Bringing the Embedded Systems Industry Towards Open Source: the SHARE project experience

G.Carrozza[1], A. Marongiu[2], L. Benini[2], M.Loffreda[1]

(1) Consorzio SESM-SCARL - Via Circumvallazione Esterna di Napoli
80014 Giugliano in Campania - Naples, Italy
(2) DEIS - University of Bologna, Viale Risorgimento, 2 - 40136 Bologna, Italy

Abstract: Open source software adoption in the embedded systems domain is gaining growing interest within the european industrial and academic communities due to the significant benefits it brings in terms of flexibility and cost reduction. Nonetheless, scepticism about open source as a viable option to support critical business functions still holds, since its decentralized and distributed development model makes quality evaluation and assessment hard to achieve. This paper reports the SHARE project experience, aimed at facilitating and promoting the use of open source software in the embedded systems industry. Performed activities, proposed methodology and achieved results are presented, along with lessons learned to exploit for enabling further initiatives in the next future.

Keywords: Open Source, Embedded Systems

1. Introduction

Over the last decade industry has shown an ever-increasing interest in Open Source Software (OSS). Nowadays large, medium and small companies are becoming more and more aware of the actual benefits that OSS provides in terms of reduced license and development costs, license flexibility, increased number of suppliers and market innovation. However, although many OSS products have reached maturity during the last few years, scepticism about open source as a viable option to develop commercial products, or to support critical business functions, still hampers the industrial world transition to open source. The decentralized and distributed development model of OSS makes quality assessment hard to achieve, thus making OSS integration within complex industrial-strength applications a risk. In fact, companies find hard to believe that open solutions are capable of delivering the same reliability level of commercial offerings and that they can actually fill the requirements gap better than commercial products. For large projects and mass market products, the situation is alleviated by the availability of a plethora of OSS software projects, supported by large communities of developers and users. Such a broad user base contributes to increase the know-how, and often makes large open source projects even more reliable than commercial solutions. This does not hold in the case of specialized markets which have a narrow set of product users, and whose requirement for specialized development expertise reduces the size of developers communities. This is especially true for embedded solutions, which are increasingly being used for developing mission and safety critical applications, e.g., for transportation systems, medical devices, critical infrastructures, and mobile computing. These systems are usually developed and tested in house by leading firms and impose strict dependability and real-timeliness requirements, as well as the need for achieving international certifications. Hence, assessing the quality of eligible OSS products to integrate into proprietary industrial solutions becomes a major concern.

So far, qualitative and quantitative evaluation methodologies have been lacking (see [QUA] for a comprehensive comparison), as well as approaches which actually address niche market sectors that can be leveraged by the embedded systems industry.

Both industry and academia are currently concentrating their efforts on how to enable the embedded systems industry to move toward OSS, all over Europe. This is also witnessed by the increasing number of initiatives and research programmes focusing on embedded systems, like ARTEMIS or European Commission objectives in the framework of 7th Framework Programme (FP7) calls.

The SHARE project, funded under the 7th Framework Programme, aims to facilitate and promote the use of OSS software in the embedded systems domain, and to foster cross communications among several classes of stakeholders, especially in the fields of e-health, nomadic applications and mission critical systems.

In order to help industry in the delicate task of choosing among several available solutions the one which best fits their requirements, a qualitative methodology for OSS evaluation is proposed which takes into account the specific needs of products targeting embedded systems.

A practical implementation of this evaluation methodology is also provided. The O4S tool – available on the SHARE web platform – allows a wide community of users and developers with expertise in embedded systems to evaluate and rate OSS solutions. This in our view helps to create an increasing awareness about OSS products, as well as specific know-how on available technologies. Moreover it strengthens the effectiveness of feedback and support actions coming from small communities.

The main novelty of the SHARE evaluation methodology – with respect to other previously proposed similar approaches – consists in the set of considered evaluation criteria. Indeed, due to the lack of mass interest in the targeted fields, most criteria encompassed by the SHARE O4S tool – which actually focus on the peculiarities of embedded systems – have not been considered before.

Through the development of a web platform, the project aims to become a virtual meeting point for the embedded system community where people can share knowledge, as well as to encourage cross domains collaborations. It also aims to promote the adoption of open standards in the embedded systems field, which would give a boost to the use and development of OSS in charge of complying with certifications requirements.

