

## **ALMA Proposal Preparation: Supporting the Novice and the Expert**

Alan Bridger, David Clarke

*UK Astronomy Technology Centre, Royal Observatory, Blackford Hill,  
Edinburgh, EH9 3HJ, United Kingdom*

Joseph Schwarz, Maurizio Chavan, Heiko Sommer, Marcus Schilling

*European Southern Observatory, Karl-Schwarzschild Str-2, Garching bei  
München, D-85748, Germany*

**Abstract.** The Atacama Large Millimetre Array Observatory (ALMA) is an international collaboration between Europe and North America to build a synthesis radio telescope that will operate at millimetre and sub-millimetre wavelengths. Other papers in this conference outline the overall ALMA Software design and cover various aspects of the software system. In this paper we describe the subsystem that will provide the Astronomer's main interface to observing with ALMA: the Proposal and Observation Preparation subsystem. Tools to handle Proposal and Observing preparation for the world's major telescopes are of course now commonplace (HST, Gemini, VLT, UKIRT, JCMT to mention a few), and we will build on the experience of those tools, but the ALMA telescope will present some novel challenges. We will outline the technical side of the subsystem, how it integrates fully with the overall ALMA software system, and also describe our proposed solutions to the challenges of the user-interface side of the system: the major requirement we have to fully support the needs of advanced users, those expert in submillimetre aperture synthesis observing, and novices, who may know very little about the domain.

### **1. The Goals and Challenges**

It is intended that the ALMA Observatory will have an Observing System similar to that used by most of the world's major ground-based Observatories: preparation of proposals and observing will be supported by software, the observatory itself will operate mostly in a queue-scheduled mode, and some level of processing will be performed automatically on the data at the observatory, the final data products arriving in an archive. This is outlined in Schwarz et al (2004).

Despite being inspired by the "cradle to grave" systems of other observatories, the ALMA software system does have some novel and complex challenges:

- We intend to implement a single tool - the *ALMA Observing Tool (OT)* - to support the users' preparation of both proposals and observing, and also to support the review process

