Requirements Engineering for Pervasive Services

Lyubov Kolos-Mazuryk
Department of Computer Science
University of Twente
P.O. Box 217, 7500 AE Enschede
The Netherlands
lkolos@ewi.utwente.nl

Gert-Jan Poulisse
Department of Computer Science
University of Twente,
P.O. Box 217, 7500 AE Enschede
The Netherlands
g.j.h.m.poulisse@student.utwente.nl

Pascal van Eck
Department of Computer Science
University of Twente
P.O. Box 217, 7500 AE Enschede
The Netherlands
vaneck@cs.utwente.nl

ABSTRACT
Developing pervasive mobile services for a mass market of end customers entails large up-front investments and therefore a good understanding of customer requirements is of paramount importance. This paper presents an approach for developing requirements engineering method that takes distinguishing features of pervasive services into account and that is based on fundamental insights in design methodology.

Keywords
Requirements engineering, mobile services, context-awareness

1. INTRODUCTION

Advances in miniaturization of electronic systems have given rise to ubiquitous computing, in which computing power is made present everywhere and at all times in our environment. The goal is to blend this computing power into our environment so naturally that it becomes invisible. An example of this are pervasive mobile services, where fourth-generation mobile networks form the basis for the development of mobile, context-aware, personalized services offered to a mass-market of end consumers. It is expected that these services will blend into consumers' lives as computing devices will in ubiquitous computing.

Developing and deploying such large scale services amounts to significant investments for a service provider. Therefore, it is very important that before a service is developed, a thorough investigation of user requirements is undertaken. The process of finding and specifying these requirements for traditional information systems is known as Requirements Engineering (RE). Pervasive systems have a number of properties which distinguish them from traditional information systems, and these properties contribute to the fact that existing RE methods are inadequate for pervasive systems. The development of pervasive services therefore requires a completely new approach to requirements engineering, and to product development as a whole. In this paper, we propose a research approach for developing such an RE method for pervasive services. As a starting point, we list the most significant properties of pervasive computing systems. These properties illustrate how pervasive services are different from traditional software systems and thus also how they impact the development process. Our approach is then rigorously motivated by combining general theory on design methodology with the inherent properties of pervasive services.

In Section 2 we present and discuss distinguishing properties of pervasive systems. In Section 3 we present a research approach to the development of RE methodology for pervasive services development. Section 4 gives an overview of the current state of our project and summarizes the paper.

2. PROPERTIES OF PERVASIVE SYSTEMS

In order to select an appropriate approach to requirements engineering for pervasive computing systems, we first have to identify their distinguishing properties. Such properties usually have a significant impact on requirements engineering, and, thus knowing them is essential for choosing the correct method.

Researchers in the area acknowledge that one of the most important and distinguishing features of a pervasive service is context-awareness [1], which means that a system is able to observe its environment using some sort of sensor and that it is aware of this environment and is able to react to changes in it [2][3][4]. According to Chen [8], context is the set of environmental settings that either determines an application's behavior or in which an application occurs and is interesting to the user.

Given our viewpoint about the importance of context-awareness for pervasive computing, we classify other properties according to their relation to context-awareness. Namely, we distinguish two groups of properties – non-contextual and contextual.

2.1 Non-contextual properties

The following properties of pervasive systems are not contextual per se, but do distinguish them from more traditional systems, and have an influence on their requirements.

Shen [1] and Pascoe [14] suggest the following:

- Users can only pay limited attention to an application over a long period of time.
- User activity occurs in spurts.
- The time to market for a pervasive system is short. Projects are under a tight schedule and are confidential, since one doesn't want competitors to know the plans before introducing a product in the market. [1][17][18][19]

Krogstie [5] adds:

- Pervasive systems do not posses a common user interface standard.
- Pervasive systems generally possess small input and output devices (i.e. small screens and keyboards).

- Compared to traditional systems, pervasive services generally posses less memory, bandwidth, processing power etc [12].
- A pervasive service should always be immediately available for use [12].

2.2 Contextual properties

To differing degrees, Krogstie [5], Finkelstein [6] and Dix [7] all mention the following contextual properties of pervasive systems:

- Dynamic environment. This might be the fact that the location of the pervasive service can change, but also relative to other devices.
- Variable bandwidth. The application is capable of adapting to changes in network conditions such as disconnection or network medium depending on availability.
- Changing display characteristics. An application may have to run on a graphics PDA but also on a text-only mobile phone PDA.
- Changing user environment. Full-screen, mouse-operated laptop vs. scroll-centered mobile phone.
- The target platform is not known in advance, that is, an application should be able to dynamically adapt itself to the new context [6].

