This final report of the SWAG, submitted in September 1990, partially addressed the question of user specifications for any new software system. In that sense it was the first written report dealing with specifications for AIPS++ and for this reason is included in this memo series.

AIPS ++ USER SPEC. MEMO 102

To: Paul Vanden Bout

From: Tim Cornwell for the SWAG

Subject: Final report from the Software Advisory Group (SWAG)

Summary

The SWAG recently met at Chicago (August 1-3, 1990) and decided to issue this final report outlining a recommended strategy for the next 5 years of NRAO data analysis software. The principal recommendation is that NRAO institute a cross-observatory initiative in data analysis software. There are two goals: to produce a major augmentation in the capabilities of the AIPS, and to improve efficiency of use of observatory manpower and hardware. Parts of the observatory staff not currently involved in the AIPS project would become involved in the new initiative. We recommend that this new effort be under the control of a manager at the Assistant Director level, who would have ultimate responsibility for data analysis software and hardware, and who would coordinate with both operations and on-line computing at the various NRAO telescopes and with project managers for new telescopes.

As well as these long-term goals, we emphasize the importance of a number of on-going projects: for example, the development of VLBA software inside the current AIPS. It is an essential part of our long-term strategy that these be completed as originally planned. In this same vein, we make three major additional comments in the area of single dish software: first, the UniPops project must be allocated sufficient resources to allow completion, second, the observatory should support use of external packages, both public domain and commercial, and third, we encourage the GBT project to make appropriate plans for its computing needs, in both hardware and software, as soon as possible.

Finally, we believe that with the submission of this report the goals of the SWAG have been met and that it should be disbanded.

1 Assessment of current status of NRAO data analysis systems

NRAO's data analysis software systems have had considerable successes, most notably the AIPS project. Indeed the AIPS project is highly regarded as an example of a large, multipurpose transportable software system. After a long history of difficulties in single dish software, a similar but much smaller scale success has recently occurred with the UniPops project. Despite these successes, we believe that overall there has been too little cohesion in the various efforts spread around the observatory. A prime example of this splintering of effort has been the duplication of VLA calibration software in the AIPS and ISIS packages. Another example has been in the abandonment of AIPS for development of sophisticated algorithms by several NRAO scientists. As a consequence, a number of state-of-the-art wide field imaging and mosaicing algorithms now exist solely in the SDE package.

In addition to these global problems, it is also clear that the AIPS project in particular is suffering from a number of difficulties, the most important being a steadily increasing work-load for AIPS programmers. The combination of increasing demands on the software together with the increasing complexity of AIPS software, particularly in the calibration/mapping package, is leading to problems in maintaining and augmenting existing software. This is a burden for both AIPS project programmers and external users wishing to add code. In addition, various parts of the AIPS system such as the user interface and graphics are becoming somewhat antiquated. Although the AIPS group recognises these deficiencies, there is currently insufficient manpower to address them on a short time scale.

Furthermore, although the UniPops project has provided badly needed functionality in single dish analysis, it is regarded by all involved as only a short term solution.

In order to characterize the opinions of the NRAO users on the long-term strategies of analysis software, we surveyed the community. This involved mailing 600 questionaires to people on various NRAO mailing lists. 106 North American and 41 foreign replies were received. The clearest deficiency of the survey is that single dish users are not adequately represented among those responding. Otherwise, we believe the survey to be reasonably representative of the opinions of NRAO users (see appendix C for a copy of the form together with a summary of results). The strongest signal from this survey is that the users expect to do most of their computing at their home institutions and put a high priority on improving NRAO support for that remote use of analysis packages. High priority was also given to:

1. improvements in the user interface,

2. improvements in the ability of users to add custom programs,

3. further NRAO research into and production of basic analysis algorithms,

4. improvements in data exchange between packages.

Users also cite the general area of workstation support as needing improved continued support. Other concerns were better testing of software (for example, CALIB), and continued support of small users with non-standard hardware. Some areas which were, suprisingly, *not* thought to be important were:

1. support for a broad range of equipment which is slowly becoming obsolete,

2. computational efficiency,

3. interfaces to commercial software such as spreadsheets,

4. supercomputer analysis.

However, we recognise that users' unfamiliarity with computing technology may require some caution in the interpretation of these views.

Overall, we see a steadily increasing burden on the AIPS project members and a pattern of inefficient use of other NRAO software manpower, together with some continuing dissatisfaction in the user community over some aspects of the AIPS package. Note however although this was not an explicit question, we conclude from the pattern of answers and the additional comments given on the returned survey forms that the users show considerable support for the continuance of the AIPS project, which is nearly universally viewed as a great success. Hence any change should be regarded as a course correction rather than a major re-think of the concept of a large, multi-purpose package such as AIPS. In this sense, we see no need for a change in NRAO's overall strategy of providing a system such as AIPS for home institution use. We do however see need for considerable changes in the tactics.

