SIMPLE EVOLUTION OF COMPLEX WEB SYSTEMS

Lech Krzanik
University of Oulu, Department of Information Processing Science
Box 3000, FIN-90014 Oulu, Finland

Keywords: Web system evolution, pervasive context sensitive applications, user-participative development.

Abstract: Usability implies effective, efficient and satisfactory interaction with a system when *using* it. Similar properties are not usually considered when an end user is faced with evolutionary *developing* or improving a system – an activity increasingly more often, particularly when the context of using a system undergoes frequent and significant changes. Project Nomadic Media investigated possibilities for user-participative, effective, efficient and satisfactory evolution of nomadic systems. We considered geographically distributed, context-sensitive, personalized systems, exploiting various web services and allowing for a general function referred to as nomadic blogging. In this paper we demonstrate general policies of evolution and the specific ways of end-user interaction with the evolving system. The overall strategy may be generally described as simplicity – effective with clear and straightforward goals, efficient in terms of resources spent at each small evolutionary delivery step and satisfactory in terms of end-user participation and reaching the explicit planned user targets. Results of experiments with such systems are demonstrated.

1 INTRODUCTION

We investigate user-participative evolution of web systems characterized as follows. Complex web systems are considered including federated context-sensitive pervasive local subsystems linked to global web services that contribute to the local behaviour. In this environment various types of local and distributed businesses are supported as well as a range of personal and social links and structures. Both server and user sides of the system are considered, and the users interact with the system by using a broad range of devices, mobile and stationary. The main goal for such a complex web system is to coordinate the offered services in a way which assures that the users may freely change their physical locations geographically and structurally - between various business points at a location, between locations within an environment, between environments, etc. This type of systems originated from the investigation of media systems for nomadic users, later extended to cover for various kinds of businesses, whether dealing with media directly or indirectly. The domain was investigated and validated in project Nomadic Media. The mission of the project was to "find solutions that allow consumers to enjoy their content and use interactive services at the times and in the places they prefer, using the devices that best suit their circumstances." The domain is currently characterized by considerable uncertainty of requirements, technologies, markets and regulations. The evolutionary strategy has therefore been selected for iterative and incremental development of systems. An important assumption is to include the end users, possibly belonging to various stakeholder classes, in the evolutionary feedback loop. Can we make such interactions simple enough to be acceptable for end users?

Usability implies effective, efficient and satisfactory interaction with a system when *using* it. Similar properties are not usually considered when an end user is faced with evolutionary *developing* or improving a system – an activity increasingly more often, particularly when the context of using a system undergoes frequent and significant changes. In order to systematically investigate evolution in such systems we introduce a general quality attribute referred to as *evolutionary simplicity*. The attribute has the interpretation of usability of evolution support. It uses similar sub-attributes – effectiveness, efficiency and satisfaction. Project Nomadic Media investigated possibilities of user-participative effective, efficient and satisfactory

Krzanik L. (2007).
SIMPLE EVOLUTION OF COMPLEX WEB SYSTEMS.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 340-346
DOI: 10.5220/0001288803400346

evolution of nomadic web systems, allowing for a general function referred to as nomadic blogging. In this paper we demonstrate several policies supporting evolution and allowing for various ways of end-user interaction with the evolving system. They should be effective with regard to clear and straightforward goals, efficient in terms of resources spent at each small evolutionary delivery step and satisfactory in terms of reaching the explicit planned user targets. We introduce three policies, respectively: For effectiveness it is the communicative specification (CS) policy. For efficiency it is the minimalist architecture (MA) policy. For satisfaction it is the feedback and cooperation (FC) policy.

Sections 2 and 3 describe the approach to evolution and the quality attributes. Section 4 presents the policies. Section 5 demonstrates selected results. Sections 6 and 7 discuss other work and formulate conclusions.

