United Kingdom Infra Red Telescope

Joint Astronomy Centre

Royal Observatory Edinburgh

Using the Gemini Observing Tool on UKIRT

Alan Bridger and Gillian Wright

orac006-gote Version 01 **Original: 24 October 1997 Modified: 17 November 1997**

This document presents an evaluation of the Gemini Observing Tool with a view to using it as a basis for the UKIRT Observation Preparation system. Conclusions and recommendations are also presented.

1.0 Introduction

Document [O-4] outlines the requirements for the UKIRT Observation Preparation system and an initial design model for the system. It also points out that the Gemini Observing Tool is a solution (currently in beta testing) to a very similar problem and thus might serve as a solution, or as a starting point for a solution, to the requirements set out for the UKIRT Observation Preparation system. This document considers that approach in more detail, pointing out the differences in the concepts behind the two systems and going on to recommend a course of action based on using the Gemini Observing Tool for the UKIRT system.

In Section 2.0 the concepts behind the Gemini system are outlined, as they are understood by the authors. Section 3.0 outlines some of the concepts behind the UKIRT system, particularly where they are different from those behind the Gemini system. Section 4.0 covers some more detailed differences and finally Section 5.0 lists a series of steps that would be desirable to produce an "Observing Tool" for UKIRT based heavily on that for Gemini. Section 6.0 summarises the conclusions.

Any mistakes in understanding the Gemini setup are the fault of the authors.

2.0 The Gemini Observing Tool

The Gemini Observing Tool [G-18] is an application written in the Java programming language[1]. Its principle purpose is to function as a "phase 2" observation preparation system, taking input from bona-fide Gemini observers and creating a machine readable description of their observing program. It can optionally take as input the output of a "phase 1" preparation tool (*i.e.* electronic proposals).

1. Note that it is an application, not an applet. It is intended to run as a stand-alone application, not from a WWW browser.

The Gemini OT is designed to do much more than is required of the UKIRT equivalent - it is in many ways designed to be the main interaction of the Gemini observer with the observing queue. For instance it will provide feedback on the progress of an observing program and some access to the data obtained. It will also give an indication of the timeline that an observation will follow, etc. These capabilities (and others) are beyond what is required for the UKIRT system. However, many of them might well be useful nonetheless. Thus if UKIRT were to adopt the Gemini OT then probably some of these additional capabilities will be retained.

The main thrust of this document is to examine whether or not the Gemini OT can fulfil the requirements of the UKIRT Preparation system, so these additional features, while they might well be useful to UKIRT, are not much considered here.

In the Gemini scenario a successful applicant for telescope time will prepare his or her observing using the Gemini OT, in both the classical and queue-scheduled observing scenarios. The tool needs to be downloaded to the users workstation for first use, but thereafter it will be kept up to date either automatically or on demand by the user. The tool is portable (to machines with a Java Virtual Machine implementation) and it is of course important that it is kept up to date to reflect changes in Gemini instrumentation etc.

The preparation may be done from scratch though a skeleton will already exist, produced by a compatible “phase 1” system. The files produced and used by the tool may be stored locally, and in general will be during the preparation process. When the files are ready they may be submitted to the Gemini site via an authenticated network connection. These files may also be downloaded from the Gemini site for modifying or reviewing.

The actual preparation process will be very familiar in many areas to those who have used the UKIRT_PREP system, except that it is considerably nicer and more sophisticated. There are, however, significant differences, and the first one is that the OT is designed to prepare for an entire observing program, i.e. the proposal allocated time by the TAC. One file is used to store all of the information for the observations making up a given program.

The system allows the preparation of instrument configurations, target lists and guide star positions for the telescope, observing sequences (*cf. EXEC*s), and site quality information (*e.g.* seeing conditions, dryness, etc.). Within a program the structure is hierarchical - a program may contain a number of observations, observation groups or observation folders, each folder may contain a number of observation groups or observations, and each group may contain a number of observations. The meaning of each of these terms may be briefly described as:

- Observation folder: A way of keeping related observations together - this is simply a convenience for the observer (cf. folders in mail) and has no effect on the scheduling or execution of observations within them. However, it does therefore provide means to structure the definition so that, for instance, a single definition of an instrument configuration may be used for all of the observations within a folder.
- Observation group: Observations that are in a group will be scheduled to be executed all together - with no other observations intervening. However, it does not guarantee any order of the observation within the group. Another feature called chaining may be used to constrain the order in which observations are taken.
- Observation: An observation is the smallest entity that may be scheduled for execution. An observation may consist of a single data frame or of a full set of frames, including calibration information, a mosaic of object frames on the sky and also iterations around different instrument configurations (e.g. JHK photometry).

