UCLA Astrophysics Program **George Brims**
Infrared Imaging Detector Laboratory **September 15, 1994**
4444444444444444444444444444444444444444444444444444444444444444

Keck II Software Workshop, Sept 1994

CARA hosted a meeting in Waimea from the 7th to 9th September, to discuss the future of Keck software, with particular reference to leaning from the lessons of Keck I and applying them to the design and development of Keck II. I was invited to represent the NIRSPEC team. Below is a series of notes I made on the progress of the meeting and important points brought up. I also have a large folder with a large quantity of material supplied before and during the meeting. Some of it is in the form of introductory documents or reprints of papers on various compnents of the system, some of it is copies of viewgraphs presented at the meeting.

At the end I have given my overall impressions and conclusions.

Day 1 - Wednesday 9th

Introduction (H Lewis)

9:00 Hilton Lewis (HL) welcomed everyone and outlined the goals of the meeting.

Session 1: Keck I Experiences (S Djorgovski)

9:10 William Lupton (WFL) gave an overview of the Keck I system. He brought up one compromise we need to make in specifying the Keck II infrastructure, between having as much in place as possible from the start, and having the flexibility to add to and modify the system as available machines and software evolve and requirements change. There is less conflict between these two equally essential qualities when there is a simple infrastructure and good isolation between layers. The Keck I system is apparently not well implemented in that regard. A huge fraction of the CARA staff's time is going on maintaining the Keck I system.

Jim Westphal (JW) made the point (WFL concurring) that better availability of simulators will reap benefits in eliminating problems before systems are brought into general use. Cited example of NASA running a huge simulation of the HST software for months and finding numerous unexpected bugs. Al Conrad (AC) pointed out that many of the problems they have been fighting have been of the sort that only show up once the whole system is running together.

Keith Matthews (KM) voiced the opinion that the Keck I system just plain isn't finished yet, and that there are still "years of work" to be done tying up loose ends without spending time on developing a "grand scheme".

9:55 "George" Djorgovski (GD) emphasized that he considers the software group at CARA to be talented and hard working (comparing favorably for instance with Kitt Peak). The bad news however is that they aren't finished yet. The telescope user interface for example is not simple and user-friendly. He particularly dislikes the amount of engineering work and monitoring that goes on during observing time.

10:00 J McCarthy (JM) discussed user interface problems, but since this discussion started to drift into too many particulars it was cut off by the chair.

10:02 Bob Kibrick (BK) gave a summary of responses he got when he asked for comments from UC and Caltech astronomers who have used the existing system. Some of the complaints were; there are still a lot of crashes; time lost per crash is from 20 minutes to an hour; there's a domino effect where one component crashing can necessitate rebooting several; the operator's display is too complex; many vital functions are too slow; too much knowledge of the internals of the software is needed to observe; many of the Motif user interface features are non-intuitive; it's hard to tell which window has the input focus and there are too many windows in any case; desired and actual values are in different windows and new ones are not automatically applied (and it's easy to forget to apply them); many important functions are not available as simple to use items, so modifying keywords individually is required (very similar to the complaints about ADAM in the early years at UKIRT).

Other problems mentioned in open discussion concerned the guider, in terms of both usability and performance. The setup of operator and observer and the location of their displays was generally recognized as needing some work. They sit too far apart to easily share information and duplication of the operator's display windows on the other screen is suspected of causing slower performance and/or crashes so it's generally not done. There was also some discussion of the ability of the observer to know how the seeing, pointing and stacking are degrading or improving during observations.

It was generally agreed that better user guides are needed, and in fact work is in progress on this facet of the system.

There was some discussion of the different data file formats which have to be used (FITS, Figaro, IRAF) and the loss of performance whenever file conversion has to be done, the extra disk space required, and the time taken to get a backup tape of data.