2 A plan

We conclude from the above evidence that there is a need for a fresh initiative in NRAO data analysis software. This initiative should cross Observatory divisions and also be larger than the current AIPS project. Its purpose would be to produce an augmented AIPS using manpower from both the current AIPS project and other computer divisions inside NRAO. The augmented AIPS must have significant improvements over the current AIPS in the following areas:

1. user interface,
2. programmability for both programmers and scientists,
3. exploitation of external software packages e.g. X-windows, PGPLOT,
4. integrated support for networking,
5. data exchange with other packages, and
6. algorithm research and production

In addition, we see need for improvements in support for users at home institutions, and for response to general user concerns, perhaps mediated through a users group. We expect two principal forms of payback: first, all of these improvements are demanded by users (including NRAO scientists) and will presumably considerably enhance user productivity. Second, these improvements will increase the productivity of NRAO programmers by making coding substantially easier and by relying upon externally-supported software wherever possible (in particular, NRAO programmers should be able to concentrate upon producing application software rather than in providing system type functionality). It is hard to estimate a payback time for an initiative such as this. We can say however that we expect that applications programmers should be able to start writing in a substantially improved environment within 2-3 calendar years, after about 5-6 man-years invested in system specification, prototyping and implementation. We would expect significantly improved applications tasks within 4-5 calendar years of starting the project. This is a conservative estimate, applicable to the whole initiative. Some improvements, such as the user interface and the DynaFITS disk format (see below), will probably be incorporated into the current AIPS project on a shorter time scale.

Our recommended plan is designed to correct or alleviate the problems described in the previous section. The specific objectives of the plan are:

- to maintain current capabilities in the AIPS and Unipops projects (e.g. existing AIPS tasks must continue to run),

- to provide an augmented AIPS package, addressing both single dish and interferometer analysis, by 1995,

The major steps in the plan are:

1. Now: start search for NRAO Off-line computing manager (see next section),
2. Now: start the data exchange initiative ("DynaFITS", see below in appendix A on technical details) immediately. This involves negotiations with other observatories e.g. STScI and NOAO. We recommend that this be handled by the Observatory's liasons to the FITS community.
3. As soon as possible: set up a group to work on specification and prototyping of the augmented system. This group should start study of other systems immediately.
4. End of 1991: prototypes of a number of major components are available for testing and comments by both programmers and users.
5. 1992 - 1993: applications programmers will see new capability and start to program new tasks,
6. 1995: user sees major enhancements in the capabilities of AIPS.

In the following sections, we flesh out our plan by providing discussion of, first, the management structure required to implement this project, second, the resources required initially, and, third, in section A, a large number of detailed suggestions for the augmented AIPS.

3 Required Management Structure

We consider the appointment of a NRAO off-line computing manager as essential both to curing some of the general splintering of efforts described above and to ensuring success of the new initiative. The manager would have ultimate responsibility for software and hardware used for analysis of data from all NRAO telescopes. In addition, he/she would coordinate with operations and on-line computing at the various telescopes. We believe that this position is only tenable if the manager is given control of all relevant resources including both staff and resources. Specifically, the manager would have control over all NRAO data analysis systems, including the AIPS, ISIS and Unipops projects. We note that this is a considerable change from the current organization where, by and large, the site directors control locally-based staff.

Our only other substantial comments concern input from users of NRAO analysis software. First, we strongly urge the manager to set up an AIPS user group to ensure that input from the end users of NRAO data analysis software plays a greater role in guiding the project than is currently the case. One possible format for such a group would be be a large group which meets once a year. Second, efforts should be made to improve the involvement of NRAO scientific staff in specifying and testing NRAO data analysis software.

4 Resource estimates

It is clear that this augmentation of AIPS cannot be acheived using current AIPS manpower alone and that a pooling of NRAO resources is necessary and desirable. Since the meagre amount of manpower available for single dish programming is fully occupied in the Unipops project, we must consider involving other NRAO staff such as the VLA computing division. We believe that an appropriate level of activity would be to involve one member of the current AIPS group and one member of the VLA computing division in phases 3 and 4 described above. During phase 5, we expect that a large fraction of NRAO's application programmers and a number of the scientific staff would become involved. Beyond this initial suggestion, we would prefer to leave any allocation of resources to the manager.

5 Appendix A: Detailed reports

At the Chicago meeting, the SWAG split into sub-groups charged with discussing specific topics. The reports from these working groups are presented here as an aid to the software designers and are not intended to constitute a plane for implementation.

5.1 User Interface

The User Survey coordinated by members of the SWAG showed that, after user support, the desire for a powerful, easy to use, flexible user interface is the primary concern. This is, indeed, the principal technical issue raised by respondents to the Survey. While NRAO's strategy for the development of second generation data analysis system must be a comprehensive one, incorporating solutions to problems such as the programming environment, data exchange, and algorithmic research, the importance of the user interface for the ultimate success of the system cannot be underestimated.

Many of the specific comments about the user interface are based on the experiences of the subcommittee members, i.e., we don't like this about AIPS, we do like this about IRAF or Unix. The decisions about how to implement a new user interface should be made within NRAO, but one item of advice from this subcommittee which would be folly to ignore is the following: NRAO should closely examine the features and implementations of other software systems, particularly of the user interface, before embarking on any work of their own. Given that many of the recommendations below are already satisfied by IRAF, this system in particular should be studied carefully. Such studies will help NRAO to minimize the costs of implementation and, at the same time, maximize user satisfaction.

