Key Technological Success Features for a Domain Specific Open Software Ecosystem for Ambient Assisted Living

Marius Mikalsen[1], Ståle Walderhaug[1], Dario Salvi[2], Geir Kjetil Hanssen[1]

[1]STIFTELSEN SINTEF
NO-7465 Trondheim, Norway
marius.mikalsen@sintef.no
stale.walderhaug@sintef.no
geir.kjetil.hanssen@sintef.no

[2]LST
Life Supporting Technologies
Universidad Politécnica de Madrid
Avenida Complutense, Madrid, Spain
dsalvi@lst.tfo.upm.es

Abstract: Ambient Assisted Living (AAL) is a domain with great potential for economic and societal impact. But market uptake of such solutions is so far limited because of market, standards and technology uncertainty. The businesses that will prevail are those that are cost effective. To support cost effective development of AAL solutions, domain specific open software ecosystems are being established. We developed a survey to investigate key technology success factors for such ecosystems. The survey was sent to 60 developers from a representative selection AAL development projects. 18 responded. Following a qualitative data analysis we found several key factors and features that must be in place to facilitate the success of such ecosystems. We found that given the nature of the AAL domain, characterised by divergent users, software and hardware, developers are seeking for support in three main areas. First, they want the artefacts available in the ecosystem to support relevant standards in the domain. Second, support for tracing artefacts available in the ecosystem to requirements (domain-fit) is needed. Third, they want support for developing, testing and emulating for complex user-software-hardware workflows in this distributed environment. The main obstacle that will scare away developers from the ecosystems is lack of documentation of the artefacts in the ecosystem. Second, not enough decoupled components, and finally, lack of proper search features. Finally, in order to be able to learn to use the artefacts, examples, scenarios and API documentation is necessary.

1 Introduction

As the western population ages, more support is needed with fewer hands to cater for their needs. This has led to a recent attention to ICT for active and healthy ageing, commonly known as Ambient Assisted Living (AAL). AAL is applications and technology that provide support for active daily life, inclusion in the society, and prevention and management of sickness and disease [VDB10]. ICT for Active and Healthy Ageing (AHA) has a great market potential, estimated to potential annual revenue of 1837 M with moderate take up, and 2576 M with high penetration [KM10].

Adoption of such technology is still limited, because it is not yet clear how the market for AAL will emerge or what kinds of applications, technology and standards are most likely to be successful in the domain. The business that will prevail are those that are flexible enough to deliver technology addressing changing market requirements more cost effectively than the competition. Cost effective in this context is to provide sufficient quality AAL applications at lower costs than the competition.

To support such cost effective development and to spark the breakthrough of AAL applications and services, the European Commission has launched the Integrated Project universAAL[1] that aim to consolidate previous EU funded research and take steps towards a *domain specific open software ecosystem* providing a standardized approach making it technically feasible and economically viable to develop AAL applications.

Hanssen defines the emerging concept of software ecosystems as [Ha11]: "… *a networked community of organizations, which base their relations to each other on a common interest in a central software technology*." The ecosystem created by universAAL is denoted *open*, which implies that it is centred on free/libre open source (FLOSS) technology.

This paper introduces the foundations of the universAAL project and reports on a survey done by the project on key technology success features needed in such an ecosystem.

This paper is structured as follows. First, in the background section, we present the foundation of universAAL in terms of a characterisation of AAL technology, open software ecosystems, as well as introducing the universAAL ecosystem itself. Second, we present the methodology used for performing the survey. Third, lessons learned so far from AAL application developers´ perceptions on what are key technological qualities that are important for the success of the software ecosystem, pointing out important success factors for the ecosystem to succeed. Forth, we discuss the lessons learned in terms of recommendations for building such an open AAL software ecosystem. Finally we summarize our experience with our concluding remarks and identify steps for future work.

