United Kingdom Infra Red Telescope

Joint Astronomy Centre

Royal Observatory Edinburgh

Report of the ORAC reviews - June 1998

Gillian Wright

orac017-rev2 Version: 01 **Created: 2 July 1998 Modified: 15 July 1998**

Comments made at the review meeting which discussed the ORAC designs and requirements presented in orac012-ocsd, orac013-seq2, orac014-chor, orac015-oiis

1.0 Purpose

This document presents a brief summary of the points that were raised at the ORAC review meetings on June 23rd 1998. The document is intended to be read by anyone with an interest in the ORAC project, although knowledge of the documents that were reviewed is required.

2.0 Overview

Present: AB, GSW, MJC, TGH, BDK, DAP, SMB, MJS, MT, ACHG, IRB, JMS.

AJA, TRG and SKR sent comments on the documents by email. SMB also sent some comments by email which were discussed at the meeting. Email comments which make additional points to those made by various people at the meeting are included separately.

Further discussions covering the same documents took place during Nick Rees' visit to the ATC in the first 2 weeks of July.

3.0 Overview Design

The meeting started with discussion of the Overview Design (orac012-ocsd) document which outlines the main ideas and describes how the system fits together overall.

Following the design reviews and discussions in December at ROE and February at JACH the design has changed so that the Sequencer is now simple in the sense that it has no knowledge of the other systems, with the intelligence moved up into the Chooser. BDK expressed concern that the knowledge of which parts of an observation definition are required by which subsystems could be being implemented at a higher level than necessary in the code. He suggested stripping out the relevant information from the SGML into a single file of keyword-value pairs and then letting the lower level software pull out the keywords it needs. However it was noted that sequences are not keyword-value pairs, and would be difficult to accommodate in such a design. The way the split up is done is partly historical and partly due to the use of the Gemini OT. AB and BDK will discuss further outside the meeting (Action 1).

GSW commented that she was concerned that by moving all the intelligence into the Chooser and doing the translation there, the process of starting the next observation definition would seem slow to the observer if they had to wait too long while the translation etc. happened. This was deemed to be unlikely - on a sufficiently powerful machine it is expected to be very fast.

The issue of data file names and UT date changes was raised - discussion was deferred until the DHS review in the afternoon.

It was noted that groups starting (and ending) would be kept track of by the DHS and agreed that this was a sensible way to proceed. AB will modify the document to reflect this (Action 2).

It was noted that the sequencer needs to know whether or not it is “allowed” to talk to the telescope - this will have to be handled through the user-interface to the chooser, so that if an observer uses the chooser to run observation definitions for two instruments at once they can specify which one is being used for astronomical observations.

In the discussion of unresolved issues, ACHG suggested that standards (e.g. spectroscopic ratioing stars) could be identified by having a “button” on the user-interface to the chooser which the user could use to flag stars which will be used as standards - if the “button” is selected a flag gets written into the header. This will also work as a flag for photometric standards to be used for re-calculating zero-points.

4.0 Sequencer

Comments on the sequencer design document orac013-seq2 followed the discussion of the overall system. AB suggested that any suggestions for changes to the names of the actual commands in the scheduler (e.g. OBSERVE) should be sent by email, so that the meeting would concentrate on the design.

It was agreed that the list of commands given in the document would be extensible - so that new commands could be added if the need arose.

For commands such as ABORT it was agreed that the sequencer should send the ABORT to each subsystem, which would be expected to carry out the appropriate action. The document should be updated to note this decision (Action 3). This would mean that if the user selected ABORT while the telescope was slewing the slew would stop, an ABORT while the instrument was taking data would ABORT the observation, - the action for an ABORT while the instrument was configuring would (for current UKIRT instruments) mean stop when the configure is complete, as IRB pointed out that stopping motors moving “half way through” would mean that the next action would have to be to datum them.

It was agreed that the sequencer would not itself have timeouts for commands. Rather the system to which the command is sent should have appropriate timeouts and the sequencer respond to the timeout error on the subsystem by reporting it.

There was discussion again about editing sequences “on the fly” (the “edit current Exec” functionality at present). Discussions at the previous reviews had agreed that this was not needed for normal observing given that “infinite” loops are possible, but it was not clear whether it was needed/wanted for engineering. It was agreed that it was not necessary for engineering.

There was discussion of the limits of the requirements for parallelism in the sequencer, and the need for only one command to be sent to a subsystem at a time. It was agreed to make the table in the document more explicit about what could/should go in parallel with what - IRB suggested re-casting it as a matrix (Action 4).

