ALMA Memo 367

ALMA Operational Model

The SSR Committee view

ALMA Science Software Requirements Committee

2001-05-04

Abstract

In this document we summarize the assumptions we have made on the ALMA science operational model, when writing the Science Software Requirements for ALMA.

Introduction

When writing the Science Software Requirements for ALMA, we had to make assumptions on the way ALMA would be operated as a scientific instrument. We have felt the need to make those assumptions available to the wider ALMA Scientific community, since our report (Science Requirements and Use Cases) does not include those assumptions in an explicit form. We try to explain those assumptions in the present document.

Proposal Selection and Ranking

We have assumed that proposals can be assigned a science rating in a uniform way. This is a requirement of Dynamic Scheduling: at a given time, only proposals requiring certain observing conditions can be scheduled, and a choice between them has to be made unambiguously, based on science rating, but also on other parameters linked to the actual observing conditions (like system temperature, phase fluctuations, available antennas, available u-v coverage, receiver capabilities, and other variables). The relative weight of all the scheduling parameters must be the same for all proposals, so this strongly calls for having all proposals treated in the same way (it would be very tricky to reconcile ratings made by local allocation committees if there were several of them). For each programme, the ALMA partner of origin can be included in the scheduling parameters, so that the scheduler may maintain for each observing season the fractions of observing time to the values agreed between ALMA partners.

Sub-arrays

We have considered sub-arrays mainly as a means to execute necessary calibrations on some antennas while the main body of ALMA executes a dynamically scheduled science programme. In the present ALMA project we have not seen a general need to have a programme split itself into two, each using part of the antennas, and doing different things. Scheduling projects to share the same time by using sub-arrays rather than scheduling them sequentially has no advantages and can have significant disadvantages, since the image quality is largely driven by the number of baselines.

The outcome is that sub-arrays will be used at the operator level to manually take a few antennas out of the main scheduling stream, to perform calibrations as needed after maintenance or antenna displacements. We nevertheless need to consider a few exceptions:

1. Simultaneous observing of a time variable target at two or more frequencies
2. Simultaneous observation of several Solar active regions.
3. Use of a frequency band which is available only in a subset of the antennas. In this case we propose to make the remaining antennas available to filler programmes, (manually or automatically scheduled)

Items 1 and 2 can be handled by scheduling in a straight-forward way due to the simultaneity of the observations in each sub-array.

It has been proposed to operate ACA as a separate array, including the smaller antennas together with a few 12m antennas, to perform single-dish and short spacing measurements on a full-time basis. This raises a few issues: that set of antennas would be used as a single interferometric array on phase and pointing calibrators, but would split on the astronomical target sources, the large antennas doing single dish work while the smaller antennas do short-spacing interferometry. This involves a change in sub-array allocation during an observing session, which we have felt was not needed otherwise.

Non-standard Observing Modes and Scripts

At the end of the whole proposal preparation phase (Phase II), the Observing Tool will output one or several control command language scripts to be executed by the observing system and the pipeline. We have foreseen that expert observers will be able to examine these scripts, and customize them to their needs.

There are several reasons for allowing non-standard observing scripts:

1. We want to keep a high level of flexibility in the observing, in particular at the high frequencies where ALMA will be a pioneer instrument.
2. Standard observing modes must be kept simple at the beginning, and progressively integrate successfully tested improvements.
3. There may be specific projects with the need of a one-time variation of standard modes; for these the full scale development of a new version of these standard modes would be a waste of time and manpower.

The implication is that there is a need to automatically validate those scripts, which can be done in two steps:

1. syntactically, by an off-line version of the observing system, that only checks the commands and their parameters. This could do some further analysis like collating statistics like time on-source, cycle times, and other parameters.
2. operationally, by using a more sophisticated simulator that uses a source model as input and produces simulated raw data, that can be processed by the pipeline, thus validating the reduction scripts too.

It is clear that these two functionalities must be made available anyway to the developers of the standard observing modes, and are thus not only needed for this 'observing mode customization' feature.

If the necessary validations are implemented we do not feel that this feature will significantly increase the load on the local staff (astronomer, operator).

