AIPS++ DEVELOPMENT PLAN: Release 1.5

Athol Kemball and Tim Cornwell (eds.)
NRAO

15 Nov, 2000

Contents

1 Purpose

The purpose of this document is to define the development plan for AIPS++ release v 1.5, currently scheduled for April 30, 2001.

2 Release priorities

The highest priority for the project at this time is scientific completeness, and planning for release v1.5 has been undertaken in keeping with this objective. Maintaining a sound infrastructure in the project is important however, and infrastructure work has been scheduled for this cycle carefully; primarily in areas which are vital for the long-term vitality of the project, or which are in the critical path for application development.

3 Introduction

Development priorities and targets are listed separately for each major development area in AIPS++. These targets cover only the development cycle through April 2001, and do not include longer-term items, which are tracked separately for consideration in subsequent development cycles (docs/project/devplan_1.6.ps.gz). There are 21 weeks between the week of November 20, 2000, when interim targets for v1.5 were first set, and the expected code-freeze on 13 April 2001, but deductions for defect correction (30%; 1.5 days per week), user support (up to 20%), and where applicable, science time (25%), have been made. The exact numbers vary by developer and institution, and the best estimate has been made in each case. Variability in these estimates is accommodated by the inclusion of medium prioirty targets, which may need to be carried over into the next cycle. Each target is assigned a priority as high (**H**), which implies that the target is expected to be completed this cycle, and medium (**M**), which is work to be done on an as-available basis. Note also, that not all developers are available to the project on a full-time basis, and have reduced commitments as a result.

Abbr.	Developer	Affiliation
AK	Athol Kemball	NRAO
AS	Anuj Sarma	NCSA
BG	Bob Garwood	NRAO
BM	Barry Maguire	NRAL
DK	David King	NRAO
DM	Dave Mehringer	NCSA
DS	Darrell Schiebel	NRAO
GvD	Ger van Diepen	NFRA
GM	George Moellenbrock	NRAO
HR	Harold Ravlin	NCSA
JB	Jim Braatz	NRAO
JM	Joe McMullin	NRAO
KG	Kumar Golap	NRAO
MH	Mark Holdaway	NRAO
MM	Malte Marquarding	ATNF
MW	Mark Wieringa	ATNF
NK	Neil Killeen	ATNF
PC	Paulo Cortes	NCSA
PT	Peter Teuben	BIMA
RM	Ralph Marson	NRAO
RP	Ray Plante	NCSA
OS	Oleg Smirnov	NFRA
TC	Tim Cornwell	NRAO
WB	Wim Brouw	ATNF
WY	Wes Young	NRAO

4 Single-dish

4.1 Priorities

Support GBT commissioning The continued development of tools required for GBT commissioning remains a high priority during this development cycle.

Calibration and imaging capabilities The provision of calibration and imaging capabilities which are integrated with the synthesis infrastructure is a high priority.

Responding to user feedback This is a high priority as many new users will be introduced to the single-dish software this development cycle through GBT commissioning.

4.2 Targets

Consistent use of flags, weights throughout dish (**JM, H, 3 wk**).

SD imaging using imager Extend and test the current SD imaging capabilities in imager. (**JM, H, 2 wk, TC, H, 3 wk**).

Spectrometer backend support (**BG, H, 2 wk**).

IF and LO1 support (**BG, H, 2 wk**).

Initial SD calibration using calibrater Initial calibration examples using a common infrastructure with synthesis. (**JM, H, 3 wk, AK, H, 2 wk**).

GBT commissioning assistance Continued assistance in GBT commissioning; total time listed here. (**JB, H, 7 wk, BG, H, 4 wk**).

dish test scripts Automated test scripts for dish. (**JM, H, 2 wk**).

Improved single-dish documentation (**BG, H, 2 wk**).

Replace sdimager (**JM, M, 1 wk**).

Bad pixel replacement Replace bad pixels by various means (**BG, M, 2wk**).

5 Synthesis

5.1 Priorities

Scientific completeness This remains the primary driver in synthesis development. Significant synthesis capabilities exist within the package at present, and are being more widely used in the scientific community with each release. This process can be aided by continued improvement in vertical integration of synthesis applications, and in the addition of scientific completeness beyond current capabilities.

Time-critical local priorities It is important that the project meet time-critical targets required for successful AIPS++ use at consortia sites, particularly where the use of AIPS++ is closely integrated into the critical path required for instrument use or operation.

Basic automated imaging Significant work in this area took place in v1.4, but needs to continue in this development cycle to include the highest level tools in this area.

Automated testing Expanding the automated testing of synthesis capabilities using simulated data is a priority during this cycle, both at the C++ and Glish level. This has benefits in speeding up release testing, and in daily and weekly checks of system integrity for end-users and user test groups.