Recommendations are given in the following areas:

- command line interface,
- command language interpreter,
- parameter handling,
- data catalog,
- process control,
- help and documentation,

- networking,
- other interface issues,
- external interfaces,
- a graphical user interface.

The recommendations are not in priority order. Indeed, since these objectives are interrelated, it may not be possible to implement one feature without also dealing with several others simultaneously.

5.1.1 Command Line Interface

The software system must support a first-rate command line interface. Although one can argue that future software systems will utilize graphical user interfaces (GUIs) extensively, if not exclusively, there are several reasons to continue support for a good command line interface:

- Astronomical institutions will still be utilizing 'low-end' hardware such as ASCII terminals, and users will want to run the software remotely (i.e., over a modem from home) where support for bit-mapped graphics is not yet practical.
- Batch processing and procedure development (in a high-level command language) must be supported. Both batch processing and the use of procedures frees users from the time-consuming and/or repetitive steps in data reduction.
- Expert users will want to have the most efficient mechanisms available for carrying out their data analysis. Despite the user-friendliness of GUIs, a command line interface is still likely to be the most direct way to get the job done.

5.1.2 Command Language Interpreter

The software system should have a high-level command language (CL) interpreter, and this interpreter should not be the host operating system command language directly. Even though there is an industry trend toward a common OS – Posix – there are likely to be significant differences (i.e., 'extensions') with different vendors' hardware. The specific recommendations include:

- A 'Unix-like' command interpreter. The CL should support Unix features such as I/O redirection and pipes. The input and output parameters of tasks should be specifiable on the command line. This interpreter could be layered on a standard Unix shell (C-shell, Korn-shell), but should not depend on a particular vendor's operating system.

- The CL must be programmable. Users must be able to tie together various tasks in the software system using procedures written in the grammar of the command language itself. In order to support the use of the CL as a procedural language, the language must provide functions such as case statements, variable declarations and assignments, and string and filename manipulation. Ideally it would be possible to compile CL procedures so that they could be executed with maximum efficiency.

- The CL should have a flexible command line recall and editing facility. Users should be able to easily review their command history, retrieve specific commands, edit previous commands, substitute different command arguments, etc.

5.1.3 Parameter Handling

The manner in which users set parameters for applications programs is a major part of the user interface. The important point here is that the software system must be flexible in this area, providing various methods of parameter setting. Users would use various methods depending on their experience, the complexity of the task, and the type of terminal or workstation they were using.

In particular, the following recommendations are offered:

- There should be no global parameters; the parameters for one task should be independent from those for all other tasks. Recognizing, however, that it can be convenient to transfer input parameters from one task to another, a simple parameter passing mechanism must be provided. One might, for example, provide a symbol substitution mechanism as part of the parameter parsing code.

- There should be various methods of reviewing and setting task parameters. This includes

  - parameter listings

- parameter editing (full-screen)
- prompting
- explicit parameter assignments (param_name = value)
- both positional and named arguments on the command line
- shared parameter sets

- The parameters used most recently for any task should automatically be saved. Users should have the ability to selective save and restore parameter sets. Task parameters should be private by user and by directory, allowing one user to keep different parameter sets for different projects, and avoiding any confusion between users sharing the same account.

5.1.4 Data Catalog

A data catalog should be viewed as an aid to the user, not as a requirement of the data structures. The data files should be independent entities, allowing a decoupling of the catalog structure from the data. The catalog can be used, or not used, at the option of the user. It should be easy to construct, modify, or delete, without any impact on the data themselves.

Specifically, the catalog should support the following features:

- There should be no restrictions on the names given to data files. The names should be up to the user, and the association of a name with a catalog slot number is simply a convenience.

- The AIPS concept of user numbers should be dropped. All modern computer operating systems provide a hierarchical directory system for managing and organizing data. The same system should be used within NRAO's software system. User numbers arose from the use of shared accounts. In this era of networked computer systems, shared accounts are major security risks and should be avoided.

- To the greatest extent possible, the file names users assign should be the same as the file names on the host operating system. Like or it not, users will manipulate their data outside the software environment provided by NRAO. The use of obscure file names makes this very awkward.

- The catalog should be easily updated. Functions that add and delete catalog entries should be available, as well as functions that will automatically generate a catalog for all files in a directory.
- Facilities should be provided to search and select files from a catalog based on certain criteria, i.e., date of observation, object name, observing frequency.

5.1.5 Process Control

In a large and complex data analysis system users must be able to control the time and manner in which their processes are executed. It should be possible to direct computationally intensive tasks to run as background or batch jobs, and in the era of networking direct such jobs to be run on specific CPUs. In addition, one should be able to

- Move foreground tasks to the background
- Move background tasks to the foreground
- Service I/O requests for background tasks
- Interrupt processes and redirect their output
- Monitor the status of all processes

5.1.6 Help and Documentation

One simple measure of the user-friendliness of a system is the ease by which users can get help information on the relevant topic. For simple tasks the ideal system does not even require help – task names and I/O parameters are self-evident, and functions and nomenclature adhere to conventions so that a user's intuition about the system grows rapidly. Complex applications, however, such as are prevalent in radio astronomy, demand good help documentation with simple and rapid access.

