

# Architecture for Adaptable Component Based Mobile Systems: End User Driven Real Life Tagging for Individualized Context-Awareness

Matthias Betz, Mahmudul Huq, Volkmar Pipek, Markus Rohde, Gunnar Stevens,  
Volker Wulf

*Faculty of Information System and New Media,  
University of Siegen, Siegen, Germany*

{matthias.betz | syed-m.huq | volkmar.pipek | markus.rohde | gunnar.stevens | volker.wulf  
@uni-siegen.de}

**Abstract.** *The palpable computing paradigm aims to support user control by composing and decomposing assemblies of devices and services, which communicate and exchange data. As a product of a research project, conducted by University of Siegen and Fraunhofer Institute of Applied Information System, we have developed a Context-Aware Adaptable System (CAAS) architecture. This architecture has both adaptive and adaptable nature. In this research paper, we would like to describe the addition of RFID technology in a prototype based on the CAAS architecture. Through the RFID technology users of the prototype will be able to define contexts of use by using small RFID tags.*

**Keywords:** Palpable Computing, End User Development (EUD), RFAID Technology, Context-Awareness, Context-Aware Adaptable System (CAAS).

## 1. Introduction

Context-awareness is the most important feature of ubiquitous and pervasive computing [1] [2]. It simplifies the interfaces we present to our mobile users, adapting them to the current situations [3]. Context-awareness is also the key characteristics of new smart information services. It includes all kinds of situational properties like spatial and temporal contexts, task context, collocation context, historical context etc. These systems also exploit their surrounding context to increase the suitability of a service or an application to the user's needs. Context-aware systems support users in dealing with complex systems of technologies and devices. This is possible because they aim to understand the user's current task, location and time contexts and provide situation adequate support.

Studying context-aware systems in a real life setting demonstrates that the contexts sensed by these systems are often ambiguous. Although some of these ambiguities may be resolved using automated problem-solving techniques, many cases still require the user's involvement for the correct handling of the ambiguous context [4]. This leads to system approaches that have to have accurate and reliable adaptive services as well as adaptable concepts in order to allow users to intervene in a simple and transparent way whenever necessary.

For the many decisive aspects of ambiguous contexts it is the end user as a domain expert that has been identified as the key role to adapt context-aware behaviors. End users are also the key people to adapt necessary data sources and constraints. Thus, the end users and the domain experts should be able to define the appropriate adaptable behavior for their application.

Therefore new design challenges in context-aware systems are to interpret the context correctly not only to inform adaptive services, but also to empower the end users to understand and configure the behavior of the system [5].

The research field of End User Development (EUD) has empowered the end users to adapt their own computer systems [6] [7] [8]. Some of the concepts already have been transferred to the construction of context-aware systems, for example, Programming By Demonstration (PBD) [9] [10].

This research paper is part of a EUD research project started in 2004 to support ubiquitous services in the health/fitness area. In this project, we started with an ethnographical study of fitness and sports activities. This study leads to several application scenarios that should be addressed by a context-aware system [11].

This research paper focuses on the technical aspects of a context-aware system that uses RFID technology. Such a context-aware system supports ubiquitous health/fitness services. We will describe how to incorporate RFID technology in a general Context-Aware Adaptable System (CAAS) architecture. By enabling the user to utilize RFID tags to mark contexts during usage, we try to adapt the idea of palpable computing in the architecture of our Context-Aware Adaptable System (CAAS).

The research paper is organized in the following order: in the second section some current trends in palpable computing will be discussed. In the third section we will provide a detailed description of the architecture of our Context-Aware Adaptable System (CAAS), which is a component framework based on the OSGi component-model. In the fourth section we will propose the integration of RFID technology in the CAAS architecture.