2 Background

2.1 Ambient Assisted Living Needs an Ecosystem

In the conclusions of its meeting of 4 February 2011, the European Council endorses the Commission's proposal for an Innovation Union, and in particular the launch of a

[1] http://www.universaal.org/

European Innovation Partnership on Active and Healthy Ageing[2], by stating that: "*Innovation contributes to tackling the most critical societal challenges we are facing. Europe's expertise and resources must be mobilized in a coherent manner and synergies between the EU and the Member States must be fostered in order to ensure that innovations with a societal benefit get to the market quicker*."

Farshchian et al has done an analysis of innovation characteristics in the AAL domain, and discuss the fact that AAL technologies operate in complex settings involving multitude of stakeholders and organizations. The target end users, i.e. the elderly and their next of kin, constitute a varied group of people with a wide range of needs [FHM12]. They conclude that: "*A range of aspects related to innovation in the marketplace are still missing*." This indicates the need for creating an ecosystem.

This need is also evident in the Lecce declaration, which unambiguously calls for the establishment of a well functioning software ecosystem[3]: "*…work should be directed at building sustainable ecosystems through targeted work, e.g., on ecosystem design, ecosystem compliance and interoperability tests, ecosystem marketing, and life-cycle management of products and services*."

2.2 What is a Software Ecosystem?

As the term of software ecosystem is emerging there is not yet any single authoritative definition of the concept. In addition to the definition presented in the introduction, Jansen et al provides the following [JFB09]: *"… a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them."*

Bosch focuses on the common interest in software and its use [Bos09]: *"the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions."*

The perhaps most well known example of a software ecosystem is Apple´s iOS and App Store, where developers can build, deploy, market and sell applications in the ecosystem owned and controlled by Apple. Application developers are drawn to the ecosystem by the potential impact their applications can have. We denote it as closed as Apple controls the hardware (iPhone, iPad etc.), the iOS operating system, the development tools, and the distribution channel (App Store).

Another well know eco-system, that is semi-open, is Google´s based on the Android OS. This consists of several hardware vendors that deliver phones and tablets running the open source operating system Android. Although Android is open source, the development of the operating system itself is closed and controlled by Google. Also, Google controls the market place, Android Market.

[2] http://ec.europa.eu/research/innovation-union/index_en.cfm?section=active-healthy-ageing
[3] www.aalforum.eu/group/leccedeclaration

UniversAAL, as we shall see in the next section, adopts a very open approach to the notion of creating an ecosystem for active and healthy ageing applications.

2.3 universAAL: Steps Towards an open Ecosystem for AAL

universAAL is a 4 year 13M European Union project that develops an open technology platform for active and healthy ageing. The goal of the project is to create a domain specific open software ecosystem to answer the needs identified by the AALIANCE roadmap [VDB10], including architectures, software components and development and design tools. All results developed in the project will be available as open source under the Apache 2.0 license. In support of the technology platform, communities are established to maintain the project results. In combination, this makes the project constitute a domain specific open software ecosystem for AAL. The main results of the project are introduced below.

Technological Results:

The universAAL Runtime Support [Taz11] is a set of libraries that must be installed on devices that support universAAL applications. At the lowest level it provides a common interface for abstracting resources, e.g., sensors pushing data through the universAAL platform (Execution Environment).

On top of this hardware level abstractions there are some basic services, such as connectivity, security, context awareness and personalization support. The AAL Platform Services enrich these services realizing functions such as secure access to medication information or shared care plan.

Two versions of the runtime support are developed. One is based on OSGi[4] and is available for download and use in version 1.0 from the universAAL Developer Depot. An Android version of the runtime adhering to the same reference architecture is planned.

The developer depot is a repository where the AAL developer community can browse and download existing as well as publish new services. Developers can reuse existing universAAL AAL platform services in the depot in order to create composite platform services [SW11]. The developer depot is accessible from http://depot.universaal.org/.

Using the Eclipse IDE[5] as the basis, new software (plug-ins) that incorporates the universAAL reference architecture and design concepts (UML Profiles from the universAAL reference architecture) is available. Part of the tools is dedicated to expressing the universAAL architecture and ontologies using model driven software engineering techniques such as model-to-model transformation and model-to-text generation [SW11].

