Towards Open Architecture System

Bahtijar Vogel
Linnaeus University
Vaxjo, Sweden
bahtijar.vogel@lnu.se

ABSTRACT

The use of diverse standards while developing web and mobile technologies brings new challenges when it comes to flexibility, interoperability, customizability and extensibility of the software systems. In addition, such systems in most of the cases are closed, thus make the development and customization process for system designers, developers and end-users a challenging effort. All these developments require further research attention. This work addresses these challenges from open system architecture perspective. The proposed approach is based on practical development efforts, and theoretical research including state of the art projects and definitions related to open architectures that we surveyed. The initial results indicate that a combination of service-oriented approaches with open source components and open standard data formats pave the way towards an open, extensible architecture. The core contribution of this research will be (a) an open architecture model and (b) the developed system itself based on the model, and (c) the benefits of applying open architecture approaches throughout the development processes.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architecture

General Terms

Theory, Design, Documentation

Keywords

Open architecture, web and mobile software, flexibility, evolvability, customizability, extensibility, model, validation

1. INTRODUCTION AND MOTIVATION

Use of diverse standards while developing web and mobile technologies brings new challenges when it comes to flexibility, interoperability, customizability and extensibility of the software systems. For e.g., both web and mobile software development is becoming fragmented with the existence of multiple browsers (that differently comply with web standards) and diverse platforms (operating systems and programming languages) and devices.

Research in web technologies indicates that open systems are more successful than the ones based in proprietary technologies [2], [15]. Moreover, it is evident that in web application development the computation is shared between the server-side and client side [15] where client is gaining more importance in producing richer web application experience. Open systems based on a modular structure and open interfaces can tackle the issues identified above [9]. Henderson [9] refers to these systems as open architecture systems.

Through the evolution of web and mobile software together with the changes of the development technologies, the emerging dynamic requirements become even more challenging to be addressed. Therefore, software flexibility and customization becomes necessary in order to modify a system to better suit user requirements for performing a specific task [13].

Motivated by these trends the aim of this research is to identify novel ways to model, design, implement, customize and deploy web and mobile software tools based on open system architecture. The main challenges to be addressed in this research are formulated as follows:

- Across all stages of the software lifecycle the fragmentation affects the costs and time, in particular the fragmentation of mobile software.
- Addressing dynamic requirements in heterogeneous environments, especially through evolution of web and mobile software.
- Integration issues between new components/modules and external systems to challenge and expand the system architecture for/in diverse contexts.
- Addressing the architectural needs for deploying flexible software tools by considering the rapid evolution of web and mobile technologies.
- In order to validate the architecture, model and tools, there is a need to identify an evaluation methodology.

In this research, we particularly address and analyze the main outcomes of our efforts from the perspective of open architecture. The system openness is tackled from an open architecture perspective for web and mobile software system taking into consideration the non-functional characteristics. All these efforts evolved from the utilization of open source components and open data formats that led towards the notion of open architecture.

Thus, the questions that guide this research are formulated as follows: What characteristics are: (a) needed to address the above-mentioned challenges? and are (b) important to model an open architecture system?

2. TOWARDS OPEN ARCHITECTURE

"Every application has architecture and an architect," moreover, "architecture is not a phase of development" [16]. Architecture is defined as: "the set of principal design decisions made about a system; it is a characterization of the essence and essential of the application" [16]. In the scope of this research the openness of the architecture is related to the ability of the system to grow in terms of new services, devices, and subsystems attached to it.

2.1 Progress, Approach and Settings

During the last four years of development efforts conducted as a part of this research, three software prototypes were implemented

ESEC/FSE'13, August 18–26, 2013, Saint Petersburg, Russia
Copyright 2013 ACM 978-1-4503-2237-9/13/08...$15.00
http://dx.doi.org/10.1145/2491411.2492407

utilizing service-oriented approaches. The software system has gone through evolutionary prototyping approach (see [8]) to become a stable and robust platform for mobile data collection, aggregation and data visualization. These efforts have been developed in relation to the Learning Ecology through Science with Global Outcomes (LETS GO) research project. One of the aims of the project is to support "open inquiry learning" using mobile science collaboratories that provide open software tools and resources, and participation frameworks for learner project collaboration, mobile data and media capture, publishing, analysis, and reflection. The architecture approach [19] that guided the design and implementation of our software prototype proved to be sustainable and expandable architecture that enabled us to meet the functional requirements. The system requirements and architectural design activities were followed within two phases, whereas the development approach was guided by following the web development life cycle and agile development (see [19]).