Many taxonomies of context have been proposed in research, several of which are mentioned here [5] [7] [8] [9] [10] [11].

Dey [9] and Schlit [11] suggest three categories.

- The computing context, such as the network infrastructure, input-output devices, and available processors etc.
- The user context, which includes the user location, location of nearby people, and social situation.
- The physical context such as the lighting, noise level, or temperature.

Chen [8] adds a time component to the above selection.

- Time context, such as the time of day, week, year etc.

Dix [7] and Rodden [10] instead use four categories

- Infrastructure context, defined as the device and the supporting infrastructure used to realize the application. This includes the network bandwidth, reliability, I/O device (display).
- System context, characterized by a device's awareness of other devices, applications, and users and the overall interaction of the (distributed) system as a whole.
- Domain context, the semantics of the application domain. This may include personalization, where an application is sensitive to a user's identity, and presents different information accordingly.
- Physical context, characterized by a device's awareness of its physical surroundings, such as knowing it is embedded in a car or mobile phone. This also includes a device's awareness of its location.

However, the one that seems the most encompassing is the one proposed by Krogstie [5]:

- The spatio-temporal context, which describes attributes like time, location, speed, direction, and the social arena.
- The environment context, which describes entities around the user such as services, temperature, noise, persons, and networks.
- The personal context, which describes the user's physiological and mental states.
- The task context, which describes a user's explicit goals, tasks and actions.
- The social context, which describes the social aspects of the user, such as information about friends and relatives, as well as a user's role, such as 'at work'.
- The information context, which describes the global and personal information space which is available.

De Heer [13] lists similar points, just without the formal framework of taxonomy.

Unfortunately this scheme verges on being too specific. One would suspect that some points such as the personal and social context could be grouped into one category, such as into the single domain context of Dix and Rodden's schemes. Nonetheless, the latter two taxonomies seem to be the ones that are considered in research today.

The utility in creating a taxonomy of context is that it simplifies the analysis on a contextual property's impact on the requirements process, assuming that each class has a different impact on the requirements.

We consider the most suitable context taxonomy the one proposed by Krogstie, which will be used further throughout the project, as explained in section 3.2.2.

3. REQUIREMENTS ENGINEERING

Despite the rich variety of existing approaches to the elicitation of user requirements, none of them has a complete set of techniques suitable for the development of mobile information systems and pervasive services in particular [21]. A preliminary study [22] showed that a new requirements engineering method is needed, which will provide a means to accommodate the above mentioned properties of pervasive systems.

3.1 Research Approach

Since we are designing a certain artifact – a requirements engineering method for pervasive services, we base our research process on the design research approach [18].

The first step in the design research model is obtaining awareness of a problem. This reflects the "problem-solving" nature of the activity. Suggestions for a problem solution are proposed from existing knowledge/theory bases for the problem area.

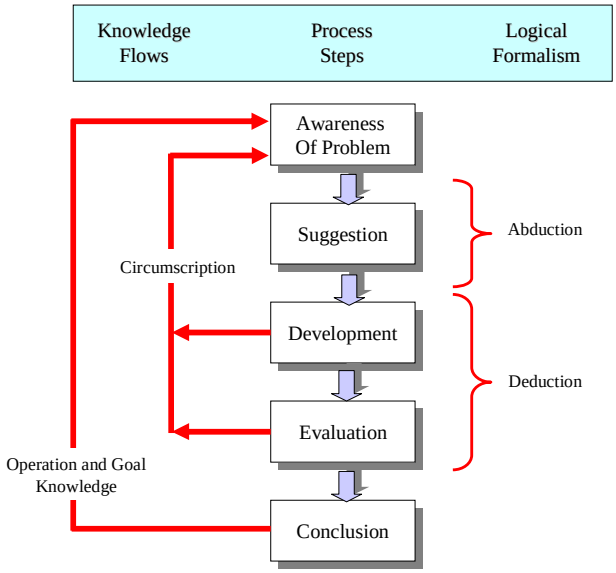

Figure 1 Design Research Process (source: [21])

Implementation of an artifact according to the suggested solution is the next step.

This stage is shown as Development in Figure 1. Implementations are then evaluated (according to the functional specification implicit or explicit in the suggestion). Development, Evaluation and Suggestion stages are frequently performed iteratively in the course of the research. The basis of the iteration, the flow from partial completion of the cycle back to Awareness of the Problem, is indicated by the Circumscription arrow. Conclusion indicates termination of a project.