## **2. Current Trend in Palpable Computing**

Palpable systems shall support control and choice by people by supporting users in understanding what is going on at the level they choose [12]. Palpable computing grounds on the notion of ambient computing, which is focused on the design of distributed, pervasive and reactive systems that are able to communicate with the users and to continuously adapt to their expectations [13] [14]. The palpable

computing paradigm addresses the way in which humans meaningfully interact with distributed computational systems at supporting users control by composing and decomposing assemblies of devices and services which communicate and exchange data [15].

Ambient computing is currently a field of strong development. It is about moving computer capabilities beyond the desktop and about constantly adapting configuration of technology to changing situations and needs. Key issues in ambient computing include [15]: invisibility, construction, heterogeneity, change, and scalability.

### **3. Architecture of Context-Aware Adaptable System (CAAS) Framework**

In the following section we would like to describe the actual software architecture for CAAS that we have developed in the EUD project. The OSGi standard has been used as the basis for setting up the context-aware adaptation. The proposed software architecture for CAAS shows the three core concepts, i. e., context-awareness, End User Development (EUD) and OSGi Service Management.

#### **3.1 Context Manager**

The Context Manager provides the application with a rich context model and associated quality information. It is responsible for acquiring context information from various sources and through a variety of communication protocols. Such context sources only provide a subset of the variety of contextual information that is relevant to a specific application. A significant increase in the expressiveness, complexity, and quality of the represented context can be achieved by transforming the acquired context data in several ways. Common context transformation techniques are fusion, derivation, aggregation and interpretation. Mainly these techniques differ in the knowledge that is required for their application. Fusion refers to the consolidation of multiple pieces of information of the same type, in order to increase the quality. Information pieces of different types may be subject to aggregation, if one piece cannot provide the required information by itself. A semantic model forms the basis for an information interpretation process, in order to gain high-level information. Another mechanism for context augmentation is the derivation of context. Deriving context means that the acquired context data serves as a basis for inferences that may vary in complexity. The context manager merges the various aspects of the acquired information into a coherent whole, which is based on any information that can be used to characterize the situation of a person or a computing entity and to identify the need for adaptation.

In addition, the Context Manager supports the determination, representation and supply of quality metrics characterizing information. The model of quality in sensed and interpreted data allows for a decision on the need for end-user involvement in the decision process with regard to the user preferences. Service developers are provided with a way to identify a loss of quality when it is present and to specify a relationship

between data and the selection of the appropriate mediation technique. Correspondingly, the quality metrics offer support for selecting among potential mediation techniques and automatically instantiating them when low-quality data arrives in an application.

A fully functional context manager for our CAAS framework is not implemented yet. At the time being, the framework needs a source for the actual context to identify context changes to trigger contextual behavior. To substitute the context manager a small web application was developed which generates output like the real context manager. This dummy context manager is not able to interpret sensor-data. It simply depends on the input from a person who uses this application.

### **3.2 Composition Adaptation Manager**

The Composition Adaptation Manager realizes the component-based approach of the CAAS. It implements the runtime environment for dynamic software components and the dynamic management of the composition along the lifecycle of the OSGi component-model. This coarse grained part of the architecture is comparable with the principles of Programming in the large [16]. Some examples of those components are already implemented for testing purposes and can be downloaded and installed dynamically into an existing application. Because of the dynamic character and the self-descriptiveness of these available components, in the context of the research project, they are called service-components.

### **3.3 Component Adaptation Manager**

Referring to the concept of Programming in the small [16] the Component Adaptation Management realizes the more fine grained approach of adaptation. This part in the architecture manages the inner adaptation of a service-component itself. These adaptations are also coupled to the current context by using the triggering mechanism. This part of the architecture is not implemented yet.

### **3.4 Mediated Adaptation Manager**

Mediated adaptation describes an adaptation technique which separates the adaptation process from the operating software artifacts. The start of a mediated adaptation can be triggered by either the system or the end user. Therefore the Mediated Adaptation Manager realizes the shared initiative between system and user driven adaptation processes. The completion of a mediated adaptation results in an accomplished and retained adaptation of the system. A mediated adaptation may be accomplished automatically, or may be driven by the end user, or may be a sequence of both can occur in combination. Every adaptation service component lists its required mediator in its own descriptor. For more specific mediators, it is also possible for service components to contribute component specific mediators in the mediated adaptation manager.