A point made strongly by BK and reiterated in a later session was that there is a need for instrument teams to have time and funding *after* delivery for a post-ship review and support for the CARA team. If we stick to our schedule and don't overrun we will in fact have some time and money to do this. I made the point that a "dry run" delivery of the interface software and possibly a transputer test station would probably help reduce the need for post-delivery support.

Session 2: Keck I Experiences (S Djorgovski)

11:20 Joel Aycock (JA) showed some viewgraphs of the screens he has to use to operate the telescope. The operators have way too many things in their field of view. They have three physical screens with many windows in each. Among his criticisms were the number of operations (in different windows) to perform a single slew and acquire. It generally takes 5 to 10 minutes to go to a new object (the physical slew takes a maximum of 2). A lot of operations are fiddly and require a lot of moving between keyboard and mouse. Different parts of the system are run under different operating systems (VMS and Unix) and have subtly different windowing setups. They even have to deal with the different keyboards having different delete keys!

A measure of the immaturity of the system is that the operators find it easier, when one component fails to bring down and restart the whole system, than to find out and perform the correct sequence of operations required to bring it back. This loses a half hour or more, and additionally knowledge of what caused the original problem may be lost by shutting down, so the same problem may well crop up over and over again. The problem is that the operator is trying to work with observers looking to have the system back on the air as soon as possible, so has no choice but to follow what appears to be the least disruptive course of action.

Joel did concede that the system is getting better in that some unnecessary information has now been removed from the screen. More work is still needed to reorganize the screens to put information that belongs together in the same windows. There is a lot of information appearing in some of the status windows. As time goes on much more of this could be hidden, but for now it's hard for the operator to tell some of the routine messages from important errors. There is some use of voice synthesis using the audio hardware of the Sun workstations, and it was felt that this could be used more for drawing attention to important error messages.

Open discussion: A lot of the problem with the user interface is a consequence of the separate development of a number of different systems. The tools available and the move toward a more unified infrastructure (using MUSIC and KTL) now make it easier to improve some of the user interface aspects, and work is continuing on improving the interface. However there is a much bigger problem when it comes to making some real changes at a lower level. For instance adding a new keyword to the DCS requires changing *7 or 8 files under 3 different operating systems*!

There was some discussion of how appropriate it is to try to re-engineer everything now. My opinion is they should build a better Keck II system and simply maintain Keck I until Keck II is fully debugged then do a retrofit. They probably don't have any other choice in any case.

Bob Kibrick emphasized the issue of feedback (of e.g. seeing changes) to the observer leading to better observing strategy.

Al Conrad pushed the idea of using remote observing to increase the amount of time software staff can spend participating in or eavesdropping on live observing, rather than testing in isolation. Software engineers can do a better job of debugging if they see the errors *as they occur*.

Hilton Lewis noted that most complaints are now concentrated on efficiency not reliability, since they were given a mandate to concentrate efforts on reliability this year. JA concurred that even in the six months since he joined Keck he has definitely seen improvements in reliability.

Keith Matthews asked what is seen as the underlying cause of the remaining unreliability. HL responded that the main causes were the lack of isolation between different components of the system, and the unanticipated complexities of tying together such a distributed system.

Session 3: EPICS introduction (William Lupton)

13:35 William Lupton gave an outline of the EPICS system. This is described in his viewgraphs and in papers in the information folder. EPICS has not yet been formally adopted but it seems likely that Keck II *will* use it for distributed control.

An important distinction between EPICS distributed database records and KTL keywords is that KTL keywords only exist at a higher level to provide abstraction between the low-level world and the user interface or other apps. EPICS database items correspond closely to real-world items such as I/O devices or channels, so they're generally much lower-level and also less abstract.

Session 4: Other products (Al Conrad)

15:30 Al Conrad described the MUSIC system, developed originally by Rick Stover at Lick, for intertask communication. Supporting material is in the information folder. MUSIC forms a part of the Keck I system, and HL and WFL both stated that had they been able to use it more, then things would have been easier.