Circumscription is a formal logical method that assumes that every fragment of knowledge is valid only in certain situations. Understanding of this process is important for understanding design research as a whole. According to [20], circumscription gives a kind of understanding of the problem, which can only be acquired from practical construction of a solution. In another word, due to the incompleteness of any knowledge base, design researcher encounters situations in which things do not work as described in theory. These situations force the research process back to the early stages (Problem Awareness and Suggestion), and in such a way valuable knowledge constraints are being discovered and taken into account during further design attempts.

Other research methods examined were action research [23] [24], Nunamaker and Chen's approach [25] and Hevner's framework [26].

Hevner et al [26] present a framework for conducting design (action) research. According to the authors, design research requires creation of innovative and useful artifacts within a specific domain. They emphasize the evaluation of developed artifacts, in order to confirm their usefulness and efficiency. Developed artifacts must be rigorously defined, formally represented, coherent and internally consistent. The research process (construction of the problem space and defining mechanisms for finding an effective solution) is iterative.

All the methodologies mentioned above are related, and define a more practical approach to research. Our method adopts a hybrid approach, based on design research. The search for a suitable requirements engineering method for service development itself suggests a design research approach.

The set of requirements to the research process defined by Hevner has the potential to complement the chosen methodology of design research by selective adoption of those Hevner requirements. According to the design research approach, a number of iterations have to be performed. Our method defines three types of iterations (project phases): "Domain Analysis and Problem Definition", "Reflection", and "Methodology Construction". These phases are related to the process steps in Fig. 1 as will be discussed at the end of this section.

In the "Domain Analysis and Problem Definition" phase our primary focus is on problem awareness activities (see Figure 1). Very few suggestion/development/evaluation tasks are performed. Those that are performed aim to better understand the problem domain. For example, small experiments are performed to see how existing requirements engineering approaches can accommodate service development.

The "Reflection" phase concentrates on evaluation and conclusion of the project as a whole (see Figure 1). Knowledge gained during the course of the research is summarized and analyzed, and directions for possible spin-off research activities are defined. The developed requirements engineering approach is verified and validated.

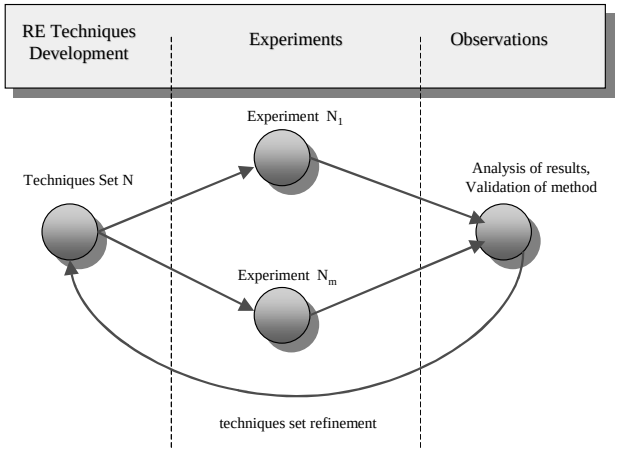

Figure 2 Research Process – Methodology Construction Phase

In the "Methodology Construction" phase focus shifts to the development of the requirements engineering methodology. The main activities during the research iterations of the second phase are depicted in Figure 2.

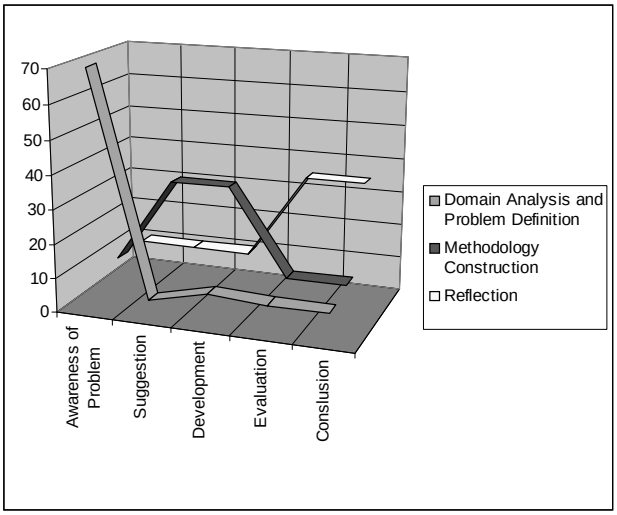

Figure 3 Shares of research activities (per phase)

At this moment, no formal boundaries between the phases have been defined. Generally, the beginning of a new phase can be detected by a shift in emphasis of research activities as can be seen from the approximation of a project workload in Figure 3. Informally, the criterion for the first phase change is when the problem is well enough understood to start the construction of the RE method. For the second phase change, the RE method should satisfy a number of criteria, which are determined in the problem analysis phase.

