



# CASA – the science data analysis package for ALMA

Dirk Petry (ESO), April 2010

## Outline

- What is *CASA*?
  - main features of *CASA*
- Who develops *CASA*?
  - development team
- What are the main requirements and how does *CASA* meet them?
  - design and implementation
- How does *CASA* look and feel?
  - the *CASA* user interface
- *CASA* status and release plans



## CASA main features

- **CASA = Common Astronomy Software Applications**
- **Development started in the 90s as the next generation of AIPS**
- **Refocussed in 2003 to be *the ALMA/EVLA analysis package***
- **Has the intention to be a *general software package to reduce both interferometer and single-dish data***
- **Internally consists of two parts:**

User interface, higher-level analysis routines, viewers  
= *casa non-core*



General physical and astronomical utilities, infrastructure  
= *casacore*

- **Implements the “Measurement Equation” (Hamaker, Bregman & Sault 1996)**
- **Internal data format is the “Measurement Set” (Kemball & Wieringa 2000)**
- **1.5 Million lines of code (mostly C++)**
- **In public release under GNU Public License since December 2009**



# CASA – development team



CASA Developers Meeting, VLA site, February 2008

# CASA – development team



Since mid 2008, two CASA developers at ESO, since Sept. 2009 three



## CASA – development team

**Originally only developed at NRAO (Socorro, NM), now**

**approx. 17 FTE developers are at work at**

**NRAO Socorro (7)**

**NRAO Charlottesville (2)**

**University of Maryland (1)**

**NAOJ (3)**

**University of Virginia (0.5)**

**Observatoire de Paris (0.5)**

**ESO Garching (3)**

+ 1 CASA manager (NRAO Socorro) = Nick Elias

+ 1 Project Scientist (NRAO Socorro) = Jürgen Ott

+ a few 5% FTEs at ASTRON, ATNF, and other places

Also involved:

ALMA Computing Managers = B. Glendenning (NRAO), G. Raffi, P. Ballester (ESO)



# CASA design and implementation

## **Overall architecture:**

- 1) A data structure
- 2) A set of data import/export facilities
- 3) A set of tools for data access, display, and editing
- 4) A set of tools for science analysis
- 5) A set of high-level analysis procedures (“tasks”)
- 6) A programmable command line interface with scripting
- 7) Documentation



# CASA design and implementation

## Overall architecture:

1) A data structure

*Tables: Images, Caltables, and the Measurement Set (MS)*

2) A set of data import/export facilities

*the so-called fillers: ASDM → MS, FITS → Image, UVFITS → MS, VLA → MS, etc.*

3) A set of tools for data access, display, and editing

*tools to load/write data into/from casacore data types,*

*Qt-based table browser, viewer, and (beta) x/y plotter, matplotlib-based x/y plotter*

4) A set of tools for science analysis

*built around the Measurement Equation (developed in 1996) = a set of C++ classes for radio astronomical calibration and imaging*

5) A set of high-level analysis procedures (“tasks”)

*special procedures for each required task such as CLEAN etc.*

6) A programmable command line interface with scripting

*Python (augmented by IPython) gives a MATLAB-like interactive language*

7) Documentation

*an extensive cookbook (500 pages) + documentation through help commands (help, ?, pdoc) + online help pages for users and developers*



# CASA design and implementation

## Overall architecture:

1) A data structure

a) → *Tables: Images, Caltables, and the Measurement Set (MS)*

2) A set of data import/export facilities

*the so-called fillers: ASDM → MS, FITS → Image, UVFITS → MS, VLA → MS, etc.*

3) A set of tools for data access, display, and editing

*tools to load/write data into/from casacore data types,*

*Qt-based table browser, viewer, and (beta) x/y plotter, matplotlib-based x/y plotter*

4) A set of tools for science analysis

b) → *built around the Measurement Equation (developed in 1996) = a set of C++ classes for radio astronomical calibration and imaging*

5) A set of high-level analysis procedures (“tasks”)

*special procedures for each required task such as CLEAN etc.*

6) A programmable command line interface with scripting

c) → *Python (augmented by IPython) gives a MATLAB-like interactive language*

7) Documentation

*an extensive cookbook (500 pages) + documentation through help commands*

*(help, ?, pdoc) + online help pages for users and developers*





# CASA design and implementation

## **CASA special features:**

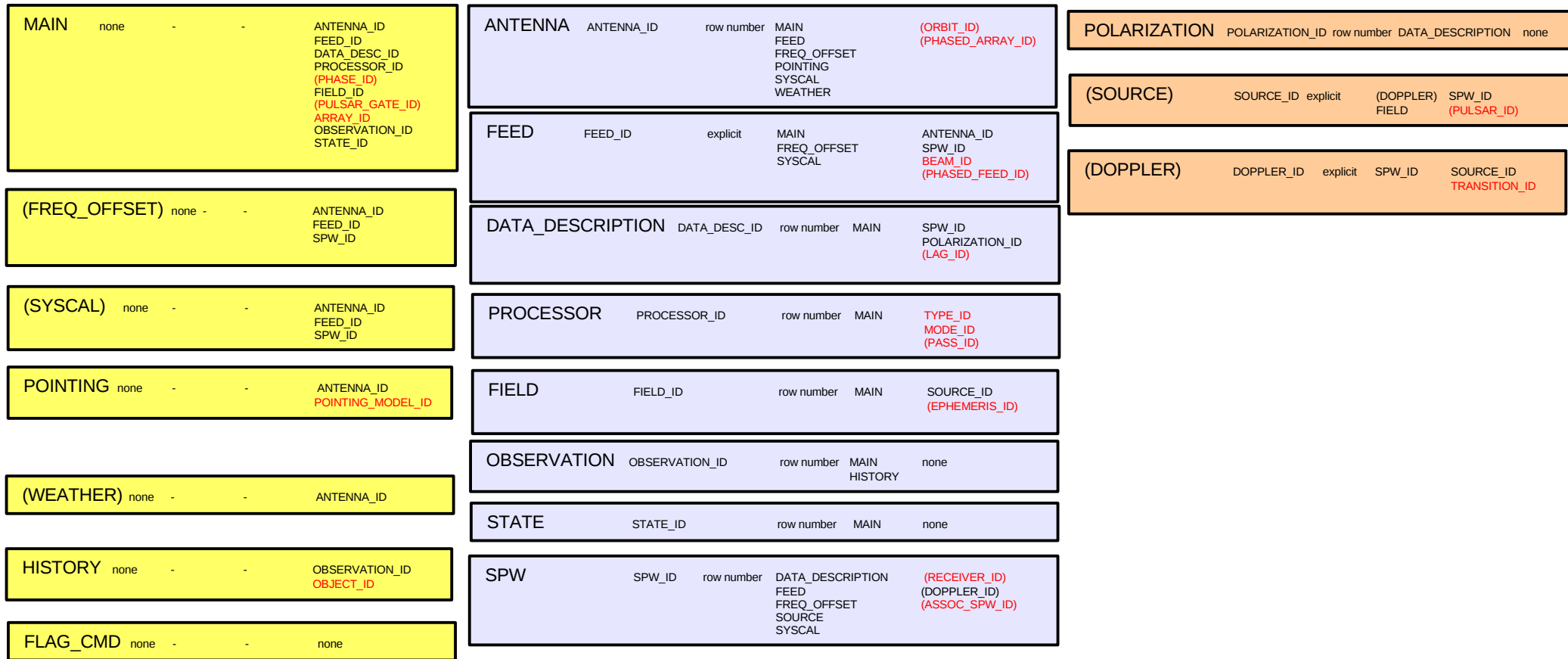
### a) the *Measurement Set* (MS)

- developed by Cornwell, Kemball, & Wieringa between 1996 and 2000
- designed to store both interferometry (multi-dish) and single-dish data
- supports (in principle) any setup of radio telescopes
- supports description and processing of the data via the Measurement Equation
- fundamental storage mechanism: *CASA Tables* (inspired by *MIRIAD*)
- *MS = table for radio telescope data (visibilities) + auxiliary sub-tables*



