ALMA Pipeline Infrastructure and CASA

Lindsey Davis (NRAO)
Dirk Muders (MPIfR)

CASA Developer’s Meeting
May 10 – 14
PDOC, Socorro
The ALMA Pipeline Subsystem

The Pipeline Subsystem is divided into 2 operational components: Quicklook Display and Science Pipeline.

The Pipeline software is divided into 2 parts: the infrastructure packages, for both Quicklook Display and Science Pipeline, and the heuristics packages.

The infrastructure packages provide the supporting interfaces, libraries, and services required to operate Quicklook Display and Science Pipeline.

The pipeline heuristics packages encapsulate the radio interferometry and single dish expertise required to pipeline process ALMA data. They support both Quicklook Display and Science Pipeline.
Pipeline Highlights over the Last Year

• Quicklook Display is in use at the OSF / AOS (April 2009 ->)
  • Monitor / display of on-line calibration results, e.g. pointing offsets (April 2009 ->)
  • Monitor / display of incoming data, e.g. amplitudes and phases (April 2010 ->)
  • Automated quick look reduction scripts for standard observing modes (April 2010 - >)
  • Alarms for deteriorating observing conditions (October 2010 ->)

• ALMA Science Pipeline has seen first light (October 2009 -> )
  • First ALMA single dish data sets were reduced end to end (October 2009 ->)
  • First interferometry data sets partially reduced (December 2009 ->)
  • Automated science pipeline reduction scripts for single dish, snapshots, single field interferometry, and pointed mosaic interferometry (April 2010)
  • User Test 6 (May 2010 -> http:alma.sw.eso.org/almasw/bin/view/PIPELINE/UserTest6 )
  • First public release of heuristics scripts including CASA task interfaces (April 2010 ->)

• Commissioning started in January 2010

• End of ALMA construction period is coming up fast
Quicklook: Calibration Monitor Display

Pointing offset chart
Quicklook: Calibration Monitor Display

Pointing offsets versus time
Quicklook Real Time Display

Plot of amplitude vs. channel of WVR data for scan 10
Quicklook Real Time Display

Plot of channel averaged data vs. time color coded by antenna1 for scan 10
Quicklook Real Time Display

Plot of amplitude vs. channel of spectral channel for scan 10
Science Pipeline

• Single dish ALMA results
  • Orion SiO map (First light from single dish heuristics pipeline)
    • http://almasv3.mtk.nao.ac.jp/~george/new/uid___X5f_X235c3_X1_ant1sp9.ASAP_out_html/index.html
  • Orion 12CO map
    • http://alma.mtk.nao.ac.jp/~nakazato/X203X24ceX1spec.DV03.asap_out_html/

• Regressions
  • Single dish, single field interferometry, pointed mosaic interferometry
    • http://alma-heuristics.mpifr-bonn.mpg.de/

• User Test 6
  • Single field and small mosaic interferometry
    • http://almasw.hq.eso.org/almasw/bin/view/PIPELINE/UserTest6
ALMA Pipeline and CASA

• Both Quicklook Display and Science Pipeline systems depend on CASA
• Both Heuristics and Infrastructure packages depend on CASA
• Pipeline developers fully committed to CASA from the start of ALMA construction (2002)
  • Participated in the short-lived AIPS++ / ACS framework experiment
  • Participated in the Glish -> Python tool migration
  • Participated in the Numeric -> Numarray -> Numpy migration
  • Participated in the rpms -> tarball distribution mechanism migration
  • Completed migration from tool only interface to task and tool interface
• Pipeline developers use CASA functionality to
  • Automate display, monitoring, and reduction functions for standard observing modes
  • Develop data driven not user driven display, monitoring, and reductions.
  • Produce data products which are portable and documented sufficiently for publication to an archive
• CASA science requirements are similar for pipeline and desktop processing but pipeline requires access to parseable output, e.g. statistics, …
• Pipeline processing imposes additional infrastructure requirements on CASA, e.g. data documentation, logging, …
Pipeline Infrastructure and CASA

• Data Models
  • ASDM, MS, ASAP tables, Calibration tables, flagging tables, intents, …
• Filler (data import)
• Data selection
  • Tasks, tools, viewers
• Logs and logging (data products)
• FITS support (data export, data products)
• CASA architecture
  • Processes, cluster interfaces, task interface, tool interface, error trapping
• Viewers and graphics tools
• Simulators
• ALMA coordination
  • ACS / CASA libraries (Python !), release cycles, regressions and testing
Pipeline Infrastructure / CASA: Data Models

• ASDMs
  • Intents (DONE), weather (DONE), spectral window naming convention (NONE), backwards compatibility (DONE)

• Measurement sets
  • Intents (DONE, State table OBS_MODE column), weather (DONE), flag data and / or by row, backwards compatibility (DONE ?), changes ?
  • Single dish, migrate from ASAP tables to MS ?