5.2 Targets

BIMA filler migration Migrate the BIMA filler to MS v2. (**PT, H, 3 wk**).

MS v2 FITS definition Complete definition of MS v2 in an external FITS binary table format. (**GM, H, 2 wk**).

Additional VLA filler features Add: (i) shadowing; (ii) Tsys and weighting; (iii) complete output filtering; (iv) new weights scheme currently adopted by FILLM. (**RM, H, 4 wk**).

MS data volume control Implement initial solution to the MS data volume problem caused by the MODEL_DATA, CORRECTED_DATA and IMAGING_WEIGHT scratch columns. (**AK, H, 1 wk**).

Automated editing Complete initial automated editing algorithms, including spectral rejection. (**OS, H, 8 wk**).

Investigate coupled solver for ATCA Complete investigate a coupled solver for ATCA data. Includes a comparison with MIRIAD. (**AK, H, 2 wk**).

Complete integration of calibration table re-gridding Complete integration of existing table smoothing and re-gridding tool. (**AK, H, 2 wk, HR, H, 2 wk, DM, H, 1 wk**).

Ephemeris-based planet flux density calibration Enable flux density calibration based on planet observations, as currently done in Miriad. (**RP, M, 4wk**).

Holography support in imager Complete general holography support in imager, excluding near-field corrections. (**JB, H, 4 wk**).

Optimal-sized FFT's for mosaicing Complete the implementation of optimal-sized FFT's for mosaiced imaging. (**MH, H, 2 wk**).

Simulator error models Expand the analytic error models in the simulator for use in correctness testing. Also, complete initial support of pointing errors(**MH, H, 3 wk**).

Automated imaging prototype Provide an initial automated imaging utility, automap.g. (**AK, H, 2 wk**).

Pipeline scientific requirements Complete the scientific requirements document for common pipeline architectures. (**AK, H, 1 wk**).

New Getting Results chapters Add new Getting Results chapters for synthesis reduction for specific instruments, including VLA and BIMA. (**AK H, 1 wk, RP, H, 1 wk, DM, H, 1 wk**).

Synthesis tests Expand test scripts for all consortium connected-element instruments (VLA, ATNF). (**GM, H, 2 wk, DM, H, 6 wk, RP, H, 2 wk**).

Errors in componentlist tables Add support for simple errors (one per existing parameter) in componentlist tables. (**RM, H, 1 wk**).

Ionosphere support in calibrater Complete FJones support in calibrater, including: i) finalize FVisJones classes; ii) test PIM-based corrections with WSRT data; and, iii) add secondary PIM corrections. (**OS, H, 6 wk**).

imager tests against simulated data Add imager tests against simulated data for all imaging modes to imagertest, and assay. (**GM, H, 3 wk**).

VLBI FITS-IDI filler Finalize existing FITS IDI filler. (**BM, H, 6 wk**).

Initial fringe-fitter Complete initial single-band, coherent fringe fitter. (**GM, H, 4 wk**).

Mosaic and wide-field demonstrations Complete demonstration scripts for mosaic, and wide-field imaging. (**MH, H, 2 wk**).

Improved total-power support in mosaicing (**MH, H, 4wk**).

calibrater infrastructure Complete calibrater infrastructure development in interpolation and parametrized solvers (**AK, H, 4 wk**).

Custom synthesis GUI Initial custom synthesis GUI. Includes variants for individual instruments (**AK, H, 2 wk, DM, M, 2 wk**).

ATCA filler enhancements i) birdies; ii) syscal flagging; iii) Hanning smoothing; iv) shadowing; v) documentation (**MW, M, 3 wk**).

6 Applications integration

This includes development aimed at improving the overall integration of applications in AIPS++ in a common framework, using common interfaces and services.

6.1 Targets

Style guide for Refman (**NK, H, 1 wk**).

Revise and update programmer doc Update programmer documentation. Includes exception-handling guidelines and a review of Glish coding rules and templates (**AK, M, 1 wk**).

7 Build and code distribution system

This covers all system work, excluding maintenance and support of the basic library.

7.1 Priorities

Stability A fundamental requirement in this cycle is the continued provision of a stable build to maximize application development and testing efficiency. The project is now operational and build failures have a significant impact on applications development efficiency and the ease of user support.

Initial developer support An initial group of external developers is starting to use the system for exploratory development. This group can be supported in the most cost-effective manner by starting to provide formal support for external development on very restricted platforms in advance of the full developer's release, which is planned for 2001.

7.2 Targets

Complete initial developer's release Document the makedefs, and provide an installation FAQ for an initial developer's release, targeted at RedHat and SuSe Linux systems of restricted version numbers. Test the resulting CDs. (**DS, H, 2 wk**).

