Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

The AIPS++ Project

J.P. McMullin, K. Golap, S.T. Myers

National Radio Astronomy Observatory

Abstract. The AIPS++ (Astronomical Information Processing System) Project has developed a codebase of libraries, tookits, and applications for the analysis of radio astronomical data. We discuss the overall architecture and core components of this package along with the current technologies in use. The existing package features many applications which are required for the next generation telescopes under development (e.g., multi-field, multi-scale, and wide field imaging algorithms, full primary beam Stokes I,Q,U,V imaging, automated and interactive statistical data editing, flexible calibration and self-calibration based on the telescope measurement equation; we highlight some of this functionality and discuss the future directions for the applications and underlying technology.

1. Introduction

AIPS++ (Astronomical Information Processing System) was developed by an international consortium of observatories (ASTRON, ATNF, JBO, NCSA, NRAO) as a toolkit for the analysis/reduction of radio astronomical data. As of April 2003, the consortium was disbanded; however, development continues based on project/instrument needs. In particular, AIPS++ is the baseline reduction package for the ALMA (Atacama Large Millimeter Array).

AIPS++ was designed as a large-scale, distributed system. Its architecture is based upon distributed components, making it easy to add, replace, or upgrade functionality. This also facilitated the distributed development within the consortium, enabling many workers to participate in the development (i.e. scalable effort). It was also developed with modern languages and modern computer science techniques (e.g. object-oriented programming). This makes the package easily expandable and parallelizable at different levels (from the core C++ libraries to the scripting level).

The underlying mathematical framework for the analysis within AIPS++ is the Measurement Equation (Hamaker, Bregman, Sault, 1996) which provides a general basis for modeling the instrumental errors in the calibration and imaging system. algorithms, and further, it makes no hardware specific assumptions, i.e., no assumptions regarding connected-element arrays or uniformity of antennas.

AIPS++ employs a well defined data model, the MeasurementSet. It was designed to meet the following requirements:

- the format must accomodate synthesis and single-dish data from a variety of telescopes, both current and future, in as broad a framework as possible

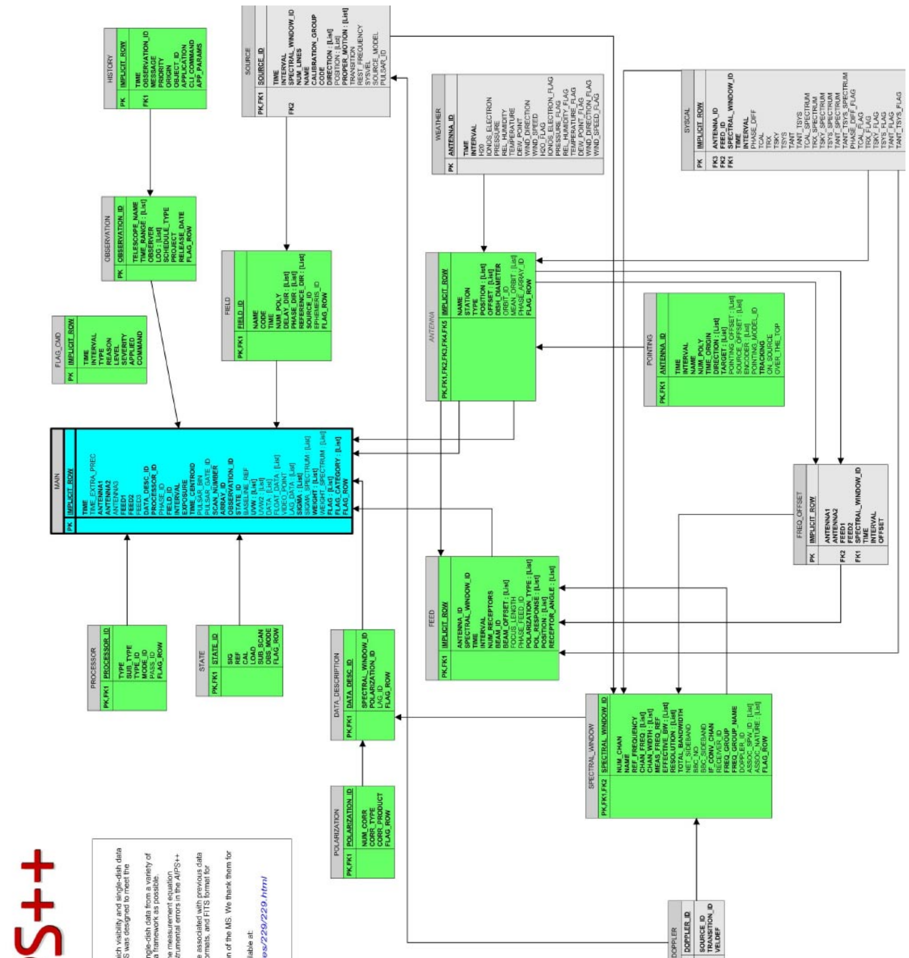

Figure 1. The AIPS++ MeasurementSet.