This paper will thoroughly describe the tools and the methodology developed in the framework of the SHARE project, as well as the lessons learned by the achieved results.

2. Benchmarking OSS

In order to enable and drive the evaluation of OSS to integrate into proprietary industrial solutions, or to be included into the industry development process, a benchmarking methodology, as well as a support tool, have been developed by the SHARE project which are described in the following sections.

2.1 The SHARE Methodology

The SHARE benchmarking process is an extension of the Qualification and Selection of Open Source software methodology (QSOS) [QSOS], aiming at creating a process tailored for the kind of systems and software benchmarking falling into the SHARE focus.

As it is shown Figure 1, the process is composed of four steps:

1. **Definition**, aiming to define the basic concepts enabling the evaluation:
    a. software families;
    b. software context (e-health, mission-critical, nomadic and others);
    c. functional grids to describe each domain;
    d. Free/Libre/Open Source Software (FLOSS) licenses classified according to Ownership, Virality, Inheritance criteria;
    e. Communities, which have been identified in insulated developer, Group of developers, developers organization, Legal and, commercial entities.
2. **Evaluation**, during which each version of the identified software products is actually evaluated. To this aim, each software is described by means of an *evaluation sheet* containing (i) a "software id card" reporting general information about the software, and (ii) software evaluation criteria. These are further divided in:
    i. *Functional coverage*, i.e., how a given product is able to fit context requirements. This is the "dynamic" part of the evaluation model based on context-related criteria.
    ii. *Users risks*, which do not depend on the *context* and which are weighted during evaluation step
    iii. *Providers risks*, i.e., risks viewed by a provider of services based on FLOSS.
3. **Qualification**, aiming at defining filters that translate requirements and constraints related to the selection of FLOSS in a specific *context*. Here is where the user's content is qualified. Filters are possible on software identity card or functionalities, as well as on users and providers risks.
4. **Selection** which allows selecting software fulfilling user's requirements and comparing different products belonging to the same family. The QSOS methodology proposes two alternative ways of accomplishing this fourth step: *strict selection*, that means to eliminate software which

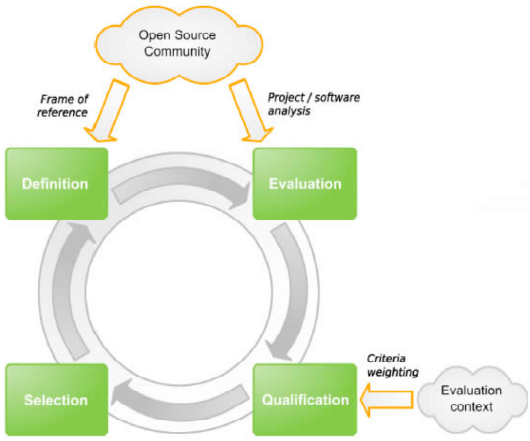

Figure 1: SHARE and QSOS methodology

do not fulfil users' request, and *loose selection* that requires, instead, to use filters built in step 3 to "weight" choices. This step results in the comparison of products in a given family.

The idea of "context" represent the added value provided to the existing QSOS methodology. It provides a means to specialize evaluations with respects to the particular area of interest and encompasses the definition of *common* and *specific* criteria which have been used to perform evaluations and comparative analysis.

The common criteria provided by the QSOS methodology follow a nested structure encompassing four groups at the first level (intrinsic durability, industrialized solution, technical adaptability and strategy) . These criteria are suitable for evaluating any OSS or system regardless of the operational field of interest. *Specific* criteria, instead, have been introduced to perform custom evaluation: they take into account different parameters for each evaluation context.

2.2 The O4S tool

The tool developed to perform evaluations has been created as an extension of the O3S tool[1] to which database facilities and several functionalities have been added to fit the defined methodology.

In this section we describe the tool, as well as the steps it enables to perform OSS benchmarking according to the methodology described in section 2.1.