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

[5] http://www.eclipse.org/

The reference architecture represents a detailed specification of functional components and the relationships among them for building systems in the AAL domain. Its most central parts are:

- Reference Use Cases: shows the main stakeholders of the system and how they would interact with it

- Reference Requirements: is a collection of requirements for a generic AAL system

- Reference Detailed Architecture: is a proposal of an architecture with its modules and functionalities that are mapped to the use cases and requirements

The uStore [BBA11]: is marketplace that enables selling and buying of AAL services. The uStore will enable service providers to specify complete AAL services (the required software, hardware and resources) and sell these through a secure and easy to use interface.

Finally, universAAL is producing a set of example services that demonstrate use of the underlying platform capabilities. The services will be used both for giving the adopters of the platform a training material for understanding the functionalities offered by the runtime support, and for reusing software components that can be included into new services. The services include an implementation (i.e. the source code) and documentation [Ibz11].

UniversAAL Community Results:

To sustain the universAAL results after the project ends, focus has been to build a community that can maintain and further develop the results in the ecosystem. To this end, the Ambient Assisted Living Open Association (AALOA) has been established and has currently, over 100 individual supporters. AALOA is available from http://www.aaloa.org/. The main mission of AALOA is to bring together the resources, tools and people involved in AAL in a single forum that makes it much easier to reach conclusions on provisions needed to achieve AAL progress. Second, make sure that all technology providers, service providers and research institutions involved in AAL are either directly involved in AALOA or (as a minimum) aware of decisions it promotes. Third, involve end-user representatives in all work of AALOA. Fourth, identify key research topics in AAL, and reach agreement on prioritization of these. Fifth, design, develop, evaluate, standardize and maintain a common service platform for AAL.

3 Methodology

We developed a survey sent out to a selected set of AAL projects to get insight that will help shape the development and evaluation of the software development tools created in the universAAL project. Developers from five different AAL-related EU projects were asked to participate in the survey (the survey was sent to 60 developers, 18 responded). The reason for asking developers in the selected projects is that our goal is to build tools that solve key challenges facing developers of AAL software, and we wanted to get an understanding of what key factors that will influence developers' acceptance of such open tools and components. Participation in the survey was motivated by the tools being based on an existing development platform (Eclipse) and that the results will be provided for free and as open source.

In accordance with Norwegian law, the survey was approved by the Data Protection Official for Research, Norwegian Social Science Data Services.

Questionnaire Design

The questionnaire was designed to address what the developers perceived as important aspects of developing software for AAL/e-inclusion projects with a strong focus on developer tools and practices. The questionnaire was designed by the researchers in the universAAL project with the goal of getting answers to the main barriers facing developers of AAL systems, and what they would see as the most beneficial features of the universAAL tool results. The questionnaire was pre-tested by participants of the universAAL project before being sent to make sure that all items were clear and understandable.

The survey had two core items pertaining to the creation of the universAAL software ecosystem:

1) Investigate current development practice, and design and expected benefits of an AAL software ecosystem.

2) Investigate possibilities and obstacles of a dedicated AAL software ecosystem, and learning strategies.

The questionnaire was based on 5-point Likert scales and fields for textual input where respondents could provide more information when and where applicable.

Subjects

Developers from five different AAL-related EU projects were asked to participate in the survey. The projects were Amigo[6], Soprano[7], MPOWER [St11], Persona[8], Genesys[9], and Oasis[10]. The invitation to participate where sent through project internal developer mailing lists. A total of 60 developers were invited. 18 responded.