16:15 Rick Stover (RS) described the TCL script language, developed by John Ousterhout at UC Berkeley, and the X-windows toolkit Tk he built using it. There are copies of papers on these systems in the information folder. These two software libraries are potential components of the Keck II infrastructure.

16:30 Al Conrad described DataViews, a user interface builder program which can be used on Unix, VMS and Windows NT systems. The product looks impressive. Keck have negotiated a software licensing arrangement whereby CARA is considered the host site and the rest of us are charged for second, third etc. copies rather than for a whole new license. Additionally they give a 50% overall discount. This brings the cost down to $1500 for us. One sore point is that there is a charge also for each target system running the resulting code, although later in the meeting AC talked to them during the lunch break and talked them down on that price. CARA will also probably pick up the tab for annual maintenance contracts.

Day 2 - Thursday 8th

Session 1: EPICS and Keck (Hilton Lewis)

9:00 William Lupton opened the discussion on how EPICS would help simplify things and how it would fit into the DCS. He also discussed how it would fit in with the optical instruments. For the latter it doesn't seem to offer as much in the way of simplification.

HL made the strawman proposal that we all agree to use EPICS to run everything on Keck II, including the new instruments. A lot of skepticism was expressed and a lot of questions asked. EPICS came out of Los Alamos and Mary Fuka and Perry McGee at JAC brought it into the telescope control community.

Nick Rees (the other EPICS guru besides WFL) pointed out that use of EPICS brings in the engineers and allows them to be more involved in configuring, testing and maintaining the system. He explained how they plan to use EPICS in the UKIRT telescope upgrade. On the down side his pet peeves are poor release control, bugginess and poor documentation. LBL are now working on a new release of the system, and he takes it as a good sign that they are running a little late with the release due to the amount of testing they are doing. His recommendation was that one should spend some time with people who have actually used the system before committing to it. They had the LANL people come over and give a one week course for UKIRT and CARA people. We will probably have another similar course for people at the workshop who didn't attend the first one. I suggested there might be a cost saving in hosting that in California, but it would probably have to be at Lick because of the hardware needed for demos.

Keith Matthews and Colin Fry were very skeptical that EPICS and KTL will give them everything they need for real time control, particularly since they have to handle a big data rate using 1024 arrays on NIRC2. (It was pointed out several times that EPICS is really for auxiliary control of hardware rather than handling the bulk data flow). They were very adamant about being independent and not being made to change things that "already work". (Since they don't yet have a drive system I don't know what the things that "already work" are.)

Session 2: Keck II instrumentation software (Bob Kibrick)

11:00 BK described the new instruments (a quick summary).

11:05 I described the transputer and the system we are proposing to use for NIRSPEC, and the way we are anticipating interfacing it. GD brought up the question (irrelevant to this meeting) of how we were planning to use the slit viewing camera for guiding. If we have to do that we don't currently have a plan for how to do so. This may influence the way we do the physical interfacing, since it will increase the kind of data rate we need to achieve.

BK and RS discussed the plans for Deimos. The software design is not far along (since they have been working on MOSS since finishing HIRES), but will probably lean heavily on HIRES, and use a similar Leach CCD controller. Some design decisions will depend on the results of the workshop and standards defined.

John Cromer described plans for the blue side addition to LRIS. Obviously since it is an add-on, it will look and feel the same as the rest of the LRIS system.

There was then some discussion of NIRC2. At the moment they are not contemplating using a Sun workstation, but instead may well buy a DEC Alpha-AXP based system. Pro: More bang for the buck, and for that matter more bang period. Con: Yet another hardware vendor and Unix flavor to support at CARA. The latter wasn't objected to in any noticeable way by the CARA people present. They do already have an Alpha based VMS system which was bought to run the telescope software when it was realized how much of a limitation the existing MicroVAX had become.

