ScienceSoft: Open Software for Open Science

An Open Community of People, Software and Services for Scientific Research

v. 0.5

Authors

Alberto Di Meglio (Project Director)
Florida Estrella (Deputy Project Director)

Geneva, 6 February 2012

Table of Contents

Overview

On September 9th, 2011 EMI started a discussion on a general proposal to investigate, design and establish an open source software initiative as part of the EMI long-term sustainability plans.

The proposal is inspired not only by the general objectives already defined in the EMI Description of Work, but also by the recommendations received by the EC-appointed reviewers after the first periodic review and the general guidelines discussed with the EC by EMI and EGI management in the context of long-term sustainability of the European research infrastructures. In addition it takes into account feedback and requirements[1] collected from software developers and infrastructure users about visibility and accessibility of software.

The main objective of this proposal was initially to create the conditions for the continuing development, support and use of the EMI software products after the end of the EMI project by establishing a broad open source community of developers and users of not only the EMI products.

However, it became rapidly apparent that similar concerns were shared by many other projects developing or using software of critical interest for the European scientific research communities. Therefore EMI decided to extend the discussion to all interested parties and try and setup a truly open community of software developers, administrators and users in the context of global scientific research.

The present document summarises the initial discussions EMI has had around the subject. The document is not a proposal to be accepted or rejected. It is just input to a more general discussion that should involve more people and more use cases.

[1] Mostly private discussions with other projects or initiatives, WLCG, StratusLab, iMarine, EDGI, EDOS and others, but see also for example
- GridPP Workshop, CERN, 15 September 2011 (http://www.gridpp.ac.uk/gridpp27/)
- Steve Conway (IDC Research Vice-President, HPC), Financing a Software Infrastructure for Highly Parallelized Codes, 9th eConcertation Meeting Lyon, September 22-23, 2011

The Open Source Initiative

The main objective of this document is to determine the stakeholders and the potential users as well as assess the value and benefits of defining and establishing an open source initiative to establish and coordinate a community of software developers and users working as part of European and international scientific research initiatives. The initiative must address clear needs, provide the necessary motivational reasons for people to join, contribute and provide solutions not available today.

Value Proposition

Problem

In order to understand the value of the proposition, it is necessary to identify what problems it is trying to address.

Most of the software developed today by research institutes, university, research projects, etc. is typically stored in local source and binary repositories and available for the duration of the project lifetime only. Subsequently, many cases have been found where important components are distributed from personal web sites of the developers. Finding a piece of software based on given functional characteristics is almost impossible. Binaries to be run on the most used operating systems are available from many different places ranging from local university repositories to mainstream community repositories like EPEL. Cases of conflicts are often found between different versions distributed by different people from different places. Source code is even more difficult to locate and access and contributing with patches and fixes, which is a very common activity in the open source world, is traditionally very difficult to do in the research communities. This has been for years a primary complaint from users.

The problem is not limited to the middleware services developed by EMI. Similar requirements have been expressed by application developers, infrastructure managers and users. Within the HPC community the organization of common repositories of application code is a known concern. Although such code is usually highly dependent on the hardware on which they are run, the lack of code sharing and availability is considered one of the reasons why the European HPC efforts are falling behind similar activities in the US and Asia (see the recent IDC report presented at the EGI Technical Forum in Lyon in September 2011[2]).

Another sensitive problem is the general lack of metadata or information about the software. Information about who develops, contributes and uses a given program is very difficult to find out and yet the widespread availability of such information would give more visibility and credibility to the software products. In addition, the EC invests considerable amounts of money into funding projects that directly or indirectly need to develop software. A single repository of information about software products would allow projects to avoid re-developing existing solutions and would provide valuable

[2] Steve Conway (IDC Research Vice-President, HPC), Financing a Software Infrastructure for Highly Parallelized Codes, 9th eConcertation Meeting Lyon, September 22-23, 2011

statistics about software usage. Such information could be used also by the EC to monitor the outcome and impact of funded projects, the extension of adoption of open source software and the compliance with OSI licenses and possible as input to future EC calls objectives and framework programmes?

The creation of links or relationships not only among pieces of software, but equally among the people interacting with the software, would foster a more active community and create the conditions for sharing ideas and skills and a more rapid improvements of the software quality. The use of modern social networking techniques would greatly help the establishment of an active open source community and focused sub-communities around specific interests. This would foster the creation of focused groups around areas where the availability of skilled experts is less widespread. For example, sub-communities could be established for people interested in testing or writing and updating documentation, communities of packagers, or experts of specific standards and so on. These principles are in fact a particular specialization of existing environments, like LinkedIn, Facebook or Tweetter for example, but with more focus on specific sectors of scientific research. Indeed, tools from such environments could be harnessed to simplify and speed up the construction of the community tools.