(i.e., it must be able to adapt to handle unforseen telescope operations, hardware devices, calibration plans, observing modes and data reduction processing).
- it must be compatible with the requirements of the measurement equation formalism.
- it must build upon the knowledge and experience associated with previous data formats such as the AIPS and MIRIAD internal formats and the FITS format for interchange.

2. Toolkit

The AIPS++ code base has extensive functionality. In particular, the core radio astronomy applications are already in place.

1. data fillers - available for many instruments and several archive and interchange data formats
2. editing and visualization - available for both visibility and image data; rasters, contours, vector, and 3-D slice displays.
3. calibration - flexible calibration of all corrupting terms
   - based on underlying measurement process
   - arbitrary parameterization and polarization basis for effects
   - visibility-plane components supported
     - P - parallactic angle correction (pre-computed)
     - C - polarization configuration (pre-computed)
     - G - electronic gain; solvable
     - T - atmospheric correction; solvable
     - D - instrumental polarization response; solvable
     - B - bandpass response; solvable
     - F - ionospheric correction; pre-computed from global, empirical model (PIM)
   - pre-computed or solved using chi-square based on the measurement equation
4. imaging - complete suite of imaging and deconvolution tools
   - supports polarimetry, spectral-line, multiple fields, mosaicing, non-coplanar baselines (simultaneously)
   - single dish OTF, holography
   - clean algorithms: Hogbom, Clark, Schwab-Cotton, Multi-scale
   - incremental multi-field deconvolution
   - Non-Negative Least Squares and Maximum Entropy deconvolution
   - flexible in image size (2^n not needed)
   - supports wide range of coordinate systems
   - tracks moving objects
5. tools for unlimited data access/exploration (e.g., TaQL, LEL)
6. good scripting, pipeline capabilities
7. infrastructure necessary for high level applications development is already in place (e.g., quanta, measures, coordinates, lattices, tables, etc.)
8. provides for development and prototyping at different levels (low level access to higher level applications)

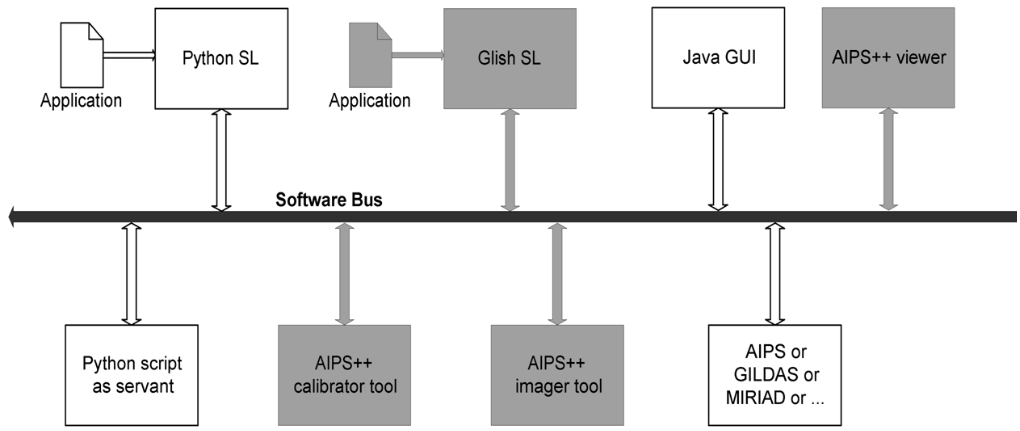

Figure 2. System Framework. The grey boxes indicate the existing framework; the white boxes indicate the additions possible through changes in the software bus.

3. Framework

The existing framework of AIPS++ is built upon proprietary technology (e.g., Glish) and as such represents a fairly closed system. It is unable to take advantage of new technologies wich have become readily available since the original design of AIPS++ over a decade ago (e.g., Python, Java, CORBA, etc). However, replacing Glish as the underlying software bus and task manager implies essentially a new system, providing an opportunity to open the architecture of the package while maintaining and enhancing the science data processing functionality (i.e., the existing C++ application libraries are maintained).

Figure 2 illustrates the existing and future framework for the AIPS++ codebase. Replacing Glish as the software bus with a version of CORBA provides interfaces to other existing languages and tools (e.g., JAVA, Python, etc) which are both well-tested and well-known in the community. Since the original design of AIPS++ used the object-oriented component concept, little work is required to preserve the existing applications/libraries and make them available in such a new framework. As a technology exploration, a prototype pipeline for ALMA has been constructed using AIPS++ applications/libraries called through the ACS environment (ALMA Common Software - a CORBA-based common software infrastructure).

For more information on AIPS++, see: http://projectoffice.aips2.nrao.edu

References

Hamaker, J. P. Bregman, J. D. & Sault, R. J. 1996, A&A117, 137