The evaluation process leverages a set of XML *templates*, which constitute a skeleton built on top of the identified evaluation criteria. A different template is defined for each target software family, and is structured in two main blocks. The first block is shared among families, as it contains a *common* set of criteria which describe generic features of the product (e.g. software maturity, availability of documentation and community support.). The second block includes specific criteria to each family, thus differentiating the evaluation based on true key features of different OSS. Criteria can be scored within a range of three values (0 to 2) which typically have the following meaning:

0. the target feature is not supported
1. the target feature is partially supported
2. the target feature is fully supported

In case this classification does not hold, and the score assumes different, criterium-specific meaning, the template provides descriptive comments to aid the evaluation process.

1 http://www.qsos.org/o3s/

The tool, which can be used by the front-end of the project website[2], allows an iterative benchmarking process consisting of the following steps:

- **CONTEXT AND FAMILY SELECTION** - Users have to select the area in which they are going to use the product (near real time and mission critical; nomadic and multimedia, e-health and generic applications) and the family of the product (e.g, compilers, operating systems, databases).
- **WEIGHTING** - Users express the weight that they want each criterion to assume thus "formalizing" somehow the requirements he is looking for the specific product.
- **EVALUATION** – Provides feedback on if and how a given product is capable of fitting user needs. By selecting among already evaluated tools (stored into the SHARE database) users get a comparative evaluation of several tools.
- **DISPLAY** – Plots the benchmarking results in the form of a graphs or tables.

It is worth to note that users are also allowed to propose and evaluate software products that have not been already evaluated. This increases the advance in knowledge sharing, thus pursuing one of the main objectives of the project.

3. The Share Platform

The Share project aims to enforce relationships between embedded systems industry and OSS communities involving researchers, developers, end users and companies, as well as to promote collaborations favouring an osmotic process of knowledge transfer. This contamination process has been realized through the development of a SHARE WEB SPACE[3], which is close in spirit to the OSS philosophy as it considers nothing better than web, and the Intranet communication tools, for sharing ideas and information. It follows the line of virtual workspaces which result from the complete integration of traditional information systems and the old idea of Intranet.

Through enhanced communication services and knowledge management tools the SHARE web space provides a working environment in charge of supporting the daily needs of people. It was designed both to provide a virtual meeting point for stakeholders interested in OSS, and to be an effective dissemination means for OSS usage in several fields, from business organizations to communities of developers. This virtual workspace

2 http://www.share-project.eu/index.php?option=com_osbenchmark&Itemid=15
3 http://www.share-project.eu

was designed to overcome spatial and temporal barriers to cooperation, and to flatten roles gap.

The WEB space is meant to provide effective support to different figures involved in the OSS software, especially in the context of embedded systems which are becoming ubiquitous in our life. Additionally, it provides the whole lot of necessary facilities to support the Share consortium, such as tools and mechanisms for internal project management procedures.

3.1 Share platform organization and facilities

Figure 2 gives an overall view of the architecture, functionalities and services which have been implemented in the SHARE Web Space.

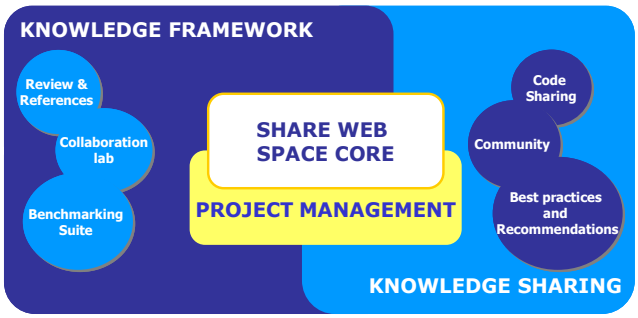

Figure 2: Overall architecture of the SHARE platform

Project management activities are managed through the so-called *Virtual Secretariat* framework. It allows SHARE consortium members to leverage several facilities for an effective project management, like data management, documents repository, calendar, internal communication and white pages. It has proven an effective means to encourage continuous information flow among the partners and to increase the cohesion of the consortium.

Platform interfaces with external users are organized into a *Knowledge Framework* and a *Knowledge Sharing* section. The former provides means and tools enabling knowledge construction whereas the latter aims to encourage interactions among users.

