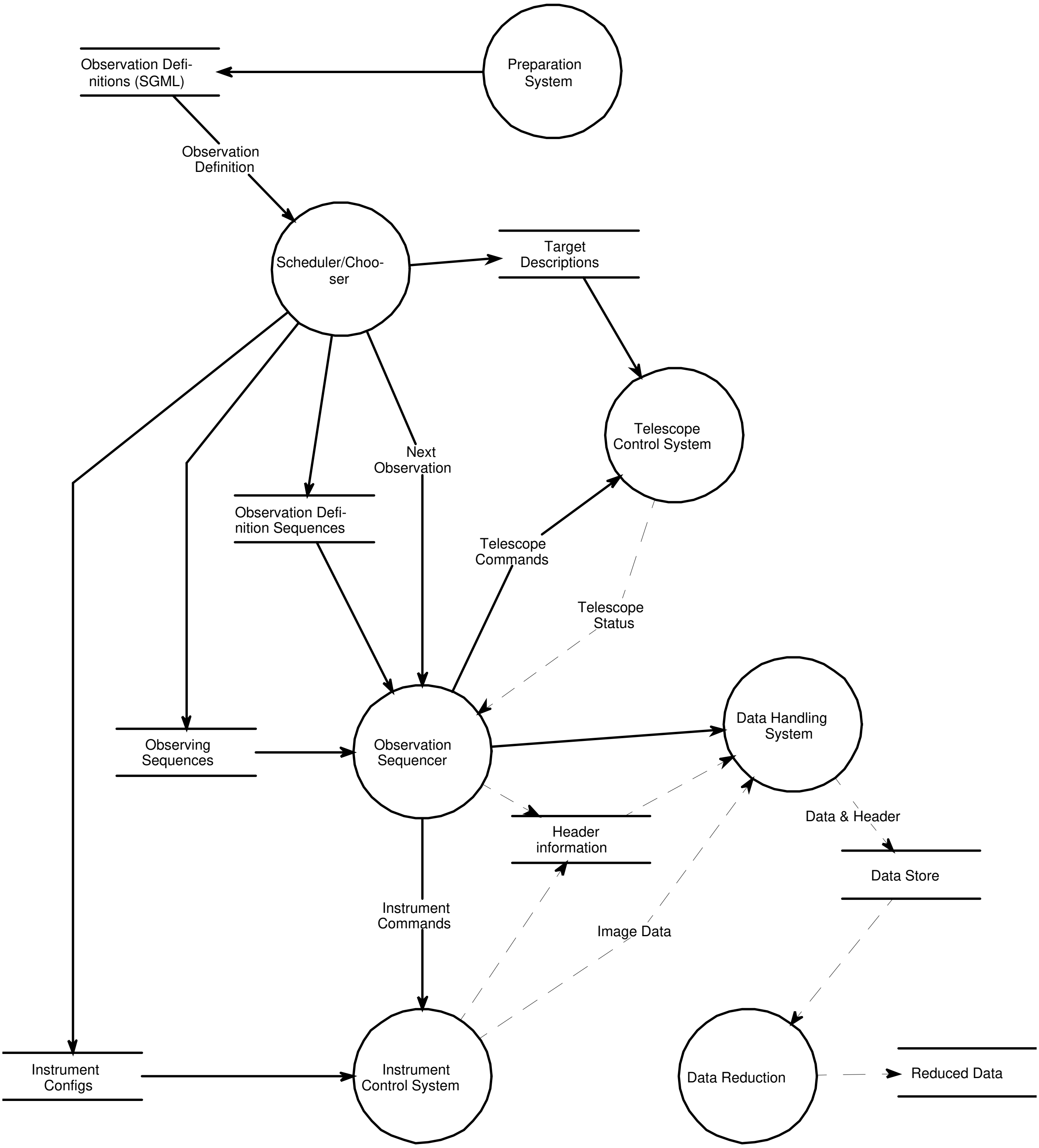

FIGURE 2. Information Flow in ORAC

What occurs can be described roughly as follows:

1. The *Preparation System* is used to prepare a number of **Observation Definitions** (as part of a complete Science Program) which are stored as **Observation Definitions (SGML)**, in the Observation Database.
2. At the telescope (i.e. during observing) the Observer will choose an Observation Definition to execute from a list presented to him or her by the *Chooser* or *Scheduler.* [NOTE: In the first version of the system this will be the *Chooser* - a tool that will allow the observer to view a list of files and choose one, in later releases it is planned that this will be a *Scheduler*, which will make the choice semi-autonomously, and pass it to the *Chooser*]
3. The *Chooser* will read the chosen Observation Definition and translate it into separate files: an **Instrument Config, Observing Sequence, Target Description** and **Observing Definition Sequence**, the latter being the definition of an Observation expressed as a Sequence of commands. All of these output data stores are temporary, i.e. they will probably be temporary files written by the system and removed after use.
4. The Observation Sequencer reads the **Observation Definition Sequence** for this Observation, and then, on instruction, proceeds to run it.
5. To run this it will command the telescope to read the **Target Description**, and acquire the target, command the instrument to configure itself from the **Instrument Config**., and then read the **Observing Sequence** for itself. It is possible that the **Observing Sequence** may already be present in the **Observation Definition Sequence** (NOTE, at least initially, and possibly always, the telescope operator will be heavily involved in target acquisition, though some help and automation can occur.)
6. In executing the **Observing sequence** it will command the telescope, instrument and other systems to acquire the requested data frames, and command the data handling system to be ready to store them.
7. During the actual acquisition of data frames the Sequencer will also instruct the data handling system to collate header information from the other systems. It (the Sequencer) will also supply its own header information, one item of which will be the data reduction recipe which it has acquired from the **Observation Definition**.
8. During acquisition the instrument control system will make data available to the *data handling system* to be stored. These data will be tagged to ensure they are collated with the correct header information.
9. The *data handling system* will store the data.
10. The *data reduction system* will detect the arrival of new data, read the data reduction recipe from its header and proceed to reduce it.

This description is of course slightly simplified and details of the exact processes may be found in the design descriptions of each part. However, it should suffice to indicate the intended execution of the final system.

One crucial step left out of this description is how remote copies of the Preparation System enter Observation Definitions (SGML) into the Observation Database. The details of this are TBD, during the design of the database.

3.1 Component Concepts and Definitions

In order to understand the system better it helps to understand the meanings of the names of its components. This section presents a short definition of each component.

3.1.1 Observation Database

This is a database containing the Observation Definitions and Science Programs prepared by observers ready for execution at the telescope. Programs and Definitions in this database are ready to be run, though they are also available for reviewing and modification. The database may or may not be a proper "database" and may be simply a directory structure (indeed in the first instance this is a very likely case).

3.1.2 Science Program

This is the single SGML file generated by the Preparation System (Observing Tool) for an observing programme's set of observations. It will normally contain many Observation Definitions, and roughly it is expected that there will be one such file for each accepted proposal - though this is *not* a technical restriction.

3.1.3 Observation Definition

An Observation Definition is a description of an Observation. In this context an Observation may contain more than one data frame (many coadds, sky and object pairs, a full mosaic), and it may also contain calibration data frames for the observation. Broadly it is roughly equivalent to the current UKIRT "group" concept, with the addition of the calibration frames.

An Observation Definition is made up of Scheduling Information, An Instrument Configuration, a Target Description, an Observing Sequence and a Data Reduction Recipe.

This usage of Observation Definitions (without SGML attached) is used to refer to the general concept and may refer to either the SGML version or the Sequence version. Exactly which one may be relevant can usually be determined from context.

3.1.4 Observation Definition (SGML)

The Preparation System writes the Observations Definition (as part of a Science Program) in SGML. This is the form in which the Definitions are stored in the Observation Database.

3.1.5 Observation Definition Sequence

This is a translation of an Observation Definition (SGML). The SGML definition may be recast in this form by adding the Sequence commands for loading and setting to Instrument Configurations and Target Descriptions, and setting the Data Reduction Recipe, to the Observing Sequence. The translation is performed by the Chooser. It is not a complete translation as the scheduling information is lost. This is not important as the Scheduler will have already made use of it.