A number of errors in the names used in the example sequences were pointed out - these would be corrected.

BDK and DAP presented two possible solutions for the sequencer. DAP has re-worked the existing Exec software into a Drama task which, as specified in the sequencer design, uses a “look-up table” to define the nature of the tasks to be communicated with (e.g. ADAM, or DRAMA) and the system they run on (e.g. VMS or Unix). He has also been able to incorporate the requirement for parallelism by introducing parameters which specify when tasks must be completed before the next can proceed. BDK has been developing the JCMT Todd in JAVA to allow parallel operation of instruments and telescope. This would also meet the requirements and is designed to allow easy manipulation for support scientists to make changes. GSW expressed concern that at first glance both might be un-necessarily complex for what the requirements were. (It no longer looked like a “simple” sequencer to her). GSW will discuss how the systems work with DAP and BDK outside the meeting (Action 5). It was noted that which route to follow would be discussed in detail with NPR, since the two options have very different pros and cons, and support implications need to be taken into account (Action 6).

5.0 Chooser

This was followed by a discussion of the Chooser requirements: orac014-chor. GSW summarised these from the users point of view and AB drew a sketch of the sort of user-interface the ORAC team had discussed with the staff at the JAC in February.

This review group did not like the additional button that allowed over-riding of the object name after an observation definition had started, and felt that it ought to be unnecessary if people had properly prepared their observations beforehand. GSW and AB said they hadn’t particularly liked it either, but pointed out that the JAC staff had felt very strongly would be necessary e.g. because people preparing observation definitions at the telescope could make mistakes with the object name, and they felt it was a valid concern in the “non-ideal” world of occasional “on the fly” observing. DAP pointed out that while changing the object name in the header for observations that had not yet been written to disk was possible, if the observer changed the name after the first few observations in a group had been taken (i.e. when they realised their mistake) the header could not be changed in real time by the DHS for the observations that are already on disk. He suggested as an alternative the use of a field in which the observer could enter comments. Such comments might be “ooops called this the wrong name its really....” in nature, but could also be useful for other things (“I think its getting cloudy”). He suggested that the DHS could write a time-stamped log (ASCII file) of each observation as it were taken to which the comment lines would be added. This was generally thought to be a good way of meeting the requirement. Further comments, invited (Action 7).

The menu which reflects the observing tool display, from which the observer chooses which observation definition to do next, should mark somehow the ones that have previously been done, without preventing it from being used over again.

There was some discussion of whether the Chooser needed any special facilities for remote observing. TGH said that so long as the system could run in full remote mode to control the instrument from Hilo (as is done presently) and there was nothing that would explicitly prevent either all or some of the displays being remote was sufficient consideration of remote observing needs.

6.0 Instrument Interfaces

There were very few comments about orac015-oiis which is a draft specification of the interfaces that new instruments would have to meet. It was noted that the items appearing for “astronomer” selection on the OT (such as wavelength and grating or resolution), would be what was in the “config” sent to the ICS. Thus, like the current system, the ICS would be expected to work out items like internal focus position and filter wheel positions from the information given. It was agreed that this was the right approach to take. The OT also allows for “hidden items” - those which the observer does not usually want to change, but might sometimes have a reason to over-ride an automatic choice of (e.g. choice blocking filter, or order for a grating). Since such items were available in the OT they would be in the Config sent to the instrument even if the “default value” were used. It was noted that the division between which instrument parameters should be available to the observer and which dealt with by the ICS only had some “grey areas” - but there was nothing in the design that prevented it from being done one way or the other.

7.0 Data Handling System

The DHS is strictly part of the Michelle project, but has to interface smoothly with the rest of the ORAC design, and the two have to be designed to work together. Michelle also has DHS requirements for its use on Gemini. Requirements for the Michelle DHS were reviewed after the ORAC meeting, with a slightly different group of people present. The following is a brief summary of points that are particularly relevant to the ORAC design.

As with the sequencer when two instruments are being used there will be one DHS per instrument.

It was suggested that the file naming convention be changed so that a shortened version of the instrument name was include in the file name e.g. f-mch19980610 for Michelle. It was agreed that this was a good idea in principal since the file is then tagged with the instrument name even if removed from its directory. AB will check with FE whether or not changing the prefix is a problem for ORAC-DR (Action 8).