To describe the current status of the system architecture we provide a hint about the workflow of our software system, in a form of an activity diagram[1]. The developed system is modular and layered architecture, based on service-oriented approaches. Furthermore, since the overall evolution of our software system was also guided from the end users perspective, it has been tested with more than 500 users during this period. The end users feedback was an important input throughout the software development lifecycle. The latest software prototype developed in this manner is based on well-established approaches such as component-based and service-oriented approaches. Furthermore, the system architecture emerged from this process utilized open source components and open standards data formats [19], [1]. Even that the non-functional requirements were not explicitly addressed by end-users, they became central for identifying characteristics in relation to the notion of open architecture. These initial results paved the way toward the development of the open architecture model from the perspective of non-functional characteristics. Thus the remaining research efforts that need to be conducted are connected toward the validity of our proposed model and system architecture.

2.1.1 Surveying Process

A surveying process was a need to synthetize the existing definitions and studies to better understand what open architecture means and how to achieve a system based on this notion. Therefore three surveying processes[2] were performed. First process was related to definitions of open architectures. Second process led us with identification of the state of the art projects in this area. The first and the second processes led us to identify some non-functional characteristics in connection to open architecture. Therefore there was a need to continue with the third process to survey definitions of these characteristics in order to deepen our understanding in this study. The survey process led to general understanding of the definitions that an open system should be flexible and interoperable so users can design and customize the system in order to suit individual requirements. In a nutshell, some of the main concepts identified are: flexibility, modular approach, customization, extensibility, interoperability, individual requirements and standards. Inspired from the definitions and the key concepts identified, a couple of research projects are included as state of the art projects that deal or contain open system architecture.

2.2 State of the Art Projects

An open modular architecture is being proposed by [2]. This open architecture combines several emerging and established technologies in order to provide tools for quickly developing prototypes of virtual worlds based on the Web. Their system is flexible due to the modular approach, which allows replacing components easily. Another implementation of an open architecture is based on service-oriented approach specifically addressed for the design of virtual organizations [3]. Cloud Computing Open Architecture (CCOA) that integrates also service-oriented architecture is presented by [21]. Cavuşoğlu et al. [4] presents architectural details of an evolving open source/open architecture software framework in the health field. Developing mobile location-based applications over the Internet also contains an open architecture [10]. Interestingly, space agency NASA uses an open architecture component-based software tool for the development, integration and deployment of mission operations software, with the challenge to integrate multiple applications into a single platform [11]. Oreizy [13] proposed a new software customization technique called open architecture software, a flexible approach to decentralized software evolution. His comparison framework is based on the concept of software open points where independent third-party developers change a software system by changing its architecture. One of the latest instantiations of a system based on an open architecture is a project called mHealth [7]. Their aim is to create an open mHealth ecosystem of reusable, substitutable modules of basic functionalities, consisting of data exchange standards where existing and new systems would be interoperable.

Above-mentioned research projects have been partial or work in progress efforts. They contributed by advancing the state of the art in the area of open systems architecture. Nevertheless, in none of these approaches the end-users and system designers are not fully considered. There are no clear extensibility approaches when it comes to make a system interoperable and integratable with other systems. An important aspect in this area is fragmentation, especially due to the fast evolvement of different technologies/platforms, where none of projects above address fragmentation issues (which lately is a huge challenge to be tackled especially in the mobile software). Yet there is no clear characterization from systems design processes that could benefit the system designers and developers. Moreover, also from end-user perspective in terms of engaging them to design, deploy and customize their own tools especially in the era of Open Web [15]. In summary, in this area there is lack of research efforts related to:

- Fragmentation issues that differently comply with web standards and diversity of platforms.
- Addressing new and dynamic requirements to be used in different contexts.
- The opportunities from systems design processes from which system designers and developers could benefit from.
- Customization process from end-users perspective and not just developers.
- Integration and Interoperability issues between web and mobile technologies.

Therefore, this work tries to also identify characteristics of open architecture, and provide insights towards building the open architecture system for web and mobile software that can be used in diverse contexts.

[1] Activity Diagram: http://tinyurl.com/cw6ny2b

[2] Surveying Process Flow: http://tinyurl.com/bt9zb63

2.3 Non-functional Characteristics

During the system development, non-functional characteristics are essential for architectural design and planning the quality aspects of the software [5], [6]. The non-functional characteristics are identified throughout this research and development process and are presented as characteristics of open architecture, mainly: **Flexibility, Evolvability, Customizability and Extensibility**. In order to have a clear understanding of these characteristics the third surveying process was performed about the definitions of these characteristics (see footnote 2). The identified definitions provided us with valuable insights by deepening even more our understanding towards the notion of the open architecture. Thus below we enlist the re-defined characteristics (notions) and interpret them into our own context in the synthetized manner:

- A **Software System is considered flexible** if it provides a system that can be used by users in wide variety of settings and situations by easily addressing different user and system requirements.
- A **Software System is considered evolvable** if the system can easily be modified to meet new, individual and dynamic requirements by minimizing costs.
- A **Software System is considered customizable** if it allows users to customize features in the system and to address their specific individual needs, usually without having access to source code.
- A **software system is considered extensible** if it offers integration possibilities with other systems and/or tools that takes into consideration the future growth by expanding/enhancing the system architecture, allowing other developers to add/modify new functionalities, also from the perspective of other projects.