3.1.6 Observing Sequence

The Observing Sequence is the description of the sequence of commands required to take the collection of data frames that makes up an Observation. For instance it may contain a command to take a DARK observation, and then to take a series of OBJECT observations separated by a number of telescope offset commands, to form a mosaic. It is prepared using a graphical interface in the Observing Tool, stored as SGML and translated to a form suitable for the Sequencer by the Chooser.

3.1.7 Instrument Configuration

This contains a description of the configuration of the instrument that will be required for a given observation. It will contain definitions of items such as the central wavelength, the filter, the exposure time etc. It may also contain variants of these definitions to be used for calibration frames. The original is stored in the SGML Observing Programme file, the one used by the ICS is stored as a set of attribute-value pairs, following translation by the Chooser.

3.1.8 Target Description

This contains a description of the target to be used by the TCS. It contains information such as the RA,DEC coordinates and name of the source, and also similar information for offset guide stars. The original is stored in the SGML Observing Programme file, the one used by the TCS is stored as a set of attribute-value pairs, following translation by the Chooser.

3.1.9 Data Reduction Recipe

This is the name of the Data Reduction Recipe that will be used by the OracDR system to reduce the Observation.

3.1.10 Scheduling Information

This is information used by the Scheduler to determine when to perform the Observation, or to provide the Observer with information to help them decide.

3.2 The Chooser

The Chooser is the system that will present the Observer with an interface to the Observation Database and allow him or her to select Observations to b performed. It will read the selected Observation, translate it into its separate components for the UKIRT system and then instruct the Sequencer to carry out that Observation.

3.3 The Scheduler

The Scheduler is a more intelligent version of the Chooser. It will be able to provide the Observer with a suggested order in which to perform Observations, or to pass them directly to the Chooser for execution, and to issue warnings and suggestions about incompatibilities between standard star observations and target observations.

3.4 Observing Control and Status Display

This will be the main user-interface for the Observer. In it's initial form it will allow access to the functions of the Chooser, including presenting the list of available Observations, and will also provide immediate control and feedback to the user of the ongoing Observation, i.e. it will form the user-interface to the Sequencer. It may also present a summary of relevant Observational information (from other systems), and must also allow access to some other functions, e.g. the ability to run "Movie".

As one of its main function is to provide feedback on the state of the execution of the Observation, i.e. the list of executing and upcoming sequence commands it must allow interaction with this as required, via the provision of the relevant control commands: pause, stop and abort. Provision must be made for more than one Sequencer interface as there might be more than one Sequencer.

3.5 Preplanned Breaks

Most observing will require pre-planned breaks in the sequence. A particular example here will be the need to complete source acquisition following a slew. The insertion of these breaks will be a function of the Chooser, where they will be inserted into the sequence during the translation process. Another example would be the “hidden” Breakpoints (Section 3.9). At what points in the sequences these breaks will be inserted will be a configuration option of the Chooser. More details of the requirements here will be found in the requirements for the Chooser ([O-14]).

3.6 The Sequencer

The Sequencer is the system that will execute the Observing Sequence. This sequence is (by this point) implemented as a Sequence of commands and thus the Sequencer is simply an interpreter for running these. The Sequencer will be table driven, allowing Sequence commands to be related to the command and system they are relevant to, along with the messaging technology that is to be used to issue them. The design of the system suggests that each Sequence command is an atomic command, it will result in only one message being sent to another system (or in a simple internal command). This results in the Sequencer being reasonably simple, but means that the more complex commands (those that require several actions to occur, usually in more than one system) are implemented in some other way. In most cases these will be Standard Sequences.

3.7 Library Sequences

These are typical Observing Sequences that are available to the user of the Observing Tool. Library Sequences will be Observing Sequences recommended by the Observatory for use by many, if not most, observers. An example Library Sequence would implement a typical jitter photometry pattern or a series of spectroscopy “pairs” using the recommended “nod” distance. Note that how changes in some of value implicitly recommended by these sequences are handled is not yet determined. As noted in Section 3.8 Library Sequences should not be confused with Standard Sequences.