One major criticism of AIPS which also pertains, at least in part, to IRAF, is that it is hard to get help on a task whose name you do not already know. This is because the help facilities for these systems are all organized by task name. Recent versions of IRAF have included an 'apropos' facility, allowing access to help information by keyword rather than by task name, and this is a major improvement. In addition, once a user has located the relevant task and its help information, facilities must exist that allow rapid perusal of what may be a large text file. One should be able to search for specific keywords (such as input parameter names), and go forward and back in the file.

There is a good deal of work in the computer industry on hypertext, and this is a promising area for exploration for an astronomical analysis system. Also, using bit-mapped screens in order to provide cleanly formatted text displays (a la Xman) is also useful.

Specific recommendations for NRAO are as follows:

- Implement an 'apropos' facility as an adjunct to the standard help by task name. This is conceptually similar to 'man -k' in Unix.
- Set up help documentation in a standard hierarchical framework, with headings such as Task Description, Parameters, Examples, Usage Notes, etc. All sections of such a hierarchy should be accessible independently.
- Help documentation must be up-to-date to be useful. NRAO must develop mechanisms by which current bug reports, etc., are distributed to user sites. A standard section for each task help file should be called Bugs, and this should be used and kept current.
- Off-line documentation is equally important. NRAO has an outstanding overview of AIPS in the Cookbook, and this and other similar efforts must be continued for any new system. The most effective documentation of this sort is often written by staff scientists rather than by the programming group.
- A command language/user interface such as is being described here will have many features, and it may be difficult for the inexperienced user to get started. A simple introduction to the CL which provides a simple way of doing things, even though it may not be the most efficient way, should be provided.

5.1.7 Networking

It is now a rare situation for a user to have access to a single, isolated CPU. More common is a network of machines (workstations and fileservers, VAX cluster, and even wide-area networks) that share both software and data. NRAO's software development plans must take the networked computing environment into account, with networked access to data and devices being an inherent part of the system design. Access to remote data and devices should be largely transparent to the user, i.e., they should not need to know about specific network protocols or the idiosyncracies of the byte order on different CPUs.

Support for a networked computer environment will include the following capabilities:

- Access to both local and remote data, with data formatting translations performed as needed, transparent to the user. In practical terms this probably requires the development of a machine independent data format.
- Access to both local and remote graphics and display devices.
- Support for data compression schemes when using networked devices.

5.1.8 Other User Interface Issues

In addition to the major interface features discussed above, there are several other topics that need to be addressed:

- Image processing histories need to be maintained. The AIPS implementation is highly regarded, and similar capabilities need to be incorporated into the next generation system.
- An area where AIPS has NOT done well is in overloading single tasks. For example, the task COMB does about 64 different things, all depending on the settings of the (heavily overused) parameters APARM and BPARM. It is much simpler and clearer to the user for tasks as diverse as image arithmetic and computing a spectral index map to be separate tasks, even if the same underlying code is used to do the computations.
- There needs to be simple access to host operating system commands, i.e., through the use of an OS-escape character. It is unlikely that any software system written at NRAO will provide ALL capabilities of the host OS, and users should be provided an easy way to access the OS. Indeed, it should be possible to define commonly used host OS commands as part of the user's environment.

- The user's environment within the software system should be highly customizable. The kinds of things that users should be able to tailor to their specific needs include
  - predefined commands and procedures
  - directory paths for data areas
  - standard text, graphics, and image output devices
  - known remote hosts
  - custom tasks

5.1.9 External Interfaces

It is not possible for NRAO to implement all of the software that its users might wish to use for the analysis of radio astronomy data. However, it is possible to provide simpler means for the exchange of data with other analysis software. While this is partly a technical issue at the level of data formats, it is also a user interface issue in that with simple data exchange mechanisms, the user can choose amongst a wide suite of software to find the analysis tools that suit the situation.

Specifically, the areas in which data exchange is most important are:

- Image data. Continue support for FITS, and drop the constraint that FITS images on disk on VMS systems must have 2880-byte records. Once the data goes through a network interchange this record length is lost.
- Work with other astronomical centers to generate a system-independent format, i.e., generalized disk FITS, that all systems can read and write directly.
- Provide better internal support for tabular data structures (table editing, for example) and I/O filters for the generalized disk FITS and for common spreadsheet programs (Lotus, 20-20, etc.). An output filter to TeX tables would also be useful.
- Provide graphics output in Postscript, allowing for its encapsulation in Postscript text files.
- In the simplest cases, allow text output from programs to be easily redirected to files named by and accessible to the user. When all else

fails, having text output can make it possible to move data from one system to another. For image/UV data this is rarely practical, but it can be useful for small data sets.

5.1.10 Graphical User Interface

All of the issues mentioned already pertain to the command language and user interface independent of whether or not there is also a GUI. Indeed, the Subcommittee strongly recommends that any work on a GUI be based on a layered approach. The GUI should not be the only mechanism available for interacting with tasks, but rather a complementary mechanism available to those with the appropriate hardware. An exception to this approach would pertain to applications where a GUI is really essential to even utilize the task, i.e., programs like TVFLG.