JW emphasized that the electrical environment in the dome is terrible, and that therefore we should all seriously consider running over optical fiber to get data to and from the instruments out at the telescope.

Action: Everyone to consider what CPU they might use in the instrument VME crate (we might answer "none"), plus what host CPU they will most likely use.

Session 3: User interfaces (J Kerr)

13:30 WFL gave a lead-in on remote and automated observing. They would like to work toward the ability to script the operation of the whole observatory, rather than just having the ability to do macros on individual instruments but not have the loop closed with the telescope. GD raised the question of how visitor instruments would hook in to such a setup. From the responses it looks like the local group want guest instruments like a hole in the head, although as BK pointed out the simple facility they have provided us at Lick would go a long way to keeping most visitor groups happy.

Open discussion of current user interfaces. Existing instruments reasonably good - yes from Sandy Faber (SF). OA's console bad - as outlined by JA earlier. Feedback to observer (TV display etc. - needs improvement (SF).

There was a lot of discussion of control room layout. Barbara Schafer would prefer the OA to have a bigger role in actually doing the observing, with the observer riding shotgun (the JCMT model). SF would like a hand paddle for pointing and focus control, and a better display of guiding camera output. Big discussion of hand paddle issues, which got out of hand and wasn't wholly relevant to the purpose of the meeting.

Discussion of remote observing. One limitation is personnel. Safety rules require at least two people be at the telescope, which ties up technicians who will then not be available for daytime work. Software wise things aren't yet fully set up for remote observing. HIRES seems to be ahead of any other instrument. Steve Vogt and BK have been running it remotely from California. To go to remote observing full time would cost another $300K for additional manpower (Jerry Smith). BK pointed out the gains to be realized from data compression techniques.

John Kerr showed some viewgraphs on the CFHT system, called Pegasus. It looked pretty slick. Pegasus is a whole system (cf. ADAM), not just a user interface. The look and feel is consistent across all instruments, and SAOimage is used for display. The UI concept is a "form" which is a Motif window with a bunch of fields for setup. Once everything is set you apply the form and the observation starts. The setup form goes away and is replaced by a smaller status window with progress bars, stop and abort buttons. They have a WYSIWYG tool for building new forms.

It was felt that enough time had been spent on the user interface issues (especially the "damned hand paddle") so we skipped the scheduled session 4 and went on to the session scheduled for the next morning.

Session 4: Standardization

AC pushed hard for the concept of code reviews (aka walkthroughs) for quality assurance (supported by me).

Version control - there are a bunch of products and standards for code management and change control. CFHT are using RCS and Gnumake. CARA are using SCCS and a set of directory structure conventions. They use CMS on the VMS system for the telescope, which WFL says is better than SCCS (but CMS isn't available for Unix systems). He proposes a public domain thing called LCCS or else CVS layered on top of RCS. Currently all instruments must use SCCS and provide the necessary make files.

There was a lot of discussion of keeping CARA staff in touch with progress as projects go along. I pointed out we had added a mid-point review and fought to keep the travel budget intact in order to keep lines of face-to-face communication open.

BK described how HIRES was thoroughly tested ("task torture") before being shipped. According to HL, HIRES is the most rugged and bug-free system. LRIS people said they were pushed too hard to deliver on time rather than when the instrument was ready. NR contrasted ROE attitudes (ship when truly ready) and RGO attitudes (ship on schedule, ready or not). I pointed out the NIRSPEC team was being led by two ex-ROE people. JW is of the opinion the 2nd generation instruments are too tight on money and time to get this right.

Action: At the design review each instrument team must define how we will deal with testing, simulation etc. CARA will provide a DCS simulator.

I pointed out that what is necessary is peer reviews not political reviews. It is much more important that we make the CARA staff happy than that we make a convincing "dog and pony show" for (unnamed) committees. Strong support from JW and just about everybody else.