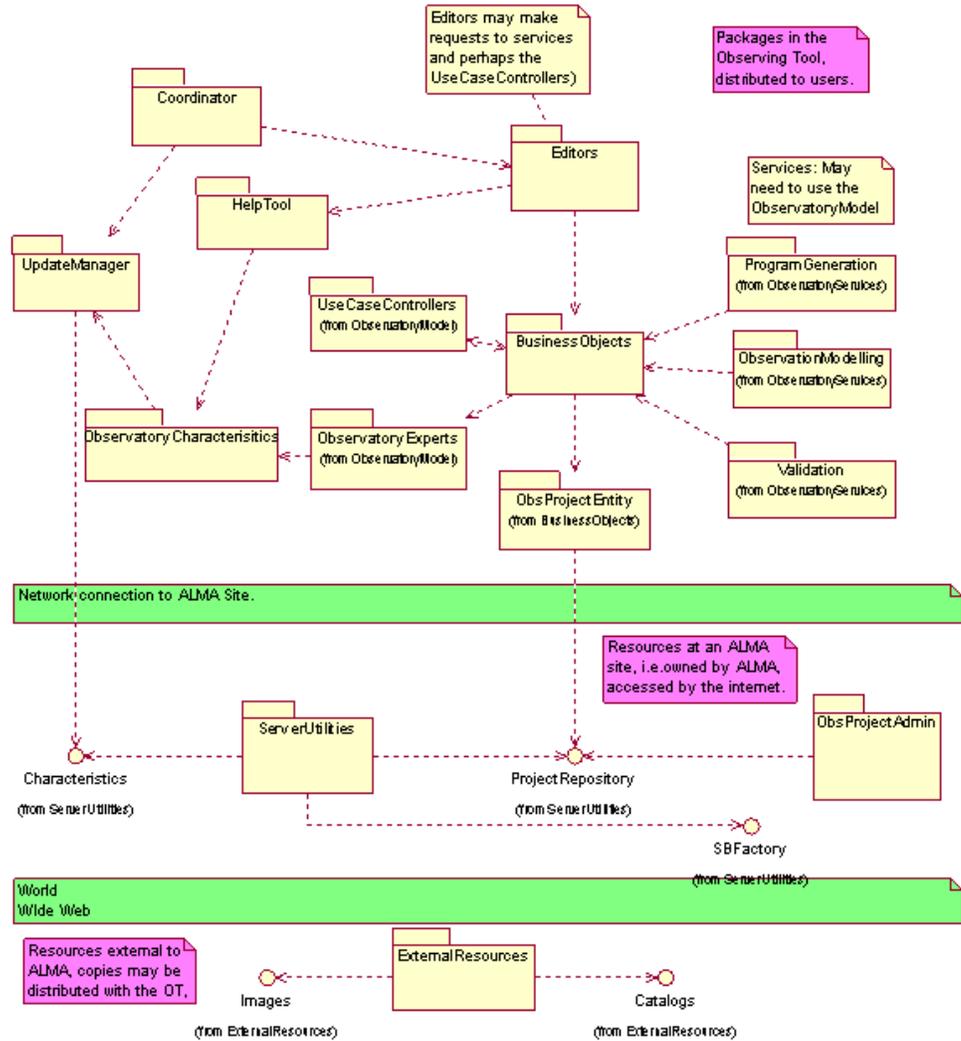


Figure 1. Package Diagram

- We support those who are new to mm/submm interferometry (with no intended offence, *novices*) in addition to *experts*, those who are very familiar with the details of such observing, some of whom will form the ALMA scientific support, and who will use the OT to create new *observing modes*.
- The OT must provide feedback on the resources required, on the optimal weather (and its likelihood), and an estimate of expected data quality.
- All projects submitted from the OT must be technically correct, *i.e.* ALMA can execute them successfully.
- The output from the OT must obviously be compatible with the ALMA Observing System. This system is driven by *Scheduling Blocks (SB)*. The SB is the “atom” of observing with ALMA. Most observing is expected to require the execution of many SBs.

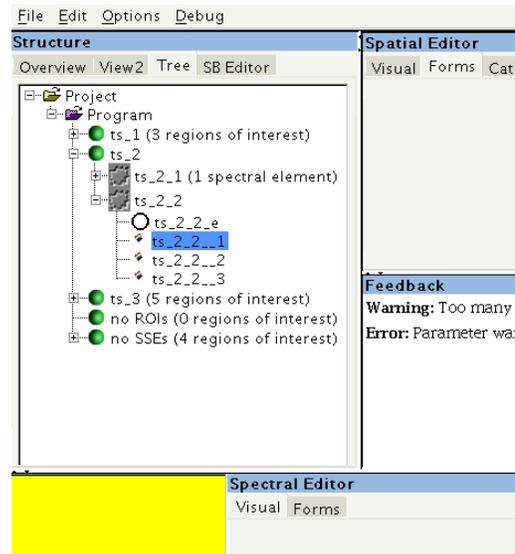


Figure 2. A view of the Observing Project tree

Thus the OT must provide a top level, goal-oriented, astronomical view of ALMA (*e.g.* mapping area, frequency, rms noise), and then transform that information into the data required to drive a complex instrument (a collection of optimised SBs). Finally, it must remain flexible to support the changing operation of the telescope as the knowledge of best use changes, and it must be able to accommodate future changes in computing technologies in response to the demands of the astronomical community.

## 2. The Solutions

Figure 1 shows a package diagram for the ALMA Observing Tool, and the server-side support required by it. Some key design features can be noted:

- Astronomical input is captured within a Program definition (Figure 2):
  1. Multiple *Regions of Interest* (ROI) lie within a *Target Space*
  2. Each ROI consists of a *TargetArea* definition and one or more *SpectralSetups* (receiver and correlator).
- Observatory Experts supply encapsulated domain-specific knowledge.
- A Program Generator service converts astronomical input to SBs lying within a recursive hierarchy that allows for very complex programs.
- Another Service supplies technical Validation of Programs.
- The generated content may be directly edited by *experts*.
- Traditional separation of business data and editors allowing different views, particularly interactive visual editing as an option to forms-based input.
- Use of the ALMA Common Software (ACS) to provide access to server-side (observatory) services, *e.g.* ProjectRepository.
- Persistence of the information via entities bound to the XML Schema.

### 3. The Technologies

The Java programming language was chosen for implementation, partly for portability but also for commonality with other ALMA system.

CORBA IDL is used to define the functional interfaces between observatory subsystems meshing with the ACS use of CORBA, facilitating distribution of server side components. ACS implements the OMG's Component Container Model.

XML Schema provide data interface definitions and binding classes are generated using the Castor tool as part of the ACS framework (Sommer et al, 2004). This also supports the persistence requirements. Ultimately persistence is implemented by the Archive subsystem (Wicenec et al, 2004).

A number of common tools are being used during the design and implementation of the software, some notable ones being Doxygen for class documentation JUnit for unit testing, and the Eclipse IDE for development. We strongly recommend these tools.

### 4. The Status

After some initial conceptual work the subsystem workpackage began in June 2002. The Preliminary design passed review in March 2003, and the first of a series of incremental Critical Design Reviews was passed in June 2003.

An early prototype proved the architecture of the system, and the first internal release of the system (October 2003) demonstrated integration with the rest of the ALMA software, by creating and storing simple SBs that could be queried and used by the Scheduling subsystem.

Annual incremental CDRs, major and minor releases for system integration will lead to a "beta" release of the OT scheduled for 2006, when it is proposed that a significant number of testers will have access. The system will be ready for use in mid-2007 in time for interim observing with the telescope.

**Acknowledgments.** We are very grateful for Steve Scott's (OVRO) contributions to the concept and early design, and to Leonardo Testi (Arcetri) for his continuing interest and advice.

### References

- Schwarz, J. et al 2004, this volume, 643
- Sommer, H. et al 2004, this volume, 81
- Wicenec, A. et al 2004, this volume, 93