3.8 Standard Sequences

The decision to remove the intelligence of many relatively complex Sequence Commands from the Sequencer design means that another way must be found to implement them. Since most of this intelligence is in fact the issuing of several commands to many systems the obvious way to implement this is by using many Sequence commands to build up a Standard Sequence. E.g. the OBSERVE Sequence command will be a Standard Sequence that might: Issue a command to the DHS to ready it for data retrieval, and obtain a data label; create and post the new observation number; instruct the DHS to collect initial header information; instruct the instrument to observe; instruct the DHS to collect final header information, and inform it of the end of the observation.

However, it is recognised that in normal observing the observer will not wish to see the details of the above sequence, they will be happy simply to know that the OBSERVE command is being executed. Thus the status display must know not to display all of the details. This is discussed in Section 3.9.

To preserve flexibility the translation into Sequences performed by the Chooser will not know of these “Standard Sequences”, it will write Sequences making use of the more complex commands, e.g.”OBSERVE”. It is on loading of the Sequence by the Sequencer that the recognition of them as Standard Sequences will occur.

Finally it should be noted that despite the similarity in implementation these Standard Sequences are not the same as the Library Sequences that the Observer will have access to in the Observing Tool. One difference is that these Library Sequences will be fully unrolled in the status display, another is that the OT and the Chooser clearly know about them as Sequences.

3.9 Hidden Commands

These are sequence commands that are used internally by the system and should not normally be displayed by the status display that is giving visual feedback on the sequence. One example was discussed above - the OBSERVE command will consist of several sequence commands, but the observer will not normally wish to know this. Thus it must be possible to mark these commands in a way that informs the status display that the next set of commands (which must be delimited in some way) are not to be displayed - only the invocation of the Standard Sequence. However, for engineering purposes it should remain possible to display the details of a Standard Sequence.

Another hidden command is the “Breakpoint” command. This is used internally to tell the sequencer where a Sequence should be sensibly broken (or stopped) to achieve a desired effect (e.g. completing a sky/object pair). Normally this command is ignored, but a selected “breakpoint” may be made “active” if a relevant STOP request has been issued. These breakpoints are not completely visible in the status display, though they may be highlighted in some way so the user gets visual feedback as to where their STOP request will become active.

These two points place demands on the user-interface, but they may in turn also place demands on the Sequencer as it is not yet clear how to flag these items to the user-interface.

3.10 Reducing the Data

It is intended that the interface between the UKIRT Observing System and the UKIRT Data Reduction system is implemented as much as possible through the file system. This has the advantage that the two systems are completely decoupled and do not depend on each other being present in any way. However, this design does present a few challenges.

In order to meet its requirements the Data Reduction system will require certain pieces of information and certain behaviours from the other systems, thus placing requirements on them. These issues are briefly considered here.

3.10.1 Format of data files.

New data files shall be written in NDF format [X-12].

3.10.2 Location of data files

It is intended to keep data files separated by both instrument and UT date of observation. Thus it is proposed that the files shall be stored as

<directory-path><prefix><yyyymmdd>_<nnnnn>.sdf

Where:

<directory-path> is the directory in which the file will be stored. This is likely to be of the form:

$DATAROOT/<instrument>/<yyyymmdd>

Where:

$DATAROOT is an environment variable pointing to the root directory of the data storage area

<instrument> is the instrument name

<yyyymmdd> is the UT date in digits, yyyy a 4-digit year, mm a 2-digit month and dd a 2-digit day

<prefix> is a prefix string. This will be simply "f" for raw data files.

<yyyymmdd> is the UT date in digits, yyyy a 4-digit year, mm a 2-digit month and dd a 2-digit day

<nnnnn> is the observation number stored in a fixed-length field 5 digits wide, padded on the left with zeros.

An example filename would therefore be:

$DATAROOT/ufti/19980819/f19980819_00042.sdf

Remember case is significant.

A decision needs to be made regarding how and where engineering data are stored.

3.10.3 Content of Data files

There shall be one data file for each OBSERVE command. Note that for chop mode observations, and the user has requested that the data from all chop beams be stored, then there will be more than one data frame in the file, one frame for each (coadded) chop beam. The OracDR system must be able to cope with this scenario. A similar situation can be envisaged for other modes, e.g. shift and add where vector map information as well as the final data frame may be returned for storage.