2 EVOLUTION

Evolutionary approaches to software development are well suited for projects of various sizes and under considerable uncertainty of requirements, technologies, market and regulations (Larman 2003, Madhavji et al. 2006). An evolutionary project necessarily consists of two interleaving processes of variation and selection. The former delivers variant instances of the system under consideration, e.g., a mobile service system, while the latter selects the instance which is best fit and most responsive to change. Software evolutionary projects are different from natural evolutionary systems in the sense that they have considerably accelerated pace, controlled variation, and the selection occurs within a limited population of specimen – hence neither the variation is as random nor is the selection as natural. Therefore even more attention must be attached to efficient and well-targeted ways of generating variants and assessing them by real stakeholders. Of most importance is such stakeholder communication which assures a successful selection process in a community of many stakeholders – that is the selection process which is the focus of this paper. An overview of the evolutionary selection process is shown in Figure 1.

Thanks to the embedded handling of variability, software product lines (Bosch 2000, Pohl et al. 2005) offer a suitable way of handling the variation process of evolutionary projects. An example implementation from Nomadic Media is shown in Figure 2.

With the focus on quality requirements, we consider the attribute-driven evolution. In the considered project the foreground quality attribute was usability – in the conventional interpretation of ISO/IEC (1998; Nielsen, 1993; Faulkner, 2000). Improvements in system qualities such as usability may result in new function opportunities – as in the demonstrators discussed in next sections. The overall function of the considered systems was nomadic media management and specifically nomadic blogging (Blood, 2004; Krzanik, 2005).

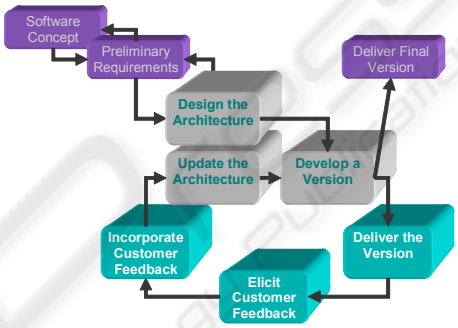

Figure 1: The iterative selection process of software system evolution.

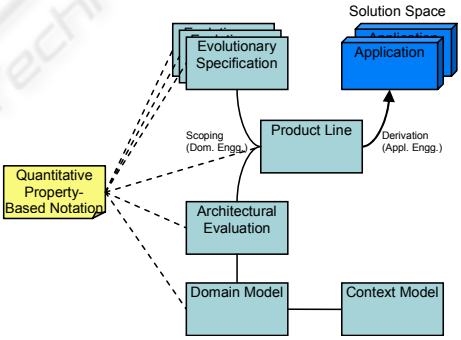

Figure 2: A product line implementation of the variaton process.

Evolution is tightly connected with the architecture of the evolving system. By architecture we mean a collection of architectural design decisions, in particular the evolution decisions, corresponding to the changing requirements and constraints. According to Jansen and Bosch (2005) architecture is the structured composition of a set of business-driven architectural design decisions. According to Gilb (2005) architecture is the set of design artefacts, which are selected or exist to impact a set of stakeholder requirements, by

constraining, or influencing, related systems engineering decisions.

3 QUALITIES OF EVOLUTION

Because of the likely fundamental role of end users in web systems evolution, we have investigated the hypothesis that the quality of evolution can be measured by the usability of the evolution support made available to the users. Usability defined by ISO DIS 9241-11 (ISO/IEC, 1998) refers to products rather than services; hence we reinterpret it as the extent to which a *service* (evolution support) "can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". By analogy to good product usability - meaning that users can easily perform the different tasks with the product - good evolution means that users can easily perform evolution tasks such as providing feedback and selecting evolution options. By analogy we consider the *evolution simplicity* attribute for user-participative projects, including the three main sub-attributes: effectiveness, efficiency and satisfaction.

Effectiveness measures the extent to which the evolution support is able to determine whether the selected or existing design artifacts provide contribution toward clear and straightforward goal specification - suitable for user interaction and supporting the evolution process. Effectiveness focuses on fundamental and critical system decisions (its architecture) and promotes easy communication between various stakeholders. Efficiency promotes accomplishing evolution tasks with least resources. Finally, the satisfaction attribute requires that the participation in the evolution process is a pleasure to the participating users. The measurement of the attribute should be based on the observations of stakeholder attitudes towards the evolution.