Dynamic Scheduling and 'Service Observing'

Dynamic Scheduling will be the default mode of operation for ALMA. In our model, the PI interacts with the Observing Tool to prepare an Observing Programme which is automatically validated; we believe that no more human intervention is needed until the programme is scheduled and executed by the observing system. That's why we had the inclination to call the PI the 'Observer', while the staff astronomer and the operator should be there mainly to react to unexpected hardware or software problems, detected by the observing system or by the calibration or quick look pipelines.

In a parallel way the Observer-PI, warned by an email message at the beginning and at the end of each observing session, would also be the first one to look at the 'final' science data (e.g. a deconvolved map obtained at the end of the observing session), in the 'no problem' case when the calibration and quick look pipelines would have proceeded without errors or warnings.

In this view the term 'service observing' would be better replaced by something like 'supervised observing' or something similar. We believe that the 'no problem' cases should be a large majority, at least for the low frequencies and after the first years of interim science operations. Having the staff astronomers look systematically at all the 'no problem' cases would be a waste of time and manpower. They should rather concentrate on the problematic cases.

Knowing that the observation will by default be done without a systematic human screening should have the effect of making sure the PI feels more responsible for his/her data. The issue of encouraging new astronomers to more involved in observing and data processing should be the task of the regional data centers, where the PI's would find assistance in the programme preparation and in the off-line data reduction.

Populating RA and Observing Condition Space

We have not addressed this question in the software requirements so far, mainly because we felt this field was highly dependent on the array configuration policy. It now looks that the change in array configurations will be kept quite flexible, leaving to the scientific staff the duty of tuning it to accommodate the actual spectrum of resolutions required by the proposers. The rest of the task should be left to the dynamic scheduler, which might need some tuning up at the beginning of each scheduling period; provided that the science reviewers have been wise enough to rate a sufficient number of proposals in each range of observing conditions (total rejections should be only for a really small fraction of programs).

We do not think that we can bring out arguments for or against continuous reconfiguration on the basis of software alone. The software model that we have outlined so far can accommodate both reconfiguration schemes.

Management of Observing Proposals and Programmes

A few questions are asked here:

1. Will the reviewers be allowed to assign different priorities to parts of observing proposals? This point is mainly a policy issue but has to be settled since it may imply additional complexity in the programme preparation software.
2. Is there a strong need to check the programmes for consistency with the allocations made by the reviewers? We think that an automated check for observed sky positions and frequencies is feasible; going much further would mean a manual check, which could be done at random of a few projects.
3. How late can the programmes be changed by the PI's? We had considered that from the software itself there is no real need to impose a time limit, since the scheduling is actually done nearly in real time. Our model strongly calls for allowing at least such a change after a break point is reached. It's clear that the change then must be limited to a policy predefined in the observing proposal, and stay within the allocation made by the reviewers. It seems that the need for a certain amount of predictability in the mid-term scheduling, to allow the determination of the array configuration policy, would impose, from the operational side, to restrict any significant changes in the observing programmes to very specific cases (like correcting a small coordinate or frequency error).

Responsibility for Data Quality

We are here assuming the ALMA has to provide a control of quality of the data-taking activity, by running a calibration pipeline and a quick look pipeline that aim to warn the staff (Operator, Astronomer) of problems that would affect the data quality. We believe that ALMA should not be responsible for PI errors (like wrong coordinates).

Responsibility for Achieving Scientific Goals

We believe that the overall efficiency and productivity of ALMA can be improved by having in most cases the PI be allocated a sensitivity level rather than a fixed amount of integration time. However sophisticated the dynamic scheduler cannot be almighty, and there is a good chance that programmes requiring unpopular LSTs will end up getting better observing conditions than actually required, and therefore will be finished in a shorter integration time. The saved time can be used to observe other projects. Obviously there will be a time ceiling for each programme to avoid overflow if the sensitivity was poorly estimated. In some cases a fixed time can be allocated (e.g. for exploratory projects for which it is meaningful to say 'let's do the best we can with ALMA in 1 hour integration').

Archive Uniformity

There appears to be some tension between the requirements of uniformity of the science data archive and of flexibility of the instrument to the observer. It is obvious that the archive cannot be regarded as a fully uniform data set, as several standard observing modes will be available, and as they are to be expected to evolve on a time scale of years. The consequence is that the uniformity of the archive will be relative, and that the archive to be useful for data mining will have to include for each data set an evaluation of data quality (noise level, dynamic range, and other metrics).