Building a world-wide open source community around a software framework. Progress, dos, and don'ts.

Jorge Ibsen[a,b], Jonathan Antognini[b], Jorge Avarias[b], Alessandro Caproni[c], Matthias Fuessling[d], Guillermo Gimenez[e], Khushbu Verma[d], Matias Mora, Joseph Schwarz[f], Tomás Staig[b]
[a]European Southern Observatory, Santiago, Chile; [b]Joint ALMA Observatory, Santiago, Chile; [c]European Southern Observatory, Garching bei Munich, Germany; [d]Deutsches Elektronen-Synchrotron, Germany; [e]Centro de Rádio Astronomia e Astrofísica Mackenzie, Sao Paulo, Brazil; [f]Istituto Nazionale di Astrofisica, Italy

ABSTRACT

As we all know too well, building up a collaborative community around a software infrastructure is not easy. Besides recruiting enthusiasts to work as part of it, mostly for free, to succeed you also need to overcome a number of technical, sociological, and, to our surprise, some political hurdles.

The ALMA Common Software (ACS) was developed at ESO and partner institutions over the course of more than 10 years. While it was mainly intended for the ALMA Observatory, it was early on thought as a generic distributed control framework. ACS has been periodically released to the public through an LGPL license, which encouraged around a dozen non-ALMA institutions to make use of ACS for both industrial and educational applications. In recent years, the Cherenkov Telescope Array and the LLAMA Observatory have also decided to adopt the framework for their own control systems.

The aim of the "ACS Community" is to support independent initiatives in making use of the ACS framework and to further contribute to its development. The Community provides access to a growing network of volunteers eager to develop ACS in areas that are not necessarily in ALMA's interests, and/or were not within the original system scope. Current examples are: support for additional OS platforms, extension of supported hardware interfaces, a public code repository and a build farm. The ACS Community makes use of existing collaborations with Chilean and Brazilian universities, reaching out to promising engineers in the making. At the same time, projects actively using ACS have committed valuable resources to assist the Community's work. Well established training programs like the ACS Workshops are also being continued through the Community's work.

This paper aims to give a detailed account of the ongoing (second) journey towards establishing a world-wide open source collaboration around ACS. The ACS Community is growing into a horizontal partnership across a decentralized and diversified group of actors, and we are excited about its technical and human potential.

Keywords: ALMA. ALMA Common Software, Open Source Development, Worldwide Collaborations

1. INTRODUCTION

The ALMA Common Software (ACS) is a comprehensive framework, on top of the operating system, offering a complete environment, data structures, design patterns and libraries originally designed to be used as the common layer for the end to end ALMA Software suite. The first ACS prototype dates from late 2000 when it was used to demonstrate its core concepts by developing a mount control for the Kitt Peak 12m Telescope [1].

Since the beginning ACS was designed to be a general system (i.e. not specific to ALMA), based on a CORBA middleware and incorporating seamless integration with three programming languages: C++, Java and Python. The use of a common software layer was introduced in ALMA as the way to enforce the use of common constructs in a highly geographically distributed development team. ACS provides a well-tested platform embedding standard design patterns and standardizing the underlying architecture to make software maintenance affordable [2].

Although the main target users of ACS were developers of ALMA applications, the general architecture made possible to use it for other projects since its beginning, like for instance ANKA by 2002 and APEX control software (APECS) by

2003 [3]. More than a dozen ACS training workshop were organized since 2004 [4], convening software developers within and outside the ALMA project. By 2006, ACS was at the peak of the construction phase, and had a medium size user community outside ALMA (Figure 1). As the activities in the ALMA Test Interferometer and later in the ALMA site in Chile started to ramp up, most of the attention of the ACS development team went to them and the support to the ACS user community was, understandably, reduced.