# CASA design and implementation

## The Measurement Set



Legend:

[Table Name]	[Key defined in this table]	[key definition method]	[referenced by]	[referenced keys] (optional) reference to table outside the MS definition
--------------	-----------------------------	-------------------------	-----------------	--

Level 1: Tables not referenced by other tables

Level 2: Tables referenced by level 1

Level 3: Tables referenced by level 2



# CASA design and implementation

## Example of a *Measurement Set*

```
$ tree AT352_A071103-K
```

```
AT352_A071103-K
```

```
|-- ANTENNA                |-- FIELD                |-- POINTING                |-- SPECTRAL_WINDOW        |-- table.f1_TSM1
|   |-- table.dat          |   |-- table.dat        |   |-- table.dat          |   |-- table.dat          |   |-- table.f2
|   |-- table.f0          |   |-- table.f0        |   |-- table.f0          |   |-- table.f0          |   |-- table.f2_TSM1
|   |-- table.info        |   |-- table.f0i       |   |-- table.f0i         |   |-- table.f0i         |   |-- table.f3
|   `-- table.lock        |   |-- table.info      |   |-- table.f1          |   |-- table.info        |   |-- table.f3_TSM1
|-- DATA_DESCRIPTION      |   `-- table.lock      |   |-- table.info        |   `-- table.lock        |   |-- table.f4
|   |-- table.dat          |   |-- FLAG_CMD        |   `-- table.lock        |-- STATE                |   |-- table.f4_TSM1
|   |-- table.f0          |   |-- table.dat        |-- POLARIZATION          |   |-- table.dat          |   |-- table.f5
|   |-- table.info        |   |-- table.f0        |   |-- table.dat          |   |-- table.f0          |   |-- table.f5_TSM1
|   `-- table.lock        |   |-- table.info      |   |-- table.f0          |   |-- table.info        |   |-- table.f6
|-- DOPPLER                |   `-- table.lock      |   |-- table.f0i         |   `-- table.lock        |   |-- table.f6_TSM1
|   |-- table.dat          |-- HISTORY                |   |-- table.info        |-- table.dat            |   |-- table.f7
|   |-- table.f0          |   |-- table.dat        |   `-- table.lock        |-- table.f0            |   |-- table.f7_TSM1
|   |-- table.info        |   |-- table.f0        |-- PROCESSOR              |-- table.f1            |   |-- table.f8
|   `-- table.lock        |   |-- table.info      |   |-- table.dat          |-- table.f10           |   |-- table.f8_TSM1
|-- FEED                    |   `-- table.lock      |   |-- table.f0          |-- table.f10           |   |-- table.f9
|   |-- table.dat          |-- OBSERVATION            |   |-- table.info        |   |-- table.f10           |   |-- table.f9_TSM0
|   |-- table.f0          |   |-- table.dat        |   `-- table.lock        |   |-- table.f10           |   |-- table.info
|   |-- table.f0i         |   |-- table.f0        |-- SORTED_TABLE           |   |-- table.f10           |   `-- table.lock
|   |-- table.info        |   |-- table.info      |   |-- table.dat          |   |-- table.f10           |
|   `-- table.lock        |   `-- table.lock      |   `-- table.info        |   |-- table.f10           |
                                                                    15 directories, 88 files
```



# CASA design and implementation

## CASA special features:

b) the *Measurement Equation* (Hamaker, Bregman, & Sault 1996 + Sault, Hamaker, & Bregman 1996) implemented as a set of C++ classes for radio astronomical calibration and imaging

$$\vec{V}_{ij} = \vec{M}_{ij} \vec{B}_{ij} \vec{G}_{ij} \vec{D}_{ij} \int \vec{E}_{ij} \vec{P}_{ij} \vec{T}_{ij} \vec{F}_{ij} S \vec{I}_v(l, m) e^{-i2\pi(u_{ij}l + v_{ij}m)} dl dm + \vec{A}_{ij}$$

where

the vectors are:  $V$  = visibility =  $f(u, v)$ ,  $I$  = Image to be calculated,

$A$  = additive baseline-based error component

the matrices are:  $M$  = multiplicative, baseline-based error component

$B$  = bandpass response

$G$  = generalised electronic gain

$D$  = polarisation leakage

$E$  = antenna voltage pattern

$P$  = parallactic angle

$T$  = tropospheric effects

$F$  = ionospheric Faraday rotation

$S$  = mapping of  $I$  to the polarization basis of the observation

other variables and indices are:

$l, m$  = image plane coordinates,  $i, j$  = telescope ID pairs = baseline,  $u, v$  = Fourier plane coordinates



# CASA design and implementation

## CASA special features:

b) *the Measurement Equation* (Hamaker, Bregman & Sault 1996)

implemented as a set of C++ classes for radio astronomical calibration and imaging

(continued)

Assuming, e.g., independence of the matrices from  $(l,m)$ , the ME can be solved for individual calibration components.

$$\vec{V}_{ij}^{obs} = \vec{B}_{ij} \vec{G}_{ij} \vec{D}_{ij} \vec{P}_{ij} \vec{T}_{ij} \vec{F}_{ij} \vec{V}_{ij}^{ideal}$$

ideal visibility known from calibrator source

⇒ have set of linear equations.

The actual calculation of the component is then a  $\chi^2$  minimization.

The calibrator (cb) tool contains a set of ***solvers*** for the different calibration components.

# CASA design and implementation

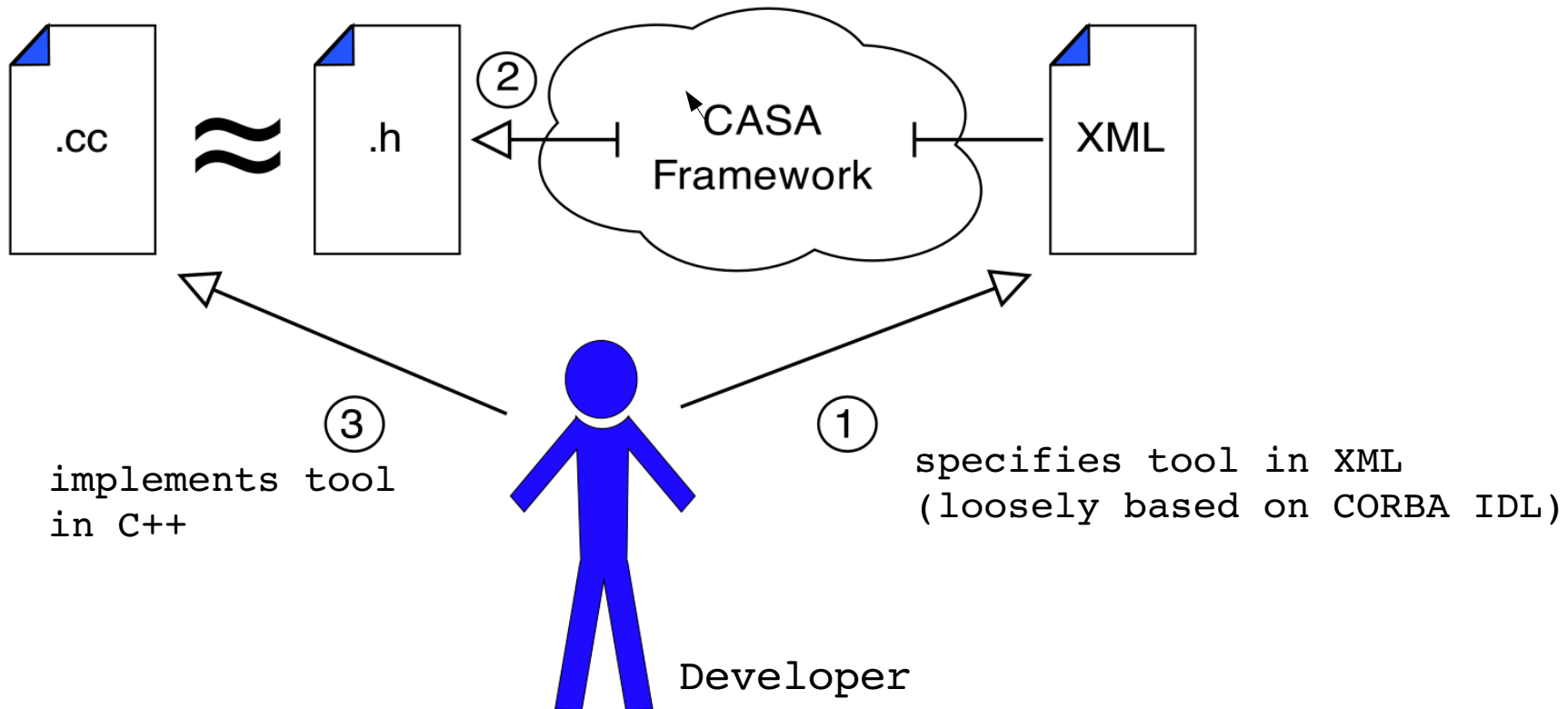
## CASA special features:

c) A programmable command line interface with scripting

Originally AIPS++/CASA had *GLISH*

Abandoned in 2006 for a **Framework Architecture of tools**

Framework generates headers + bindings to scripting language + documentation





# CASA design and implementation

## CASA special features:

c) A programmable command line interface with scripting ([continued](#))

*Framework Architecture of 17 tools can be bound to any scripting language, presently selected is **Python (augmented by IPython)***

at – atmosphere library

ms – Measurement Set utilities

mp – Measurement Set Plotting, e.g. data (amp/phase) versus other quantities

cb – Calibration utilities

cp – Calibration solution plotting utilities

im – Imaging utilities

ia – Image analysis utilities

fg – flagging utilities

tb – Table utilities (selection, extraction, etc.)

me – Measures utilities

tp – table plot

vp – voltage patterns

qa – Quanta utilities

cs – Coordinate system utilities

pl – matplotlib functionality

sd - ASAP = ATNF Spectral Analysis Package (single-dish analysis imported from ATNF)

sm - simulation



# CASA design and implementation

## **CASA special features:**

c) A programmable command line interface with scripting

(continued)

*Python (augmented by IPython)*

*Gives features such as*

- tab completion
- autoparenthesis
- command line numbering
- access to OS, e.g.
  - Lines starting with '!' go to the OS.
  - `a = !ls *.py` to capture the output of 'ls \*.py'.
  - `!cmd $myvar` expands Python var `myvar` for the shell.
- history
- `execfile()`
- comfortable help





# CASA design and implementation

## **CASA special features:**

- c) A programmable command line interface with scripting  
(continued)

***In addition to toolkit: high-level tasks for the standard user***

toolkit (implemented in C++) —► tasks (implemented in Python)

e.g. the task *importfits* is based on the tool *ia* (image analysis):

```
#Python script
casalog.origin('importfits')
ia.fromfits(imagename, fitsimage, whichrep, whichhdu, zeroblanks, overwrite)
ia.close()
```



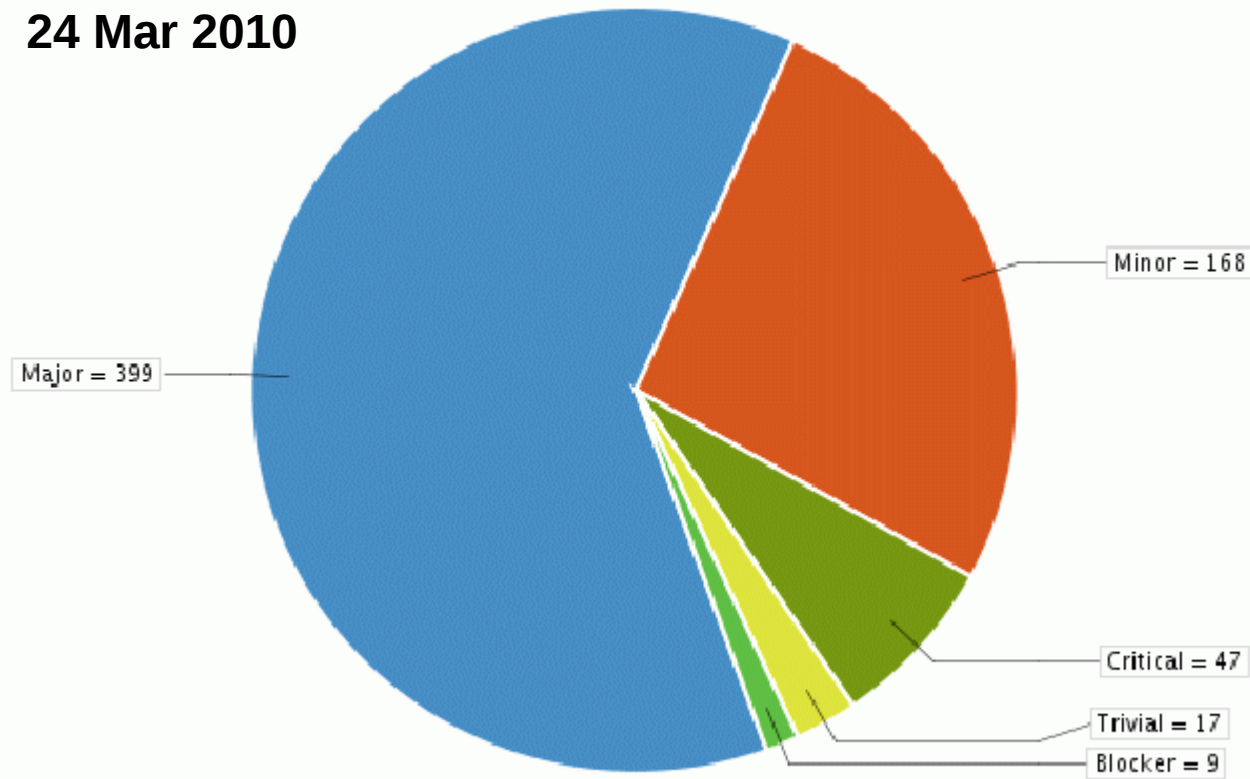
## CASA status

- Since Dec 2009 in public release under GPL = anybody can download, no warranty (see <http://casa.nrao.edu> ), limited support (help desk, needs registration)
- Tutorials for the user community regularly given
- Naming scheme:
  - release X.Y.0 - “major release” (once or twice per year)
  - release X.Y.Z, Z≠0, denotes “patch release” = major release with urgent improvements
- The first public release was CASA 3.0.0 (Dec 2009)
- Release 3.0.1 has been published this month.
- *At this tutorial you are going to use release 3.0.1*
- Development platforms: Linux (RHEL) + Mac OS X
- Supported platforms (binary distribution): RHEL, Fedora, openSuSE, Ubuntu, Max OS X
- Code kept in *svn* repository at NRAO, Socorro
- Presently have approx. 4200 modules, 1.5E6 lines of code, 1E6 lines of comments
- The core functionality (mostly derived from AIPS++) was split off into *casacore* for use by other projects (also available at <http://code.google.com/p/casacore/> )

# CASA status

- Status in terms of JIRA tickets (bug reports and requests for new features)

