ALMA Software Project Management Lessons Learned

G. Raffi[1], B.E. Glendenning[2]

[1]*European Southern Observatory, Garching, Germany*
[2]*National Radio Astronomy Observatory, Socorro, New Mexico, USA*

ALMA Software Project management, Lessons learned - G. Raffi, B.E. Glendenning

ALMA Project in Summary

- 54 x 12m + 12 x 7m antennas, 30-950 GHz
- Array configurations:150 m-14 Km
- Near S. Pedro de Atacama, Chile at 5000 m
- EU and North America as equal partners, East Asia at 25%
- Construction phase 2003-2012
- Early Science foreseen for mid 2011

- ALMA is the largest ground based astronomical project under construction.
- Upcoming astronomy projects include 30-42 m optical telescopes but also radio projects (e.g. SKA)

Software Scope

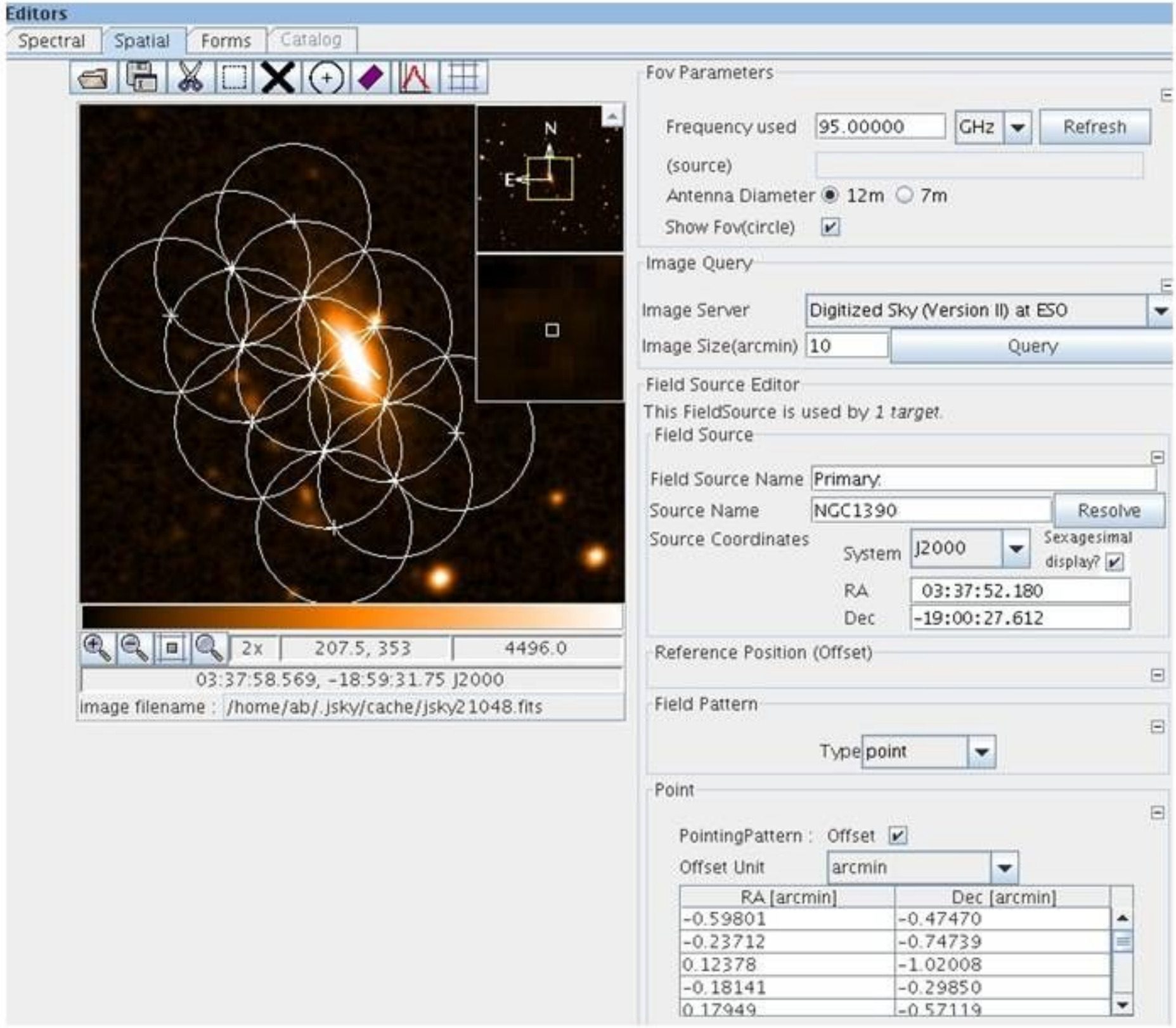
end-to-end (e2e) software system

- From the cradle…
  - Proposal Preparation
  - Proposal Review
  - Program Preparation
  - Dynamic Scheduling of Programs
  - Observation
  - Calibration & Imaging
  - Data Delivery & Archiving