Update the project compiler Evaluate gcc 2.95.2 as the project compiler; adopt if review and associated change proposal are accepted. (**AK, H, 2 wk**).

8 Glish

8.1 Priorities

Maintenance The work on Glish is in a stage of consolidation and maintenance. In general, Glish's capabilities are sufficient both for developers to implement higher-level applications and for users to explore their data. The language is also at about the right level; it is capable enough for complex scripts, while still being approachable by general users.

8.2 Targets

Understand speed of TCL/Tk and Glish/Tk There is currently a perceived performance problem in Glish/Tk, which is especially pronounced when drawing large GUIs. The specific origin is unknown. This target involves determining the specific cause by direct comparison with TCL/Tk, including investigating what fraction of the speed differential is caused by byte-compiling. If simple remedies are apparent short of multi-threading, this target includes their implementation. (**DS, H, 4 wk**)

9 User interface and tasking

9.1 Priorities

Consolidation and evaluation Views on user interfaces are inherently subjective; a wider pool of scientific users needs to use and comment on the current automated GUI system, as implemented in toolmanager, before undertaking a second revision of the user interface. This feedback needs to be actively sought, and reviewed during this cycle. This factor, combined with the limited available resources for toolmanager development, argues for consolidation only during this development cycle.

New technology evaluation It is appropriate at this point to evaluate new technology for the Tasking and UI systems.

9.2 Targets

toolmanager Maintenance of toolmanager, and addition of new capabilities in the area of integration and scientific usability. Consolidation of related code in identified areas, and speed improvements (**RM, H, 6 wk**).

Requirements document for revised Tasking/UI system Draft requirements for an upgrade of the current Tasking and User Interface system. (**AK, H, 1 wk**)

Implementation proposal/prototypes for revised Tasking/UI Draw up an implementation proposal and schedule for moving to a revised Tasking/UI system, including proposed middleware, such as CORBA, and the associated Glish interface. Includes simple prototyping and related evaluation. (**DS, H, 5 wk**).

10 Basic library

10.1 Priorities

Consolidation The basic library, which covers all fundamental infrastructure, is mostly complete. Development in this cycle has been scheduled in the area of library maintenance, and in the provision of limited new capabilities required for application development.

10.2 Targets

Large-file support in Tables Add large-file support in the Table system. Includes evaluation of trial LFS available for Linux (**GvD, H, 3 wk**).

Sub-table locking options Add expanded sub-table locking options (**GvD, H, 0.5 wk**).

Arrays in TaQL Support arrays in all TaQL functions (**GvD, H, 1 wk**).

Column removal in TSM Allow column removal in TSM (**GvD, H, 1 wk**).

Deep copy and actualDesc() Provide additional information on the storage manager in use and the actual table description; allow deep copy of tables. (**GvD, H, 1 wk**).

Time comparison function in TaQL Add a time comparison function to TaQL, to specifications provided by synthesis. **GvD, H, 1 wk**).

Exception-handling clean-up Remove CanDelete and Cleanup from the library (**DS, H, 0.25 wk**).

File descriptor limit Change aipsinit scripts to set ulimit/limit on maximum number of file descriptors appropriately (**DS, H, 0.25 wk**).

complex, string, and iostream Move to standard library compliance in these areas. Build stability to be maintained in this process. (**WB, H, 2 wk**).

Measures and Quanta i) tests of baseline, antenna positions and uvw calculations; ii) arrays of measures in Glish; and, iii) quanta improvements per defects and release testing. (**WB, H, 4 wk**).

Fitting i) Image-fitting support; ii) requirements document for component model changes for uv-fitting. (**WB, H, 3 wk**).

Measures chapter for Getting Results Chapter in Getting Results describing end-user Glish use of Measures. (**RP, M, 2 wk**).

11 Images

11.1 Targets

Imagefitter i) add support for error propagation to component models; ii) corrections for spatial correlation. (**NK, H, 3 wk**).

Stokes coordinate handling Fix Stokes coordinate problem relating to the use of the reference value (**NK, H, 0.5 wk**).

Relative coordinates Support relative coordinates (**NK, H, 1 wk**).

DSS FITS images Handle DSS FITS images correctly (**NK, H, 1 wk**).

Extension regions Improve extension region handling (**NK, H, 2 wk, GvD, H, 1 wk**).

Spectral fitter Fit spectra in images (**NK, H, 2 wk**).

Coordinates review Finish coordinates review (**NK, H, 0.5 wk**).

Prepare LatticeStatistics and LatticeHistogram for review Prepare these classes for code review. (**NK, H, 1 wk**).