Data Analysis

The rather low number of respondents (*n*=18) reduces the effect of doing statistical analysis (like regression analysis etc.). The result does however lend itself quite well to

[6] http://www.hitech-projects.com/euprojects/amigo/
[7] http://www.soprano-ip.org/
[8] http://www.aal-persona.org/
[9] http://www.genesys-platform.eu/
[10] http://www.oasis-project.eu/

data analysis using descriptive statistics and frequencies. The survey also allowed for free text input to allow developers to provide input important to them on selected items. As the survey is meant for gathering input to shape the development and evaluation of the software development tools created in the universAAL project, we see the results as valid for their intended purpose.

For the analysis of the free text input we report the text provided to us as well as the researchers analysis of the provided input. The text were analysed based on the principles of "Grounded Theory" [SC98], which can be used to categorise or find central concepts from textual data. We used an open coding approach where we identified categories of phenomena within the topics selected for the survey.

4 Results

4.1 Findings on Current Development Practice and Expected Benefits of Tools and Components in an AAL Software Ecosystem

Summary of data on programming languages and tool usage

We found from the self-reported programming skills that the highest skills were reported for Java (a mean score of 3.44 reported). Knowledge of OSGi development was rated lower. Design and modelling knowledge (such as UML), gets a lower rating (a mean score of 2.72 reported). The most frequently used developer tool in our sample is Eclipse.

Summary of data on the importance of use of standards

The statement "Using standards is important for my company/organization" got a mean score of 3.94. The statement "Standards are relevant in my work" got a mean score of 3.89.

Summary of data on design and development tasks that would benefit from tools support and reusable components

We found from the categorization of responses that a main thing that developers would benefit from having support in is design, documentation and traceability support. This can be in the form of traceability from requirements, through design and to the components, or by documenting use cases. The respondents also wanted to get support in requirement analysis towards finding AAL specific services, that is, a link between requirements and available components in the ecosystem. This is also reflected in the feedback on the user interface artifacts developers would like to have available in the ecosystem, where the respondents highlight a need for reusable user interface components for divergent devices (e.g. PCs and embedded) and proven user interface designs.

The mean value reported for use of design patterns is quite high, 3.88, indicating that developers use design patterns. Also quite high is the use of code conventions, 3.59. A mean of 3.47 suggests that the development process tends to be repetitive and follows a certain pattern or structure.

Equally important in terms of number of responses is the request for support for deployment to divergent smart devices, sensors and other hardware. Developers also want support for configuring gateways and integration of sensors into their solution. Finally they want to be able to emulate and test on emulators for smart devices/sensors. Two respondents that report need for testing and automated testing, and the reported need for workflow management and messaging in distributed OSGi environments.

4.2 Findings on Possibilities and Obstacles of a Dedicated AAL Software Ecosystem, and Learning Strategies.

Summary of data on how to access to software components

We found that the majority (15 of 18) would prefer to have access to development resources (code, designs etc.) directly through the development tools they use. The alternative is to use a web browser to navigate such repositories.

We also investigated what would be the main features of a dedicated AAL repository. The primary feature required was to have documentation and examples. This includes descriptions of components, manuals, APIs, tutorials and functional code examples.

Second, the components that developers wanted are in three main categories. These are, support for different platforms (hardware and software), libraries for AAL specific reoccurring tasks (including user interface components), and examples real application (that has been in use) that can be used for evaluating new project ideas.

Only two developers mention open source and only one mention the possibility to get user ratings of components in the repository.

Summary of data on current obstacles to code reuse

The primary obstacle to reusing code in terms of number of responses is lack of documentation and bug tracking. The second most important obstacle is that components that are available for reuse are not open enough in that they have to many interdependencies with other components and are not properly decoupled. The third most important obstacle is that it is hard to find appropriate components due to lack of search support ("*Google is not enough*")

Summary of data on assets that would help a developer to implement a successful application

The results show that examples are the primary source of information developers would need. This ranges from example of use of the application (including demonstrations, screencasts and step by step descriptions) and also examples including code samples. The second most important asset is scenarios, use cases and requirements in the domain. These should also be accompanied by code examples. The third most important asset is API and other documentation including standards and UML models. Other sources of information are feedback from experts in the domain, tutorials and tool support.