KM made a strong statement about documenting the as-built system and not delivering with out of date support docs. We *must* put a procedure in place to make sure we do this for both hardware and software. We must use formal change requests and approvals, with a tracking system to make sure each change has been fully implemented (i.e. both done and written up). Documentation must be done ahead of or simultaneously with the coding process and not left to the end (or worse still left undone). For mechanical stuff it's simpler since you won't get a change performed by the shop without a modified drawing (must tell Al Casillas to refuse to take any back of an envelope change drawings!). For software we must maintain a module list with a change/debug history as well as dataflow diagrams, flowcharts and a subroutine dependency tree.

Day 3: Keck II standards and management

Session 1: Software coordination/management (H Lewis)

This should have been session 2 as per the original agenda.

Infrastructures in use at other telescopes include ADAM, DRAMA (being developed from ADAM at AAO), Sloan Digital Sky Survey using a combination of TCL, VxWorks.

On the subject of coordination, the SSC came up last year with the idea of hiring a coordinator for software, performing the same function as Ed Romana did for the first generation instruments for mechanical aspects. Ed is perceived as having done a very good job in that respect. This new person would organize reviews and walkthroughs and help facilitate integration of the new instruments by helping get systems coordinated as we build the new systems.

An issue given some discussion time was the benefit of design reviews. Are they of any use if people don't follow the recommendations the reviews generate. Obviously there was some of this going on last time around. Interesting quote from Jerry Smith - "If you're held accountable to a committee you're not being held accountable".

There was a general reluctance to set standards right now in the workshop. HL made a good point (following WFL comments on use of ADAM at WHT) that you have to work with standards even when you don't like them rather than fighting them or going off and coming up with something else.

Two points that became clear to me during this discussion: We should try to either acquire or develop a way to code manage and automatically build the transputer code (something beyond just having make files), and we should get ourselves the new Sun workstation as soon as possible and get the various tools installed and start working with them.

HL recommends we attend the Orlando SPIE conference next April and go to the session on telescope control systems.

Session 2: Keck I lessons (again)

HL - the DCS is now much improved, with the UI getting simpler with time. At the moment there is information overload. As reliability improves a lot of the information now presented can be tucked away in log files and not displayed all the time. A lot of serious reliability problems have now been solved, especially a couple of really big ones in the last six months. However his feeling is that the final product will *never* be quite good enough, and some radical re-engineering will be needed in the long run.

The hand paddle was discussed again and passed around for inspection. The upshot is that it's there and it works even though people have been complaining there isn't one, and some of the OA's don't like it so they never use it.

We then went through a list of suggestions (7 of them) and questions compiled by Djorgovski, Faber and Matthews. Questions and responses below:

Q1. How are you going to find out what to do to improve the situation? From the Keck users? Other observatories? Be there during observations?

Q2. How do you set priorities with software tasks? Combined software/hardware tasks? How do you monitor your own progress and efficiency? How well can you estimate your manpower and other resource needs?

A1./A2. HL - night logs are always read by the software staff, HL sets the priorities. They do spend some time up there when people are observing. AC pushed again for more remote obs to expand this process. KM suggested they actually work as OAs for periods. They tend not to do so. NR, GD and I all pushed this same point.

Q3. How and when will you get the remaining Keck I tasks done, especially the combined software/hardware ones (e.g. automatic focus comp.)

A3. HL - system reliability is priority 1. Management will give him feedback from multiple directions in order to set other priorities. Really big jobs requiring a project setup, and fast fixes to small problems tend to get the majority of attention. Intermediate size jobs are the ones that may stay on the list longer.

Q4. The present code is apparently difficult to modify; even small changes take a long time and many tries to successfully modify. Why is this? Is there something in the present architecture that causes this?

A4. HL - it's hard to make changes because of lack of an infrastructure. There is too tight coupling between systems and applications. Changes have to be made in too many places and it's hard to know whether a change is harmless, e.g. a change in the DCS may require altering 7 modules and a lot of recompilation and relinking. GD made the point that reversion is something the OA's should be able to do. They don't know how at the moment. HL - made the point that new releases are generally put in place on engineering nights. WFL - the process needs to be better documented. HL - they have actually hired someone to produce a troubleshooting guide.