4 POLICIES FOR SIMPLE EVOLUTION

We propose three policies aiming at contributing to the three above attributes of effectiveness, efficiency and satisfaction respectively, for evolving complex web systems: communicative specification (CS), minimalist architecture (MA) and feedback and cooperation (FC). In the proposed configuration CS is the basic policy, MA and FC depend on it.

4.1 Communicative Specification

The goal of the communicative specification (CS) policy is to contribute to the effectiveness of the evolution processes. The solution provides clear and straightforward web system specifications, which are suitable for system evolution and can serve as easy communication media between various stakeholders. To support effective communication the specifications shall include elements of both (1) requirements and (2) design, (3) be able to define multiple stepwise requirement targets for an evolving system such that at each delivery step the process explicitly defines software architecture with the respective (4) impact on critical requirements so that evaluation of the effects of the steps is immediate. Moreover (5) quality of the specification can also be assessed. Thus the specification aggregates elements of both problem and solution spaces. The structure of the specification is shown in Figure 1. The policy assumes a three-partite structure of the requirement set including functions, and attributes (non-functional qualities), extended with features that associate the two.

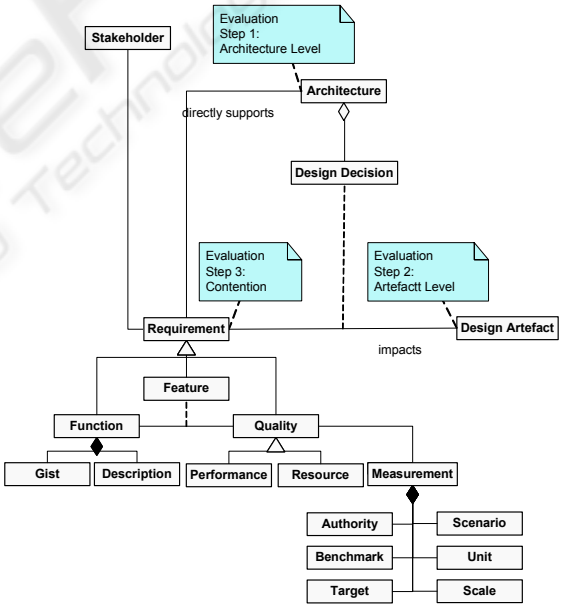

Figure 3: The structure of a communicative specification.

On the solution space side the respective software architecture is defined as a collection of architectural design decisions to meet the specified requirements (Jansen et al., 2005; Gilb, 2005). The approach includes online architecture evaluation, which allows for immediate reaction to stakeholder feedback. The prospective evaluation methods for

the policy are extensions of popular software architecture assessment methods such as SAAM and ATAM (Bass et al., 2003; Kazman et al., 2004) as well as innovative product line evaluation methods such as the BAPO method (van der Linden et al., 2004). Evolution can be expressed in terms of both the requirements and design decisions. The CS policy provides for a suitable support of the other proposed policies.

4.2 Minimalist Architecture

The goal of the minimalist architecture (MA) policy is to contribute to the efficiency of the evolution process. The solution is to optimize the size of the architecture. It is expected to allow for architectural decisions that are accomplished with the least resources, including as little effort as possible. For complex web systems software architecture may include thousands of architectural decisions. Optimal target architecture should match the current delivery steps and minimize the number of involved architectural decisions by including only active and critical decisions. Moreover not all the remaining decisions have to be dealt with explicitly and some can possibly be deferred until later in the iterative lifecycle. For example Malan and Bredemeyer (2002) argue that as few decisions as possible should be actually made.

4.3 Feedback and Cooperation

The goal of the feedback and cooperation (FC) policy is to contribute to the satisfaction attribute of the evolution process. The solution is to provide suitable media to support (a) feedback collection, (b) formulation and evaluation of candidate and final architectural decisions, and (c) cooperative decision-making, with regard to new or revised decisions. The support is arranged so as to promote continued and enhanced participation in the evolution activity. The aim is to ensure that the stakeholders have positive feelings towards the activity. The solution we propose is based on visualization and sharing of the architectural specifications and on providing access to evaluations which adapt and enhance selected known assessment methods.