24 Mar 2010

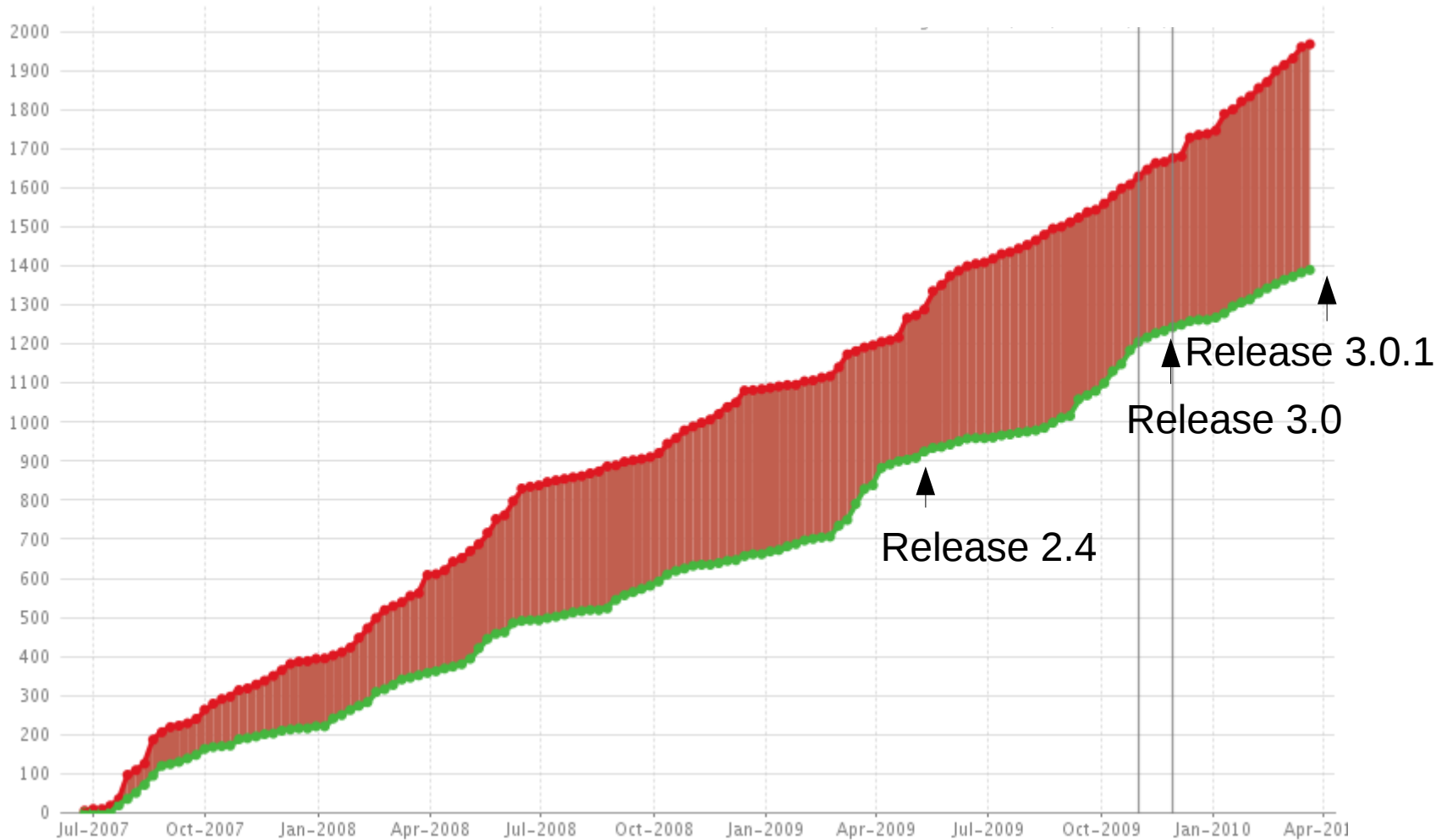


	Issues	%
Major	<u>399</u>	62%
Minor	<u>168</u>	26%
Critical	<u>47</u>	7%
Trivial	<u>17</u>	2%
Blocker	<u>9</u>	1%

*(there has been a certain inflation in the priority classification of tickets)*

# CASA status

## JIRA tickets



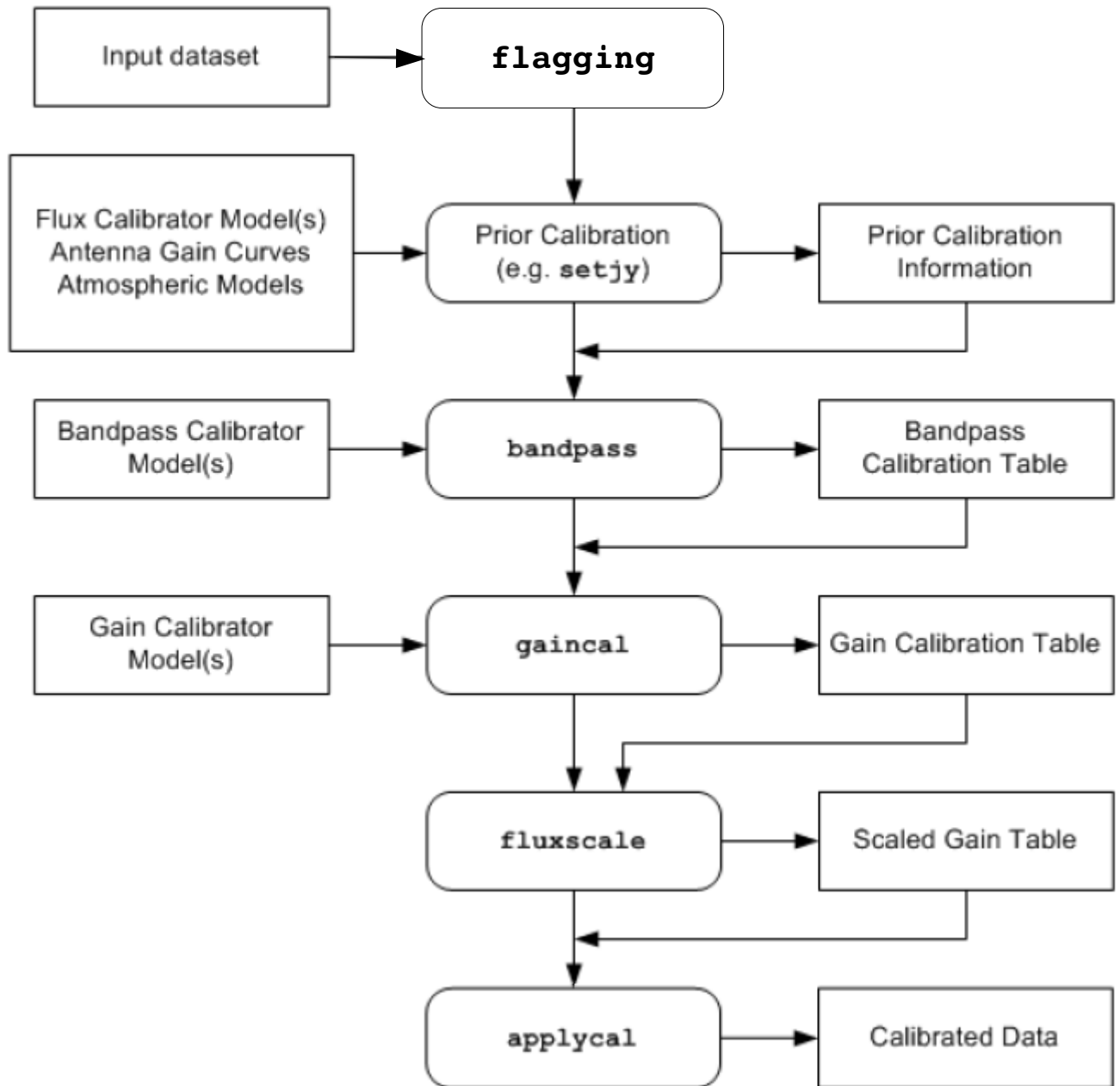
*For release 3.0 approx. 225 issues were resolved/closed, another 180 for 3.0.1. These include not only bug fixes but also new features and improvements. Testing has intensified leading presently to an increased rate of issue reports.*