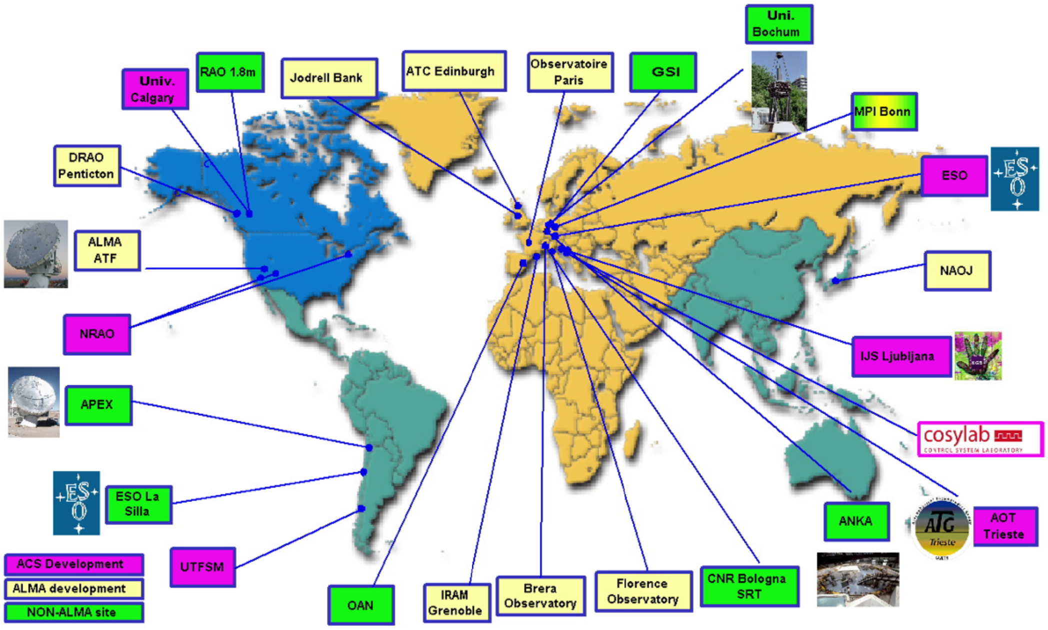

Figure 1. ALMA Common Software installations by 2006

In June 2014, an impromptu meeting was held at the end of SPIE to discuss revamping a collaborative effort to extend ACS by and for non-ALMA resources. The rest of the sections on this paper describes what has been achieved and learned during this exercise.

2. WHY DEVELOPING AN OPEN SOURCE COMMUNITY?

In short, why not? It is clear that a number of features of a common software infrastructure developed for more than sixteen years to support the largest radio telescope in the world could also be used to support software development in other projects. It is close to be a code reuse 101 principle, isn't it? In fact, the idea of sharing ACS with others was there since the beginning, as it was explained in the previous section, and for that reason the project adopted a LGPL license [5] for it at the early stage of the project. This allowed the source code to be easily shared welcoming others to contribute to it. During most of the construction years, the coordination was actively led by the ACS team at ESO headquarters and several contributions were made to it during that time.

But code reuse comes with a price: the learning curve. The effort involved may range from simple to extremely difficult depending on several factors such as developer/programmer experience, easy access to the documentation, possibility to exchange information with domain experts, and your own availability of resources, namely talented human beings as well as financing for both their activities (i.e. software development) and the required infrastructure to support it (mostly machines and work space). As ALMA started the system integration and commissioning phases in 2008, the resources available for this (secondary) activities became scarcer leaving little room to be "distracted".

The key motivation to develop an open source community around the ALMA Common Software (the ACS Community from now onwards), is a mixture of good intentions to support other projects, the idea of reaching for resources outside ALMA to explore some new ideas and/or further develop some aspects of the software infrastructure that are not currently a priority in ALMA, and the curiosity to learn through a practical and concrete example how this should (or should not) be done. The main difference with respect to the (many) ACS collaborations led by ESO in the past is simply allowing a group of independent advanced ACS users and volunteers to organize themselves to take over further development of ACS related features by them for them.

Concrete and simple initial scope, resources, and schedule were set up early to correctly manage expectations (Figure 2).

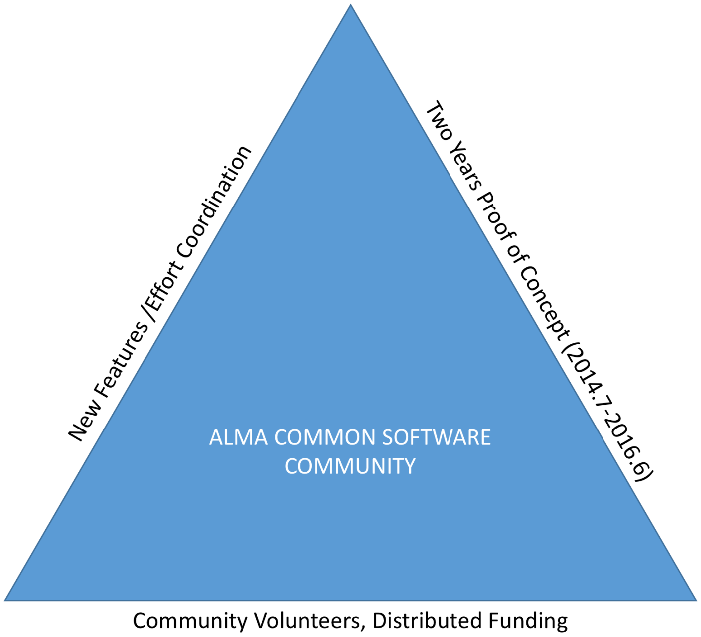
New Features /Effort Coordination
Two Years Proof of Concept (2014.7-2016.6)
ALMA COMMON SOFTWARE COMMUNITY
Community Volunteers, Distributed Funding