5 DEMONSTRATORS

We consider a complex web system including federated context-sensitive pervasive local subsystems linked to global web services that contribute to the local behaviour. In each of the local environments various types of local and distributed businesses are supported as well as a range of personal and social links and structures. The users interact with the system by using a broad range of devices, mobile and stationary. The mission of the project Nomadic Media has been to “find solutions that allow consumers to enjoy their content and use interactive services at the times and in the places they prefer, using the devices that best suit their circumstances.” The content may be linked to individuals, groups, local or distributed businesses, etc. The domain of such an exploratory project is characterized by considerable uncertainty of requirements, technologies, markets and regulations. Hence the evolutionary strategy has been selected for project deliveries. We considered a user-participative evolution process. Figure 4 shows the architecture of a federated context-sensitive pervasive local subsystem. The evolution of both server and user sides of the system were considered.

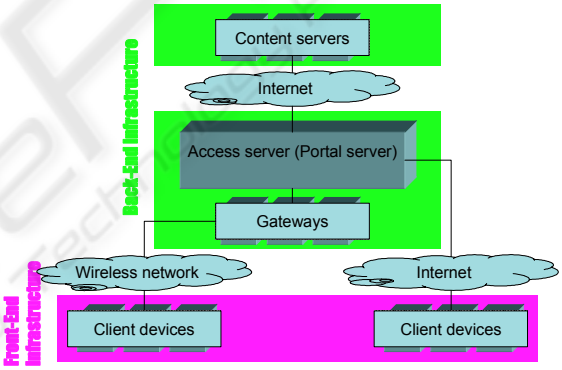

Figure 4: A multi-tier architecture of the federated context-sensitive pervasive local subsystem for nomadic blogging applications.

There were two variants of the subsystem that provided support for user interaction for system evolution: (I) One was based on automated regular end-user interaction analysis and evolution planning and evaluation facility intended for system architects and developers only. That system did not allow all regular end-users to directly access the evolution support function and was used as a benchmark for (II) the other variant which provided direct evolution support for all important categories of stakeholders. The implementation of both variants was based on a cooperative wiki engine that followed the FC policy. Not all devices used in the system had equal access to all features (a)-(c) (cf. section 4.3) of the evolution support applications. The results have shown that variant (II) performed significantly better

in terms of the introduced evolution criteria (conventional product usability). Detailed analyses of these and other results, also taking into account the exploratory nature of the project, are currently in progress. A relatively stable nomadic blogging functionality (Blood, 2004; Krzanik, 2005) had been defined and a product line platform, the PORO system (Krzanik et al., 2005), implementing the variability process for context-sensitive nomadic blogging has been built. It has been used for systematic evolution of nomadic media systems. Examples of the functionality are visualized in Figures 5-6.

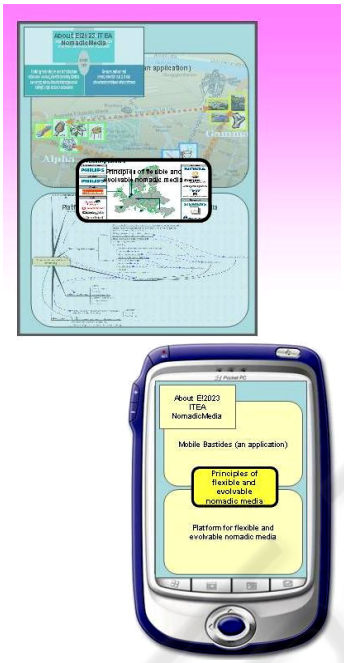

Figure 5: Public screen control: An application derived from the nomadic blogging functionality.