JS noted that the filenames are getting long for those who are not running DR scripts.

As currently designed the DHS automatically changes the file name when the UT date changes. So if the system is being run in the afternoon in Hawaii (e.g. for engineering) the filenames will change at 2pm. Whether or not this is desirable was the subject of some debate. The implications for ORAC-DR need to be examined by FE and AB (Action 9).

A related issue to file names is the naming and directories for engineering data. GSW pointed out that identifying engineering data was an example of a more general problem of identifying data belonging to different runs, which the ORAC project was charged with suggesting a solution for. As currently designed if there has been daytime engineering the file numbers for the night time observer will not start at one. While breaking with tradition at UKIRT this should not be a problem so long as the observers data can be identified (for half nights one observer does not start with observation #1 at present). It was concluded that some further thought on filenames and directories and discussion with the JAC was needed. GSW will ensure this happens soon (Action 10).

The DHS will be capable of writing the headers in the “human friendly” order described in the UKIRT FITS headers document, with blank line separators.

It was noted that UKIRT would also require WCS information to be written into the FITS headers of the files, although the calculation would be different for UKIRT. DAP pointed out that for UKIRT the calculation can be handled by the DHS drama task, but on Gemini it must be done in EDICT.

8.0 Additional comments from Andy Adamson

There were some comments in the email from AJA which were not covered during the discussions, these are noted here with some responses.

OCS Overview :

3.10.3 - are there other examples of frames which will have multiple entries ? I was going to suggest storing the star-sky subtracted frame as the top level data array and storing the two halves of the chop separately (in the same file of course), so that standard data reduction could be carried on despite the observer asking for both halves.

response : Good idea. Another example would be a shifted&added frame for which the observer would also like to see the frame of shift&add vectors, as well as the final data. A better known example might be the set of interleaved spectra that are taken to achieve full sampling with a spectrometer that undersamples in a single shot. CGS4 is a clear example, though we may not retrofit, but it is also possible that this may happen with Michelle (we hope it won’t, but it might for various reasons).

3.10.4 - data files occurring before they’re complete; yes, temp files are one solution, but aren’t file locks another, which would at least require less copying of data around the place ? i.e. the dr system sees a new file there, and goes into a loop to open it which stops once the file is completed and unlocked by the acquisition system ?

response: We’ll investigate (Action 11). However, I wasn’t envisaging a copy of the data, merely the equivalent of a unix mv or VMS rename, i.e. the data stays in the same physical place.

page 13, normal completion - why not just have a keyword ENDGROUP or some such in the file header, which is TRUE if the file is last in a group, and FALSE otherwise? the dr may be aware enough to tell if normal completion has happened, but it would be optimistic to assume that someone re-reducing offline will be as savvy. I mean people doing things manually with figaro etc. - a header item which explicitly flagged an end would be useful, maybe, whether or not the dr uses it, and could also be used in those pathological cases on the next page?

response : The GRPMEM and GRPMAX keywords solve this for normal completion, for other cases the problem we've found is in coping with what happens when the observer manages to stop between observations - through lucky timing or getting in during a telescope offset or something - you cannot affect the header of the data that has just been taken, only that is still in progress or yet to come. This would mean that sometimes an ENDGROUP=T might get into the last frame, and sometimes it wouldn't - this inconsistency is potentially confusing for someone looking at the headers.

page 16 - suspect that at least in the early days, you'll want to make engineering stuff available on a quicker timescale than would be involved in running down the observer's OT version and starting up the jac version. how about a passworded button to bring up a screen of engineering functions within a single standard OT? *response*: Will investigate. Note that you wouldn't need to run one down, the run-up is fast, and we mean here only the preparation of engineering test sequences, not engineering level access to the instrument or telescope control for troubleshooting during observing (which the TO will have via the Epics database).

page 16, para 3 for my tuppence worth, i think you'd be much better not implementing a direct user interface to the sequencer.do it through the OT, and make sure all the commands are available there.

page19 - DRRECIPE SPEC-QUADS - this seems to be a posting operation ? if so, shouldn't it be POST DRRECIPE=SPEC-QUADS ?

response: Probably it should. I'm still thinking of exactly how this is implemented - perhaps "POST"s are just commands to the DHS, but the general idea seems acceptable.

Sequencer :

- tuppence worth - go for the drama system. sounds like a real risk to go with something as developing as java. *Response* : OK. This I intend discussing a lot with Nick in the coming week or so. There are pros and cons, but I take your point very seriously. The main pro is that JCMT will have the same system.