Figure 2. ACS Community Project

2.1 Scope

In simple terms, the scope of the ACS Community is to develop new features for projects that have chosen to use ACS as the common software layer for their own purpose, and it serves as a coordinating body to ensure that tasks are done avoiding duplication of effort. New features can be proposed by anyone participating in the community, but priorities are assigned by a governing ACS Community Board, which is composed by different representatives of participating projects.

2.2 Resources

Human resources are essentially volunteers from different institutions wanting to participate for a variety of reasons. This includes people from the projects that decided to participate (essentially LLAMA and CTA at the time of this writing) in the community, ALMA software engineers that decided they love their work so much that they wanted to contribute to it during their free time, and a number of students coming from different (mostly Chilean) universities looking for an opportunity to learn and/or to contribute for the better good. Their activities are funded mostly by their own institutions.

2.3 Schedule

The short term schedule for the ACS community is to proof concept it and review the progress after an initial period of two years, starting the clock by June 2014.

3. WHERE DO WE STAND TODAY?

The ACS Community was formed immediately after the initial meeting took place. The initial software repository contains all ACS modules released with a LGPL license and it is hosted in github [6]. This early decision was taken mostly because github is a well-established platform for open source development, there was already an initial site created for that purpose from a previous attempt, and the support tools provided by the platform (issue tracking, wiki, analytics). This official community repository contains a working version of ACS with tested and peer-reviewed changes and the intention is to update it regularly.

Following the standard open-source software philosophy each participating member is welcome to get a copy the repository and make changes as needed to fulfill the requirements of his/her individual application. They are encouraged to keep in mind that general purpose patches should be contributed back to the community repository, through pull requests in github jargon.

The initial list of tasks (meaning bugs, enhancements, feature requests, etc) was based on a list that was historically maintained by the ACS development team, with the addition of specific items within the interest of the participating projects, and monitored through the github built-in issue tracker. Relevant documentation is aimed to be maintained within the built-in github wiki for starters.

Finally, a simple structure was developed for decision making and organizational purposes, consisting in an ACS Community Board (ACS), and a Community Control Board (CCB).

3.1 ACS Community Board

The role of the ACB is to develop the initial ACS Community policies and procedures and to ensure it grows as an all-inclusive, fruitful and useful collaboration for its members. The ACB membership is planned to be active for an initial period of two years, after which an election process to replace/confirm the original Board members will be in place.

Among the initial tasks of the ACB:

- Review and update the community roadmap draft.
- Define community infrastructure and development priorities.
- Define ACB and CCB election process.

ACB members are assigned a corresponding administrator role in github.

3.2 Community Control Board

The role of the CCB is to be in charge of receiving, reviewing and approving new patches/contributions from the community to the code base, to ensure that the integrity of the framework is preserved.

Among other things the initial tasks of the CCB are:

- Define code formatting standard and minimal testing rules.
- Define Community release structure.
- Define pull request procedure.

CCB members are also assigned a corresponding administrator role in github.

3.3 Volunteers recruitment

Obviously somebody has to do the work and recruiting volunteers essentially willing to work for free, is where most of the effort has been focusing. At the moment there are about 20 members in the github community, including ALMA staff wanting to contribute during off hours (they really love their work, don’t they?), a few numbers of persons from other projects, and engineering students (mostly from Chilean universities). This is still a very small community, and just a few of the members are really active in terms of contributing code.

Recruitment is still on-going and a few additions have resulted after the three ACS workshops that have been organized in the last couple of years (either by ALMA or the community itself).

3.4 Main achievements

To a large extend the focal point for the discussions that were carried out during the first year, where about what needed to develop and when needed to be delivered. It was agreed that the initial priorities were essentially two: the key definitions for the supporting infrastructure and a couple of development tasks in direct interest of the participating projects.

There are currently eleven labels defined in the issue tracker, but in only six are actively used (community visibility, documentation, enhancement, help wanted, language consistency, usability). A total of 36 issues have been created, from which 25 are still opened.