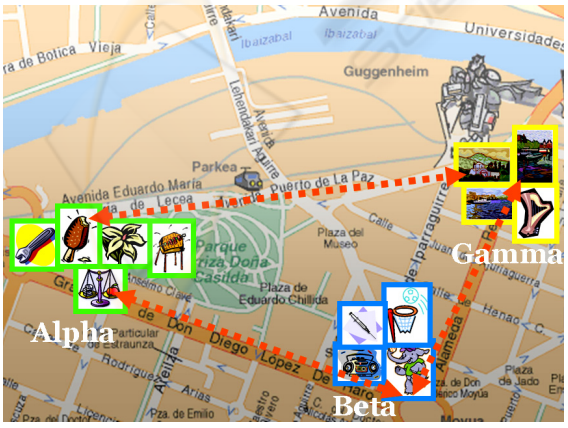

Figure 6: Virtual city: An application derived from the nomadic blogging functionality.

The evolution experiments followed an attribute-driven strategy with usability as the most critical driver. Six different experiments and deliveries are illustrated in Figure 7. The experiments demonstrate the enabling nature of usability: the incremental quality improvements not only lead to better systems, but also were instrumental in discovering new functional opportunities visualized in Figure 7. It should however be stressed that the functionalities of all systems shown in Figure 7 were essentially very close to one another. In the evolution paths the sub-attributes of efficiency, effectiveness and satisfaction were main drivers. It is not yet known to what degree the inventions of new functions are to be attributed to the learning process associated with using the evolution support.

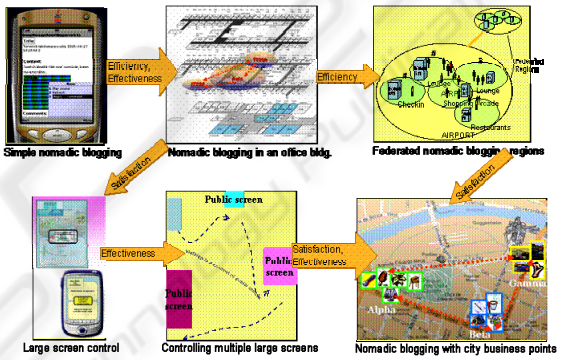

Figure 7: Generating application variants driven by usability and functionality changes.

We found that the domain vocabularies, associated with requirements and design engineering (and the revised orthogonal variability model included in PORO), were critical in supporting the stakeholders in the evolutionary process. Vocabularies as well as other introduced support services and their positive effect on the process demonstrated the importance of explicit consideration of stakeholder communication. Various investigated applications were integrated into innovative packages such as the Mobile Bastide package (Krzanik, 2005) implemented as a dedicated product line for nomadic media.

6 RELATED WORK

The main elements of the proposed approach include the user-participative evolution, the tailored approach to usability and the innovative approach to software architectures. The revised approaches to architecture assessment, to stakeholder cooperation

in project evolution, the exploratory wiki application and the innovative nomadic blogging functionality are all in a sense consequences. All these introduce a number of changes to existing practices.

The user-participative evolution is essentially a voluntary activity. Not only it differs from conventional approaches (Madhavji et al., 2006) that establish routine evolutionary processes but also requires specific approaches which promote user cooperation – in our case through improved usability of the evolution support function. Our architectural approach is based on the decision-oriented variant of architecture definition (Jansen and Bosch, 2005; Gilb, 2005), and explicitly integration of problem and solution spaces - different from conventional approaches to software architecture (IEEE 1471, 2000). It has had far-reaching consequences for architecture assessment methods in our evolution support – the possibility for evaluating architectures with regard to both requirements and design artefacts, in terms of end-user characteristics, e.g., system properties such as usability. The revised methods could be used to equip the collaboration tool such as wiki with more support for various stakeholders as compared with more conventional approaches, e.g., by Bachmann and Merson (2005). The nomadic flavour created many new interesting applications of the conventional blogging function (Blood, 2004) but perhaps more interesting is the innovative exploratory and evolutionary way in which we propose to discover, analyse and validate these new applications.