In addition to the data frame or frames all of the header information relevant to this observation must be stored in the same file.

3.10.4 Announcement of New Data

The Data Reduction system needs to know when a new data file is completed in order to be able to reduce it. With the DR system and acquisition system decoupled there will be no queue of observations to be reduced, and no message will announce the presence of new data. Thus the DR system will simply monitor the relevant directory and spot the appearance of new data. There is one danger in this - for various new files can appear before they are complete. The baseline solution to this is to design the data storage system so that data files are constructed with scratch names and then "moved" (renamed) to their final location when complete. This must be included in the design for the storage system, which is the responsibility of the Michelle project.

3.10.5 How to reduce a data file?

The Orac Data Reduction system requires that each file has a header item “RECIPE” which has as a value a string containing the name of the data reduction recipe that will be used to reduce it. This recipe name is part of the Observation Definition and so it will be available as data at the time the Sequencer begins to run the Observing Sequence. A likely solution is to use a Sequence command of the form “DRRECIPE name”. This header item will be made available for storage in the data header by the Observation Sequencer.

The DR system will also need information on what type of data it is, along with other relevant data (e.g. exposure time). All of this data must be stored in the header of the data file. It is the responsibility of the DR designer to define what must be in the header in order for the DR system to function.

3.10.6 Reducing a group of files

This will be done in the same way as a single file - the recipe name will decipher into information that the file is part of a group to be reduced in a certain way.

3.10.7 Relating groups of data

The baseline design for relating groups of data is to include in the headers a “group number”, a “sequence number” and a “maximum in group” value so that the DR system knows how to relate data files to each other. Again it is the responsibility of the DR system designers to define more fully what is required. It is the responsibility of the OCS and DHS as a whole to ensure that these values are correctly calculated and made available for storage in the data header.

3.10.8 Determining the calibration frame

The DR system has a requirement to be able to determine what is a sensible calibration frame to use. E.g. it will need to find a FLAT-FIELD frame taken nearest in time with certain instrument configuration details identical to those in the current data frame. To forestall searching the header of every data file it is sensible to keep an index of data frames that may be used for this function. This index will be written and maintained by the Data Reduction system

3.10.9 Normal completion of a group

The “recipe” for a group may be related closely to the observing sequence such that the exact number of data files in the group is immediately known. If not then the recipe may expect to be parameterised, e.g. a parameter describing the number of pairs to be added. These parameters will be found in the header of the data file. Again details are to be provided by the DR system designers.

Given one or other of these pieces of information then the DR system knows when the data frame that has just appeared is the last one.

3.10.10 Premature completion of a group

There are common circumstances when a group will finish prematurely - for example when enough signal to noise has been obtained, or when enough of an area has been mapped. In addition some observing sequences will call for indefinite integration (when the expected signal is not well-known) until stopped. These scenarios all correspond to use of the STOP directives described in Table 1 of [O-13]. The problem is how to inform the DR system when it is completely decoupled and will be waiting for the next data frame in the group.

The Orac DR system will function by always performing all of the reduction that it can on new data, this means that, for instance, a complete pair of “nod” positions for spectroscopy will be flat-fielded, subtracted, coadded and divided by a standard, and if relevant, the spectrum extracted, even if more data are expected. The consequence of this is that the Orac DR system does not need to know that a group is “complete”, it is always treated as complete when sufficient information is present. The upshot of this is that no special action has to be taken by the rest of the system in order to inform the DR that a “group” has finished. The DR will work this out for itself eventually by the appearance of data with a new “group” number. In the meantime the observer always sees data reduced as far as possible.

3.10.11 Aborting an observation.

Occasionally it is necessary to abort a data frame, which may of course be part of a group. It may be that the entire sequence is being aborted, a case treated in Section 3.10.12, but it could also be that a single observation is being aborted with a view to repeating it immediately. A case in point might be because the telescope pointing strayed - it will be useful to quickly abort the observation and simply ask the system to repeat it after the problem is resolved.