# How does CASA look and feel?

## A typical analysis session

### Part 1: flagging and calibration

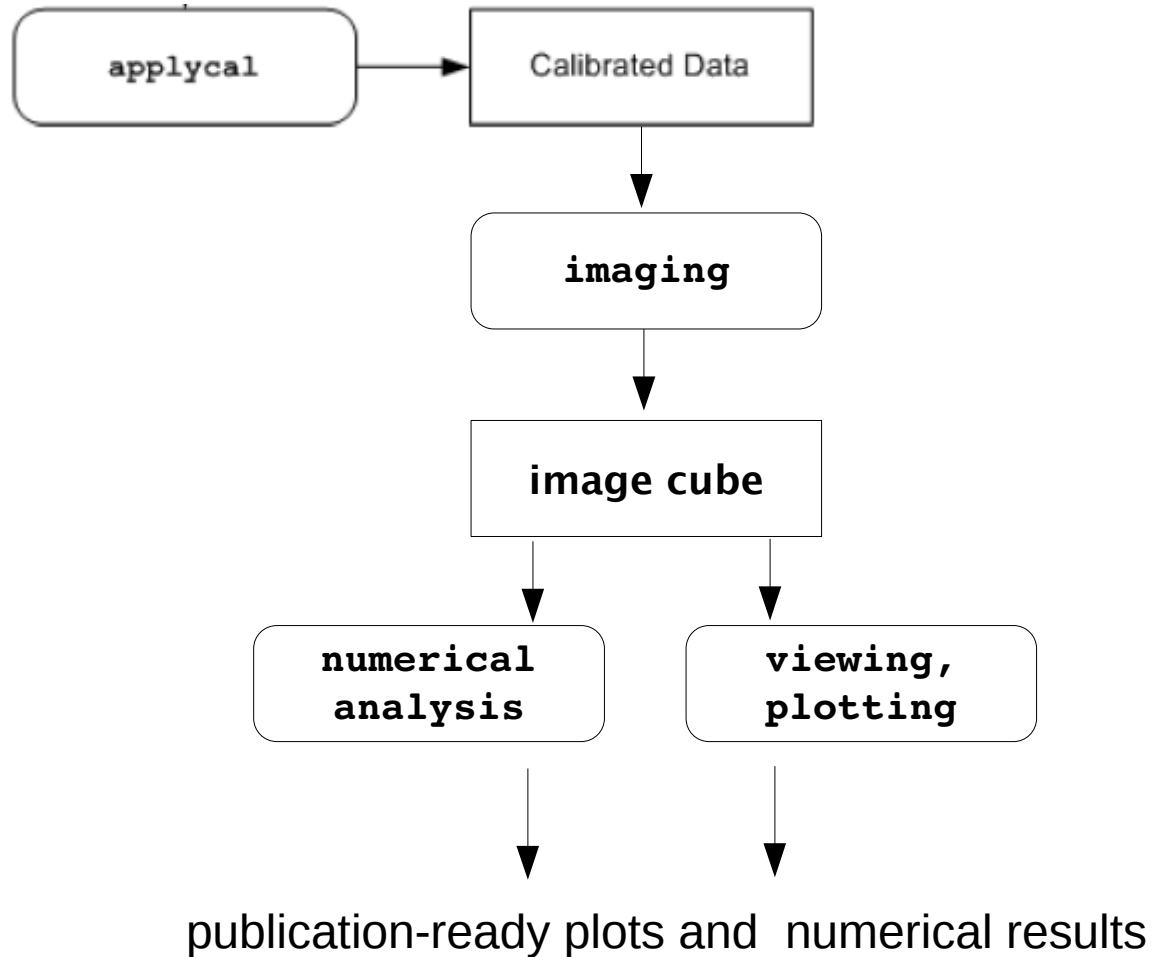




# How does CASA look and feel?

## A typical analysis session

Part 2: imaging and  
image analysis





# How does CASA look and feel?

## Pictures from a typical analysis session

- 1) Startup:  
open terminal and start *casapy*

*Available tasks and tools are listed and the logger window is opened.*

```
dpetry@M83:~/temp/casa-bologna2010
[dpetry@M83 casa-bologna2010]$ casapy
CASA Version 3.0.1 (r11099)
Compiled on: Thu 2010/04/15 04:08:39 UTC

-----
For help use the following commands:
tasklist           - Task list organized by category
taskhelp           - One line summary of available tasks
help taskname      - Full help for task
toolhelp           - One line summary of available tools
help par.parametername - Full help for parameter name
Single Dish sd* tasks are available after asap_init() is run
-----
Activating auto-logging. Current session state plus future input saved.
Filename          : ipython.log
Mode               : backup
Output logging    : False
Raw input log     : False
Timestamping      : False
State              : active

CASA <2>: █
```



# The CASA user interface

*The logger provides functionality for monitoring and debugging command execution.*

The screenshot shows a window titled "Log Messages (M83:/export/home/M83/dpetry/temp/casa-bologna2010/casapy.log)". The window contains a table of log entries. The columns are Time, Priority, Origin, and Message. The log entries show the execution of tasks like plotms and plotxy, and the start of a calibration process. The calibration process includes selecting baselines, fields, spectral windows, and scan numbers.

Time	Priority	Origin	Message
2010-04-23 12:04:03	INFO	plotms:::...	##### Begin Task: plotms #####
2010-04-23 12:04:03	INFO		plotms:::casa
2010-04-23 12:04:04	INFO		plotms:::casa
2010-04-23 12:04:04	INFO	plotms:::...	##### End Task: plotms #####
2010-04-23 12:04:04	INFO	plotms:::...	#####
2010-04-23 12:08:11	INFO		plotxy:::casa
2010-04-23 12:08:11	INFO	plotxy:::...	#####
2010-04-23 12:08:11	INFO	plotxy:::...	##### Begin Task: plotxy #####
2010-04-23 12:08:11	INFO	plotxy:::...	plotxy:::casa
2010-04-23 12:08:11	INFO	plotxy:::t...	Switching to GUI mode. All current plots will be reset.
2010-04-23 12:08:11	INFO	plotxy:::...	Adding scratch columns, if necessary.
2010-04-23 12:08:11	INFO	calibrate...	Opening MS: ah847_1-k-selected-flagged-calibd.ms for calibration.
2010-04-23 12:08:11	INFO	Calibrate...	Initializing nominal selection to the whole MS.
2010-04-23 12:08:12	INFO		Data to be selected from matches the following:
2010-04-23 12:08:12	INFO	+	Baselines: *ALL pairs of* -- VA01, VA02, VA03, VA04, VA05, VA06, VA07, VA08, VA09, VA10, VA11, VA12, VA13, VA14, VA15, VA16, VA17, VA18, VA19, VA20, VA21, VA22, VA23, VA24, VA25, VA26, VA27, VA28, VA29, VA30, VA31, VA32, VA33, VA34, VA35, VA36, VA37, VA38, VA39, VA40, VA41, VA42, VA43, VA44, VA45, VA46, VA47, VA48, VA49, VA50, VA51, VA52, VA53, VA54, VA55, VA56, VA57, VA58, VA59, VA60, VA61, VA62, VA63, VA64, VA65, VA66, VA67, VA68, VA69, VA70, VA71, VA72, VA73, VA74, VA75, VA76, VA77, VA78, VA79, VA80, VA81, VA82, VA83, VA84, VA85, VA86, VA87, VA88, VA89, VA90, VA91, VA92, VA93, VA94, VA95, VA96, VA97, VA98, VA99, VA100
2010-04-23 12:08:12	INFO	+	Fields: *ALL* -- 12190+47182, 12191+48299, 1331+305
2010-04-23 12:08:12	INFO	+	Spectral Windows: *ALL* --
2010-04-23 12:08:12	INFO	+	SPW 0: *ALL Channels* -- 1 to 1 with a step of 1
2010-04-23 12:08:12	INFO	+	SPW 1: *ALL Channels* -- 1 to 1 with a step of 1
2010-04-23 12:08:12	INFO	+	Correlations:
2010-04-23 12:08:12	INFO	+	Corr. ID 0 - RR, RL, LR, LL
2010-04-23 12:08:12	INFO	+	Corr. ID 1 - *NONE*
2010-04-23 12:08:12	INFO	+	Time Range *ALL* -- 2004/5/22/01:06:05 to 2004/5/22/03:32:25
2010-04-23 12:08:12	INFO	+	Scan Numbers: *ALL* -- 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100
2010-04-23 12:08:12	INFO	+	UVRanges: *ALL*
2010-04-23 12:08:12	INFO		Preparing data ...
2010-04-23 12:08:21	INFO		Now get the data
2010-04-23 12:08:23	INFO		Done Processing data ...
2010-04-23 12:08:24	INFO		Now get the data
2010-04-23 12:08:25	INFO		Done Processing data ...
2010-04-23 12:08:26	INFO		plotxy:::casa
2010-04-23 12:08:26	INFO	plotxy:::...	##### End Task: plotxy #####
2010-04-23 12:08:26	INFO	plotxy:::...	#####