This is also supported by the numbers we found when we asked what resources the developers would use when learning new platforms and code components in a new domain (such as AAL). First, the majority (n=16/18) look at example code to learn. Second, they use API documentation to learn (n=15/18). Third, quite few look at designs (n=5) and use project wizards and templates (n=4) to learn. Finally, one respondent lists other sources for learning, namely to ask colleagues.

5 Lessons Learned and Preliminary Recommendations for Building an Open AAL Ecosystem

5.1 Current Development Practice and Expected Benefits from an AAL Ecosystem

Programming language

We found that Java was the programming language where most developers reported their highest skills. However knowledge of OSGi (which is written in Java) is rated lower. Attention therefore must be paid to provide sufficient levels of training material and documentation as more advanced technologies than plain Java APIs are introduced in the ecosystem. Eclipse was the integrated development environment (IDE) that most developers reported using, this was expected, as Eclipse is a very popular open source IDE, it is extendable (plug-in model) and as such should be a good choice for providing tool support in an AAL software ecosystem.

Standards

For standards, we found that a majority of the developers found the use of standards important and relevant for their work. An AAL ecosystem therefore will benefit from providing software artefacts that adhere to relevant standards in the domain, such as for example the Continua standard for communication to medical devices (IEEE 11073), or HL7 or similar CEN TC 251 standards for exchanging information with dedicated health systems. An AAL software ecosystem benefits from providing software components that work with these standards, and should as a very least have a flexible design so as to allow the use of relevant standards, facilitating divergent business models, e.g. both CEN and HL7.

Tools and design support and reusable components

From the tools, developers want support in four main areas. First, support will be appreciated in discovering requirements in the domain (understanding the domain), and then to be able to have traceability from these requirements to the applications created, so as to be sure to implement the right functionality. Second, as the developers use design patterns, relevant and applicable patterns in the domain should be available. Third, tool support is needed to emulate different kinds of sensors, hardware and gateways and manage complex workflow in distributed environments. The domain of AAL is characterised by a mix of divergent user needs, hardware and software. These findings indicate that developers are seeking for guidance in the form of best-practice AAL specific design patterns (domain specific languages), components, and traceability from requirements to designs and components in this fluctuating reality. Finally, they need emulators and testing facilities to be able to test complex distributed applications.

5.2 Possibilities and Obstacles of a Dedicated AAL Software Ecosystem, and Learning Strategies

Possibilities

Developers would like to have access to certain resources from the ecosystem directly from the development tools. The resources they would like to have access to include documentation, tutorials, and functional code examples for the software artifacts available. The most wanted software artifacts were support for different hardware, for reoccurring tasks, and examples of actual applications that have been in use and that are possible to run out of the box and experiment with. Not so many was concerned with the resources being open source. Again we find that the developers are looking for ways to cope with a domain of divergent user needs, software and hardware, and good example applications that has been in use, that are available for use and tailoring are appreciated. The fact that open source is not a requirement many posed, is presumably that other concerns related to quality is more important, as we shall see next.

Obstacles

What we found are best suited to scare away developers from entering the ecosystem is first to not document the resources properly. Second, if there are too many interdependencies in a software artefact, so that it is difficult to start using it, they are simply not going to do it. Finally, if they cannot find the components, they will not be used. These are classical errors the open source projects make, and it is equally true for AAL ecosystems. Resources need to be easy to find, well documented, and decoupled.

Learning

In order to learn to use resources in the ecosystem, developers would prefer to see example use of resources (including demonstrations and walk-throughs) and example code. Scenarios and use cases are necessary to trace to needs in the domain. APIs are another source of information. This is easy to forget, but to be able to use the resources in the ecosystem, these sources of learning are essential, and will contribute to bringing more users into the ecosystem.