The implication of this is that the Storage system should not store data for the aborted observation but then must re-use the data label and observation number it had planned to use for the new data frame that is taken instead. In this way the DR will not be aware of any problems and no further special action is required.

3.10.12 Aborting a group

Occasionally it is necessary to abort an entire observing sequence, or group. In this case the data reduction system will act in the same way as described in Section 3.10.10, except of course that the group may not be at a “tidy”, i.e. completed, point. As the group is being aborted for some reason this is not regarded as a problem.

The implication of this is that the Storage system should not store data for an observation that is aborted, and does not need to perform any further special action for the DR system.

3.11 Backwards Compatibility

Orac must be able to run the existing suite of UKIRT instrumentation (CGS4 and IRCAM3 in particular, but also including UFTI), though it is not a requirement of the Orac project to implement this, merely to show how it can be done. The problems fall into two main areas, the control of the instrument and the receipt and storage of data from the detector controllers.

3.11.1 The Instrument Control Systems

These will be considered in more detail in [O-13]. However in brief CGS4 and IRCAM will remain VMS-ADAM based, probably controlled using ADITS. This will require some modifications to the existing code, but these should be small.

UFTI could be controlled in the same way. However, it is a sufficiently simple instrument, now with an EPICS interface, that writing a Drama task to control it should be a fairly simple task and might present the best way forward.

3.11.2 Data Storage

In general Data Storage is not part of the ORAC project. However, in order to meet the backwards compatibility requirements it should be considered.

The simplest approach to take for IRCAM3 and CGS4 is to leave the data storage exactly the same as it is at present. However, this will present problems when interfacing completely to the new Data Reduction system. One approach would be to layer a new system above the old that converts the old-style data files into new ones ready for the new Data Reduction. This is very similar to what is already done for IRCAM by the ACVT task. The addition of a facility to ensure the file was not available until fully converted might be required.

A more sophisticated approach would be to interface the old data acquisition system to the new data handling system. The new data handling system will be set up to serve the Gemini DHS interface ICD3 [G-4], however there would seem to be no particular reason for it not to be capable of responding to the interface presented by the ALICE system, which is in fact a fairly simple socket-based system.

There should be a very convenient solution for UFTI. The UFTI data acquisition system is now in a vxWorks system and thus it should be easily possible to fit the Gemini ICD3 interface to it and thus use the new data storage provided by Michelle.

If required this will considered in more detail in a later document.

4.0 Engineering Functionality

As an *Observatory* control system the OCS will not have much engineering functionality and in general required engineering functions should be provided by the individual systems. However, the engineering functionality needs to fit into and be consistent with the overall system.

An important question is how “engineering type” Instrument Configurations and observing sequences might be created?

It is envisaged that instruments may have two levels of engineering configuration (like CGS4) or just 1. The higher level of these is what is known for CGS4 as an “intermediate config” - this allows the setting of non-standard instrument configurations by setting the position of each motor in the cryostat, by selecting either the name of a component (*e.g.* name of a blocking filter, or slit) or its position in calibrated units (*e.g.* focus position = x mm, slit angle = y degrees). This level of configuration definition can be provided from the OT by treating it as a different instrument and providing an appropriate GUI. The intention is that these “engineering instrument” screens would only be available in the JAC version of the OT, and NOT to the general astronomer. Iterators could be provided for each item, allowing the easy creation of sequences repeatedly moving a single item.

The lower level of engineering instrument control is to set the instrument parameters directly to a position in motor steps. (available for CGS4 in engineering/direct motor control). For the existing instruments this would be provided in the same way as it is now, and for future instruments using Epics databases the functionality would be provided by entering values directly into DM screens controlling the database.

Many engineering tests consist of sequences that the JACH version of the OT will be capable of writing, assuming that the “intermediate versions” of the instrument configurations is available. These sequences typically consist of commands like Config eng1, object, Config eng2, object etc. However, there is a need for some specific engineering sequencer commands which the Gemini OT does not (currently) have, such as DATUM. Whether this is achieved by a direct user-interface to the sequencer, or via the “JAC” version of the OT is TBD.

For troubleshooting the OCS will provide an ability that is not available with the current system - namely that the status displays could be expanded to show the details of all the sub-commands in OBSERVE so that the point at which it fails can be readily identified.