• Calibration tables
  • Finalize format ?
  • Standard format across gain solutions, e.g. splines, e.g. SNR, …?

• Flagging tables
  • No formal association with parent MS, one to one correspondence by row assumed
Pipeline Infrastructure / CASA: Filler

• importasdm

• Auxiliary table support (DONE)

• Intent propagation (DONE)

• Weather integration (DONE)

• Logger integration (DONE)

• Error handling (NONE at Python level, C++ level ?)

• Apply the temperature scale (Will not do as decided at single dish meeting ?)
  • See current importvla
  • Where does CASA do this ?

• Separations of calibrator and target data on filling (?)
  • Old importvla used to do this, prototype pipeline used this
  • We now have a proper intents model

• Averaging on filling ?

• Indexing / Sorting / Splitting on filling ?

May 10-14, 2010, PDOC
CASA Developer’s Meeting
Pipeline Infrastructure / CASA: Data Selection

- **Common syntax and set of selection options**
  - Tasks (DONE)
  - Tools (PARTLY DONE ?)
  - Viewers (PARTLY DONE ?)
  - Supported at C++ level and available to tools (YES ?)

- **Add selection by intent**
  - Intents now supported in core MS
  - Especially viewers, e.g. select by calibrator / target or select by gain calibrator

- **Add selection by subscan**
Pipeline Infrastructure / CASA: FITS Support

- Data export
- Images
  - Standard headers (spatial coordinate system, spectral coordinate system flux calibration, beam parameters, DONE …)
  - Header editing (DONE but inefficient ?)
  - History
    - Full logs end up in image history and get transferred to FITS
    - Too CASA specific ?
    - History editing (NONE ?)
  - Quality assessment
  - Data discovery
- Tables ?
  - Can we create a FITS binary table from a CASA gain table
Pipeline Infrastructure / CASA: Architecture

• Changes with the latest patch build #11097
  • 2 Casapy processes, 4 viewer processes, need to understand this

• Cluster interfaces, HPC
  • Need an update
  • Issues, e.g. Python interface, errors, logs

• Cleanup

• Task interface
  • User mechanism for writing Python tasks (DONE, we use it)
  • Task packaging mechanism (NONE, “We have one of our own …”)
  • User documentation (MINIMAL, guidelines for writing CASA task)
  • Tasks from other tasks (Best way to do this ?)
  • Use of builtin tools in core CASA tasks
Pipeline Infrastructure / CASA: Architecture

• **Tool interface**
  - Pipeline uses private instances of tools in its tasks and packages
  - More user friendly mechanism for creating our own tools (DONE ?)
  - Like to have done methods on all tools so they can be properly dismissed
  - No user mechanism for creating a tool as there is for tasks ?
  - Memory leaks ?

• **Error handling**
  - Tool level (DONE, using exceptions), Pipeline assumes untrapped errors are bugs
  - Core tasks currently don’t throw exceptions ? How then to write scripts based on tasks ?
  - User tasks ? Follow CASA patterns or throw exceptions ?
  - Viewers should not block on exceptions

• **Logger mechanism**
  - Need to be able to define our own logger tool and log file
Pipeline Infrastructure / CASA: Graphics

• Multiple use cases (casaplotms, casaviewer, …)
  • GUI (Quicklook Display, on-line event / data driven, interaction with GUI)
    • Standalone GUI interface only, C++ casaplotms, logging, errors
    • Standalone Python tool + GUI, private pm tool, logging errors
    • Casapy Python tool + GUI, builtin or private pm tool
  • No GUI (Science Pipeline, no interaction with GUI, scripted plots)
    • Standalone Python tool, no GUI, private pm tool, batch mode
    • Casapy Python tool, no GUI, builtin or private pm tool, batch mode

• Remote client support ?

• Requirements
  • Fully scriptable, data selection, plotting options
  • Support error trapping, no blocking on exceptions
  • No foreground / background issues
  • Functionality same as interactive mode

• Matplotlib replacement ?
ACS / CASA libraries versions should be the same (Still issues ?)
  - Python (!!!) , Matplotlib, Numpy, …
  - Minimize third party library dependencies …
CASA cross platform libraries should be the same (DONE ?)
Release cycle coordination with ALMA ?
Please no non-backwards compatible algorithm changes for at least one release cycle (mosaicing)
Regressions
  - Use simulations ?
  - Helpful to include pipeline scripts and data sets as part of regressions ?
  - We need to work on this to make tests more modular
More attention to Pipeline use case …
  - Would benefit us but would also benefit CASA and overall scriptability