- observer giving commands direct to the sequencer - not sure this is a good idea. shouldn't they always go through the chooser? if you decide to allow engineers to talk direct, then perhaps an engineer could be added to the context diagram, with the observer talking to the chooser and the engineer where the observer currently is? *Response:* I agree, this is a left over which predates the introduction of the chooser.

page 6, point 7 - will called sequences still be expandable ? *response:* Yes, they should be. How to implement this is still in debate, but it should be possible.

- 4.8 - one command at a time - bit restrictive if you have an instrument which is capable of moving various motors at once. better make it clear in documents that the designers of ics for such instruments should make a "config x y z" command available ? *response:* The intent is that a "set" will be a single command which will allow the ics the freedom to move several things at once. Having said this I think the review meeting indicated I should rewrite the description of parallel commands and also that this restriction can probably be removed (both possible sequencers are flexible enough).

4.9.4 - why not have an abort table which lists, for each currently executing command, what should happen if it is aborted ? *response* : yes, good suggestion.

4.9.6 - beware multiple instruments? *response*: Indeed. It shouldn't be a worry.

9.0 Additional Comments from Suzie Ramsay

The meeting was held while SKR was away, so her comments were sent by email. Those which were not covered during the discussions are noted here with some responses.

General:

There are lots of comments throughout the document about engineering functions, and I want to add my support for well-thought out engineering functions. A few of the comments are a bit worrying - e.g. p15 of the overview section 4 about ORAC not having much engineering functionality. Also, I hope the 'JAC' verions is a UKIRT version and not just a Hilo version.

Response Yes. JAC in this context means "the version available to JAC staff" - not where it is running. You would run the JAC version here for instrument tests. The reference to not much engineering functionality is not quite correct. ORAC overall does provide for engineering, and it is the intent that engineering sequences of the "datum, take data, move motor x, take data, wait 20 secs,....." type will be run in the same way as any other sequence. What this comment was really referring to was the fact that at the moment the Gemini OT, which we will be using to create the sequences as part of the observation definitions, does not have any engineering sequencing commands (but they are planning to do something like this soon). So the problem is not how you would use ORAC to run an engineering test sequence/take engineering data, its that right now there is no way to write it unless you edit SGML. We've also been coming to this conclusion over time, so in the documents there are probably fragments of earlier thoughts that engineering might have to be a separate function.

Are we going to be able to run a subset of ORAC in the lab here in Edinburgh, so that when carrying out the engineering tests the interfaces to the data-handling, data-reduction etc. are all as they will be at UKIRT? *Response* Yes that it is the intention.

I believe that the software used for engineering should be as identical as possible to that used for the science. There are a few places in the text where this issue comes up.

Response - After considering this over time we've found that we agree wholeheartedly! The stumbling block for some time was how to do it in the context of the Gemini OT, and we feel we're now working out how to do this.

- Feedback from the DHS to the TCS. Since the DHS is decoupled from the TCS, there is no opportunity to have the DHS deduce a number (for example, which pixel is the object on and how far to I have to offset to get it on the slit) and get the TCS to act on it. So, how would I do this for UIST acquisition? If I get the DHS to calculate the number, can I insert it into the observing definition?

Response. Assuming by "DHS" you mean ORAC-DR - then you are right that the telescope can't be sent it, because the DR is completely independent. If you wanted to you could put an offset x,y command into your observing sequence part of the observation definition. If its a number that stays constant then that might be the way to do it. If it were constant for all runs, not just yours, then it could be put into a standard sequence. However if its something you have to measure every time you slew the telescope then the TO would enter the value you measure. Also note that there is an issue about whether or not you need to do real data reduction and fits to an image (which will require ORAC-DR) or whether the offset can be calculated in some simple way from a raw image (in which case maybe quicklook could be extended to do it). We are assuming in the ORAC design that you need real data reduction to do this, and plan to have suitable scripts.