In addition there will be logging capabilities in most, if not all, systems (details are still TBD).

5.0 Handling of Errors

In general each system in the overall control system will be expected to detect, report and handle its own error conditions. However, these systems must also report failures or errors to any system that has issued a command to them. In particular the Sequencer will be issuing commands to a number of systems and must be made aware of any failures to complete these commands.

It is intended that the Observer control the observing from a single screen, thus any errors that occur in one of the systems must be related back to the Sequencer and onto the user-interface where the Observer can respond. Occasionally a system will require resetting in some way to correct a problem. This reset will, in most cases, require action on the DM or SMS screen controlling and monitoring that instrument, which will be under the control of the telescope operator.

6.0 Taking Observations

The description presented in this document attempts to describe an OCS that will enable Orac to take data in all of the modes present in current and planned UKIRT instrumentation. However, it is useful to “step back” and present a description of what happens during a typical observing sequence. Two examples are presented here, a single image, to demonstrate what happens during an observation, and a sequence of two “nod pairs”.

6.1 A Single Observation

The following is a simple sequence to take a single image of the source FS18. It is assumed that the Preparation System has been used to prepare the observation and that the Chooser has selected the observation and performed the translation into the relevant Instrument Configuration, Target Description and the Sequence we see below. Each step in the sequence is accompanied by a description of what happens as a result of the sequence command. It will be noted that the steps necessary to take an actual observation are the most complicated.

Sequence Command	Action(s) taken
LOADCONFIG FS18	Send command to ICS to read in the configuration called FS18
TARGETDESC FS18	Send command to TCS to read in the target description called FS18
SET DARK	Tell the instrument to set its configuration to that relevant to a dark, do not wait for this to finish before proceeding.
SLEW	Tell the telescope to slew to the position described in the target description. This may also include acquiring guide stars and a significant amount of interaction with the telescope operator
OBSERVE DARK	Do nothing until all previous commands are complete. Then (each command must complete before the next is issued): **1.** Inform the DHS that an observation is about to begin, and receive a frametag in return **2.** Inform the collector task that header information relevant to the beginning of an observation should be collected. Provide the frametag with the information. **3.** Instruct the ICS to take an observation of type DARK, giving it the frametag to use. **4.** Inform the collector task that header information relevant to the end of an observation should be collected. Provide the frametag with the information. **5.** Inform the DHS that the observation using the frametag has completed.
SET OBJECT	Tell the instrument to set its configuration to that relevant to an object, do not wait for this to finish before proceeding.
DRRECIPE SINGLE_IMG	Post the name of the DR Recipe to make it available for the DHS to collect.
OBSERVE OBJECT	Do nothing until all previous commands are complete. Then (each command must complete before the next is issued): **1.** Inform the DHS that an observation is about to begin, and receive a frametag and group sequence number in return (sequence number is irrelevant if not in a group) **2.** Inform the collector task that header information relevant to the beginning of an observation should be collected. Provide the frametag with the information. **3.** Instruct the ICS to take an observation of type OBJECT, using the frametag. **4.** Inform the collector task that header information relevant to the end of an observation should be collected. Provide the frametag with the information. **5.** Inform the DHS that the observation using the frametag has completed.

TABLE 1.

6.2 A Group of Observations

This will be slightly different in that the grouped observations must be identified as such.