The definitions of site quality, instrument configurations and target lists (these are observation components) may be created at any layer in this hierarchy. Inside the hierarchy a form of inheritance occurs - when a component is defined at some point then that definition remains valid at all levels below it in its branch of the hierarchy, unless it is overridden by another definition.

Thus the OT allows the observer to create the definition required for a program of observations with a great deal of flexibility and control. (Note - a much more detailed description of the Gemini Observing Tool and its use may be found at http://www.gemini.edu/science/ObsTool/index.html)

Once thus created a program file is stored (submitted) to the Gemini observing database onsite. This submission is done via an authenticated connection to a server at the Gemini site. Once in this database the file may be retrieved at any time for further adjustment, up until the time it is verified and put into the *active* database by the Gemini staff (this might typically be just before the beginning of a semester, but transfers at other times will certainly occur). After this the file may no longer be changed as the observations within it are available for scheduling. (Actually they may be modified with assistance from the support staff.) As observations in the program are obtained then new items are added to the database, and the program file, indicating the status of the observations, and indeed providing links to previews of the data obtained. This information may be viewed remotely using the OT.

The remote access to the database (which is an ObjectStore object-oriented database) uses Corba/IIOP, handled by a Corba server (written using IDL). For security reasons initial access goes via an apache web server.

When the Gemini OCS needs the information that is necessary to execute a particular observation from a program file then it needs to unwind the hierarchy in order to work out the particular configurations that are required. Having obtained the information as a series of atomic attribute-value pairs the OCS sets up and executes the observation by passing this information to the other principal systems (tcs, ics, dhs) and initiating the required actions in their control systems. (This is a deliberately very simplified brief description!). The OCS knows which values to put where and which actions to initiate by parsing the parameter definition formats that have been created by the principal systems groups.

This concludes a very brief description of how the observations defined using the Gemini OT will be executed.

3.0 The Present UKIRT system

Most of the readership of this document know the basic concepts of the current UKIRT system, so this description will be very brief.

Before observing (though sadly not remotely with the present system) the observer may create instrument Configs for configuring the instrument during observing and EXECs which control the observing sequence:

- Config: definition of an instrument configuration. It includes items such as filter required, exposure time desired, etc. It does not contain information pertaining to the target object. It is an ASCII file.
- EXEC: A user-defined sequence of observing commands, e.g. Configure the instrument, take an observation, “nod” to sky, take a sky observation, etc. EXECs contain com-

mands to control both the instrument and the telescope and act as the main interface between the observer and the control system. Typically EXECs will "call" Configs. They may also call other EXECs.

It should also be noted that Configs also contain the information necessary to take the calibration frames related to the observation (UKIRT does not have a facility calibration unit, but each instrument tends to have its own). Also note that EXECs may also issue commands to the facility polarimetry waveplate and the Fabry-Perot etalon.

In the present system there are no pre-defined target lists (except that personal lists of objects names with RA,DECs may be maintained at UKIRT for direct use by the telescope operator).

EXECs and Configs are maintained as separate files in a directory structure (one directory structure per user).

In execution the EXECs are read in and then "executed". In the course of this a named Config may also be read in and the relevant configuration applied to the instrument by passing on the relevant parameters and initiating a "configuration" in the instrument control system. Similarly telescope offsets are sent to the telescope control system to be carried out.

This system is quite similar to the Gemini approach (though much simpler). Apart from the obvious differences in scope and ease of use (between the Gemini OT and UKIRT_PREP) the main differences are:

- the UKIRT system is based around defining what is required for a group of related observations (including the calibration frames, indeed this is almost identical to the Gemini *observation*), not around defining the plan for an entire program;
- Configs and EXECs are stored in separate files;
- The "database" of observations is very simple (a VMS directory structure!);
- Remote preparation is not possible (though advance preparation is);
- As they are in separate files Configs and EXECs are directly reusable to form other observations - there are standard UKIRT Configs and EXECs and also users sometimes share (with permission) Configs and EXECs with each other ("I've got a good set of Configs for covering the H and K windows...")