Because the development of a GUI is a wide-open area, the Subcommittee was reluctant to make specific recommendations in this area. We feel it is important to begin work in this area, and that users should be involved in provided quick feedback concerning the strengths and weaknesses of various approaches. Also, NRAO should interact with other groups, both within and outside of astronomy, and compare progress with GUIs. The entire computer industry is very much on the learning curve with GUIs, and it is not reasonable to expect NRAO or any other single astronomical institution to development the 'best' GUI in a short time.

The GUI will undoubtedly be based on X windows, although the choice of a particular implementation or desktop (Motif, OpenLook, etc) is best deferred until experience is gained in this area. This is discussed more fully in Section 5.5 below.

The kinds of features that should be explored in a GUI include, but are certainly not limited to, the following:

- Point-and-click interface to commonly used commands
- Point-and-click interface to help, with help available from within task menus (i.e., by parameter name)
- Parameter setting via enumerated lists (options pull down), dials and sliders (e.g. clean loop gain from 0 to 1), drag-and-drop (drag icon for a file onto the icon for a task; this sets the input for the task)
- Graphical file manager/catalog

```
    Calib ⇒
            vla     ⇒
            vlba    ⇒
            at      ⇒
            gbt     ⇒
            hst     ⇒
    Analysis⇒
            images⇒
            spectra⇒
etc.
```

Figure 1: Task menu bars

- Task menu bars, with hierarchical organization: see figure 1.
- Procedure construction via icons (i.e., draw lines between processes indicating data flows); process status could be shown by blinking icons, problems by turning icon to red, etc.).

These suggestions merely scratch the surface, but are meant to be indicative of the types of things that could be done with a GUI.

5.2 Interferometer Applications

5.2.1 Manpower/Management

We see a problem at present in that users have a perception that there is little point in asking for new applications because such requests are generally not satisfied for some years, if at all, presumably because of insufficient manpower. We suggest the following measures to rectify this.

- An AIPS usergroup should be formed as a means of giving NRAO feedback from users on what new applications are needed.
- Easier programming interfaces within AIPS will enlarge the potential manpower base available for writing new procedures.
- NRAO scientists should be integrated into the AIPS testing and development cycle.

- Code checking and distribution facilities should be upgraded to remove the anomalously large numbers of "non- standard" tasks.

- A facility should be provided (bulletin board? usergroup news?) for distribution of "unsupported" user- generated code. This would be distributed unchecked by NRAO on a caveat emptor basis, and no guarantees or support would be available for such code.

- Application development should take account of engineering (e.g. for tracing on-line VLA problems) as well as astronomical needs.

5.2.2 Spectral line and multifrequency observations

- At present, not all AIPS tasks will cope with spectral line data (data cubes). In future, all tasks must cope with this generalised format in a natural way, since this will become a more widely used format because of

  1. Increasing spectral line use
  2. Multi-frequency synthesis of continuum sources
  3. Wide field mapping (to avoid bandwidth smearing)
  4. Very high dynamic range images (to calibrate bandpass phases)

- Better support is needed for tasks that are specific to spectral line work. In general, everything that is needed for single dish spectral line work (interactive baseline subtraction, line profile fitting, etc.) will also be needed for interferometry.

- Better support is needed for slices and generalised one-dimensional problems (e.g. display the intensity of a slice along a curved radio jet).

- Support for interfaces to other packages

5.2.3 Other requirements

- Arithmetic manipulation of images should be provided. What is needed is a parser to interpret and execute any valid Fortran line such as:

  A = B**2 + C*SQRT(COS(D)) + F(X,Y,OBSFREQ)

  where A, B, C, D are images or data cubes, X, Y, are coordinates in the image, and OBSFREQ is the observing frequency.

- Desirable but clearly more difficult (and thus lower priority) would be a corresponding function for u-v data.
- More sophisticated applications would increase the power and ease of use of complicated functions. This might be achieved by packaging procedures to run in the background without global adverbs.
- A better (more sophisticated and less error-prone) batch facility is needed.
- The slow start-up time of tasks causes considerable frustration, particularly for users wanting to run a sequence of short tasks. An example is the user who wants to have a quick look at his/her data in a variety of ways (UVPLOT, TVPL etc.).

5.2.4 More difficult changes

A more difficult long term goal, but desirable if possible, is the provision of an event-driven run file linking two (or more) windows.

Example 1: Display u-v data (channel 0) in window 1 (a la TVFLG), and display the spectrum of a selected segment of data in window 2. Moving the cursor around in window 1 interactively shows the spectrum of each point in window 2.

Example 2: Run TVFLG in window 1, display resulting map in window 2, and see the effect of flagging data interactively, with something like MX running continuously in the background linking the two windows.

5.3 Single Dish Applications

5.3.1 Support/Enhancement of UniPops Effort

The Observatory must provide the hardware and manpower resources necessary to implement UniPops, the observatory-wide version of the POPS single-dish analysis program. This effort will require procurement of several (perhaps as many as 4) more work stations before it is freely available to observers, visitors, staff, and even its programmers.

Our analysis capabilities are now artificially split into separate toolkits for single-dish, array, spectral line, and continuum observations even though the manipulative abilities needed are often the same in each case. Future software development should entail a closer coupling of single-dish line and continuum data-handling software.