3.2 Constructing an RE Method

The previous section described our project methodology. In this section we describe how we expect to realize the RE method.

Throughout the project we will perform a number of case studies in order to elicit the requirements for a number of mobile, context-aware, proactive and personalized services.

Too many variables affect the measurement of the usability of various requirements elicitation techniques. Therefore the most trustworthy results are obtained by the practical validation of the mentioned techniques. Prior to iteratively applying our design research approach [18] and refining our techniques on our case studies, we evaluate the candidates against the properties of pervasive systems as described in the previous section. In other words we try to estimate to what extent a candidate technique would be able to cope with the unique properties of pervasive systems.

3.2.1 Experimental Studies

We apply the following approach to the experimental studies. We select a number of existing or invent new requirements elicitation techniques, which we think promising for pervasive services and apply them in a project in order to derive a set of requirements. These are then evaluated as follows. First of all, the requirements themselves –do they fit the initial needs of the stakeholders? Secondly, how chosen techniques were applied in the project – what does work well, what doesn't, and what needs improvement. Based on such reflections we improve our requirements elicitation techniques to arrive at an improved set for application in the next case study. After a number of iterations, no more substantial improvements in requirements elicitation techniques can be made, at which point we have achieved a set of requirement engineering methods suitable for service development.

3.2.2 The proposed RE method

The starting point of our requirements engineering method is the following. We observe the various stakeholders of the systems under development, and try to ascertain their needs with respect to the system from various sources, such as through interviews, diaries, user testing, and workshops [27].

We use semi-structured interviews to learn about the users' environments. The six different notions of context distinguished by Krogstie [5], as presented in Section 2, serve as a guideline in the development of these interviews. For instance, these different notions can serve as the top-level structure of the interviews, thus ensuring that the interviews are complete with respect to context information. Information obtained from them allows us to understand the daily routine of the users, and obtain insights into their goals that they seek to achieve with the help of the system. The next step is to build a more detailed and complete model of the environment through the composing of diaries based on the information gained. This lets us capture all aspects of the daily routine that were overlooked during the interviews, allowing the refinement of our requirements model.

Since pervasive systems are conceptually new and not yet available on the market, we combine the diaries with a game, in which every participant is asked to draw a picture of how they imagine the system without any limitations at all. This allows us to see the mental model a participant has of the system. Such a model can play an important role in the early stages of development helping to align the ideas of developers with those of participants in a playful and informal way.

Another important type of activity suited to idea generation for pervasive systems is a workshop. The goal of the workshop is facilitate concept creation for future services. Issues such as technology usage, but also the users' motivations regarding utilization of future services, are revealed. All the participants are divided into groups and brainstorm for ideas. They select their best two, and discuss them when the groups come together. Then all the participants receive fake money, which can be spent on anything they want, including the new services they proposed. This determines and prioritizes the more realistic ideas, since people tend to be more critical when financial issues are at stake. These ideas are then transformed into goals that the user means to attain with the help of the system being developed.

4. CONCLUSIONS

Pervasive computing is a relatively new application, which is rapidly developing. Hence, it has not matured yet, and needs a new design and development approach, which would better suit the specifics of pervasive systems. In our work we focus on requirements engineering – one of the initial steps in the development of any product. Here we summarized a number of so-called distinguishing properties of pervasive systems, as described by various authors. In our opinion, these properties should have an instant influence on the RE methods designed specifically for pervasive applications. In addition, we presented

an approach to the solution. Since we actually are trying to solve a design problem, a design research approach was taken as a basis. Then we considered how to construct the RE method for pervasive systems.

Currently, this project is in the initial phase. A number of requirements engineering methods have been studied, and the need for a new method, which focuses on the development of context-aware personalized mobile proactive services, was confirmed by the results of this study. An approach to the development of the method was proposed, and the initial set of techniques was selected.

5. ACKNOWLEDGMENTS

This work is part of the Freeband A-MUSE project. Freeband (http://www.freeband.nl) is sponsored by the Dutch government under contract BSIK 03025.

6. REFERENCES

[1] J. Shen and X. Shen, "User Requirements in Mobile Systems", Proceedings of the 2001 Americas Conference on Information Systems, August 2-5, 2001, pp. 1341-1344.

[2] Dey, A.K. & Abowd, G.D. (1999). Towards a better understanding of context and context-awareness. GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology.

[3] Pascoe, J., Ryan, N.S. & Morse, D.R. (1999). Issues in developing context-aware computing. Proceedings of the International Symposium on Handheld and Ubiquitous Computing (Karlsruhe, Germany, Sept. 1999), Springer-Verlag, 208-221.