Additionally, two important milestones were defined in conjunction with the community stakeholders:

- Milestone 1, encapsulating several the task related to the initial organization in the repository, where five out of six tickets are already completed.
- Milestone 2: encapsulating two development tasks identified as priorities for CTA control software prototyping that are about to be completed.

Most of the eleven issues closed are related to definitions, standardization and developing the required IT infrastructure to support development activities. The tasks of the second milestone are close to be completed at the time of this writing.

Revamping the community also allowed to reactivate the organization of ACS workshops and update the workshop material. Three of them where organized between 2014 and 2016 by ALMA but with a clear intention to train and trigger discussions with community users.

4. THE DOS AND DON'TS

4.1 Clients are a must. A mixture of success and failure.

To explain this point is necessary to provide a short historical background.

By 2006, a group of students at the Technical University Federico Santa María in Chile that had started an informal collaboration with the ACS development team in 2004 received a Chilean governmental grant to fund their activities for an initial period of three years, which was subsequently extended for two more after that. This funding allowed a significant number of students to be exposed to yearly ACS workshops supported by the ACS development team and organized in Chile between 2007 and 2010, as well as directly contribute to ACS development [7].

The win-win scenario was based on very simple premises:

- Students and faculty members were thrilled to get involved in a large software development carried out by international organizations
- The ACS development team, and later ALMA operations in Chile, wanted to supplement available effort to do some additional tasks and on a longer term to ensure that there would be trained engineers at the time of the operations hiring ramp up in Chile.

During the period of five years between 2006 and 2010, ALMA was the main beneficiary receiving both deliverables and professionals from the existing collaboration. A group of around 20+ people at the university, between faculty members and students, was participating on it. Many developers coming from this collaboration were later hired by ALMA between 2009 and 2012.

By 2012, where most of the recruitment for ALMA computing was about to be completed, a first attempt to organize an open source community around ACS was organized and presented as a project. Under the assumption that ACS would also be used for the E-ELT early prototyping development, students created the github space to coordinate porting efforts to other platforms including windows and different linux flavors, applying for funding from different agencies. The outcome was not successful as there was a decision not to use ACS for this purpose that weakened the funding request. As a direct consequence, all the activities fade away in time due to lack of clients and funding.

The second attempt to develop an open source community reported here is therefore directly motivated by LLAMA and CTA independent decisions to adopt ACS for their control software development and it is reflected in the current composition of both the ACB and CCB.

4.2 LGPL, open source and the proprietary software mind set.

ACS was and still is develop within the ALMA partnership for the purposes of ALMA software development and deployment, therefore formally speaking the intellectual work belongs to ALMA partners, and more specifically to ESO, the executive responsible for this deliverable.

As the source code is distributed with a LGPL license, it is possible for a third party to copy and distribute verbatim copies of the source code as long as the copyright notice, disclaimer of warranty and license are kept intact and distributed with it. A similar condition applies to the modifications such third party can introduce to the source code.

Sounds clear, isn't it? To our surprise, clarity was only apparent and many discussions were carried out in relation to property, potential support requests, and warranties. Clarifying these aspects took a while.

The positive outcome was a common understanding that the intent all along was to exercise the right to control the distribution of derivative or collective works based on the distributed source code and not to claim or contest rights of the rightfully owners of the intellectual work as the LGPL license explicitly explains. In short, it takes time to understand the whole picture, so be patient.

4.3 The human resource

For the projects participating in the community, the challenge is to balance their specific activities with the ones needed by the community. The recipe is, not surprisingly, discovering commonalities between them and lots and lots of coordination to ensure the collaboration benefits them all. Fortunately for us, it has been possible to make progress in this direction and there are a few concrete successful examples.

Recruitment of additional developers is still an on-going effort and has been attempted mostly in Chile given the previous successful experience carried out by ALMA [7]. In this case the history is a little bit more elaborated.

Once upon a time, motivating talented students in Chile to work in control software and/or other applications for observatories was a simple task. You just needed to mention that this was going to be a direct contribution to any specific project installed in the region and voilà, around a dozen top notch software engineers to be were knocking at your door. In fact, during the 2006-2012 period where ALMA was directly involved in developing the collaboration, internal selection processes at the university were carried out to select an elite of students to participate on it. The motto of the group working with ALMA at that time was "Just for fun!".

When we started this revamp, it was thought that the situation was going to be similar and it would be pretty much straightforward to recruit hands for it. After all it has been done before, right? … Wrong!