Our present array-based software is widely recognized to be deficient in its handling of individual data vectors (slices). The capabilities expected of single-dish packages, if extended to AIPS, would provide a solution for this problem.

5.3.2 Software Tools/Algorithms Available For Single-Dish Analysis

The Observatory should widen the range of analysis tools available for data processing, supporting a broad range of software created within the scientific community (IRAF, SAOImage, NCSA Visualization, etc.) and by commercial vendors (IDL, for example). Use of such externally-developed software is important on three counts; to provide added capabilities, to provide programmers the ability to keep abreast of the general state of developments, and to refine estimates of which needs must be met internally and which can be satisfied "off-the-shelf" without extensive programming efforts.

Array receivers now used for continuum observations in Green Bank and soon to be in widespread use at the 12-meter require a multitude of new data-handling abilities. At the present time, there is no manpower devoted to development of sorely-needed capabilities. Unless more effort is focussed in this area, the usefulness of our telescopes will be severely compromised.

5.3.3 Long-Term Developments

It has always been the opinion of this group that the UniPops package is not the correct basis for a long-term data-processing strategy and it would be severely disappointing were the GBT to open with only the UniPops package for support.

Development of a long-term strategy must consider two related questions:

- Can we provide a comprehensive analysis package which is sufficiently nimble to be used on-line by both observers and support staff or must the on-line and off-line efforts be divided? During observing or hardware development the greatest need is to accomodate new circumstances and the full range of analysis is seldom employed. Offline, in a more remote and relaxed environment, the emphasis is quite different.
- If a new system is developed for VLA/VLBA off-line data-handling, should new single-dish software be incorporated into or merely inter-

faced to it? As noted above, similar data-handling abilities will need separate implem- tations if the two systems are too loosely coupled.

In the long term we support integration of the single dish analysis system into an exportable, observatory-wide software system following the general guidelines of this memo, with the proviso that the need for a competent online system must be fully addressed in any future developments.

5.3.4 Green Bank Telescope

We encourage the Green Bank Telescope project to consider its computing needs, both software and hardware, at the earliest possible moment. In light of the comments made above it is obvious that the Green Bank Telescope can not expect to draw on the existing software/software development effort for more than very partial support.

5.4 Programmer Interfaces

The following describes the suggested applications programmer interface to disk data structures. The implemention of this programmer interface is to provide compatibility with existing software and allow continued development of applications software until the new interface is ready for general use.

Disk resident data will appear to the application programmer as an Abstract Data Type (ADT) with an interface which hides all of thede details of the actual storage. The ADT should be a level on top of the AIPS data structures to allow backward compatibility with existing software. If an agreement on the shared data disk FITS format is reached with other image processing groups then the ADT interface will also support this format as an alternate internal format.

Processing control information (e.g. windows in images and weighting functions for gridding uv data) will be attached directly to ADTs rather than passed thru commons or call sequences. Data in an ADT will be referenced by label (i.e. a description of the data) rather than by position in the file. There will be c and Fortran callable versions of the ADT interface routines. The interface to tabular data will allow "atomic" access as well as larger structures.

Basic operations on ADTs will be performed by functional modules. These operations are those such as image arithmetic and operations on uv data sets and will be the basis for implementing applications algorithms and will be made available directly to the user as command language functions.

A set of standards needs to be developed for the use of c in applications software and the standards for Fortran need to be better defined. The use of NSE and other packages for software managment needs to be examined.

5.5 Graphics and Image Display

The subcommittee recommends a number of changes in the Graphics/TV interfaces used in AIPS with the goals of improving maintainability providing a more friendly programmer interfaces, and providing new graphics functionality for the purpose of rapid communcations between user and program.

The recommendations are mostly of a general nature; specific requirements or design issues are left to the NRAO teams.

5.5.1 TV communications

All low-level TV communications should occur via X-Window interfaces. Devices for which no X-drivers exist (I2S, DeAnza etc.) should be abandoned, or X-drivers (possibly limited in functionality) should be written or purchased.

5.5.2 Interfaces

The programmer/system programmer interactions with workstations and with plotting devices (graphics terminals, laser printers, film writers) should proceed through the following, or a similar hierarchy:

A. High Level functional modules, for example:

1. Draw a contour map of a given sub-image with given contours
2. Load a given sub-image with given lookup table into a workstation, and label it, including coordinate grid
3. Create a standard pop-up menu allowing user to enter information for a task, or for AIPS as a whole

B. High Level Programmer Tools

1. Line Graphics (Draw Axes, Draw Contours given a 2-D array of numbers)

2. Image Graphics (Load a rectangular array to TV, Request Definition of a Polygon Area on TV, Read TV cursor in specified coordinated system)
3. Graphics User Interface (create push buttons, text windows etc.) These interfaces should be created in the context of one of the more or less public Toolkits for workstations like Open View, or MOTIF, or the Athena Widget set.