The core of the platform is the *Collaboration Lab* which is the actual meeting point for founding collaborations among people interested in the same topic. It provides:

i. the *Wall*, where users can post their ideas or advertisements

ii. the *Business Card Store*, where users are allowed to upload their professional information

iii. the *Collaboration Tracker*, which is a framework for supporting and keeping alive collaborations promoted by the projects. This has been particularly appreciated by users that have been provided with dedicated files repository, maling tools and calendars to manage their activities.

Within the *Knowledge Framework*, besides the already described *Benchmarking Tool* a *Review and References* section has been included. In fact this is a documentation area, where documents (related to OSS initiative review, user guides, papers and best practices) can be uploaded and stored by the users. This section is continuously updated to keep the community alive and always aware of current progresses in the state of the art.

The spirit of community has been actually realized by delivering the *Knowledge Sharing* section of the platform where several there have been provided

i. a code sharing framework for allowing users to share their OSS projects

ii. a forum and newsletters for encouraging discussions

iii. best practices and recommendation for those OSS and embedded systems practicioners which are going to select products to use.

A *Content Management System* has been chosen as the technical means for accomplishing these functionalities, and to allow all the users to have editing rights, according to given privileges. This way, the SHARE community can grow continuously thanks to users contributions. The Joomla framework Joomla!, version 1.5.7[4] Production/Stable, has been chosen to develop the site. It is a powerful Open Source Content Management Systems, which can be extended both via installable add-ons authored by the developer community and by customized solutions that users can develop by their own.

The O4S tool and several other extensions are going to be released in the form of Joomla add-ons publicly available. This represents a concrete outcome of the project activities.

4. Project outcomes

This section describes the main achievements of the SHARE project, regarding the OSS benchmarking activities, as well as the achieved results in terms of community constructions and knowledge sharing.

[4] http://www.joomla.org/

4.1 Benchmarking results

The main aim of the benchmarking process discussed in this paper is that of supporting the creation of active communities of embedded OSS users and developers. The O4S tool is meant as the main technical means to share and spread knowledge and information among the community.
For each of the three main market sectors considered as the basis for OSS classification within the project, several evaluations have been carried out. A total of 80 products have been evaluated by selecting representative applications in the mission critical, nomadic and mobile multimedia applications, and e-health thematic areas.

These evaluations are stored in a database, available for consultation through the web platform. Users can also provide their feedback (i.e., their own evaluation) to the SHARE community. The tool allows for a convenient and readable display format, where the user can decide whether to display the scores of a single target evaluation (i.e. a single product), or a comparison of all the available evaluations as shown in Figure 3.

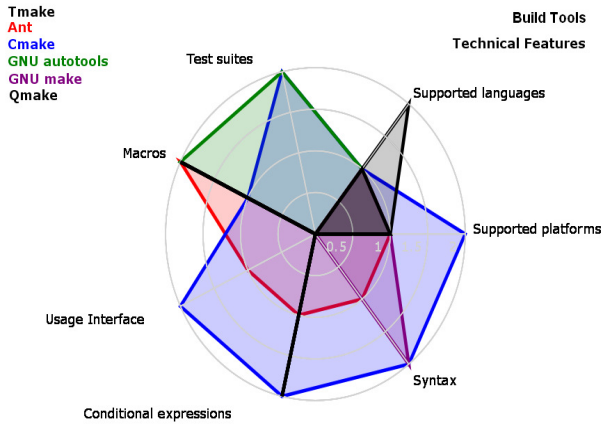

Figure 3: Graphical exploded view of the *Technical Features* criteria subset for the *Build Tools* software family. Comparison of 6 evaluated products.

Similarly, a comparison can be done among several products belonging to a same family. We believe this is a valuable feature for users which are looking for a particular application template, but do not have insight on the specific technical features of different available OSS matching that template. The evaluation criteria are organized as nested subsets of related features (Figure 4).

compilers	GNU Compiler Collection (GCC) - Score	Weight
⊞ **Generic section**	**1.15**	**2**
⊟ **Languages**	**2**	**2**
C support	2	2
C++ support	2	2
Java support	2	2
Ada support	2	2
Multiple frontends support	2	2
Verbosity	2	2
⊞ **Targets**	**1**	**2**
⊟ **Multi Core Optimizations**	**1.17**	**2**
⊟ Automatic Parallelization	1.33	2
Loop Parallelization	1	2
Loop Vectorization	2	2
Reduction Parallelization	1	2

Figure 4: A subset of the criteria tree. Each level has an associated graphical plot.