Sequence Command	Action(s) taken
LOADCONFIG MYGAL	Send command to ICS to read in the configuration called MYGAL
TARGETDESC MYGAL	Send command to TCS to read in the target description called MYGAL
SET DARK	Tell the instrument to set its configuration to that relevant to a dark, do not wait for this to finish before proceeding.
SLEW	Tell the telescope to slew to the position described in the target description. This may also include acquiring guide stars and a significant amount of interaction with the telescope operator
OBSERVE DARK	Take a DARK observation
SET OBJECT	Tell the instrument to set its configuration to that relevant to an object, do not wait for this to finish before proceeding.
DRRECIPE SPEC-QUADS	Post the name of the DR Recipe to make it available for the DHS to collect.
STARTGROUP <grpnum>	Inform the DHS that a group of observations is beginning, receive a group number in return.
POST GRPMAX=<grpmax>	Make the maximum group number information available to the DHS.
BREAK	Stop to ensure that the target is properly acquired.
OBSERVE OBJECT	Take an OBJECT observation, as in Section 6.1
OFFSET 48.0 48.0	Offset the telescope
OBSERVE SKY	Take a SKY observation
BREAKPIOINT	Optional break point for sequence
OBSERVE SKY	Take a SKY observation
OFFSET 0.0 0.0	Offset the telescope
OBSERVE OBJECT	Take an OBJECT observation
BREAKPIOINT	Optional break point for sequence
OBSERVE OBJECT	Take an OBJECT observation
OFFSET 48.0 48.0	Offset the telescope
OBSERVE SKY	Take a SKY observation
BREAKPIOINT	Optional break point for sequence
OBSERVE SKY	Take a SKY observation
OFFSET 0.0 0.0	Offset the telescope
OBSERVE OBJECT	Take an OBJECT observation
ENDGROUP <grpnum>	Inform the DHS that the group has completed

In this example the internal details of the OBSERVE commands are omitted for clarity, but they will be similar to those in tableSection 6.1.

7.0 Unresolved Issues

The following is a list of the known issues that are yet to be fully resolved.

1. The Observing Database needs to be designed.
2. Ensuring that switching instruments is rapid and easy.
3. Flagging and separation of data from different programmes.
4. Possible reuse of calibration observations.
5. How are Standards identified?
6. It is not yet completely resolved how standard “nod” offsets will be entered. The current plan is that Library Sequences (from a OT Library) will be used by the Observer, but since the Observer gets a *copy* of these, rather than a pointer to, this does not resolve the problem of them changing later. Things like standard rows can be interpreted as part of the target acquisition and their values made available by the TCS. Details of these areas are still TBD.

8.0 Acknowledgements

There are too many people to acknowledge reallly, I have chatted with a large number of scientists, TOs, software engineers and others at JAC and ROE, all of whom have made valuable contributions. However, in addition to the Orac project team of Gillian Wright, Frossie Economou, Malcolm Currie and Andy Adamson I am also grateful to Alan Pickup, Nick Rees and Dennis Kelly for significant contributions to this overall design.

A.I An Example Observation Definition

In the system presented here an Observation Definition *as read by the control system,(*i.e., not as produced by the Preparation System) is envisioned as a sequence that will be interpreted by the Sequencer. This is one of the outputs of the translator tool which has parsed the information produced by the Preparation system. This tool produces *Instrument Configurations Target Descriptions* and *Observing Sequences* from the SGML file which will then be used by the control system. In addition it produces *Observation Definitions*, one for each observation, which are also implemented as sequences.

The Observation Definition will thus be a sequence which also includes commands of various types that are to be sent on to other systems or processed internally but are not actually part of the *Observing Sequence*.

A simple example, for some jitter-style photometry:

```
! Observation Definition myPhotom1.OD
TARGETDESC target1          ! Inform the TCS of the target description
LOADCONFIG config1          ! Inform instrument system of configuration
DRRECIPE jitterPhotom       ! The DR recipe name to be put in the header.
SEQUENCE JITTERPHOTOM       ! The Observing Sequence to use
! This is the JITTERPHOTOM Sequence, expanded in detail here for convenience
! The next two commands will execute in parallel
SLEW                        ! Start telescope slewing
SET DARK                    ! Set instrument moving to its dark position
! Take a DARK, this will not execute until the SET DARK is complete
```

```
DARK
SET OBJECT                      ! Set instrument to object position
! Take object data. Will not begin until the SET OBJECT and the SLEW are both complete
STARTGROUP <grpnum>             ! Tell DHS we are starting a group, grpnum returned
OBJECT
OFFSET 8.0 0.0
OBJECT
OFFSET 0.0 8.0
OBJECT
OFFSET -8.0 0.0
OBJECT
OFFSET 0.0 -8.0
OBJECT
ENDGROUP <grpnum>               ! Tell DHS we have finished group.
! This is the end of the expansion of the SEQUENCE JITTERPHOTOM
```