4.0 An Evaluation of the Gemini OT for use at UKIRT

Given the present UKIRT setup the original idea for developing it further, via evolution rather than revolution, involved extending the use of separate files to include a truly separate EXEC (rather than an EXEC which included the name of a Config) to represent the observing sequence; a new target description file which could be used as a telescope "configuration"; and a data reduction "recipe". A complete description of an observation would then involve linking these descriptions (Config, EXEC, target description and DR recipe) along with some "scheduling information" (= site quality) to create a description of an observation. A vastly improved user-interface would also be created, along with full remote submission, a slightly better database, etc. This approach allows the flavour of the current system to be retained, and in fact it would also be backwards compatible, whilst creating an environment where more automatic and efficient observation scheduling and execution could be attained.

Clearly the Gemini OT is so close to being an implementation of the UKIRT thoughts that it must be evaluated as a potential solution to the problem. In this section the major differences in the approaches are considered and their impact assessed.

4.1 "Whole Programme"

The hierarchy is different from what we had in mind in the sense that the tool is designed to define a whole programme at once, whereas we'd been thinking in terms of defining the observation of a particular target and then afterwards building these into a programme. This is probably more a "how you think about it" than a real difference. The "programme information" can be very simple - a title, PI name and whether its for the queue or classical. At this level it is just a wrapper for the observations within it.

Within the programme you can create multiple observation definitions that each have a target name, a position, a priority, an instrument configuration, etc. - which is very similar the UKIRT model. Because the users edit and change things largely at the observation level, the fact that the results are all contained in a single file would be transparent to them. UKIRT users would get used to starting from the programme level very easily, and indeed a few regular UKIRT observers already use a different "user_id" for different programmes or observing runs in the current PREP system.

4.2 "Name-able re-useable components "

The Gemini OT does not use this concept - all the information for a programme is in one file. The UKIRT_PREP concept was to use a graphical editor to cut and paste and arrange the named components for instrument configuration, observing sequence, etc. to build up an observation and a programme of observations. In the Gemini Observing Tool you use a graphical editor to cut and paste items such as an observing sequence between programmes and between observation definitions within a programme. Each component of an observation such as an instrument configuration, or an observing sequence has a separate "component editor" that can be invoked if you want to change the individual components. In addition there are "iterators" which are a powerful tool for dealing with the problem of wanting to change only one item, e.g. on chip exposure time, in an instrument configuration (see Section 4.4).

The main differences between these two concepts is in what lies behind them, not in what the user sees or does. The user can use the OT to create a sequence called "my_jitter" and cut/copy and paste it where-ever they need to - the fact that it is not in a separate file shouldn't matter to them. Anything you can do with named components you can do with the OT, and their editor design and layout is in general very nice and powerful to use. If UKIRT PREP were to "go its own way" in developing a graphical tool for creating and tying together observation components then it would not look very different to the user than the Gemini OT component editors.

What was not so clear from the beta release of the Gemini OT is how it will handle "observatory procedures" in terms of how a user chooses them and copies them into their own programme. Gemini plan to have a library of such procedures from which the user can select the one they want. The concept behind this is again very similar to the current UKIRT use of "standard Execs" which we intended the new PREP to extend. A UKIRT version of the OT could have a library of standard UKIRT observing sequences, instrument configurations and data reduction recipes.

A requirement for the new PREP is that it will work for the existing CGS4 and IRCAM3 control software - this means it has to be able to write out ascii Config and EXEC files. The translation from a Gemini OT programme definition file into several Configs and EXECs may not be simple, though it is clearly possible in principal since all the required information will have been specified. Translation from an "observation component" into a Config and EXEC might be easier, at least conceptually.

4.3 Instrument Setup

Each Gemini instrument will have its own menu in the instrument component editor for defining what UKIRT currently calls the "Config". The basic NIRI example would adapt well to any of the UKIRT instruments, with the addition of some sort of auto-expose button.

Michelle may need to have different UKIRT and Gemini components if it uses its own calibration unit on UKIRT and the Gemini facility CU on Gemini.

