

## I) GENERAL REQUIREMENTS

The AIPS++ system will represent a major investment in software development by a consortium of radio observatories from around the world. Since these observatories hope that the proposed system will be able to be adapted to changing user requirements for years to come, the philosophy of the AIPS++ design is extremely important. In our opinion the new AIPS++ must meet the following general requirements:

- a) The design must not impose unnecessary structure on data.
- b) The design must allow the easy incorporation of new operations and algorithms into the system.
- c) The design must allow an extremely flexible selection of data subsets.
- d) The design should take into account the anticipated growth in parallel processing, both in machine architectures and computer algorithms, in the near future.
- e) The design must take into account the growth in network computing and must allow the use of remote displays, remote batch processing, subsets of parallel processing on different machines in the network, etc.

## II) USER INTERFACE

- a) At least three and possibly four user interfaces must be provided.

- i) An X-windows based icon system must be provided for the neophyte user.

- ii) Both question and answer, and command line interfaces must be provided. A possible example of how to do this is shown in the Khoros image processing system: if the user issues the command 'xyz -P', application xyz starts up and prompts the user for input. If the user types 'xyz -a=432 -b=123 -c=34' then values for a, b and c are directly fed to application xyz; default values would be used for any unspecified parameters.

- iii) A menu style interface must be provided. At a minimum, this must be something like the AIPS1 'setpar' programme, which displays a list of parameters and default values which can be modified by the user. A complex application might have a hierarchy of menus.

- b) Applications must have the ability to save user input into 'parameter' files. The application should be able to read these parameter files so that it can be run again at a later time without the user's having to re-specify the input.

For example, in the question and answer input procedure, questions might be displayed on one half of the screen and answers would be stored in a table which is displayed on the other side of the screen. The next time the programme was run, answers could be retrieved directly from the table and modified before the programme was run.

- c) An application must validate all user input. Also, an application must be 'smart' in the sense that it should warn the user if the combination of parameters that has been specified will cause the application to run for an unreasonably long time. For example, a warning message of the form 'Your input parameters will cause this application to run for about 30 hours on a convex C1. Do you wish to continue?' would be useful.

- d) Both general help for an application as well as specific help

for each parameter to be specified in an application must be available.

e) A good log facility must be provided. The log facility must include the following features:

- - there must be a tool to edit the log.
- - there must be a tool to allow the user to insert notes about what the user has done.
- - if the user wishes, he/she should be able to obtain a complete 'script' of the user's session.
- - the log should, in general, not be as verbose as that of the current AIPS, which contains too much detail. A 'verbose' flag might be provided for users who want every last bit of information.

### III) DATA DISPLAY

a) At a minimum, the equivalent of SAOimage must be provided to display two-dimensional images.

b) Cursor feedback in a variety of coordinate systems, including user-specified ones, must be provided.

c) The user must be able to do comparative astronomy. For example

- - overlay a contour map of one image on top of a gray-scale display of a second image
- - display a contour map and gray-scale display of an image side by side.
- - blink a series of images
- - compare two images side-by-side
- - rotate and stretch (or coordinate transform) one image so that it can be properly compared with a second image. For example, one should be able to read in and scan a Palomar Sky Survey print, pop up the print on the display and align the print with a radio image by clicking a mouse on common features.
- - display the positions of stellar, or other objects', positions on top of a gray-scale image.

d) The system must provide a 'movie' capability to scan through 3-d data cubes. The movie system must have the ability to freeze a frame, reverse, run at different speeds, etc.

e) The system must provide a slicer / dicer facility to examine a data cube from an arbitrary angle.

f) The system should be able to integrate (render, voxel view) the emission from a data cube to show how the data in the cube would appear as seen from an arbitrary position angle. In the longer term, as graphics hardware prices drop, this facility must be expanded so that a movie of a rotating data cube can be displayed in near-real time.

g) The data display should be able to do crude smoothing of an image (and in 3-d, if the user is examining a data cube).

h) The system must allow the user to click on a button and dump the display to either a postscript file or directly to a postscript printer, if the system has one. In addition the movie display system described in item d) above must allow the user to dump each individual frame.

i) The system must allow the astronomer to define areas on the screen such that

- - coordinates can be saved to a file
- - locations and sizes of boxes can be saved to a file
- - integration within a specified polygon region can be done

j) The system should allow the astronomer to perform various filtering functions interactively - for example the user could click on an edge-detection button, and the system would display the edge-detection 'image'.

#### IV) DATA ANALYSIS

a) The system must have a facility to produce publication quality graphics, including gray-scale and contour maps of images as well as data supplied by the astronomer that may come from outside AIPS (e.g. DRAO's PLOT package). Graphics must include coordinate grids, labels, multiple superimposed boxes, display of cuts, multiple graphs, etc. The package must support a variety of output devices, including colour devices.

b) The system should provide an interactive plotting package with, for example, the ability to edit tabular data in one window, and show a modified plot in another window. The reverse operation, putting a cursor at a location on the plot and having the corresponding tabular data show up in the other window, should also be available.

c) The system must provide a tool for the astronomer to manipulate images and perform general mathematical operations on images (equivalent of DRAO MADR package).

d) We require an AIPS++ equivalent of SQL to allow the astronomer to make randomised queries of data, including both UV and image data. The astronomer should be able to retrieve data using commands similar to 'list all pixel values > 15 mJy and < 30 mJy' or 'retrieve all uv points with u > 15000 wavelengths and v < 20000 wavelengths'.

This requirement implies that data must be handled by means of a database management system.

e) AIPS++ should investigate image compression techniques - for example wavelet transforms may allow an image to be compressed by a factor of about 90% with little loss of detail. Data compression techniques will become important when users begin to analyse data over wide-area networks.

f) Standard synthesis telescope data processing algorithms such as Fourier transforms, map making, image deconvolution (e.g. Hogbom clean, Steer-Dewdney clean, maximum entropy), self-calibration, etc. must be supported.

----- End of forwarded message -----