These can be easily displayed at once by simply navigating the criteria tree at different depths through the graphical interface. An example of the exploded view of two nested criteria levels is shown in Figure 5.

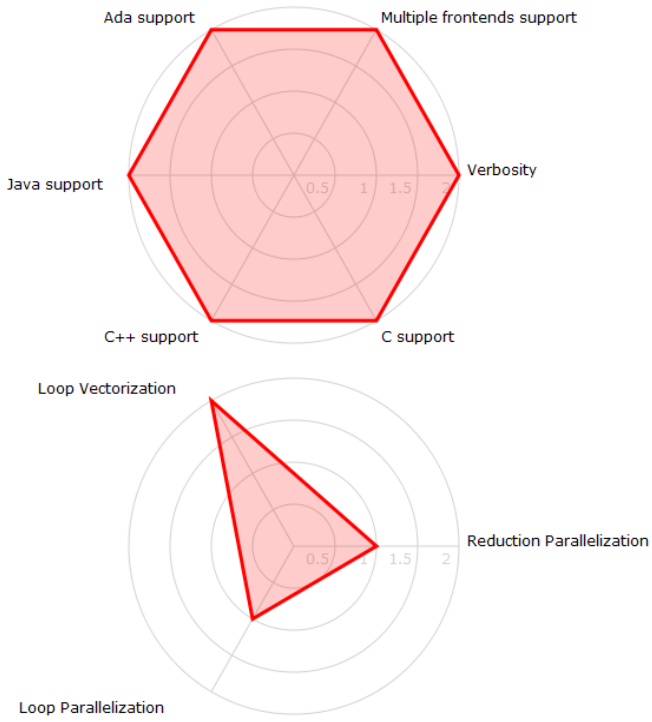

Figure 5: Exploded graphical view of two levels of the criteria tree.

4.2 Share community and created knowledge

During the project, partners have been strongly committed into dissemination activities, thus increasing the visibility project activities and interim results. This has been particularly helpful to allow an incremental community construction. This section summarizes the achieved results in terms of SHARE community results, knowledge base created and collaborations that have risen exploiting the SHARE support.

At the time of this writing, the SHARE web community counts more than hundred registered users, coming both form universities and industries from several European countries, and almost 30.000 visits.
The SHARE consortium is proud of this results as well as the actual collaborations that have been created thanks to the support provided by SHARE.

An interesting case study comes from the joint collaboration that SESM[5] and STM Products s.r.l[6], an Italian SME active in embedded software development, have created. The collaboration was about the evaluation of an industrial solution (from STM) that was to be tested and assessed in terms of quality and reliability. The collaboration has been particularly fruitful and it will continue over the end of the project.
Interesting results have also been achieved in terms of users' interest in the feedback about funding opportunities provided and continuously updated through the Share website. This activity has led to an increasing number of partnerships created through SHARE and aiming to deliver project proposals for the incoming research calls in this field.

Encouraging results also came from the experience that the SHARE consortium put in place with academia. Two real cases are worth to cite. First, a strong collaboration has risen with the *University of Naples* and the *CINI* (the Italian Interuniversitary Consortium for ICT) which is witnessed by the *3rd International Workshop* which has been jointly organized focusing on embedded systems and OSS[7]. Second, a collaboration has been created with *Roma Tre University*: students from a master Embedded Software course have been invited to test and use the O4S tool, in the context of a set of seminars held by SHARE partners. Feedback and opinions from students are believed crucial in terms of future enhancement of the Platform, and are going to be published in the form public deliverables on the SHARE website[8].

5. Recommendations and best practices

At the end of the project activites, some guidelines for each of the three main areas of interest have been derived. The scope of such achievings can be broadened and generalized to other application fields.

The sense of what quality means with respect to OSS is clearly stated in a short sentence by a famous Linux kernel maintainer: ***"It takes time to make things work in Linux, but it takes even more to make the code clean and acceptable by the community".***

In fact, the lack of attention to the generation of "high quality" code is the main reason for manpower effort requirements planning underestimations and deadline misses, which make OSS adoption still dangerous. Remaining in Linux kernel domain, one of the most evident and important Open Source Software success stories, it has been recently noticed that more and more companies asking for Linux kernel base ports are expecting code deliveries to be fetched from public repositories and are not going to accept local branches or custom patches over *vanilla* kernels. This is because having the code integrated into the mainstream Linux tree is a guarantee to the customer for excellent code quality and reliability.