4.4 Iterators

The iterators serve two purposes: making sequences of offsets at which to observe, and "variations on a theme" changes to the instrument configuration chosen. E.g. the NIRI iterator lets you change just on-chip exposure time or filter without re-defining everything else. The offset iterator lets you enter a series of offsets to observe at each. Changing the order of the iterators in the sequence changes the observing sequence. For example to change from: J+ pattern of telescope moves followed by K+ pattern of telescope moves, to: Pattern of telescope moves with J and K at each position, you would change the order of the NIRI component iterator and the offset component iterator in the observing sequence.

Although the iterators are an extremely useful and elegant way of building up sequences, UKIRT may need to limit their use for instrument configuration changes initially, to simplfy translation into a Config and EXEC sequence.

We would have to add iterators for the FP and the polarimeter. This is the logically consistent way to handle these given that we observe using them in much the same sorts of sequences as we do the telescope ("move" them, then take an exposure).

UKIRT may need changes to the offset iterator, since the observing sequences that result in the Gemini OT have offsets but no information about how the offsetting is to be carried out. Usually at UKIRT offsetting is done by moving the cross-head while guiding, however in cases where there is no guide star, or the offset movement is larger than the cross-head range the telescope encoders are used for offsetting. Having said this, it is possible that this could be taken care of by the UKIRT control system - the UKIRT TCS would decide how it was best to implement a given requested offset, indeed this may be the preferred option. There may be requirements for more detailed control, but these are likely to be engineering options.

4.5 Position Editor

This is one of the most attractive features of the tool, but it is not one of the explicit requirements for the UKIRT PREP system. (It is perhaps implicit in the sense that the new PREP is supposed to provide a friendly way of writing Observing Sequences for mosaics). The Position Editor's capabilities are far in excess of what had been envisaged for a first release of a new “PREP” - in the “nice to have, maybe add it later category”. This implementation is very nice though, and it would be hard to do better even if a lot of effort were available (which it isn’t). For making mosaics and editing them to suit source shapes it would be very useful. Given that Gemini will have a tool like this and people will get used to using it, there will be a demand for a visual display of things like map areas and guide stars for UKIRT PREP as well.

Would a simpler tool meet UKIRT’s requirements just as well? This is important because to make good use of the position editor for UKIRT we would need to make a number of changes:

1. UKIRT has only one guider so the buttons on the position editor would need to be changed to show just one position, the cross-head offsets for the guide star.
2. The tool draws the fields of things like the OIWFS and PWFS onto a digitised sky survey image centered on the observer’s field and then allows users to click on stars to select guide stars. For this to make sense for UKIRT, the items to draw on the image would be cross-head range and dichroic area. Given the frequency with which users seem to choose guide stars at the edge of the dichroic, this would be extremely useful !
3. We would also need to change the software that decides if something is bright enough to be a guide star and checks that it is in an acceptable offset range.
4. One of its most powerful uses is to display the science field onto an image. If there are a list of offsets for a mosaic it will draw the area or mark the centre of each position onto the image. We would therefore need to change the field drawn from that for NIRI (their example) to IRCAM, Michelle, UFTI etc.
5. It also calculates offsets from the chosen position to stars in the HST guide star catalogue and highlights them on the sky survey image. Using these offsets to move the UKIRT crosshead to a guide star would be fine for imaging if the precise centre of the field doesn’t matter too much, but for spectroscopy (and some imaging applications) you really need to measure the coordinate of the object and of the guide star on the same system or same image. The tool does let you read other FITS images in, so there is some accommodation for this.

Depending on how much work is involved to implement these changes, it may be that they cannot be justified at the moment, since a tool like this is not the highest priority for UKIRT PREP. None-the-less a graphical display for mosaic editing is a “wish list” item for any future development path for PREP, and if UKIRT were to adopt the Gemini OT for PREP then it would want a fully functioning UKIRT-adapted version of the position editor eventually.

Even in the short term some changes would have to be made to the position editor because if it were made available to UKIRT users with only Gemini relevant buttons available on it then it would be frustrating and/or confusing for them. Two possible solutions:

1. Initially use the OT without the position editor at all, with the intent of adding a UKIRT-relevant one later.
2. Include an emasculated position editor that has just the ability to display the sky survey image, and mark the target and science field positions on it, with the rest of buttons removed for now. This would be the preferred option. Again the intent would be to include additional UKIRT relevant features later.