On the implementation side the web and pervasive solutions tend to use conventional technology (Hansmann et al., 2004). On the other hand, with reference to the conventional model of product line engineering (Pohl et al., 2005), we introduce a revised orthogonal variability model that better supports variation and selection processes of evolutionary development for software systems. Another proposed change was a closer cooperation between the processes of domain and application requirements engineering. We also offload some non-functional attribute process areas from product management and shift them to requirements engineering. Consequently, the product roadmap artefact that links the two process areas (Pohl et al., 2005) changes respectively. On the other hand product management takes responsibility of more extensive evolutionary variation support. It is also proposed that responsibilities regarding other new process areas, such as evolutionary selection support, are divided between product management, integrated requirements engineering and the variability model.

7 CONCLUSION

Evolutionary approaches can be applied to software development projects of various sizes under considerable uncertainty of requirements, technologies, markets and regulations. In doing so we should carefully investigate issues connected with realizing the evolutionary strategy. Revisions to conventional approaches may prove necessary. We propose a user-friendly evolutionary methodology for complex web systems. The methodology is still in its validation stage. We illustrated the concept and proposed solutions with selected results from a recent project.

ACKNOWLEDGEMENTS

This work was partly supported by project E!2023 ITEA Nomadic Media.

REFERENCES

Bachmann, F., and Merson, P., 2005. Experience Using the Web-Based Tool Wiki for Architecture Documentation. *Technical Note* CMU/SEI-2005-TN-041, Software Engineering Institute, September 2005.

Bass, L., Clements, P., and Kazman, R., 2003. *Software Architecture in Practice*, 2nd Ed. Addison-Wesley

Blood, R., 2004. How Blogging Software Reshapes the Online Community, *Comm. ACM*, 47, Nr. 2.

Bosch, J., 2000. *Design and Use of Software Architectures*. Addison-Wesley.

Faulkner, X., 2000. *Usability Engineering,* Macmillan Press, London.

Gilb, T., 2005. *Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*, Butterworth-Heinemann.

Hansmann, M., et al., 2004. *Pervasive Computing*, 2nd Ed., Springer.

IEEE 1471, 2000. ANSI/*IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software Intensive System.* IEEE, October 2000.

ISO/IEC, 1998. *DIS 9241-11. Part 11: Guidance on usability*. ISO/IEC.

Jansen, A., and Bosch, J., 2005. Software Architecture as a Set of Architectural Design Decisions. *5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005.* IEEE, pp. 109 – 120.

Kazman, R., Nord, R.L., and Klein, M., 2003. A Life-Cycle View of Architecture Analysis and Design Methods. *Technical Note* CMU/SEI-2003-TN-026, Software Engineering Institute, September 2003.

Krzanik, L., 2005. Mobile Bastides: Flexible and Evolvable Business-Oriented Contextual Media, *Proc. 11th Intl. VSMM Conf.*, Ghent.

Krzanik, L., et al., 2005. Usability-Driven Evolution of Context-sensitive Nomadic Media, *Proc. MUM'05 Conf.,* Christchurch.

Larman, C., 2003. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley.

Madhavji, N. H., Fernandez-Ramil, J., and Perry, D., 2006. *Software Evolution and Feedback: Theory and Practice*. Wiley.

Malan, R., and Bredemeyer, D., 2002. Less is More with Minimalist Architecture. *IT Professional*, Sept./Oct. 2002.

Nielsen, J., 1993. *Usability Engineering*, Academic Press, Inc., San Diego.

Pohl, K., Böckle, G., and van der Linden, F.J., 2005. *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer, Heidelberg.

van der Linden, F., Bosch, J., Kamsties, E., Känsälä, K., Krzanik, L., and Obbink, H., 2004. Software Product Family Evaluation. In: Frank van der Linden (Ed.), *Product Family Engineering. Proc. of PFE-5, Siena*, Springer Verlag, Heidelberg.

van Gurp, J., and Bosch, J., 2002. Design Erosion: Problems & Causes. *Journal of Systems & Software*, 61(2), pp. 105-119, Elsevier, March 2002.