Acquiring an object via and image and then moving it to be aligned with the slit will be a big change for MICHELLE and UIST when compared with CGS3/4. I tried to think how this fits into the observation definition. I suspect you will have thought this out, but it occurred to me that acquisition might be a good thing to have as a standard sequence. The steps you want are:

1. load the instrument config (for spectroscopy)
2. get the target
3. slew to the target
4. adopt the instrument config
5. take a dark
6. move the slit and grism out (uist), select the pick-off mirror (michelle)
7. take an image NB with different exposure times than the spectroscopy
8. offset the telescope to acquire the target
9. reset the config for spectroscopy as step 1.
10. start an object group

For all targets, steps 6-9 should be common for acquisition, perhaps with difference exposure times and read-out modes, and could be a sequence, couldn't they? The exposure times and perhaps read-out modes will be different than for the real observation. And I can think of reasons why you might want to change filter for the acquisition. Could there be an 'acquisition' config associated with the object config, the way there is a dark config associated with it? I'm thinking aloud, here - I should check what you have planned.

It would be just great to be able to enter the object configuration file and the target and have ORAC handle all the image taking and offsetting without involvement of the observer

Response: Yes, this is exactly the sort of thing we want to build standard sequences for. The project scientist or support scientist (who would be familiar with the instrument details and the OT) would be able to use "iterators" to define a sequence that iterated the items you mention (in fact anything in the instrument configuration), which observers could then use. In this scenario only offsetting the telescope would be a "manual" step. For simple acquisition (e.g. point sources) this would probably be handled by the TO using an ORAC-DR script in much the same way as we anticipate the setting of the telescope to the best focus will be done. It also means that for sources with complex morphology its easy for the observer to have the option of selecting a position by hand if they want, the offset for which would again be entered by the TO.

Telescope pointing. UIST may have a pointing model specific for UIST alone, required to remove the effects of differential flexure between the guide camera and the UIST slit. How can this be handled within ORAC?

Response: This a problem for the specific telescope pointing model - which ORAC does not know anything about. i.e. once you have acquired your target and are guiding, it affects how the telescope tracks, not how ORAC acquires data. One likely effect is that ORAC may have to ensure that the telescope knows that it is UIST that is in use and thus the UIST pointing model is required. This would be straightforward.

Communications. Perhaps I'm over-intepreting these flow charts, but the one in the instrument interface document shows the ICS communicating with the TCS, but the one in the overview document shows not such communication. Is this just simplifying the diagram? Will instruments talk directly to the TCS?

Response Basically it was to simplify the diagram. Not much communication goes between the ICS and TCS. At the moment all we are aware of is chop synchronisation (or control). It is plausible that other types of interaction could happen here, e.g. an aspect-like scanning. Essentially anything at this level would be an observation *mode,* i.e. happens as a result of the OBSERVE command (and wholly within it).

Specific comments on orac012-ocsd

p4 - will full remote observing be possible with ORAC if UKIRT ever goes that way? *Response*: Yes

p7 - what's SGML? can it go in the glossary? *Response* Ooops. Standard Generalised Markup Language, yes.

p7 - data reduction recipes in the header. I made comments about this in my reply to the MICHELLE DHS doc, saying that it will be hard to stop people doing daft things like leaving the wrong recipe in the obs. definition. Can the DHS cope with this and let people re-reduce a botched set of observations?

Response. Data reduction (ORAC-DR) is completely independent of the DHS. One advantage is that if you get the wrong recipe name into the header, inspite of all the user-interface protection against doing so, then you can re-reduce your data by specifying the name of the recipe to use on the startup command line. None-the-less it is our intention to use the OT in such a way as to prevent this as much as possible.

p8 - I had a question as to whether calibration frames could be shared. On reading further I find that this is TBD. It's highly desirable. I think the most compelling case might be for service observing - you might well want to share a load of calibrations between observations taken for a number of different observers. I think this might test the ORAC system, especially your thinking about how to flag and separate data for different observing definitions.

Response: Noted. We agree its desirable, but haven't settled on the best solution yet.

p9 - information on offset guide stars. Nick and I were discussing the possibility that the colour of a guide star may be important for UIST guiding. This information would have to be passed to the TCS.

Response There's nothing to rule out adding guide star colour to the observation definition. It would be passed to the TCS which would have to know what to do. The observation definitions can handle this - the problem is whether or not the colour information is available in the catalogues.

p12 content of data files. This part really had alarm bells ringing! I suspect you might be thinking of storing data from different chop beams in multiple arrays in an NDF or FITS structure or some such, are you? I had data from ESO once which was stored like that, and it was awful. Mostly it was terrible because the software at the telescope could only handle one of the arrays, so I could only reduce one third of the data! As long as the DHS can handle it, I guess it's OK, but how does this interface with the fact that people can write reduction scripts in whatever package they fancy. I'd like to know what you're planning, here. Apologies if it's not as bad as I think! Also, does anyone really want all that chopped data? and can we handle storing and transporting it from ALICE/EDICT? This could be too much hard work.

