NOTE 208 – AIPS++ TALK SLIDES

Tim Cornwell

June 16 1997

What is AIPS++ and why are we doing it?

- *Consortium* of radio observatories, building package to handle processing for new telescopes, and improved processing for existing telescopes.
- All observatories facing same problem of old packages leading to user dissatisfaction, and limited technical development:
  - Need better tools for end-user, improved interfaces, and new packages
  - Astronomers need to be able to program at *e.g.* IDL-type level
  - Programmers need richer set of tools and packages to work with

Data Uniform access to data

Tools General operations, array math, FFTs, fitting, plotting, display, *etc.*

Tasks Applications for doing radio-astronomy *e.g.* processing of data from radio-telescopes, both single dishs and synthesis arrays

See *http://www.nrao.edu/aips++* for more information

What is AIPS++ being used for now?

ATNF At the Parkes telescope for Parkes Multibeam observing.

NRAO At Green Bank, for support of the Green Bank Telescope engineering.

NFRA At WSRT, integrated into the Telescope Management System.

AIPS++ Project timeline

First beta release *Feb 97* :

- Continuum synthesis imaging and self-calibration
- General Glish-based tools

Second beta release *September 97* :

- Spectral line additions to synthesis processing

Third beta release *Early 98* :

- GUIs for many standard objects
- More synthesis functionality

V1.0 *Mid 98* : Limited Public Release

V1.5 *Late 98* : Code development release

- Code development system (documentation, examples, templates).

V2.0 *Early 99* : Full release

- Completed connected element synthesis package
- Initial VLBI capabilities
- Mosaicing package
- Contributed code

AIPS++ Personnel

ATNF : Neil Killeen, David Barnes, Wim Brouw, and Mark Wieringa

BIMA/NCSA : Dick Crutcher, Dan Briggs, John Pixton, Harold Ravlin, Doug Roberts, and Peter Teuben

NFRA : Jan Noordam, Michael Haller, Friso Olnon, Ger van Diepen, Henk Vosmeijer

NRAO : Tim Cornwell, Bob Garwood, Brian Glendenning, Athol Kemball, Ralph Marson, Joe McMullin, Pat Murphy, Darrell Schiebel, Jeff Uphoff, Kate Weatherall and Wes Young.

What's available in First Beta Release (0.8)?

- The synthesis capabilities of AIPS++ are as follows:
  - Filling from and writing to a UVFITS file,
  - Filling from WSRT format,
  - Full imaging, deconvolution, and self-calibration,
  - Joint deconvolution of Stokes IQUV,
  - Robust, uniform and natural weighting,
  - Flexible windowing in the deconvolution,
  - Non-Negative Least Squares Deconvolution,
  - Flexible construction of models for self-calibration,
  - A sophisticated multi-component model for gain effects,
  - Interactive editing,
  - Flagging of gain solutions by antenna, spectral window, and time interval.
  - Writing of final images to FITS files.
- AIPS++ has a powerful command line interpreter called Glish.
- AIPS++ has a number of general purpose tools that are accessible from Glish.

- Access to all data in AIPS++ via the table module.
- Catalogs of directories, interpreting the contents to show different types of files, and catalogs of potentially everything in Glish (*i.e.* functions and variables).
- Plotting of Glish variables.
- Assorted mathematical capabilities such as statistics, random numbers, polynomial fitting, interpolation, Fast Fourier Transforms and convolutions.
- Manipulation of measured quantities with units and reference frames either from the Glish command line or via a gui.
- Display of AIPS++ or FITS images or Glish arrays using the aipsview program.
- Conversion of images to and from FITS image format, reading and writing AIPS++ images to and from Glish. Statistics, histograms and moments of images may be calculated. Subimaging and padding are both allowed.
- Logging of messages to a gui window and a table, also printing to a postscript printer or to ghostview.

  - Miscellaneous useful utilities including: obtaining AIPS++ configuration information, the help utility, a printer tool (including a gui), execution of shell commands, reading and writing to external files using C-like commands, *etc.*

- All user capabilities of AIPS++ are documented via the AIPS++ User Reference Manual and online help is available from the command line.

Responses to First Beta release (0.8)

- Announced on February 26 1997.
- All consortium sites, and Caltech, University of Iowa, Kapteyn Astronomical Institute, and the National Optical Astronomy Observatory.
- Most feedback on the beta release is conducted either via the AIPS++ Bug Tracking system or or via an e-mail exploder. A fair amount also via private e-mail.
- The responses to beta release can be summarized as follows:
  1. Installation seems relatively straightforward apart from some difficulties with shared libraries.
  2. Configuration does not require much work.
  3. The initial verification of the system via an assay function seems to have gone straightforwardly.
  4. The documentation seems comprehensive but hard-to-understand. Some of the OO terminology is either not explained well or has crept into places where it is not wanted. Testers

have found a (not-unexpected) number of errors in documentation.