- Afterlife:
  - Archival Research & Virtual Observatory (VO) Compliance

ALMA Computing

Observation Preparation: Spatial Editor



Project Management Aspects

- Large but extremely distributed team. About 80 Full Time Equivalent including the support group in Chile.
- Staff in 15 locations in 4 continents
- Geographical distribution with this size & pace is difficult

… but we believe to be a very good example of a collaborative project

- ➢ Computing Subsystems mixed across continents (sometimes)
- ➢ Acceptance of common software (optimized for system, not for everybody's taste & mandatory) => *Requires team spirit.*
- ➢ Some inefficiency is inevitable (but better checked design)

✓ Most important: Wiki, CVS, regular telecons, face to face meetings

Project Management Approach

Requirements (and Use Cases):

- Requirements working group to be recommended
- There will be still missing or late requirements, but design is done upfront
- Tracking requirements completion to show progress (planned vs. actual) (we use Telelogic's DOORS)

Project Management Approach (cont.)

Using a software framework is essential (ALMA Common Software - ACS #)
(but most of this would apply also to EPICS, TANGO etc):

- Allows collaborative work, results in an homogeneous system
- Provides a solid debugged base of software
- Enforces also hardware standards and operating system versions
- Makes large distributed projects manageable and maintainable
  - .. But requires team discipline and managerial support
  - And learning (yearly ACS courses)

(#) Based on the container-component paradigm and using CORBA. The system allows the use of C++, JAVA and Python on Linux operating systems.

Started with a collaboration with M.Plesko &Co (JSI, now Cosylab) based on ANKA ... around ICALEPCS'09 (10 years of ACS !) This is still on-going.

- *ACS is free under GNU LGPL license..*
- *And has its own circulation, also outside ALMA.*

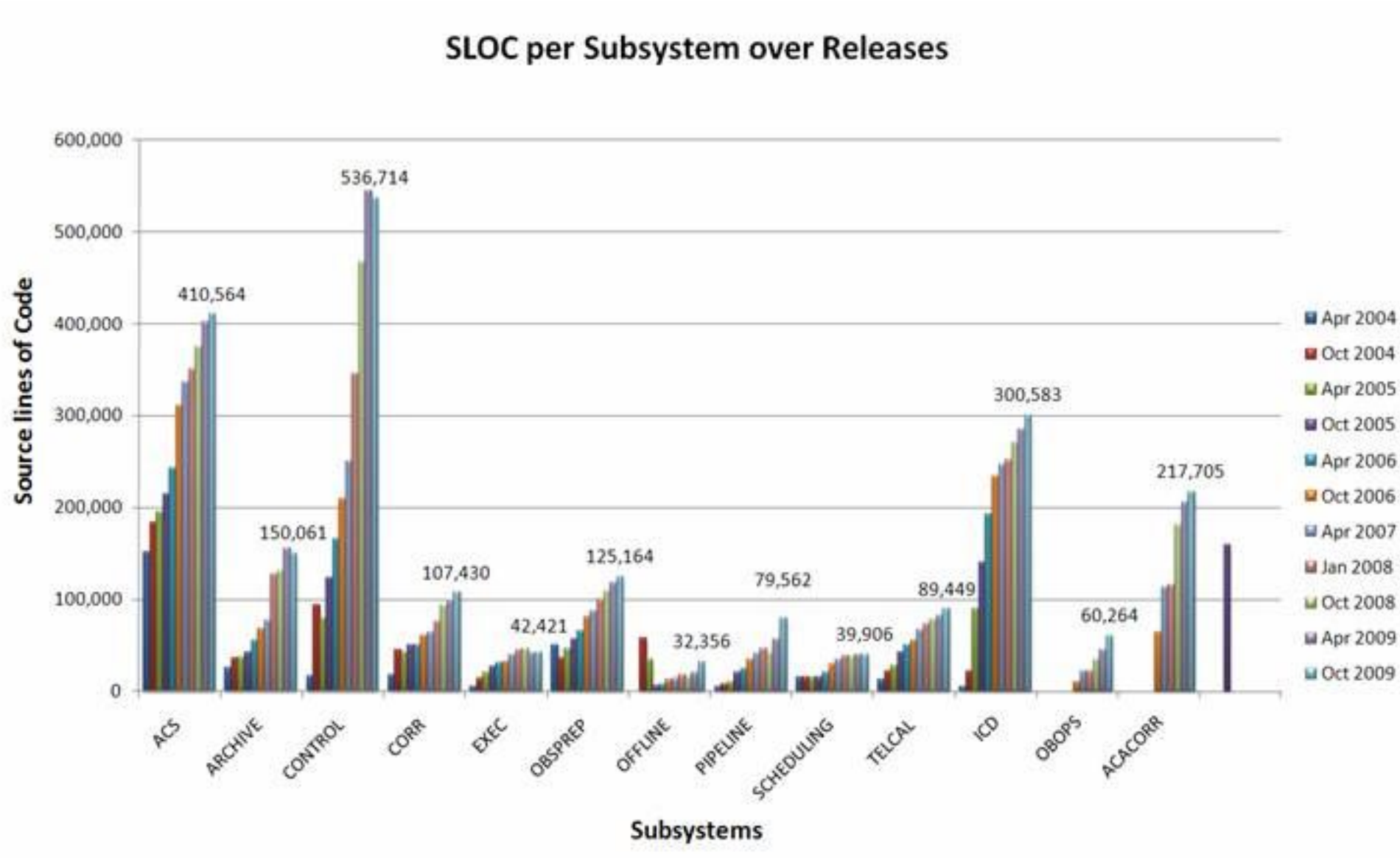
Project Management Approach (cont.)