6 Conclusion and Future Work

In this paper we have reported the results of a survey that set out to find the key technological success factors that would attract developers to an open AAL software ecosystem. We found that given the nature of the AAL domain, characterised by divergent users, software and hardware, developers are seeking for support in three main areas. First, they want the artefacts available in the ecosystem to support relevant standards in the domain. Second, support for tracing artefacts available in the ecosystem to requirements (domain-fit) is needed. Third, they want support for developing, testing and emulating for complex user-software-hardware workflows in this distributed environment.

The main obstacles that will scare away developers from the ecosystems is lack of documentation of the artefacts in the ecosystem. Second, not enough decoupled components, and finally, lack of proper search features.

In order to be able to learn to use the artefacts, examples, scenarios and API documentation is necessary.

This work gives a synopsis of what a sample of AAL developers rate as technological enablers and barriers for a successful AAL software ecosystem. More work is needed to relate this work to the larger work on theories of software ecosystems, in order to extend the knowledge base. Also, more qualitative studies in the form of e.g. focus groups with developers are needed to find deeper and more concrete insight into the mechanisms that motivates developers are needed to provide even more relevance for the industry.

7 Acknowledgements

This work was partly funded by the EU Seventh Framework Programme, Theme ICT-2009.7.1, ICT & Ageing, Grant agreement no. 247950.

References

[BBA11] Burla, A., Bakalov, I., & Arnaulov, V.,: universAAL Deliverable 3.4 - Tools for Deployment and Management - uStore, 2011 Available at: http://www.universaal.org/index.php?option=com_content&view=article&id=66&Itemid=15

[Bos09] Bosch, J.,: From software product lines to software ecosystems. In Proceedings of the 13th International Software Product Line Conference. SPLC 2009. Pittsburgh, PA, USA: Carnegie Mellon University, s. 111–119.

[FHM12] Farshchian, B. A., Holbø, K., Mikalsen, M., Reitan, J.: AAL Technologies in Rehabilitation – Lessons Learned from a COPD Case Study. In Handbook of Ambient Assisted Living – Technology for Healthcare, Rehabilitation and Well-being. Ambient Intelligence and Smart Environments. IOS Press, 2012.

[Ha11] Hanssen, G. K.: A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. In Journal of Systems and Software, 8699(7465), 2011, s.1–12.

[Ibz11] Ibanez, G., universAAL Deliverable 4.2 – Innovative AAL Applications on universAAL Platform., 2011 Available at: http://www.universaal.org/index.php?option=com_content &view=article&id=66&Itemid=15.

[JFB09] Jansen, S., Finkelstein, A. & Brinkkemper, S.,: A sense of community: A research agenda for software ecosystems. In Proc. 31st Int. Conf. on Software Engineering - Companion Volume, 2009. ICSE, s. 187–190.

[KM10] Kubitschke, L., Müller, S., Gareis, K., Frenzel-Erkert,U., Lull, F.: ICT & Ageing - European Study on Users, Markets and technologies, European Commission, 2010

[SC98] Strauss, A.C. & Corbin, J.M.,: Basics of Qualitative Research: Second Edition: Techniques and Procedures for Developing Grounded Theory 2nd utg., Sage Publications, Inc, 1998.

[St11] Stav, E., Walderhaug, S., Mikalsen, M., Hanke, S., Benc, I.,: Development and evaluation of SOA-based AAL services in real-life environments: A case study and lessons learned. In International Journal of Medical Informatics, 2011.

[SW11] Stav, E., & Walderhaug, S.,: universAAL Deliverable 3.1 A - Tools for design and development, 2011, Available at: http://www.universaal.org/ index.php?option=com_content&view=article&id=66&Itemid=15.

[Taz11] Tazari, S.,: UniversAAL Deliverable 2.1.A - universAAL Execution Environment Installation packages and hardware abstraction layer., 2012 Available at: http:// www.universaal.org/index.php?option=com_content&view=article&id=66&Itemid=15.

[VDB10] Van Den Broek, G, Cavallo, F. & Wehrmann, C.: AALIANCE Ambient Assisted Living Roadmap, 2010.