This is a clear symptom of the need for a renewed development process in line with the OSS philosophy, even if it could somehow be in contrast with organizations' internal development procedures.

The use of standards in OSS becomes then crucial for high quality products development, also in terms of development process. In fact, the high variety of products and the need for a short time to market make the reusability of the software elements mandatory.

So far, OSS leverage two kinds of standards, namely *industry driven standards,* mainly delivered by industrial consortia like *Khronos* [KHR], and *de-facto standards* which are related to widely used OSS which become in fact a standard due to their large diffusion.

[5] http://www.sesm.it
[6] http://www.stm-products.com/
[7] At submission time the event still to come. Reports and results will be made available on the Share platforms after April 9th, 2010-03-27

[8] At submission time the activities are still on going. Results will be made available on the Share website at the end of the project (April 30th, 2010)

Industries accepting to adhere to both these kind of standards would be really allowed to deliver cross-platforms products, i.e., to sell the same software for different platforms, as well as to reuse the code in different products.

This is the reason why the SHARE project is committed to encourage any process driving a software into a standard especially in the fields of mobile and nomadic, real time and e-health applications.

In the context of mobile application development the need for open standards is steadily increasing. Nowadays several mobile devices are available on the marketplace, as well as several software development solutions, which are based on a number of wide-spread operating systems such as Linux, Windows Mobile, Symbian. Besides proprietary systems such as Windows and Symbian – which do not even claim to embrace open source (although many components of their SDKs are based on OSS), many Linux-based systems cannot be considered "open", being ad-hoc solutions, tailored on the characteristics of a specific device. Indeed, just as it happens with other proprietary platforms, not all the features of the device are accessible from the software stack due to the lack of specific APIs.
A representative initiative aimed at the adoption of open standards in the mobile area is represented by the Open Handset Alliance (OAH) [OAH], a business alliance composed of 65 participants from several areas including mobile operators, handset manufacturers, software, semiconductor and commercialization companies. The goal of the alliance is that of allowing faster and cheaper development of innovative mobile solutions, which is achieved through "*a commitment to openness, a shared vision for the future, and concrete plans to make the vision a reality*". Android [AND] software stack for mobile devices is one of the initiatives started and promoted by OAH.
Another remarkable initiative in this field is the OpenMoko community [OPM], which supports a completely open solution, from the hardware[9] to the software stack, which includes a rich set of OSS to develop mobile applications. The OpenMoko community is strongly committed to develop Linux-based solutions for mobile programming that ensure interoperability and shared technologies.

In the field of mission critical and near real time systems, the need for standards is even more exacerbated by certification issues. Companies developing critical systems often encounter serious difficulties in satisfying reliability requirements, in many cases imposed by certification standards, at competitive and acceptable cost and time.
In fact, engineering disasters such as the 'Three Mile Island' and the 'Therac-25' incidents let software in this area move from a disordered development field to one with tight, prescribed methods and controls. Standards such as MIL-STD-2167 and DO-178B/C give guidelines for producing safe and reliable software which if followed with well defined methods do have a marked improvement on process quality. Industries delivering solutions for critical domains like aerospace, air traffic management or safety critical systems are required to develop software products exhibiting high Software Assurance Level (SWAL) to be compliant with the these standards for certification.
As for the nomadic application area, industrial communities pursuing standard compliance for software products also exist in this area.
Open Source Real time Operating Systems (RTOS) (e.g., OSEK[10]) standardization boards go in this direction trying to encourage the delivery of OS operating systems that could be used in several mission critical and real time scenarios, providing the same reliability level than commercial alternatives. This way they can be integrated into a variety of industrial solutions without mining the overall quality of delivered products and reducing the time to market. Although these standards are not the only factor in producing high quality software, following a well defined process (similar to processes in other engineering disciplines) unquestionably improves the end product.

The best recommendation, then, is for industries to actively participate to these joint initiatives not only to gain visibility but, more important, to put on the table their requirements and open needs than can be shared and accepted by the community.