[4] Schilit, B.N., Adams, N.I. & Want, R. (1994). Context-aware computing applications. Proceedings of the Workshop on Mobile Computing Systems and Applications, pp 85-90. IEEE Computer Society, Santa Cruz, CA.

[5] J. Krogstie, "Requirement Engineering for Mobile Information Systems", Proceedings of the Seventh International Workshop on Requirements Engineering, Interlaken, Switzerland, 2001.

[6] A. Finkelstein and A. Savigni, "A Framework for Requirements Engineering for Context-Aware Services", First International Workshop from Software Requirements to Architecture (Straw 01) 23d International Conference on Software Engineering, 2001.

[7] A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, and K. Palfreyman, "Exploiting space and location as a design framework for interactive mobile systems", ACM Transactions on Human Computer Interaction, September 2000.

[8] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research", Dept. of Computer Science, Dartmouth College, 2000

[9] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness",
Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands, April 3, 2000.

[10] T. Rodden, K. Chervest, N. Davies, "Exploiting Context in HCI Design for Mobile Systems", First Workshop on Human Computer Interaction with Mobile Devices, Glasgow, 21st & 22nd May 1998.

[11] B. Schilit, N. Adams, R. Want, "Context-Aware Computing Applications", IEEE Workshop on Mobile Computing Systems and Applications, 1994.

[12] A. H. M. Cramers, and M. A. Neerincx, "GigaMobile: Usability Requirements for Customization", TU Delft, 2000.

[13] J. de Heer, A.J.H. Peddemors, and M.M. Lankhorst. "Context-aware mobile business applications", Position paper for the first CoCoNet workshop, Zurich, Switzerland, October 2002.

[14] J. Pascoe, N. Ryan, and D. Morse, "Human-Computer-Giraffe Interaction: HCI in the Field", First Workshop on Human Computer Interaction with Mobile Devices, Glasgow, 21st & 22nd May 1998.

[15] K. Väänänen-Vainio-Mattila and S. Ruuska, "User Needs for Mobile Communication Devices: Requirements Gathering and Analysis through Contextual Inquiry", ", First Workshop on Human Computer Interaction with Mobile Devices, Glasgow, 21st & 22nd May 1998.

[16] P. Brown, W. Burleson, M. Lamming, O. Rahliff, G. Romano, J. Scholtz, and D. Snowdon, "Context-awareness: Some Compelling Applications", International Symposium on Handheld and Ubiquitous Computing, 2000.

[17] H. Hjelmeroos, K. Riho, and P. Ketola "Coping with Consistency under Multiple Design Constraints: The Case of the Nokia 9000 WWW Browser", Personal and Ubiquitous Computing, 2000.

[18] K. Väänänen-Vainio-Mattila and S. Ruuska, "Designing mobile phones and communicators for consumer needs at Nokia". Interactions, vol. 6, no. 5, ACM Press, New York, NY, USA, 1999.

[19] G. Herzwurm, S. Schockert, U. Dowie, and M. Breidung, "Requirements Engineering for Mobile-Commerce Applications", presented at the First International Conference of Mobile Business, Athens, Greece, 2002.
http://www.ebusinessforum.gr/content/downloads/033_dowie_requirements.pdf

[20] Vaishnavi, V. and Kuechler, W. (2004/5). "Design Research in Information Systems" February 20, 2004, last updated June 5, 2005. URL: http://www.isworld.org/Researchdesign/drisISworld.htm

[21] L. Kolos-Mazuryk, R. Wieringa, P. van Eck. Development of a Requirements Engineering Method For Pervasive Services. RE '05 Doctoral Consortium. Paris France. August 2005.

[22] Kolos-Mazuryk, L., Wieringa, R., P. van Eck. "A Survey of Requirements Engineering Methods for Ubiquitous Computing Systems". CTIT Technical Report, 2005, to appear.

[23] Rapoport, R.N. "Three Dilemmas in Action Research," *Human Relations*, (23:4), 1970, pp. 499-513.

[24] Harris, R.T. "Improving Patient Satisfaction Through Action Research," J. Applied Behavioral Science, 14(3): 382-399, 1978.

[25] Nunamaker, J.F. and Chen, M. (1990). Systems development in information systems research, IEEE, 631-639.

[26] Hevner, A., March, S., Park, J. and Ram, S. (2004). "Design Science in Information Systems Research." MIS Quarterly 28(1): 75-105.

[27] S. Consolvo, L. Arnstein, B. R. Franza. User Study Techniques in the Design and Evaluation of a Ubicomp Environment.