Vector DisplayData Implement an initial display data class for plotting vectors. (**NK, H, 2 wk**).

12 Visualization

This includes work in the Display Library (DL), and in applications using this library.

12.1 Priorities

Higher-level image visualization The image visualization interface is sufficiently advanced to allow the development of initial high-level image applications.

Provision of uv-visualization services The DL has provided excellent support of image visualization, and continuation of the effort to expand this capability to uv-data visualization is a high priority.

12.2 Targets

kslice3d equivalent Design and implement a kslice3d equivalent (**MM, H, 4 wk**).

Color-wedge DD (**MM, H, 2 wk**).

viewer support of PanelDisplay and MWCAnimator Move from WC in viewer to PanelDisplay and add panelling to viewer. Add MWCAnimator to viewer (**MM, H, 3 wk**).

Relative coordinate support (**MM, H, 1 wk**).

Improve SkyCatOverlay Use TableMeasures. (**MM, H, 1 wk**).

Datamanager improvements FileBrowser and drag-and-drop capabilities (**MM, H, 2 wk**).

Implement DisplayShapes as base class Re-implement rectangle DD, finish ellipse, and add polygon, line and crosshair (**MM, M, 2 wk**).

Vertical integration of uv-DD Make uv-DD visible in the viewer, starting with TableAsRaster (**DK, H, 3 wk**).

Completion of MS uv-DD Completion of MSAsXY, MSAsRaster and MSAsContour (**DK, H, 3 wk**).

Editing EH Basic editing in uv-DD using DL EH (**DK, H, 3 wk**).

Coordinates on WC Implementation proposal for coordinates on the WC (**DK, H, 2 wk**).

Initial interactive editor using TB display data Complete the first TVFLG-like interactive editor using the TB display data. (**DK, H, 3 wk**).

OpenGL or vtk PixelCanvas Investigate OpenGL or vtk PixelCanvas (**HR, M, 6 wk**).

13 Parallelization and high-performance computing

13.1 Priorities

Scientific application The parallelization effort needs to demonstrate a scientifically useful capability to address the most challenging problems in radio astronomy where supercomputer resources are required.

These include wide-field imaging problems at low observing frequencies, mosaicing and the largest VLBI observations, amongst others. In addition, this also includes new algorithms which have not been widely used to date due to limited computing resources.

Parallelization infrastructure A central goal of the parallelization effort is to ensure that infrastructure is developed within the AIPS++ system as a whole to support parallel and distributed computing without expensive ad hoc modifications. This requires that the parallelization infrastructure be compatible with the overall project design, and also that the mainstream project development consider parallelization when implementing algorithms. It is also imperative that the parallelization capabilities be presented using the same user interface as the conventional package.

High-performance computing in AIPS++ The parallelization effort has a strong vested interest in the serial performance of AIPS++ for problems of the largest size, which are defined to be those with exceptional I/O, memory or CPU requirements. It is considered the responsibility of this group to profile the serial performance in these specialized cases, make any changes required to support these large problem sizes, and optimize overall serial performance in these cases.

13.2 Targets

Cluster Linux and IRIX build maintenance Continued maintenance of the existing NCSA/NRAO builds under IRIX and on the AHPCC Linux cluster. (**WY, H, 3 wk, DM, H, 3 wk**).

Key project processing Processing of five key projects (including at least one each of mosaiced, wide-field or large spectral line). Candidates include the existing M33 dataset (Westpfahl), TXCam (Kemball), and a selection of low-frequency VLA projects in A-configuration. Generation of user liaison documentation and user support at NCSA. (**RP, H, 2 wk, AS, H, 2 wk, WY, H, 2 wk**).

Complete multi-field parallelization Complete implementation of a prototype parallelization for mosaiced or wide-field imaging. Candidates include field- or facet-based gridding, model prediction or residual image computation. (**KG, H, 4 wk**).

Parallelization of other deconvolution methods Extend the Clark CLEAN parallelization to other deconvolution algorithms. (**KG, H, 4 wk**).

Initial parallel I/O implementation Implementation of multi-process I/O on the same file, multi-iterator access in a single process, and ROMIO asynchronous I/O using MPI-2. Evaluation of performance in these cases. (**WY, H, 4 wk**).

Complete NT migration document Combine work by P. Cortes in NT migration to formalize incorporation of these changes in the main code distribution (**WY, H, 1 wk, PC, H, 3 wk**).

Modify test suite for large problem size Modify the current bigimagertest to use simulated data. (**DM, H, 1 wk**).

Cluster parallelization Expand evaluation of parallelization methods specific to cluster architectures, with a specific focus on Linux systems. Port AIPS++ to IA64 when available. (**PC, H, 7 wk**)