4.6 Ascii Files

The current UKIRT system writes out ascii files. These have the advantage that they are easy for a support scientist or engineer to edit with an editor when the need arises, which it does occasionally for testing and engineering purposes. The Gemini SGML files are a bit harder to understand because of the markup language tags. None-the-less they are readable and editable, and the format makes it very obvious what each item is, so there should be no problem with this. User editing of observation files should not be required, and indeed even with the current ascii files it is *very* rare for a visiting astronomer to edit a CGS4 Config file.

4.7 Engineering and "Intermediate" Configurations

The Gemini OT currently has no equivalent to what for CGS4, UKIRT would call an "intermdiate" or an "engineering" configuration, as opposed to the "astronomical" configuration normally used by observers.

For CGS4 an "intermediate config" is one in which the user is allowed to select/set instrument parameters in more detail than the usual method - e.g. by selecting a specific slit by name ("-4m") instead of "1-pixel", selection of a blocking filter by name ("B4" when the default is "B6"). This allows some flexibility for more unusual choices to be made , but assumes that the user knows a good deal more about CGS4. The use of intermediate Configs is now almost wholly for engineering, although the possibility of a user needing a non-standard setup can arise at any time. The control system does less checking of the legitimacy of intermediate Configs in some areas. All of the additional flexibility for defining CGS4 Configs provided by the current "Intermediate Configuration" menu in UKIRT_PREP could be covered in the instrument component editor for CGS4 that would have to be written for the OT. For example a menu item "blocking filter" could offer a choice between **"Standard"** (which would automatically select the appropriate blocking filter based on wavelength, grating and order as in the current "Astronomical Configuration") and **"Advanced"** (which would allow the user to select the filter by name from a list). The only reason that this merging of "Astronomical" and "Intermediate" instrument configuration choices did not happen in the current UKIRT_PREP is that SMS is a very limited menu system, and it would have meant too many heirarchies cramped into a small space.

The CGS4 style "Engineering Config" which allows the positions of the motors to be set directly is perhaps more of a problem. However, Gemini are aware of the need to be able to set the instrument configuration via a low level engineering method and then take data with the higher level control system. For Gemini instruments and for future UKIRT instruments such as Michelle and UIST, the low level instrument setup will be via Epics screens in anycase. In the case of CGS4, engineering configurations are not available as part of UKIRT_PREP and are never used by visiting astronomers - they are a part of the control system menus only. This probably means that this is an issue for the eventual integration of CGS4 into the new OCS and is not relevant to the PREP system.

4.8 "Breaks" and Pauses in the Observing Sequence

Current UKIRT observing practice makes frequent use of the Exec command "break" to pauses the execution, for example after slewing to a target position so that the telescope operator can centre up on the target, fine tune the guider position etc. This is currently written explicitly into the Exec by the observer. The observing files and sequences written by the Gemini OT do not (currently) have the ability to put an explicit pause in the data taking into them. Gemini intend to handle most of the need for manual setups and interrupts to their queue slightly differently. The Gemini plan is to allow the TO to set up the execution of the observing queue to include breaks and pauses at convenient or necessary points. For example the TO will be able to set it up to "pause after a configuration and before the start of a sequence iterator". Since the UKIRT OCS will have a queue manager, this solution could also be adopted, wth the setup under either TO or observer control. For compatibility with the current IRCAM3 and CGS4 system it would be possible to implement a "rule" for inserting a "break" into Execs in the translation from OT output to Configs and Execs. This would probably result in a more standardised style of Exec - e.g. depending on the "rule", there might always be a "break" in a CGS4 Exec after a "Config" command.

4.9 Data Reduction

We would have to add a component editor for data reduction recipes. How this fits into the observing sequence needs some thought. The orginal concept was that after selection of a standard observing sequence the user would be guided/prompted with the name of the appropriate data reduction recipe. The OT already has the ability for changes in one component editor to be reflected in another window, so perhaps this can be used to allow the selection of DR recipes to be semi-automated?