# The CASA user interface

## Pictures from a typical analysis session

2) enter commands in a MATLAB-like environment

recall previous settings

list present settings for given task (includes parameter verification)

```
dpetry@pc014720:~/temp/radio-analysis/cqtau+mwc480 - Shell - Konsole
Session Edit View Bookmarks Settings Help

CASA <15>: fluxscale(vis='AT352_A071103-K', caltable='AT352_A071103-K-gain', fluxtable='0', transfer='1')

CASA <16>: applycal(vis='AT352_A071103-K', gaintable='AT352_A071103-K-gain', field='2')

CASA <17>: tget clean
-----> tget(clean)
Restored parameters from file clean.last

CASA <18>: inp
-----> inp()
# clean :: Deconvolve an image with selected algorithm
vis                = 'AT352_A071103-K' # name of input visibility file
imagenam          = 'cqtau-3-target'  # Pre-name of output images
field              = '2'              # Field Name
spw                = ''              # Spectral windows:channels: '' is all
selectdata        = False            # Other data selection parameters
mode               = 'mfs'           # Type of selection (mfs, channel, velocity,
niter              = 500             # Maximum number of iterations
gain               = 0.1             # Loop gain for cleaning
threshold          = '0.0mJy'       # Flux level to stop cleaning. Must include
psfmode           = 'clark'         # method of PSF calculation to use during min
imagermode        = ''              # Use csclean or mosaic. If '', use psfmode
multiscale         = []              # set deconvolution scales (pixels), default:
interactive        = True            # use interactive clean (with GUI viewer)
npercycle         = 100             # Number of iterations before interactive pro

mask               = []              # cleanbox(es), mask image(s), and/or region(
imsize             = [512, 512]     # x and y image size in pixels, symmetric for
cell               = ['0.03arcsec', '0.03arcsec'] # x and y cell size. default unit
phasecenter        = ''             # Image phase center: position or field index
restfreq           = ''             # rest frequency to assign to image (see help
```



# The CASA user interface

## Pictures from a typical analysis session

3) where needed, tools have GUIs:

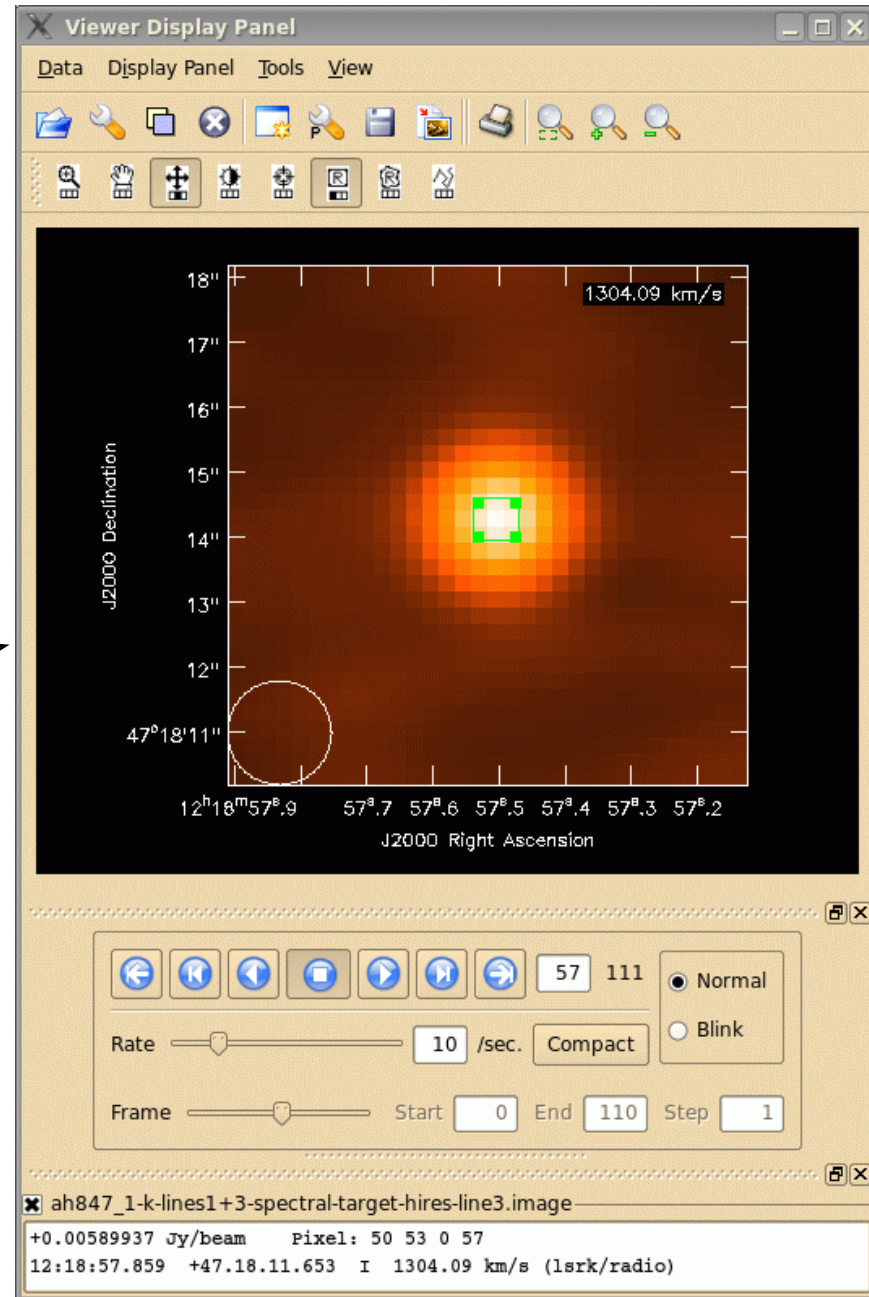
plotxy, plotcal, browsetable,  
viewer, clean

(started in separate threads)

The **viewer** is a powerful multi-function tool for data selection and visualization.

Uses Qt widget set  
(but 80% independent)

Rendering based on pgplot



# The CASA user interface

## Pictures from a typical analysis session

3) where needed, tools have GUIs:

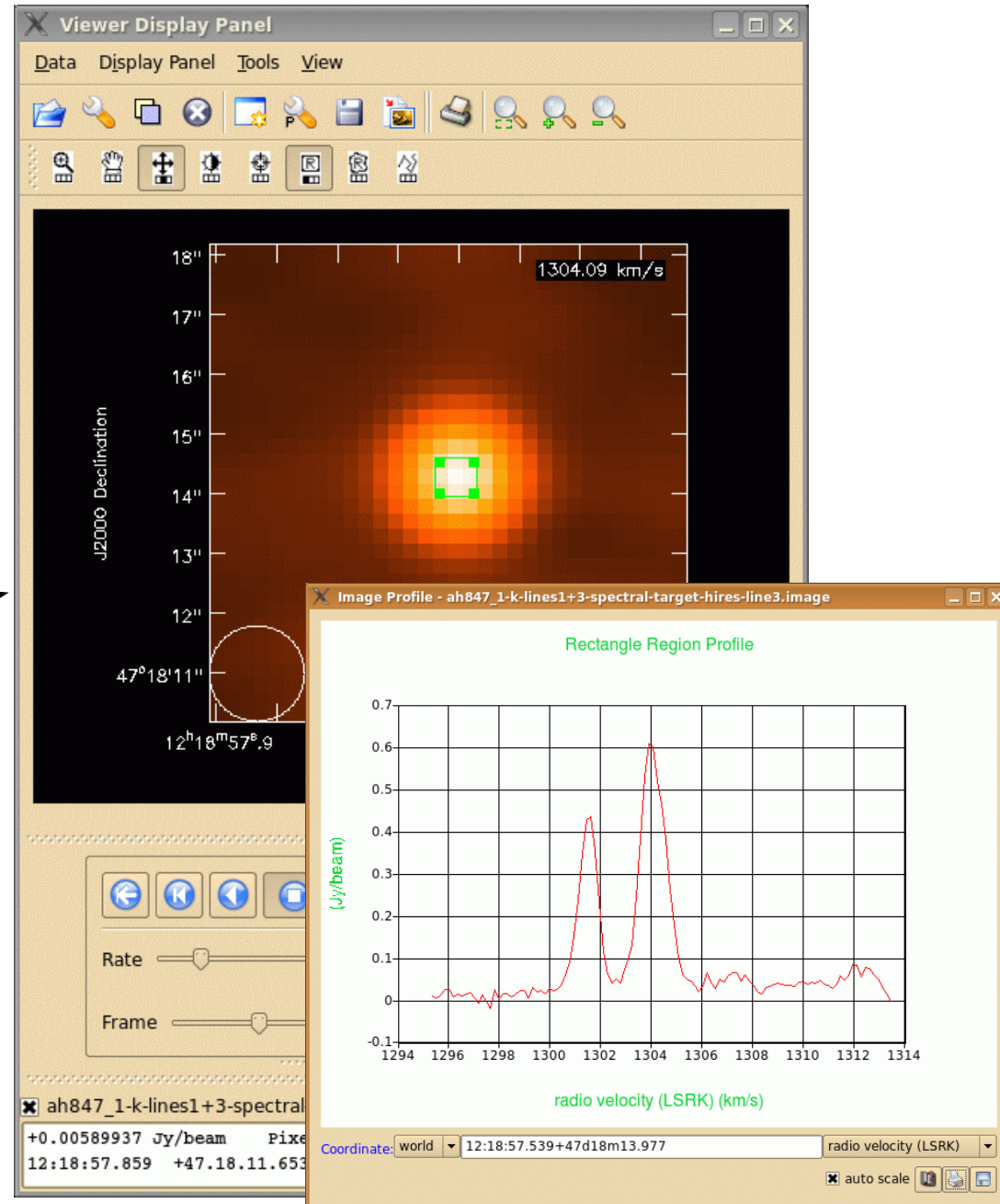
plotxy, plotcal, browsetable,  
viewer, clean

(started in separate threads)

The **viewer** is a powerful multi-function tool for data selection and visualization.

Uses Qt widget set  
(but 80% independent)

Rendering based on pgplot





# The CASA user interface

## A typical analysis session

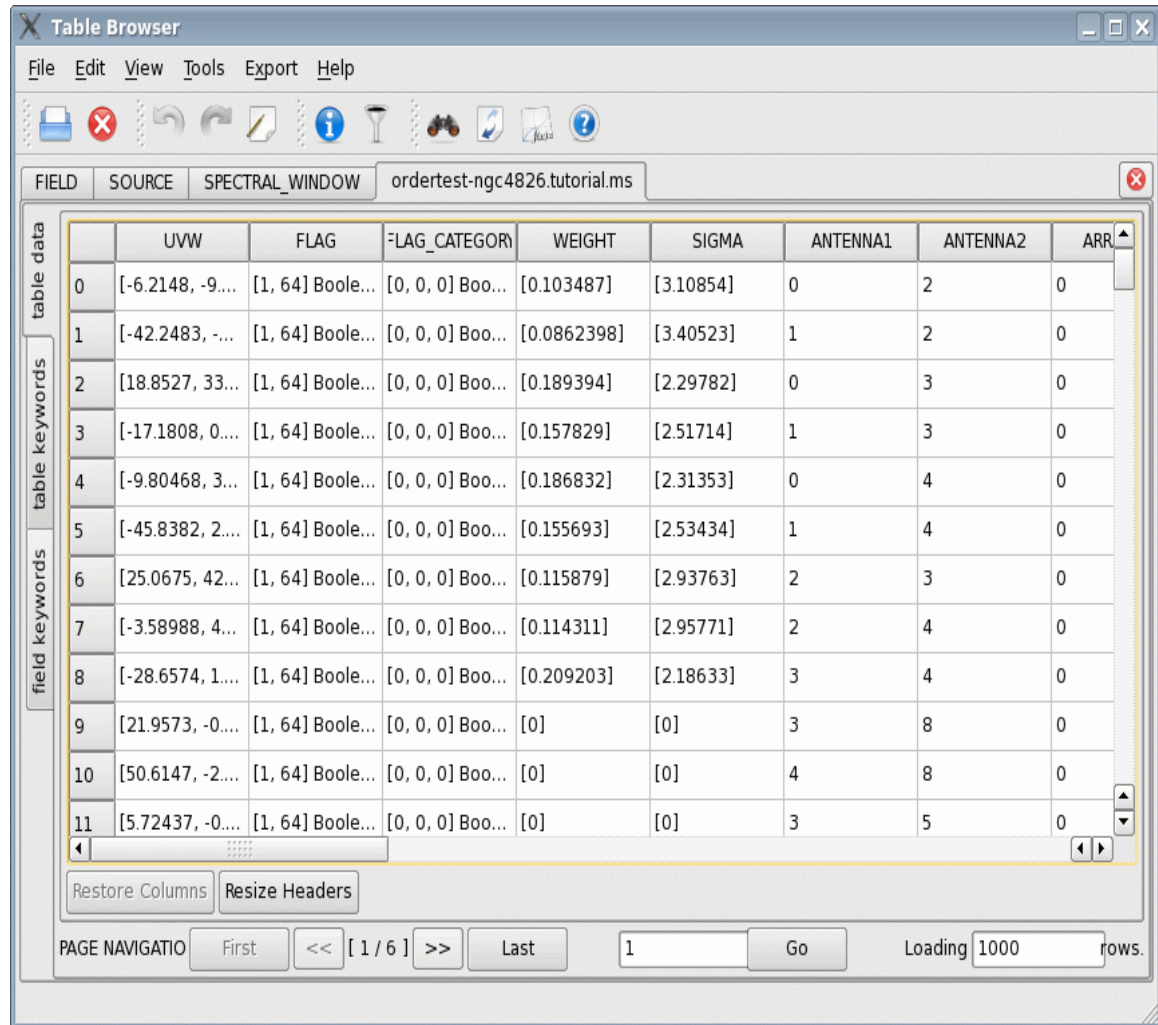
3) where needed, tools have GUIs:

plotxy, plotcal, browsetable, viewer, clean

(started in separate threads)

***browsetable*** permits you to explore any CASA table, e.g. Measurement Sets

Also Qt-based.