Incremental Releases at fixed dates (vs. fixed content) twice/year

- ❖ Software is developed incrementally in 6 monthly steps
- ❖ Easier integration, predictable dates for the rest of the project
- ❖ Releases are an integrated e2e system
- ❖ Patches (typically one per 6 month development cycle) allow to upgrade a few computing subsystems
- ❖ Planning work is for 6 months and can be tuned to accommodate project priorities
- ➢ Give priority to testing and making releases over development when deadlines approach

Lines of code by Computing subsystem at Release 7.0 (Oct.2009)

Total ~2M SLOC (comments, doc., adopted/modified packages not included).



Project management Approach (cont.)

Integration tests (by independent team)

- ❖In addition to subsystem tests (build-in test time up front)
- ❖Regression tests, eventually mostly automatic
- ❖Require good test models (several computers)
- ➢… but cannot replace tests with real hardware
- ➢defend towards the rest of the project the need for significant test time on the system, to discover/fix issues before software gets used
- ➢.. You will get anyway criticism later and it will be your problem if you did not follow your procedures

Cross-subsystem Function Based Teams (FBTs) (~3 months)

- ❖Implement important functionality reducing impact of changing interfaces
- ❖Make integration easier, as inter-subsystem issues get sorted out continuously.
- ❖Integrations are more frequent, which is important with a geographically distributed team.

Lessons Learned

- Project management tools: risk analysis, earned value measures
  - ❖ Risk analysis helps project to assess software risks
  - Earned Value (apart from Requirements tracking) was difficult to apply in a meaningful way
- Reviews to monitor progress:
  - ❖ Internal reviews , like Releases, are incremental. This allows incremental design and flexibility towards project priorities
  - ❖ External reviews are good.

    They require preparation and thinking and result in obtaining comments that help in the remaining work.

Lessons Learned (cont.)

- ➢ Planning, Control
  Plan coming year in detail according to priorities
- ➢ Verify (trace) feature completion via integration and user end tests before delivering software for real use.

❖ Problem reporting (JIRA in our case)

- Important to track bugs/improvement request
- JIRA is good. Whatever the system, follow up is important.
- Weekly meeting to discuss issues and flag blocking ones