In the meantime, a number of "competitors" appeared including aggressive recruitment from big international companies such as Google or Amazon as well as startup incubators at the universities promising to become the next Mark Zuckerberg. This by no means implies there isn't people interested in working on this, just that you need to look harder and come up with a better selling speech. In summary, the "Just for fun!" motto became "Just for (add here your own preferred noun)".

Fortunately for us, at the same time this happened, there was an increasing interest from Chilean governmental agencies in supporting the development of engineering, particularly the ones relevant to observatory operations. This interest created several forums suitable for recruiting volunteers.

4.4 Funding

At present there is no single funding source for the development of community activities and each party contributes with effort and supporting infrastructure independently from their own respective budgets. This obviously adds complexity as work for the community competes for priority with work within each individual project. Although some of it can be mitigated through frequent coordination among the projects, developing a joint budget to finance the effort, or at least agreements to secure sustained development effort, will have to be a priority moving forward to succeed.

5. CONCLUSIONS

Initiating an open source collaboration is really hard. It requires a significant number of consecutive hours to get it organized, setting up the group rules, and aligning the different project and individual interests into a common goal. The next step is commitment or in simpler works the agreement to set aside infrastructure and a number of man-hours to actively contribute. This also implies looking for funding as well as recruiting long term collaborators. Finally, you need to set up both expectations and quantifiable results to account for progress. As the whole endeavors is basically on a voluntary basis, the schedule sets up opportunity time windows for success of failure as they are open for a limited period of time.

Regarding this particular collaboration, it is clear that people first discussing it where incredible optimistic in thinking that two years was enough to get everything going. Reality is that it takes much longer for a number of different reasons including among other things the time it takes to precisely agree in common goals, the maturity of each project to be able to do long term commitments, and finally personnel movements as people in software tends to move quite often (although it has to be noted that in this domain the turnover does not seems to be as dramatic as in the industry).

Finally, will it continue or is it over? This is to be discussed in June 2016. Resource wide, the horizon for the following couple of years looks promising as the interest, at least in Chile, to support this kind of activities seems to be increasing given the abundance of telescope and astronomical instruments already installed or to be installed in the country and the governmental interest to use this type of activities as a driver for local innovation [8]. In terms of the collaboration itself, some initial gains have already occurred, so at least there is an indication that if sufficient critical mass is developed, this community will strengthen the coordination between projects and further ACS development for their own purposes.

ACKNOWLEDGMENTS

The authors of this paper would like to acknowledge the hard work of past, present and future members of the ALMA Common Software team and thank the Integrated Computing Team (formerly known in construction as the Computing Integrated Product Team) management for licensing the source code as LGPL. Without their overall interest in fostering collaborations with others, this community would not had started. We would also like to thank all the people who have spent a non-negligible amount of time and effort conducting this experiment and taking it to its present state. We hope this has been as fun for them as it has been for us.

REFERENCES

[1] G. Raffi, G. Chiozzi, B. E. Glendenning, "The ALMA Common Software (ACS) as a basis for a distributed software development", Proc ADASS XI, Victoria, BC, Canada, September 2001.

[2] G. Chiozzi, H. Sommer, J. Schwarz, "ALMA Common Software Architecture", April 2009, document available at http://www.eso.org/~almamgr/AlmaAcs/OnlineDocs/ACSArchitecture.pdf

[3] Muders, D., Hafok, H., Wyrowski, F., et al., "APECS - the Atacama pathfinder experiment control system," A&A 454, L25–L28 (2006).

[4] Links to past ACS workshops material is available at https://github.com/ACS-Community/ACS-Workshop/wiki/Workshop-Editions.

[5] https://opensource.org/licenses/lgpl-2.1.php

[6] https://github.com/ACS-Community/ACS/wiki

[7] M. Mora, J. Ibsen, G. Chiozzi, N. Troncoso, R. Tobar, M. Araya, J. Avarias, A. Hoffstadt, "Integrating a university team in the ALMA software development process: a successful model for distributed collaborations"Proc SPIE 2010, 7740-130

[8] Addere Consultores, technical study (in Spanish) "Capacidades y oportunidades para la industria y academia en las actividades relacionadas o derivadas de la astronomía y los grandes observatories astronómicos en Chile" developed for the Innovation Division of the Chilean Ministry of Economy (2012) available at http://www.economia.gob.cl/wp-content/uploads/2012/06/OPORTUNIDADES-ASTRONOMIA-EN-CHILE-INFORME-FINAL.pdf