4.10 Other issues

4.10.1 Distribution

The Gemini group has solved the issues of distributing up-to-date software, sending plans to the data base, etc. - we'd be re-inventing the wheel if we started from scratch. However, the Gemini solution for submission to the database does involve a commercial database which is probably beyond UKIRT's current plans (though could be a future option). In the shorter term a modification of this solution to allow use of a simpler database (a directory structure?) would need to be investigated. Even this might well leave UKIRT in the position of using a Corba server - is this a support worry?

4.10.2 Commercial Software Dependencies

The Gemini OT also uses two pieces of commercial software. Marimba's Castanet is used for software distribution and Marimba's Bongo is used as a development tool. The use of Castanet is not seen as a problem. Bongo relies on proprietry class libraries and influences the style of the user-interface. These are not necessarily problems but of course it does carry the usual worries about long-term support and that it might place restrictions on what future changes to the system might be possible.

4.10.3 Phase 1 submission

PATT are currently debating what to do about electronic proposals and there is a possibility that they will recommend adopting a version of the Gemini Phase 1 tool. Since the phase 2 tool discussed here will be capable of reading the output of phase 1 there would be advantages for UKIRT in this too if the Gemini phase2 tool were being used for UKIRT preparation.

4.10.4 Effects on the UKIRT System

Use of the Gemini OT will have implications on how the UKIRT control system will look in the future. The ORAC project is aiming to deliver a Unix-based system which will replace and enhance the current VMS-based system. Though this will be a new system it was originally intended that it would be implmented in a similar way to the current system, namely that the components of the observing definition would be separate items, probably in separate files, and that the responsibility for understanding the information contained in these components would rest with the relevant system - the instrument control system would understand the instrument configurations, the observing sequencer would understand the observing sequences, etc. Other approaches, like the Gemini OCS one of sending atomic attribute/value pairs toother systems have been considered and in the light of the Gemini OT use of a single file for all the programme information it seems sensible to consider them again. This will impact on the design of the observing sequencer.

5.0 Recommendations

The Gemini observing tool meets all the requirements for the UKIRT preparation system set out in [O-4], and provides a number of additional facilities that would be very useful at UKIRT. There are obviously benefits in terms of commonality between UKIRT and Gemini if the observing tool were used to prepare for UKIRT observing as well . Using the tool at UKIRT would save much “re-invention of the wheel” by the ORAC project, and provide a means for Gemini to get real user experience of the tool prior to commissioning. Thus this should be a mutually beneficial arrangement for UKIRT and Gemini.

We strongly recommend that the Gemini Phase 2 Observing Tool be adopted as the basis for the new UKIRT preparation system, subject to agreement by Gemini to the necessary changes and to its use by UKIRT observers prior to Gemini commissioning.

The recommended changes that would be necessary to use the OT on UKIRT are

1. Changes to the position editor for UKIRT relevancy - see section 4.5 for options.
2. Addition of instrument component editors and iterators for CGS4, IRCAM3, UFTI and Michelle, with later additions for UIST and other future UKIRT instruments.
3. Addition of iterators for FP and IRPOL control.
4. Addition of Data Reduction Recipe components
5. Production of a translator between OT output files and Configs and Execs (for initial backwards compatiblity).

In addition there should be an investigation into the following:

- the best way to interface to a (probably simpler) UKIRT observing database.
- whether the best longer term option for UKIRT is for the control system to make use of the SGML files produced by the Gemini OT, rather than continue to use a translator (as above) into separate files.

The Gemini OT team have clearly put a lot of thought and effort into the design of the OT and it is important that UKIRT and the Gemini team can agree a method by which Gemini is given full credit for this.

A clear understanding regarding support of the tool as it is used by UKIRT observers also needs to be reached. Clearly Gemini cannot be responsible for any changes or additions made by the ORAC project. Also it should not be expected that suggestions for improvements or bug reports made by UKIRT users about the OT will be responded to as quickly as they might be by an operations team, since the Gemini staff will have other priorities.

6.0 Conclusion

The Gemini Phase 2 Observing Tool is an excellent tool meeting the basic requirements for the UKIRT Preparation System and coming very close to the original concepts for this system. It is recommended (subject to agreement by the Gemini team) that the Gemini OT be adopted by UKIRT for its preparation system. Some changes to the tool itself will be required to reflect its use on a different telescope. The slightly different paradigm in use by the Gemini OT will require some changes to the design of the UKIRT Observatory Control System.