# The CASA user interface

## A typical analysis session

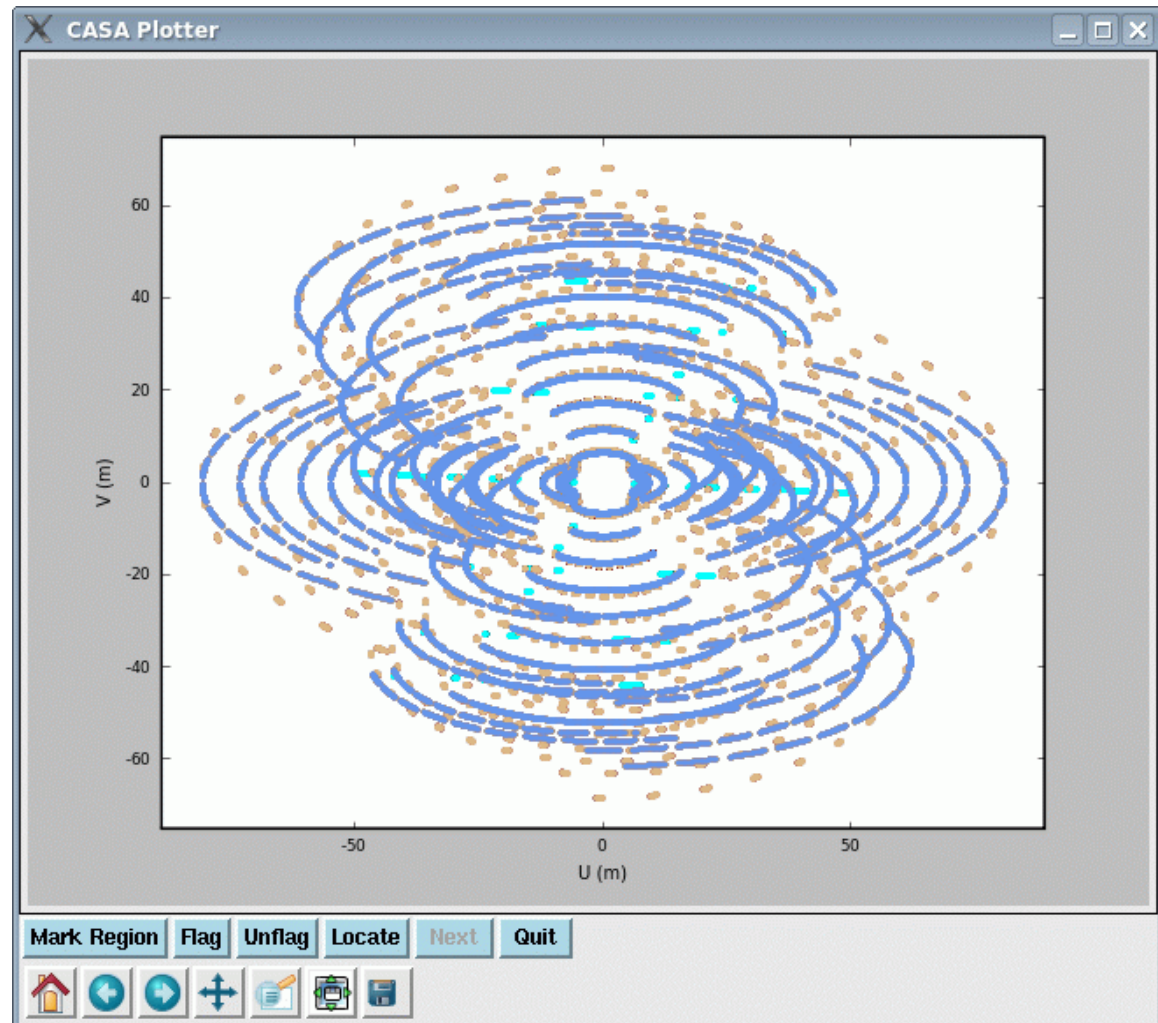
3) where needed, tools have GUIs:

plotxy, plotcal, browsetable,  
viewer, clean

(started in separate threads)

*plotxy* is a specialized tool  
for diagnostic plots and  
data selection

To be phased out.



# The CASA user interface

## A typical analysis session

3) where needed, tools have GUIs:

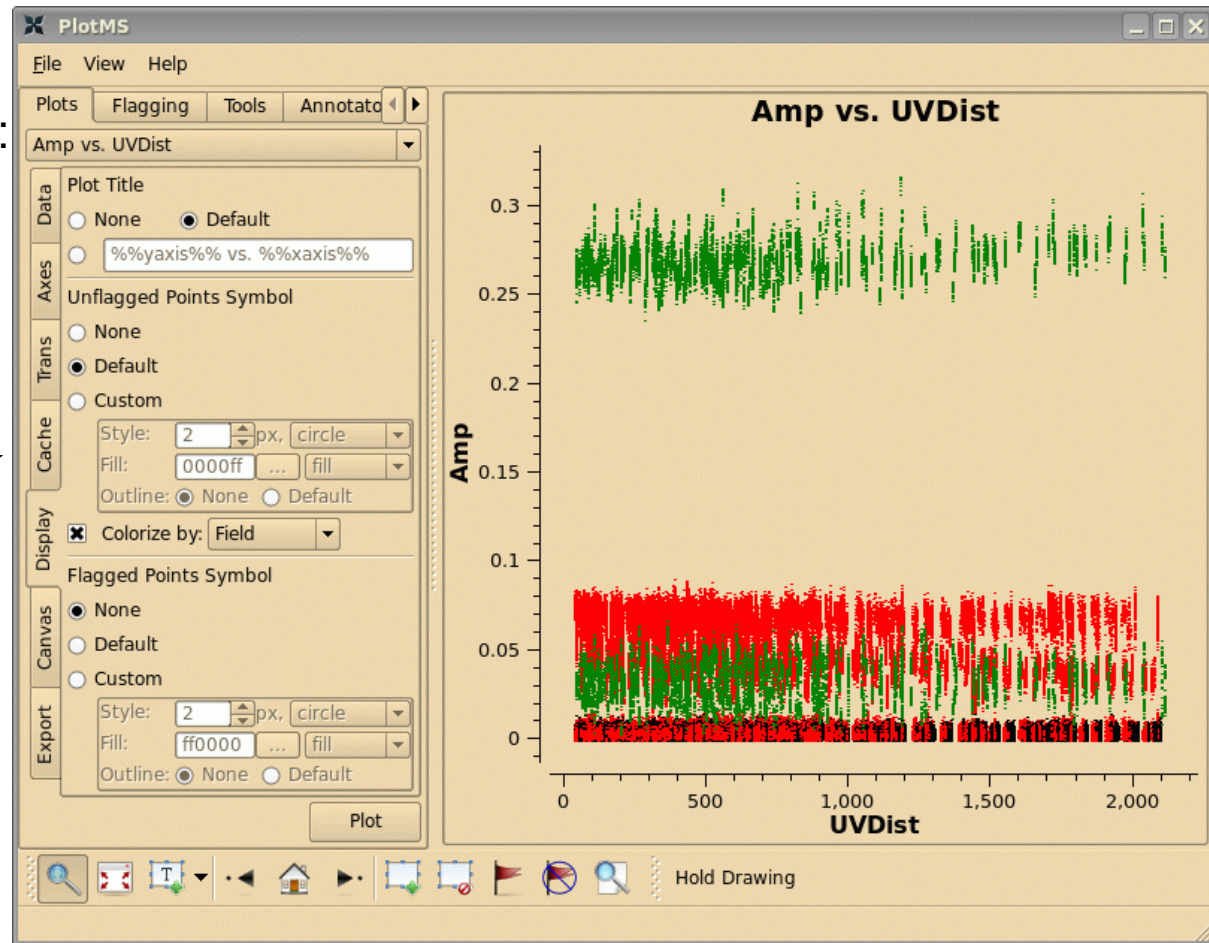
plotxy, plotcal, browsetable, viewer, clean

(started in separate threads)

*plotms* is going to replace plotxy. Release 3.0.0 contains beta version.

plotms is Qt-based and much faster than plotxy.

Uses generic plotting class which in turn uses **Qwt**.





## Summary

- The standard science data analysis package for ALMA and EVLA is **CASA**
- Data from other observatories can also be processed, e.g. VLA, BIMA, ATCA, ...
- CASA derives from AIPS++ (partially survives in **casacore**)
- approx. 20 people are working on CASA  
in North America, Europe, and Japan
- CASA is a **toolbox** with
  - MATLAB-like user interface
  - GUI tools for data selection, browsing, and image processing
- the heart of the science analysis code is the **Measurement Equation**
- the internal data format are **CASA Tables**
- the **Measurement Set** is the CASA data format for visibility data  
(it is technically a Table with several well-defined sub-tables)
- CASA is publicly available under GPL for **Linux and Mac OS X**
  
- The first public release of CASA (version 3.0.0) became available in December 2009
- The latest release is version 3.0.1