Thus motivated by these notions and to better clarify them, we put the non-functional characteristics in context by structuring them into a model with some initial artifacts/attributes and/or constraints.

2.4 The Initial Model

The non-functional characteristics can play a crucial role when designing and developing a software system based on open architecture. Moreover, they are critical for the success of the project [8]. We thus present the non-functional characteristics as attributes of a system or constraints [8]. To better clarify the non-functional characteristics, we are introducing an initial conceptual model that provides an abstract depiction of the open architecture model[3]. This model utilizes aspect-oriented representation [8], which in our case is based upon the four key non-functional characteristics - flexibility, evolvability, customizability and extensibility – followed with artifacts that flow between them. Each of the attributes/artifacts and/or constraints are structured starting from the main object, which than is expanded with the four key non-functional characteristics. Afterwards, they are grouped as: requirements, actors, contexts, operations, systems, and measures.

2.5 Goals

Reflecting upon our current development efforts in relation to the case presented in section 2.1, these definitions (section 2.3) are closely connected. The notion of flexibility and evolvability enabled our software system to be used in a wide variety of settings and for different activities (for instance activities related to soil quality, water quality, geo-audio notes) [19, 20]. Furthermore the notion of customizability in our case has been tackled by utilizing open source components and open data standards [18, 19], these made our system to easily address needs for different settings, that substantially have shortened the development time [18, 19]. The notion of extensibility in our case was manifested with the extension of our solution with other existing and new tools and robust systems, for example making use of web APIs. One of our latest activities is related with the integration of our software system with National Geographic Society's (NGS) FieldScope GIS tool, also with the easy extension of our software with Microsoft Surface table [20]. Our software system is also extended to other robust systems such as with other web-based systems/tools [12].

The goals of this research are based and motivated from the identified non-functional characteristics of the open architecture introduced earlier and included in the model. Thus, these include the challenges that rose in introduction and motivation, and deficiencies elaborated in state of the art projects and our own development case. Hence the goals to be addressed are:

GOAL 1: Provide an Open Architecture System to explore the possibilities for offering flexible and evolvable software system for system designers, developers and end users in wide variety of contexts.

GOAL 2: To analyze the extensibility possibilities of Open Architecture System with other systems and/or tools/modules by expanding/enhancing the system architecture, allowing other developers to add/modify new functionalities, also from the perspective of other projects.

GOAL 3: To analyze the possibilities of Open Architecture System to allow users to customize features for addressing their needs for different settings.

Demonstrating that the goals stated have been met, we need to follow methodological steps to validate them.

2.6 Validation Plan

Measuring software processes and products is an important aspect throughout the development process (lifecycle) and evolution of the software system, which 'requires a measurement mechanism for feedback and evaluation' [17]. Since this research is goal oriented, the validation process is based on Goal, Question, Metric (GQM) approach [17]. GQM paradigm deals with goal-oriented measurement, 'defines a certain goal, refines this goal into questions, and defines metrics that should provide answers to these questions' [14]. With this approach metrics for measurements are defined from a top down perspective, and from bottom-up the data is analyzed and interpreted [17]. Therefore, the goals of this research follow specific methodological steps from GQM in three levels [14]:

Conceptual level: A GOAL defined for an (1) Object, with a (2) Purpose and (3) Focus, from various (4) Viewpoints and from particular (5) Environment**;**

Operational level: QUESTIONS that characterize the object of measurement and the achievement of the goal to be performed**;**
Quantitative level: to identify METRICS with every question in order to answer it in quantitative way.

We already arranged the goals using GQM approach; questions and metrics are underway. We posses logs from our previous studies from which we have data to be analyzed. As a final effort

[3] An Initial Model: http://tinyurl.com/bocdq3u

we plan to conduct a study in multiple steps using our earlier scenarios. We plan to collect data from logs; the study will cover tests, thus a number of tasks will be introduced to both end-users and developers. Based on this, questionnaires will be distributed to both parties. Finally, we use GQM as a mechanism to interpret the data collected and understand if we reach the goals in operational and measurable way.