C. Low Level Programmer Tools

1. Line Graphics would consist of calls to a standard, public domain graphics package like PGPLOT.
2. Image graphics should occur through a modified set of Y-like routines. As described below (section III) the functions of these routines should be somewhat more limited than the current set.
3. GUI elemental calls, taken from the above Toolkits

D. System calls

1. Line Graphics via PGPLOT or similar drivers
2. Workstation/Image via X-drivers

5.5.3 Low and Intermediate Level Image Functions

A. Low Level TV functions (e.g. turning a window into an ICON, positioning of an image on the screen) are the responsibility of a Window Manager (WM) on the display device, and should not be supported by any applications code.

B. Intermediate functions (zoom/pan, standard interactive modification of BW/Pseudo-Color transfer functions, reacting to "exposure events" on a workstation) are not usually handled by Window Managers, but should not be the responsibility of the applications programmers. For these functions a seperate process, an "Image Manager" (IM) should be created. Possibily one IM should exist for each image, or alternatively, one for each host/workstation pair.

The IM will provide the above functions in response to standard requests by the user on his/her workstation. In addition the IM can handle the following functions:

1. Be the home for the interface between applications calls/ application data (Y-routines, image formatted data) and device calls/data (X-routines)
2. Be the reservoir of Astronomical information regarding images, such as coordinate transformations, image catalog entries, physical units, and pointers to the original disk files.
3. Be a coordinator of TV to and from line graphics interactions so that line overlays can be correctly placed and scaled on images. For example when an image is zoomed, an overlay should zoom correspondingly in scale, but the line segment widths and character size should not increase in size.

Thus the Y-routines, or at least the subset available to the applications programmer can be relieved of the functions supported by the Image Manager.

Probably pure line graphics to graphics screens or laser printers should also be handled by similar managers, so that coordinate or intensity information can be tracked, an so that the programmer interface for such graphics and for image overlays are consistent.

5.6 Data Exchange

A model was proposed that defines the image analysis part of AIPS to work in close cooperation with other packages. This requires that all packages can read one another's data in binary form and file headers without copying (as in FITS via AIPS tasks IMLOD/FITTP).

A header identical with FITS in logical contents but spread over a number of diskfiles might be the best solution. As the FITS format is already accepted throughtout the astronomical community, we feel that agreement of "dynamic FITS" or "DynaFITS" could be reached quickly between NOAO and NRAO and that suitable software to the reading and writing of header and data could be written with only a few man-months. We feel that once these two sister organizations have taken the lead, the authors of other packages would follow shortly.

The net result would be a virtual explosion of easily accessible functionaility for the astronomical community: functionality which comes without too much burden on each package's author.

6 Appendix B: SWAG membership

The SWAG is:

Bob Burns	NRAO, Charlottesville	
Tim Cornwell	NRAO, Socorro	Chairman
Bill Cotton	NRAO, Charlottesville	
Geoff Croes	DRAO, Penticton	
Darrel Emerson	NRAO, Tucson	
Steve Grandi	NOAO, Tucson	
Bob Hanisch	Space Telescope	
Gareth Hunt	NRAO, Socorro	
Walter Jaffe	Sterrenwacht, Leiden	
Harvey Liszt	NRAO, Charlottesville	
Ron Maddalena	NRAO, Greenbank	
Pat Murphy	NRAO, Tucson	
Ray Norris	Australia Telescope, CSIRO	
Frazer Owen	NRAO, Socorro	
Bob Payne	NRAO, Socorro	
Dave Roberts	Brandeis	
Arnold Rots	NRAO, Socorro	
Larry Rudnick	UMN, Minneapolis	
Don Wells	NRAO, Charlottesville	

7 Appendix C: User survey

See attached sheet.

Dear NRAO User:

The NRAO is engaged in a comprehensive review of its long-term software strategy. We would greatly appreciate your filling out the attached survey (Right Now!), folding, stamping, and returning it no later than July 12. You are welcome to make any additional comments you wish, and can mail or call any one of us with your concerns or questions. Thanks for your prompt help!.

Sincerely,

Larry Rudnick 612-624-3396 larry@ast1.spa.umn.edu (128.101.221.1)
Bob Burns, bburns@nrao.edu; Tim Cornwell, tcornwell@nrao.edu; Dave Roberts, roberts@brandeis.bitnet
for the NRAO Software Advisory Group

BACKGROUND INFORMATION

Name ______________ Institution ________________

1. How many significant users of NRAO are at your institution? ____

#users	#respond
1	6
2	13
-4	18
-9	29
-19	21
≥20	5

2. Regarding your use of NRAO software, are you (check all that apply):

11 Non-user 87 Regular user 40 System installer/manager 26 Programmer

3. Please allocate your intended observational research work (for next 5 yrs) into the following categories (to total 100% of effort): [estimated means] (bin biased low)

a. Single-dish continuum	8 %;	b. Single-dish line	8 %;
c. Interferometry continuum	21 %	d. Interferometry line	14 %
e. Very Long Baseline	12 %	f. Non-radio observations	14 %

4. What % of your radio-observational computing (over next 5 years) would you like to carry out at: (means- bin biased low)

a. NRAO facilities (personal visit) 12 %
b. NRAO facilities (remote login) 5 %
c. Your home institution 62 %
d. NSF supported supercomputer centers 3 %
e. Other (please specify) 2 %

5. If you use NRAO computing facilities, what are the reasons? (check all that apply)

Nothing else available	11
Convenience	32
No cost	15
Large computing power	23
Latest software	25
Expert computing assistance	32
Scientific contacts	32
Other (please specify)	10

- continued -

-1. Bad idea, don't do. 0. No opinion. 1. Lowest priority for support. -->> 5. Highest priority.