Status

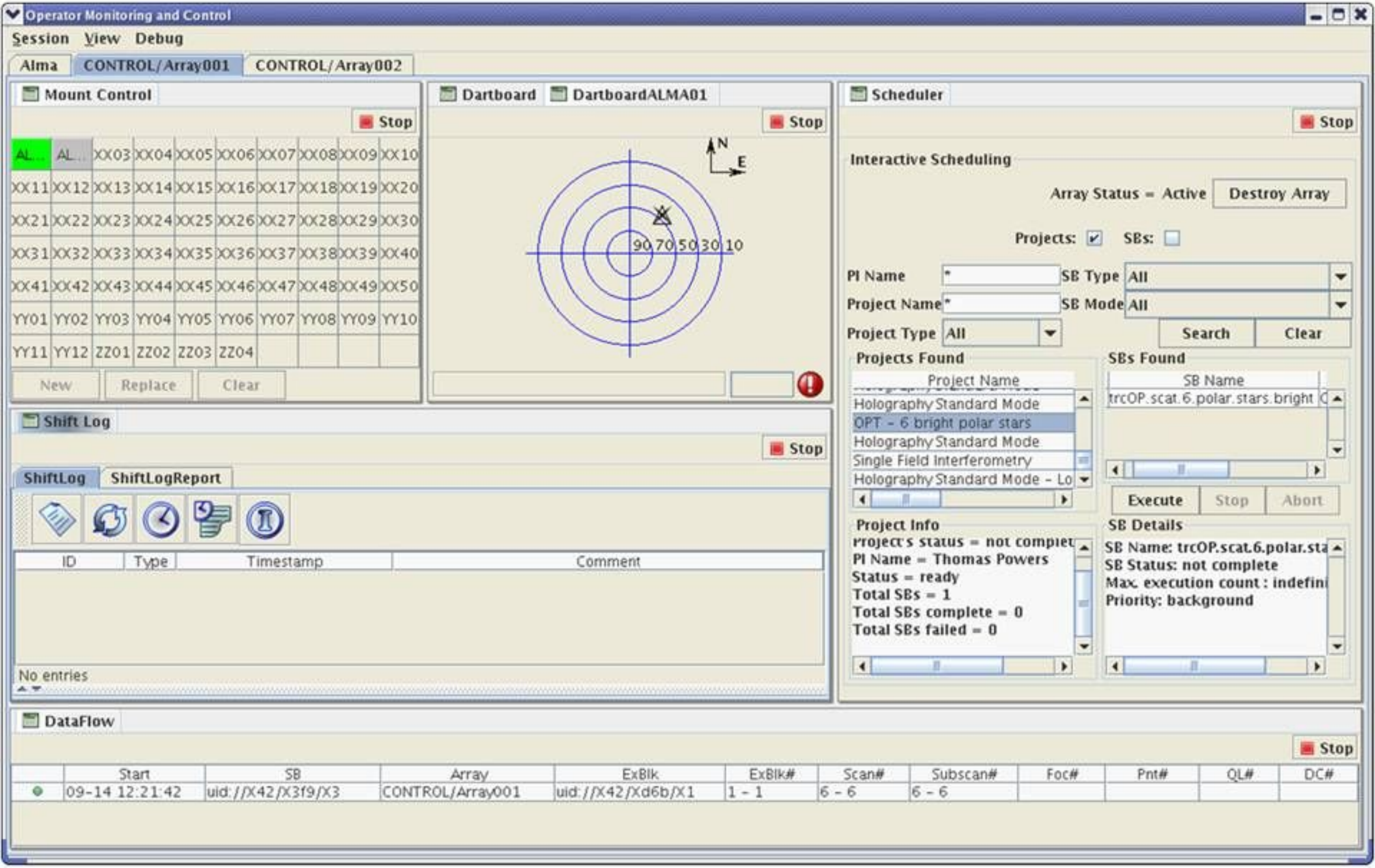
- Passed external PDR (03) and three external reviews- last one Oct.08
- Incremental internal Critical Design Reviews CDR1 (‘04) – CDR6 (‘09).
- Delivered R0-R6 release (+Rx.1 Releases & Patches).

✓ The ALMA software is regular use for commissioning the ALMA observatory

ALMA Observatory - Operation Site Facility (OSF) (2900 m)

Antennas under test

Project management, Lessons learned - adenning



Operator User Interface

ALMA antenna moving up to the high plateau of Chajnantor (5000m). ALMA base camp –OSF in the background.


Otto

More information on ALMA at:
www.almaobservatory.org

There is a video on the Web about the transport of the first ALMA antenna to the site at 5000 m.

Acknowledgments

We are grateful to ALMA Computing staff, who are the real authors of the ALMA software and are located in:

- Europe (Munich, Bonn, Edinburgh, Grenoble, Madrid, Manchester, Paris, Trieste),
- North America (Charlottesville, Socorro, Calgary),
- East Asia (Tokyo, Taipeh),
- ALMA Observatory in Chile (OSF- San Pedro de Atacama).

For more on ALMA and ACS at this Conference:

–TUP101 ALMA Common Software, status and development by G. Chiozzi (ESO) et al.

–WEA006 Data Distribution Service as an alternative to CORBA Notify Service for the ALMA Common Software by G. Chiozzi (ESO) et al.

–TU048 This paper