3. EXPECTED CONTRIBUTIONS

The idea of this research is to identify novel ways to model, design, implement, customize and deploy web and mobile software tools based on open system architecture. We aim at modeling and producing system architecture based on non-functional characteristics. Upon validation the model introduced provides an abstract depiction of the open architecture system that can guide the design and development of the architecture based on these non-functional characteristics. We strongly believe that the importance of stressing these non-functional characteristics into a model can play a crucial role when designing and developing web and mobile software system based on open architecture, from which system designers, developers and end-users will benefit. Therefore, the above-mentioned model helps addressing emerging requirements for an open, extensible architecture for deploying web and mobile software systems across heterogeneous environments. In summary, the expected contribution of this research could be formulated as follows:

- Identifying proper approaches for addressing dynamic requirements in heterogeneous environments.
- Developing an instance of the open system architecture that reflects upon the non-functional characteristics that relies on flexibility, evolvability, customizability and extensibility.
- Decreasing the fragmentation impact between different platforms / applications / devices with open system approach.

4. REFERENCES

[1] Anokwa, Y., Hartung, C. and Brunette, W. 2009. Open source data collection in the developing world. October (2009), 97–99.

[2] Byelozyorov, S., Pegoraro, V. and Slusallek, P. 2011. An Open Modular Architecture for Effective Integration of Virtual Worlds in the Web. 2011 International Conference on Cyberworlds (Oct. 2011), 46–53.

[3] Carrascosa, C., Giret, A., Julian, V., Rebollo, M., Argente, E. and Botti, V. Service Oriented MAS : An open architecture (Extended Abstract). 1291–1292.

[4] Cavuşoğlu, M.C., Göktekin, T.G. and Tendick, F. 2006. GiPSi:a framework for open source/open architecture software development for organ-level surgical simulation. IEEE Engineering in Medicine and Biology Society. 10, 2 (Apr. 2006), 312–22.

[5] Chung, L., Cesar, J. and Leite, P. 2009. On Non-Functional Requirements in Software. A.T.B. et al. (Eds.):, ed. Springer-Verlag Berlin Heidelberg. 363–379.

[6] Doerr, J., Kerkow, D., Koenig, T., Olsson, T. and Kaiserslautern, G.- 2005. Non-Functional Requirements in Industry – Three Case Studies Adopting an Experience-based NFR Method. Proceedings. 13th IEEE International Conference on Requirements Engineering (2005), 373–382.

[7] Estrin, D. and Sim, I. 2010. Open mHealth architecture: an engine for health care innovation. Science(Washington). (2010).

[8] Glinz, M. 2007. On Non-Functional Requirements. 15th IEEE International Requirements Engineering Conference (RE 2007). (Oct. 2007), 21–26.

[9] Henderson, P. 2009. Modular Open Systems Architecture. April (2009), 1–15.

[10] Josc, R., Moreira, A., Meneses, F. and Coulson, G. 2001. An Open Architecture for Developing Mobile Location-Based Applications over the Internet. (2001), 500–505.

[11] Lindsey, A.E. 2009. Component-Based Tool for Mission Operations Software Deployment. (2009), 1–8.

[12] Milrad, M., Kohen-Vacs, D., Vogel, B., Ronen, V., M. and Kurti, A. 2011. An Integrated Approach for the Enactment of Collaborative Pedagogical Scripts Using Mobile Technologies," Proceedings of the International Conference on CSCL 2011, Hong Kong, China.

[13] Oreizy, P. 2000. Open architecture software: a flexible approach to decentralized software evolution. PhD Thesis.

[14] Solingen, R. Van and Berghout, E. 1999. The Goal/Question/Metric Method: a practical guide for quality improvement of software development. McGraw-Hill Publishing Company.

[15] Taivalsaari, A. and Mikkonen, T. 2011. Objects in the cloud may be closer than they appear towards a taxonomy of web-based software. 2011 13th IEEE International Symposium on Web Systems Evolution (WSE). (Sep. 2011), 59–64.

[16] Taylor, R. N., Medvidovic, N. and Dashofy E.M. 2009. Software Architecture: Foundations, Theory, and Practice, Wiley, John & Sons, USA.

[17] Victor, B.R., Gianluigi, C. and Rombach, H.D. 1994. The goal question metric approach. Encyclopedia of Software Engineering. 2, (1994), 1–10

[18] Vogel, B. 2011. An Interactive Web-Based Visualization Tool: Design and Development Cycles. 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops (Jul. 2011), 279–285.

[19] Vogel, B., Kurti, A., Milrad, M. and Mikkonen, T. 2012. Architectural Concepts: Evolution of a Software System Across Design and Implementation Stages in Dynamically Changing Environments. 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops (2012).

[20] Vogel, B., Kurti, A., Milrad, M., Johansson, E. and Müller, M. 2012. Mobile Inquiry Learning in Sweden: Development Insights on Interoperability, Extensibility and Sustainability of the LETS GO Software System," submitted to Special Issue of "Powering Up: Insights from Distinguished Mobile and Ubiquitous Learning Projects across the World" Journal of Educational Technology & Society, on December, 2012.

[21] Zhang, L.-J. and Zhou, Q. 2009. CCOA: Cloud Computing Open Architecture. 2009 IEEE International Conference on Web Services. (Jul. 2009), 607–616.