Please remember that with limited NRAO resources, choices and priorities must be made. So - you may mark only a maximum of 3 items in the whole survey as Highest priority (5).

prio rank		mean	#5's
	USER INTERFACE		
1	6. How important is a powerful, easy, flexible user interface?	4.0 *	40 *
	7. What priority is a user interface with a menu-driven capability?	2.8	7
	8. question-answer?	1.7	3
	9. user command line?	2.8	10
	10. graphical (e.g., icons?)	2.1	6
	11. Is it important to execute system level commands without exiting the program?	2.8	7
	ENHANCED FUNCTIONALITY / APPLICATIONS		
5	12. How important is further NRAO basic algorithm research?	3.5 *	28 *
9	13. How important is continued NRAO production of "applications"?	3.3	18
	14. Of what priority is a built-in "map/data arithmetic" capability, e.g., Map2(i,j)= f(i,j,freq)*Map1(i,j) + g(blank(i,j))*Map3(i',j')/Map1) ?	2.6	6
	15. How important are interfaces of data (e.g., slices, outputs of gaussian fits) to pc/Mac packages like Lotus,dBase, Graph?	2.4	6
	16. How important are interfaces to mathematical analysis packages like Mathematica?	1.8	6
	USER PROGRAMMABILITY		
4	17. How important is the ability for you to add custom programs?	3.4 *	29 *
	18. ability to add programs in FORTRAN ?	3.0	15
	19. ability to add programs in C?	2.3	7
7	20. How important is *improving* the ability for users to program within NRAO software?	2.8	20
	RELATIONSHIPS to OTHER PACKAGES		
	21. How important is combining single dish and interferometry software systems?	2.1	6
6	22. FITS currently provides a method for converting data from one package / format to another. How important is it to improve this transportability?	2.9	24
10	23. IRAF is being used / planned for much of the optical and NASA communities. How important is it for NRAO software to be merged with this effort?	2.4	17
	24. A number of other radio astronomy packages exist around the world. Is it important for these efforts to be better integrated or combined?	2.5	7

-1=Bad idea, don't do; 0=No opinion;1=Low....5=High Priority

HARDWARE / SYSTEM CONSIDERATIONS

		#s's
25. NRAO software may evolve to make greater use of emerging industry standards. How important is it to continue supporting a broad range of hardware that may become incompatible with these standards?	2.3	6

PROCESSING EFFICIENCY

26. Given the primary role of hardware in determining execution time, should much effort be spent in improving software execution time efficiency?	2.4	4
27. How important are workstations to your computing needs?	3.8 ★	34★
28. How important are supercomputers to your computing needs?	1.8	4
29. How important is improving efficiency through use of large memories?	2.3	4

GRAPHICS / IMAGE DISPLAY

30. Given the emerging importance of X windows as a networking display standard, how important is continuing support of special displays such as I^2S?	1.4	2
31. How important are interfaces to image display / analysis packages such as apE or AVS?	.6	0
32. Should NRAO put high priority on support for PostScript devices	3.2	20
Film recorders	1.6	3
Color laser printers	2.1	5

SYSTEMS

33. Is it important for NRAO to drop its in-house 'control languages' (e.g., command interpreters, file managers) in favor of publicly available standards? (e.g. POSIX, X/Open Portability Guide, SVID)	1.7	7
34. Should NRAO put priority on adopting C++ as a programming standard?	0.9	0
35. Given the emerging predominance of UNIX/POSIX should portability of NRAO software to other operating systems continue to be a high priority?	2.2	8

STORAGE MEDIA

For which storage media will you require support for the next 5-10y?

36.	1600bpi mag tape	1.1	1
37.	6250bpi mag tape	3.3	17
38.	Exabyte (8mm) tape	2.3	10
39.	DAT (4mm) tape	1.2	3
40.	QIC	0.6	2
41.	TK50	0.3	0
42.	Optical disk	2.2	8
43.	Other	0.2	0

-1=Bad idea, don't do; 0=No opinion;1=Low....5=High Priority

USER SUPPORT

			#s's
(2) 44.	How important is NRAO support for users at their own institutions?	3.9 ★	38★
45.	Distribution of regularly updated software?	3.6	1[illegible]
46.	Use of electronic networks for updates?	2.8	8
47.	Use of tape/other media for updates?	2.6	[illegible]
48.	Problem / bug reporting system?	3.1	1
49.	User documentation?	3.6	2
50.	System management documentation?	2.9	
51.	Programming documentation?	2.9	[illegible]
52.	System management software tools?	2.2	4
53.	Assistance for local problems?	3.0	9
54.	Scientific / analysis consultation?	2.6	9

(45–54: sub-categories)

WHAT ELSE ? Please tell us any of your other concerns/ideas.

5.1/resp

PLEASE FOLD SHOWING RETURN ADDRESS, STAMP & MAIL NOW. THANKS!!