The establishment of a software rating system based on technical criteria (Is a given platform supported? Is a certain package format available?) and subjective criteria (What do existing users think of it? Is documentation up-to-date and well written?) would allow filtering mature products from less mature products and would increase the developers' motivation to improve certain aspects of their software like documentation that traditionally receive less effort that the actual code writing.

Information about what licenses are used and related compatibility charts would address a number of usage and legal issues that have been so far only marginally discussed and would allow potential users also in the commercial sector to find out more efficiently which products are worth considering for further exploitation.

Solution

A possible solution to the problems described above is the creation of a top-level information hub where metadata about software and people can be easily registered and looked up. The implementation of this solution can be split into two activities:

1) **Technical implementation**: the possible format of the hub is a web-based portal where users can register products, look up products, link products among each other, register and connect to people, find documentation, usage tips, success stories, etc. The main assumption is that existing code and binaries cannot be moved from wherever they are stored at the moment. Therefore the need arises for a common broker of information about software for easy look up and sharing. Registration should be subject to a number of basic entry criteria to enforce a minimal level of quality and commitment. Registering or managing projects should be very intuitive. A source code or binary repository can be easily provided to host projects without such facilities by using any of the existing local or even better community repositories like github[3].

[3] https://github.com/

Simple ways of performing standard checks like verifying compatibility with Fedora using mock or the use of a valid OSI license must be provided.

2) **Sociological implementation**: the benefits of this initiative come from the creation of a strong and active community of people accessing the information, maintaining the software, looking up products and people to collaborate with. Such a community provides advantages to both developer and end users. Developers get more visibility for their products, comments, feedback and patches; they can get the help of experts in specific technological domains for solving difficult problems, find libraries and tools useful for their development activities. End users get information about software products, ratings, documentation; they can submit requirements and get help from other users or technical experts. One of the benefits of creating such a community is also the establishment of a "natural selection" environment able to promote mature products and filter out unsustainable products, where maturity is defined based on a practical set of criteria (functionality, reliability, documentation, support, innovation, etc.)

Benefits

For EMI

The EMI project needs to clearly define and implement a sustainability strategy to allow products currently developed and supported within EMI to be continually evolved, supported and used after EMI. It is of course clear that the key to sustainability is in the availability of resources to carry on the activities. Resources become available when there is a demand and a will to pay for that demand to be satisfied. The creation of an open software community satisfies the sustainability requirements of establishing a growing base of users and streamlining available funds and resources onto products for which a real demand exists. This process goes also through an investigation of which products have the best chances for further funding or for generating revenues through support and consultancy activities. The proposed initiative addresses this problem in four ways:

1) Provides a widespread database of information about the software products and their functionality and a channel to advertise the products, collect feedback, propose additional services to a wider audience, feeding information into search engines and other taxonomy systems
2) Provides a single entry point to accessing source code and binary package from wherever they are stored, easing the tasks of creating distribution for specific communities like WLCG and EGI (UMD)
3) Provides a system to collect and formalize information about the software products, which will help in understanding how the products are used, by whom, in which contexts, etc.
4) Provides a social networking environment where people can advertise their skills or find the skills they need for their research or development projects

For the Scientific Community

There are two primary benefits for the scientific user communities depending on whether they need to find software and skills for new or existing activities or to get information and support about software they are already using. The creation of an active software infrastructure is essential to promote the usage and sharing of software and related information to help in both cases.

A comprehensive catalogue of software and services is fundamental in finding the right tools or checking whether solutions to problems already exist before committing resources to new initiatives. The possibility of accessing professional profiles of people helps to create the right conditions for collaborations and improve the chances of locating the right persons for a job based on their past achievement and recommendations from other people.

One of the major issues users face today is the difficulty of finding software and documentation and getting help for problems. The "users-help-the-users" aspect is very often underestimated, but it is not only beneficial for the developers in terms of reducing their support load, but also very effective for the users, who can talk to people having the same issues and "speaking the same language". Available solutions can be shared; problems with enough users complaining about them can receive more attention from the developers and help set priorities. This process results in an improvement of software quality through standard open source contribution methods. In addition the natural selection of products through the use of rating systems and recommendations is another important benefit that can rapidly lead to a wider use of the best products.