#### 4. Next Steps: RFID Based Real Life Tagging

Currently the Context-Manager is implemented as an interface and the core functionality is substituted by a third person, who observes the user of the system and triggers changes in the user's context. This solution was chosen to test the implemented system in the early state. In the next step of the project we will replace this through the integration of RFID reader units into the mobile device and connect them with the software interface of the Context Manager. The idea is to utilize the readers to detect RFID tags in the surrounding environment. Therefore those tags can be used to mark certain places which will be identified as contexts of use by the user himself.

In mobile settings RFID seems to be a good option to detect certain contexts in mobile applications to enable context-awareness. This is because RFID tags are cheap, small and easy to handle. In addition they provide a definite possibility to identify contexts of use by the end user. After the integration of the RFID reader into the implementation of the CAAS (as an implementation of the Context Manager), this idea will be evaluated in real life settings within a fitness studio scenario with service-components related to the fitness domain. Nevertheless the framework defined through the described architecture is open for other scenarios and applications. The target is the creation of a multi purpose framework to realize context-awareness and adaptability for mobile applications where the definition of usage contexts is apprehensible, transparent, easy to use and finally palpable.

#### References

1. Dey, A.K.: Understanding and using context. *Personal Ubiquitous Computing*. 5(1) (2001) 4–7
2. Dourish, P.: Seeking a foundation for context-aware computing. *Human-Computer Interaction* 16(2-4) (2001)
3. Want, R., Pering, T.: System challenges for ubiquitous & pervasive computing. In: ICSE '05: Proceedings of the 27th international conference on Software engineering. (2005) 9–14
4. Dey, A.K., Mankoff, J.: Designing mediation for context-aware applications. *ACM Transaction Computer-Human Interaction* 12(1) (2005) 53–80
5. Bellotti, V., Edwards, K.: Intelligibility and accountability: Human considerations in context-aware systems. *Journal of Human-Computer Interaction* 16(2-4) (2001) 193–212
6. Fischer, G.: Beyond "couch potatoes": From consumers to designers. In: APCHI '98: Proceedings of the Third Asian Pacific Computer and Human Interaction, Washington, DC, USA, IEEE Computer Society (1998) 2
7. Lieberman, H., Patern&#242;, F., Wulf, V.: *End-User Development (Human-Computer Interaction Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
8. Wulf, V.: Anpbarkeit im prozes evolutionrer systementwicklung. *GMD – Spiegel* 24(3) (1994) 41–46

9. Lieberman, H., ed.: Your Wish is My Command: Giving Users the Power to Instruct their Software. Morgan Kaufman Publishers, Inc. (2000)
10. Dey, A.K., Hamid, R., Beckmann, C., Li, I., Hsu, D.: a cappella: programming by demonstration of context-aware applications. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (2004) 33–40
11. Stevens, G., Wulf, V., Rohde, M., Zimmermann, A.: Ubiquitous fitness support starts in everyday's context. In: Proceedings of the 6th World Conference on the Engineering of Sport. (2006)
12. Kyng, M., Mogensen, P.H.: Palpable computing: a new perspective on ambient computing (Februar 2007)
13. Weiser, M.: The computer for the twenty-first century. *Scientific American* 265(3) (1991) 94–104
14. Weiser, M.: Some computer science issues in ubiquitous computing. *Commun ACM* 36(7) (1993) 75–84
15. Ingstrup, M., Hansen, K.M.: Palpable assemblies: Dynamic service composition for ubiquitous computing. In: Proceedings of the Seventeenth International Conference on Software Engineering and Knowledge Engineering. (July 2005)
16. DeRemer, F., Kron, H.: Programming-in-the large versus programming-in-the small. In: Proceedings of the international conference on Reliable software, New York, NY, USA, ACM Press (1975) 114–121