Q5. How much recoding is necessary if EPICS (or another high-level system) is brought in? Can it be done incrementally?

A5. Keck II will be incrementally different from Keck I (e.g. no VAX, KTL, VxWorks 5.1), and EPICS will be slipped in once *checked out thoroughly*. WFL - he believes the changes can all be done incrementally.

Q6. Are there any lessons learned from the major problems solved so far?

A6. The big lesson is to make things more independent, so you don't have to run up the whole system to test e.g. the rotator. They would give more thought to the complexity of the distributed system. Fault tolerance needs to be given more priority, e.g. more retry ability and recovery procedures, and not having to shut down the whole system if one part breaks.

Q7. What degree of involvement of instrument teams is desired after the acceptance? How much instrument software upgrade is CARA willing to take up (e.g. the wish-list of fixes to Figdisp)?

A7. BK - action CARA to set up mailing lists for observatory in general and each instrument individually. Thinks there should be some level of funding for the instrument teams after delivery for tailing off of support. JS - that's a question for the SSC.

Session 3: Definition of instrument infrastructure (H Lewis)

HL suggests a useful standard from the point of view of reliability and quality control would be to run all code through Purify (a code checker program for C). CARA will be purchasing both Purify and Dataviews. Possible standards were discussed with reference to a list provided by WFL.

Item 0: Host computers, operating systems? Sun/SunOS, Sun/Solaris, Alpha/OSFn, possibly HP/HPUX, Silicon Graphics/??. Basically any sensible Unix system. There was some discussion of the XPG4 standard produced by the X/Open consortium. Most Unix versions (and as of this week VMS!) conform or will soon conform to this standard. There was discussion of whether VxWorks is necessary. LRIS - they wouldn't have used it although he still likes it. BK - it cost a lot of time on HIRES due to bug-ridden early versions. Now better and has better tools (Stethoscope and Windview). KM - not sure whether or not it was necessary since Berkeley did all the programming on the IRE. Richard ?? thinks VxWorks was a big liability on the ACS. HL - DCS still requires it. Overall it looks like it will stay. We don't have to use it for NIRSPEC (and in any case the VME crate and software were taken out as the budget was revised).

Item 1: Directory structure and management conventions. We need a common way to manage versions during development, which will be the same as that into which we deliver. **TBD**.

Item 2: EPICS: for the DCS; for AO; for ACS; for anything else? This needs further clarification. CARA will fund another training course so that we all know more about it.

Item 3: KTL: keyword libraries for all instruments and telescope subsystems. The NIRC2 team were dubious. They need more information to make up their minds. Everyone else (including me) committed to using KTL.

Item 4: KTL kernel: for all logging and error reporting. Still needs some work on current implementation. Further development at CARA, then tech group will define.

Item 5: Motif and X11: for all user interface development. A done deal. But see items 6 and 7.

Item 6: DataViews: as a GUI builder. Yes. We buy (at $1500/machine via CARA negotiated agreement), CARA pays for maintenance.

Item 7: TCL/Tk: for scripting and simple user interface development. TCL is the default command language for control and for scripting control sequences. NIRC uses "crisco" which has been a success but is completely unique. TCL not a perfect choice but *very* widely used (Lick, Starlink, Sloan DSS). There's a need to better understand the effect of using TCL on quick-look facilities, since it's not good at math.

Item 8: Data analysis tools: IRAF, IDL, SAOimage, ximtool, pgplot. Discussion was deferred since this is a whole workshop in itself.

Item 9: VxWorks: for DCS; for AO; for ACS; optional for instruments? Discussed earlier, under Item 0.