State of the Art

The source code and binary packages used across the research infrastructures are currently hosted in a multitude of source code and binary repositories ranging from in-house institutional repositories to standard open source community web sites. The following main categories can be distinguished:

1) **Institutional source code repositories**: most of the source code developed in EC funded projects is stored in source code repositories hosted by the research institutes contributing to those projects. Typically the repositories are based on CVS, Subversion, GIT or similar distributed version management tools. This solution is usually efficient and easy to maintain since it doesn't add particular overhead on top of existing local procedures. However, these systems may require giving access to people external to the Institute to be able to contribute with the additional burden of managing accounts for people without legal affiliation with the Institute.
2) **Institutional binary repositories**: many Institutes offer the possibility of distributing binary packages from local web sites or full-fledged APT or YUM repositories. Also in this case the administrative overhead is usually small, however an excessive number of such repositories creates overhead for the users, who have to locate and configure multiple sources in their deployment procedures.
3) **Operating Systems Repositories**: binary packages of a high enough level of maturity and maintained by expert open source packagers are distributed via officially endorsed repositories like EPEL, Debian, Fedora, Maven Central and others. This is usually the best distribution channel,

since it is normally pre-configured in standard Operating Systems and guarantees a certain level of interoperability

4) **Open source repositories**: a number of products are distributed from third-party open source repositories like SourceForge or github. This solution is efficient in terms of hosting, but it presents difficulties in terms of organization of a community, since such repositories do not require any special criteria to be satisfied for submitting new projects and do not represent a specific community, but a generic open source community. However, they can and should be used to integrate the repository functionality in the proposed open source initiative.
5) **Open source foundations**: a number of "non-for-profit" foundations exist to coordinate and promote the development and support of open source software like the Apache Software Foundation, the Eclipse Foundation, the Drupal Association, etc. These foundations can distribute software, but they either provide reference distributions that end up being rebuilt and packaged for specific operating systems by experts within the community (like Apache) or are independent from specific operating systems and are distributed as modules within a reference framework (Eclipse or Drupal). The license to be used is often mandated for all projects within the foundation.

None of the above described solutions are able today per se to provide the higher level functionality that this initiative proposes to build. Indeed no single repository or foundation can, if they only take into account the content that they "own" or modules written for a single reference framework. This is not currently the case with most of the software we deal with, which is hosted in many different repositories, written in different languages and based on different technological solutions. Moving the entire codebase into a single global repository could be an ideal long-term solution (and github could provide a very interesting, well managed and free service), but it's presently not achievable in the short-term. Hence a federating or brokering architecture is desirable for the time being. What creates a commonality among the different software products is not their technological attributes, but the purpose they are written for, namely the execution of scientific research. In addition these solutions are repository-centric, while the community needs more functionality in terms of creating relationships, which this initiative can provide.

Mandate

The mandate of the proposed organization is to coordinate the activities of an open community of users and developers of software to be used in scientific research endeavours. The software can include middleware, applications, tools, testsuites and any other product providing useful functionality to scientists within research activities or software engineers and developers. The software can be based on different technologies and be written in different languages, as long that it provides a function of some type within a scientific research project. Such software can be based on grid, cloud, high performance computing, volunteer computing or any other suitable computing model with a user community behind it.

The coordination include screening of contribution requests to verify they respect a set of minimum quality criteria (relevance, usage, compatibility with major supported operating systems, availability of support, documentation, etc.) and the organization of online and live events and activities to stimulate sharing of information and the establishment of active communication channels between developers and users. The organization is also responsible for managing the web-based information hub and a minimal set of tools to allow users to contribute software and track its status within the community (feedback, patches, usage statistics, etc.) Additional functionality can be added in the future if needed and if effort is available.

Organizational Structure

The organizational structure should be as simple as possible and similar to existing successful open source organizations. In particular the Apache Software Foundation is taken as main reference. It is envisaged to have a President or Managing Director and a small set of experts (Vice-Presidents) responsible for the various functions or areas of the organization (grid computing, cloud computing, HPC, web site and community tools, public relations, licensing, etc.) It is important to keep the size of the managing team small and to involve committed professionals able to dedicate their time and expertise to actively coordinate their areas of responsibility. The President or Director and the Vice-Presidents are elected each year among a set of suitable candidates selected by invitation or spontaneous candidature.

The structure and operations must be open and transparent and the interests of the stakeholders must be safeguarded. This can be done by establishing an advisory board composed of representatives of the major user and developer communities. The election mechanism also ensures direct participation in the strategy and management of the initiative.

Legal Form

Although establishing a legal entity may not be necessary from the beginning, it is considered important to make sure that the organization mandate and structure don’t prevent such an entity to be created in the future. The most suitable form would be that of a “not-for-profit” organization (foundation, association or other type of legal entity). This would allow the organization to take part into fund-raising activities and in turn fund activities within its community.