5. The user interface is viewed as overly verbose and unfriendly.
6. The synthesis code is viewed as being powerful but hard to master.
7. The synthesis code is slow and subject to memory bloat in some circumstances.
8. There have been a number of bugs, perhaps slightly more in code written in Glish (as opposed to C++).

What's new in Second Beta Release (0.9)?

- Announced on September 11 1997.

Aipsview Many improvements to facilitate control of the program. Can also provide flexible slicing through multidimensional images.

Glish Now considerably more robust.

Image DO Production of moments images, Hanning smoothing and numerous small improvements have been made to the Glish/image interface. FITS history cards are now preserved for images. Asynchronous execution is possible (as for other Distributed Objects). No transpose necessary.

imager Many changes involving spectral line processing. A demonstration of spectral line processing is available as `imagerspectraltest()`. There have been improvements to the speed of all types of processing, and a number of changes to the interface. Imager now traps errors more carefully, and can be interrupted (by Control C) and restarted (by `di.restart()`. Filling of MeasurementSets is now possible from BIMA and WSRT data sets.

Messages Some unnecessarily obscure messages have been eliminated or made less threatening.

Spectral Cubes The FITS/spectral axis interconversion is more robust now.

Responses to Second Beta release (0.9)

- Some errors in FITS processing
- Some worries over channel zero processing
- That's it!

What's planned for the Third Beta Release?

GUIs for standard Distributed Objects : Beta testing showed that we needed a better interface. There will be a standard format interface, autogenerated for most Distributed Objects.

A GUI for imager : The imager interface will be redesigned and a GUI added.

Visibility visualization tool : Glish-based Visibility visualization tool *written by an astronomer with no C++ needed!*

More synthesis capabilities : Need more functionality asap

Single Dish processing : GUI-based tool for Single Dish processing

Improved tools : *e.g.* much improved table browser with editing, selection of columns to display, user-controlled format, SQL-like queries, flexible plotting, *etc.*

AIPS/AIPS++ Interoperability : Can access AIPS Tasks from Glish, also AIPS++ Data and Functions available via client-server interface

Cookbook : Once the look and feel of the interfaces has stabilized, we plan to start on a cookbook.

What's in the works?

Synthesis : to include wide-field (3D) imaging, mosaicing, improved calibration methodology. Also working on parallelization of synthesis code. Great team effort: Briggs, Brouw, Kemball, Marson, Noordam, Wieringa, and Cornwell.

Visualization library : C++-based visualization library. Available to application programmers as a service. Some initial applications will be available soonish. Eventually plan *e.g.* better visibility visualization tool based on this library.

Internals : Rearrangement of internal directory structure, *etc.*

Developer's release : Sometime late 98 or early 99.

Why would I want to use AIPS++ now?

Synthesis unique capabilities:

- **Non-Negative Least Squares deconvolution**
- **IQUV Simultaneous CLEAN**
- **Polarization leakage self-cal**

Ad Hoc processing *e.g.* Strange image or visibility processing.

AIPS GUI Tool Can run AIPS from a GUI (in next beta, or in daily version now).

Examples of Ad Hoc processing

ASCII Table : can convert ASCII table to AIPS++ Table then use AIPS++ and Glish-based math, plotting, browsing tools.

Image :

- Convert FITS to/from AIPS++ Image
- Read/write AIPS++ Image to/from Glish
- Do strange math in Glish

Visibility Data : For example, removal of Cygnus A from 75 MHz data:

- Can convert UVFITS to/from AIPS++ MeasurementSet
- Use `componentmodel` DO to hold model of Cygnus A
- Use `measures` DO to do angle conversions, *etc.*
- Use `ms` DO to read/write **any** part of visibility data to/from a MeasurementSet
- Use Glish to do math
- Use `mathematics` DO to help math

AIPS++ Operations

- Starting to think about Operations of AIPS++:

  Observatories Observatories will use AIPS++ as the vehicle for providing support for particular telescopes. a number of components:

  Software for observing preparation, monitoring of observations, and reduction of data.

  Distribution of ancillary data for the telescope *e.g.* source lists, antenna coordinates, instrument history, physical parameters (such as ionospheric menasurements), *etc.*

  Support for astronomers using the telescope.

  Astronomers Astronomers will use AIPS++ for access to common telescope reduction packages, and for analysis tools that are not the specific responsibility of any one observatory.

  Programmers Programmers (who may be active astronomers) will use AIPS++ as a resource for tools to solve problems and as a distribution mechanism for their products.

- More details in AIPS++ Note 209.

- Core functions:

  Maintenance : Bugs, tracking environment changes

  Distribution : System, Data, Knowledge, Update schedule

  Development : Core library, Applications, System

- Astronomer and programmer support:

  Help :

  Education and training : User and developer workshops, tutorials and demonstrations

  Feedback : email, phone, bug-reports

  Library and contributed code : core, documentation, consortium, contributed, ad hoc.

- Quality Assurance group: Unit testing, application testing, package testing, benchmarking, code reviews.

- Advice and Oversight: AIPS++ User Group and Technical Working Group