Response. Well hopefully it will really not be as bad as you think. Yes there will be recipe that is capable of reducing the two chopped beams. Unfortunately there are some people who really do think that they still need the ability to write the two halves of the chop beam separately. The default would be for the difference only to be written (and we hope that the difference will always be written as the main data frame even if the other beams are also there), unless a project scientist/support scientist found out that this would not be enough for their instrument. So hopefully this is provision of a facility that's needed for flexibility but not often used. Other data reduction packages can cope with FITS extensions, so if in the future an observer wanted to run IRAF reduction, then they would need to write a primitive for combining the two beams, with support from JAC - i.e. it could be done in principal if it was needed. Note that initially all ORAC DR scripts will use Starlink software and NDFs.

p13 3.10.7 relating groups of data. Really 'maximum no in group' or actual number, as I picked up from the DHS doc. And didn't like. MAX is OK - presuming you can set it to a daftly large number.

Response If you are doing an indermined number (the go until you decide to stop scenario) then GRPMAX will be set to 0. You may also set an unrealistically large number. In either case you STOP when you've got enough. We are also looking in to whether or not an underestimated number can be implicitly increased, i.e. extend the observing sequence. We think this is desirable, but its not trivial.

3.10.8 determining calibration frame. Can you ever over-ride having a sensible calibration? This may depend on what you decide about shared files, I guess. And on similar lines, is there much checking of the obs. def to see that it comes with the correct calibrations? complicated, that.

Response. Yes and yes. However if, for example, you want to use an inappropriate "dark" then to over-ride it you would use a recipe that called the "subtract a frame" recipe rather than the "subtract a dark" recipe. Or you can choose or write a recipe that does not do that calibration step. If you have a bunch of suitable darks and just want to specify using a different one from the auto-selection then you name it on the command line. Full checking for the correct calibrations depends on there being a scheduler - but there will be a lot of help at the observation definition level (e.g. reminders when you select a DR-recipe about which calibrations are needed).

p14 Aborting. I think it would be good to store even aborted observations. Weeks later when the rest of the data look odd, you might have forgotten that you had aborted one. (With reference to comments on the DHS, I am now happier/understand about the handling of incomplete groups)

Response. Experience at the JAC suggests that aborted observations that are saved tend to cause problems, because they are weird and people forget! They also cause problems for the current on-line data reduction. Since the file you aborted doesn't exist, then since the next data frame that is not aborted will be after the problem has been fixed there should be no weird data. If you find the problem after taking several bad frames then its no worse than now, perhaps better, since there will be a chance to add a comment to an electronic log that will get saved with your data.

Comments specific to orac013-seq2

p7, Table 1: I think I eventually figure out how all the various 'stops' happen and that there isn't actually any redundancy, though it appears that way at first. However, if you want to stop in a hurry, it should probably be a one word command, not something with a parameter. I don't know enough about how this is to be implemented.

Response. It will be from a GUI. Details are TBD, but it is unlikely that it will ask you to enter a parameter - it will just look like there are different types of stop.

10.0 List of Action Items

1. AB and BDK to discuss the level at which knowledge of which parts of an observation definition are required by which subsystems is required.
2. AB - Modify overview design to reflect agreement that having the DHS keep track of the starting (and ending) of groups will be all that's needed
3. AB - design of sequencer to specify that ABORT should be sent to each sub-system which would be expected to carry out the appropriate action
4. GSW/AB - make the table in the sequencer design document more explicit about what should be allowed to go in parallel with what. (matrix).
5. GSW - discuss with DAP and BDK how their possible sequencer designs work.
6. AB - discuss support and other implications of sequencer design choice with NPR.
7. All (especially JAC) - consider and comment on the suggestion that the DHS write a time stamped ASCII log file of each observation as it was taken with the user able to enter (short) comments from a field in the Chooser user-interface.
8. AB - check whether the changing of filename prefix to reflect instrument used is a problem for ORAC-DR
9. AB/FE - investigate the implications for ORAC-DR of filenames changing automatically at 2pm (in Hawaii) when the UT date changes.
10. GSW/All - have further discussion of directory names, identifying data belonging to different programmes, identification of shared calibration frames.
11. ORAC team - consider use of "file unlocking" instead of "file appearing" as a way for ORAC-DR to know when a new data file is ready for reduction.