Operations

As described in the proposed mandate, the organization is responsible for managing the community, running a web site and providing an efficient set of tools to allow the community member to interact. The set of features offered can grow over time if required, but at the beginning the following operations are considered necessary:

1) A community web site to showcase the organization activities and act as community hub
2) Functionality to submit new projects and collect all required information about software properties and people, manage existing projects, submit patches to hosted projects, etc.
3) Data mining, statistics and visualization tools
4) Social networking tools, like a forum, blogs, profiles, etc., with the possibility of setting up ratings, recommendations, followers, establish sub-community having common specific interests, etc.

Ergonomic considerations are to be taken strictly into account. It is important to design and implement the portal to be very easy and intuitive to use with a few clearly visible and clearly accessible functions. Such a design is part of the value proposition and a critical factor for the success of the initiative.

Submission of projects should be very simple and based on existing open source mechanisms. It is envisaged that the organization would not require hosting source code and binary products in its own repositories, but would on the contrary encourage to use existing well-established repositories (like github for source code or EPEL and Debian for binaries) or existing Institutional repositories. However, it is also envisaged to offer a time-limited repository service for projects not currently having their own facility or not yet mature enough to be directly contributed to other mainstream OS repositories. This can be done using any existing reliable open source repository. However, the owners of such hosted projects must commit to adhere to the recommended guidelines within a fixed period of time.

The organization is also mandated to coordinate high-level community events, such as conferences or workshops and to actively work on fund-raising activities to support the organization. Such fund-raising activities can include setting up sponsorships, advertising, paid partnerships and memberships and other suitable schemas. A brokering function between service providers and service consumers could be established to match user needs and service offerings to the benefit of both users and developers. Commercial providers could also be motivated to advertise their services for a fee if the community is large and profitable enough.

Partners and Customers

The organization should be managed by individuals committing their own time and expertise and endorsed to do so by their home Institute or Companies. Institutes and Companies can be partners in the organization with advisory functions, but partnership does not entail automatically the attribution of roles within the organization, which are assigned by election as described earlier. The partners are expected to actively contribute to the activities of the organization by providing resources, expertise, networking and public relations and general advisory functions on the organization strategies and activities.

Customers are in general any person or entity interacting with the organization to contribute software or use the hosted software.

After a first round of informal discussions, the following parties may be interested in being involved in various forms in the initiative:

EMI Partners (who? What level of commitment? For what benefits?)

EGI (the community of site administrators at large is asking for better and more structured ways of submitting patches to fix issues found in the deployed middleware services and applications. EGI interest and commitment must be secured in order to set up a successful initiative)

WLCG (software used by WLCG must be readily available and there must be a simple way to contribute to it)

PRACE (the HPC community at large is interested in sharing and advertising code, PRACE could be involved in as advisory or coordinating role)

StratusLab (interested in finding a place to make its software available and visible)

EDGI (interested, but also already working on establishing a similar foundation dedicated to desktop computing, what overlaps or opportunities do we have?)

OpenAIRE (considered a natural extension of the broader digital storage activities, interesting to see how existing tools used in OpenAIRE and OpenAIRE+ can be reused in area of software. In addition OpenAIRE is very influential with the EC and can help in promoting the initiative)

OSOR: The Open Source Observatory and Repository project (http://www.osor.eu/) is an EC-funded project tasked with establishing a network of repositories and application in the e-Government domain. Its goals are similar to what the EMI initiative proposes, although there is much more focus on the repository aspects than the social aspects. Nonetheless there could be opportunities for sharing ideas and tools and the EC would certainly expect a connection between the two initiatives.

User communities: developers (applications and tools developed by infrastructure and user communities are potentially the main targets of this initiative. Early discussions with projects like StratusLab, iMarine, DNET showed interest in using the proposed software infrastructure)

User communities: end users (need to understand clearly who the users are and how to involve them from the beginning. Having a single well-known place for finding software and support and interacting with other users and experts is a very strong aggregation and motivational element as demonstrated by existing open source organizations. However, the hosted software must be useful and of high enough quality in terms of functionality, documentation and support to be of interest to the end users.)

The **EC** is necessarily an important stakeholder in this initiative. Involvement of the EC in endorsing and promoting its benefits and the activities must be secured. The EC would benefit from the establishment of the proposed open source community in terms of visibility and exploitation of the outcome of the funded projects, cross-collaborations among projects, collection of statistics on actual usage of the software and promotion of a wider open source culture within the research communities.