Regarding e-health applications, the unquestionable improvement that the new technologies have produced within the medical sector has accomplished the inclusion of ICT in diverse medical tasks which, of course, strongly rely on embedded systems. Even if this is a younger field of application, if compared to the nomadic and mission critical domains, there exists a lot of outstanding activities related to OSS.
Great attention has been devoted to medical images and vital signs processing which have encouraged the development of several OSS applications.
Since signals and image processing have large application, several initiatives are already ongoing to promote OSS, in different engineering fields.

[10] http://osek-vdx.org

What has been observed in the course of the SHARE project, is the lack of OSS specifically supporting the treatment of a given illness (e.g, cardiovascular issues) and, more in general, of highly specialized scientific software for medical diseases.

The benefits that OSS could bring to these areas is great but it has not being exploited so far due to the lack of common initiatives and cross fertilization among different companies delivering proprietary, closed, solutions.

6. The future of SHARE

It is always a tricky task to quantify the real success of research projects. In the case of SHARE we believe that several measures can give an idea of the real impacts that the project actually got like the collaborations that have been created thanks to the project support, the size of the community that has risen around the project, the number of shared OSS projects. However, what we are really concerned about is the future of the SHARE community, and which ways have been paved for future initiatives. Of course, the trickiest issue will be that of keeping the community alive once that project has finished. The great attention that the SHARE consortium devoted to exploitation is a concrete evidence of such a concern.

Even if each member of the SHARE consortium has a different perspective with this respect, all partners agreed that "*exploitation*" actually means which strategies to adopt to have the SHARE initiative survive and grow. SME partners within the consortium are mainly interested in reusing SHARE results and methodologies in their production chain to improve development process and OSS selection activities. Academic partners are mainly interested in training and further research that may arise from the project outcomes. As for large industries, project results constitute a starting point to promote standardization and gain visibility in international scenarios as well as to lead industrial research and business in their fields of interest.

Several actions are foreseen from the whole consortium which, taking into account feedback gathered though technical questionnaires and user forums, will be undertaken to try and keep the community active:

- ***Collaborations with other research project and initiative raised during the project life will be kept alive (hArtes [HRT], FLOSSMETRICS [FSM], QUALIPSO [QUA] among the others).*** Interesting collaborations rose with MORFEO [MRF], which is a significant open collaborative framework where industries and academic organizations have a common point to share knowledge as well as software. Through the fusion with this large community, interesting input will come to the SHARE project. With particular emphasis on embedded systems, the SHARE project devoted great attention to ARTEMIS[11] initiatives which aims to help European industry to consolidate and reinforce its world leadership in embedded computing technologies. ***The natural evolution of these actions will be the creation of larger communities with focus on Open Source and embedded systems, as well as the formulation of larger research projects that will consider the SHARE experience as a starting point***. Dissemination actions performed during the project, aimed at enlarging the network of contacts will be exploited and particularly helpful to create new partnerships and research consortia.
- ***The creation of a network of excellence aiming at fostering cooperation among research groups and spreading funding opportunities awareness***. Since there are many research projects and initiatives focusing on open source, both in academia and industry, creating larger alliances is a powerful way to spread knowledge even across different fields and scientific communities, as well as to increase raising the attention on common topics and needs.
- ***The creation of joint collaborations with standardization organizations. This will give a wider visibility to the SHARE outcomes, reaching all the practitioners in the field of OSS.***

References

[QUA] *Deprez, J. and Alexandre,* S. Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS. Proc. of the 9th Int. Conf. on Product-Focused Software Process Improvement. Lecture Notes In Comp. Science, vol. 5089. Springer-Verlag, 189-203.

[QSS] Method for Qualification and Selection of Open Source software (QSOS),www.qsos.org/download/qsos-1.6-en.pdf

[KHR] http://www.khronos.org

[OAH] http://www.openhandsetalliance.com/

[AND] www.openhandsetalliance.com/android_overview.htML

[OPM] http://www.openmoko.org
[HRT] http://www.hartes.org
[FSM] http://www.flossmetrics.org
[QUA] http://www.qualipso.org
[MRF] http://morfeo-project.org/

[11] http://www.artemis-ju.eu