Item 10: mosaic tools and viewers as basis for all on-line help and documentation. Used a lot already at CARA for all kinds of guides. Yes.

Item 11: Higher-level model: overall system control, programmed observing, session managers etc. Taken over by KTL and TCL/Tk to a first order but something else may come along later (e.g. DRAMA).

Item 12: Archiving: at least define a model that people can work to. Security copies are made of data, although not uniformly. Nothing beyond that was mandated. Currently it's not fast or convenient.

Item 13: Databases: identify standard database. CARA and Lick use Sybase. Non-issue.

Item 14: Compilers: ANSI C, Gnu C, C++? Much talk of Gnu C, Gnumake, Gnu everything. Generally recommended. ANSI C definitely. Gnu decision deferred. Gnu products, for the uninformed, are a set of public domain tools produced by a set of talented wackos who don't believe in making money for their intellectual outflow. I believe you get what you pay for. They're free.

Item 15: Coding standards: SDSS standards, Spencer C standards. CARA has a set of coding standards (a 2 year old document) Also use WFLman which automatically extracts the documentation blocks from the header of code modules. WFLman produces LaTEX, which can then be used to produce man pages for Unix or HTML files which can be put on a mosaic server and accessed from anywhere over the net.

CARA will produce a Keck II programming standards document and we will all stick to it.

Item 16. Productivity tools: All code will go through Purify if HL has his way. Stethoscope and WindView will both have to be considered too.

Item 17: Word processing tools. No standard software defined, but at the very least PostScript output is necessary, and the ability to make HTML files would be desirable.

Item 18: DSP development tools. There are no standard tools because there is no standard chip. We will have to use the Inmos tools for the transputers.

There was some discussion also of the messaging system. CARA line is that they should have used MUSIC in the first place (and thereby saved a lot of grief). Will probably continue with MUSIC but also looking at channel access mechanism of EPICS. DRAMA is another possibility.

Following this discussion there was a summary of the meeting by GD. Most people seemed to be taken aback a little by him declaring the meeting a failure. What he was driving at is that there still remain a lot of decisions on standards. According to him, the SSC wants a software technology monitoring committee to be set up and act as a successor to the old DAWG. This time however the committee will have teeth (many bad jokes about toothless DAWGs ensued). According to GD the SSC is very keen on this idea, and will go as far as *cutting funding to groups not complying with standards* once they are defined. His suggestion is that we form this committee from the people attending the workshop, and have the first meeting in a couple of months once the Keck II DCS software PDR is over. NR suggested we move faster and try to define as many standards as possible *before* that PDR.

Conclusions

My overall impression of the meeting is that it was an enormous success, despite the damper put on at the end by Djorgovski. However I agree with him that the job is not done. There remain a number of standards issues to be clarified and decision to be made. I learned a tremendous amount, not just about the current software and where it is likely to take us in the future, but also about the people involved. There was a refreshing lack of name-calling or finger-pointing in the discussions of Keck I shortcomings. The CARA group had no hesitation in taking responsibility for things they got wrong. This gives me a lot of confidence that they are equipped to get the job done in the end. Projects succeed or fail based on the attitudes, not just the abilities, of the people involved.

As far as our own project is concerned, I am a bit less worried now about the challenges facing us in designing a working system, since I believe there is now much less chance we will be aiming at a "moving target" in trying to integrate what we build with the observatory system. In addition, I will be on this new committee and therefore able to exert influence on the standards defined.

I have been strengthened in my view that we ought to start work as soon as possible to try to transfer Gemini over to a Unix machine, and thereby do some advanced prototyping for the NIRSPEC system. There are options we need to explore, and either adopt or eliminate, before we can proceed to completion of the design.

Also it has become clear to me that since this is to be a delivered system we need to move onto a new level of professionalism as far as design documentation and documentation of the final system is concerned. We will be helped in this regard by the CARA group and the new oversight committee giving us a clear indication of the standards they would like us to achieve.