POSTER
Towards Proactive Adaptation in Pervasive Environments

Sebastian VanSyckel, Gregor Schiele, and Christian Becker

Universität Mannheim, 68131 Mannheim, Germany
{sebastian.vansyckel,gregor.schiele,christian.becker}@uni-mannheim.de
http://proactive.bwl.uni-mannheim.de

Abstract. Pervasive computing applications can adjust their behavior to a multitude of information deemed to be relevant for their situation, their so-called context. Thus far, however, adaptation in such context-aware systems is reactive and limited to the application itself. These restrictions inevitably delay adjustments to events. They cause frequent reconfigurations, and may result in inferior overall system configurations. To remedy these shortcomings, we propose a framework for proactive adaptation that supports applications in preparing for, or counteracting, upcoming context events. The framework consists of (i) a context management component with prediction capabilities, (ii) an application model for calculating adaptation alternatives, and (iii) a pool of adaptation strategies for decision-making.

Keywords: Proactive Adaptation, Context-aware Computing, Pervasive Computing.

1 Introduction

Today, context-aware pervasive computing applications detect the current state of their environment – their so-called context – and adapt themselves to it, dynamically. To do so, a multitude of sensors are deployed, measuring conditions like user location and speed, light level, and temperature. When a predefined context event occurs, the application reacts to this, e.g. by adapting its deployment or behavior. However, this *reactive adaptation approach* has a number of important shortcomings. First, adaptation can only happen after the event has occurred. This leads to inevitable delays before an application can adapt, e.g. due to the complexity of computing the best adaptation. During this time, the application may act inappropriately, annoying users or even causing system failures. More importantly, in some cases, it might not be acceptable for a given context event to occur at all. A reactive system can never prevent specific context events. It can only try to minimize the damage. In addition to this, reactive adaptation can only optimize for the current application context. It has no knowledge about the development of future context states. Therefore, an adaptation decision might later on turn out to be

A. Puiatti et al. (Eds.): MobiQuitous 2011, LNICST 104, pp. 214–218, 2012.

suboptimal or even invalid. In dynamic environments, this may lead to numerous reconfigurations over a short period of time.

We believe that *proactive adaptation* can resolve these shortcomings. In a proactive system, applications can prepare themselves to future context events, e.g. by precomputing the best adaptations. In addition, they can optimize for series of future context events, e.g. minimizing the number of necessary reconfigurations. Finally, unacceptable context events can be avoided by adapting the chain of events leading to them. In our work we examine how to realize such proactive pervasive computing systems. To do so, we develop a conceptual framework for proactive adaptation that identifies all necessary components and algorithms, and integrates them in a single system.

In this paper we identify three main components, namely *context prediction management*, *application model* and *adaptation strategies*. We analyze the challenges resulting from proactivity for each of these components and propose how to address them.

2 Proactive Framework

Figure 1 shows our approach to proactive adaptation, which is based on a trisection of the research question. The *context prediction management* (CPM, Section 2.1) is responsible for providing suitable access to context information as well as predicting context, whereas the *application model* (AM, Section 2.2) is responsible for finding all possible configurations based on the provided context and the set of context requirements posed by the application. Finally, the *adaptation strategies* (AS, Section 2.3) can determine the best chain of adaptations by applying their predefined policy. In the following, we describe the three components in more detail.

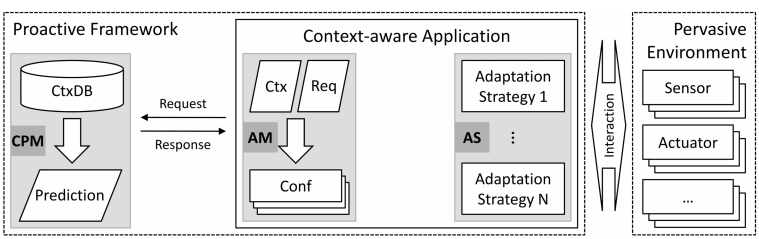

Fig. 1. A Framework for Proactive Adaptation

2.1 Context Prediction Management

Context management – acquisition, representation, preprocessing, and provision of context – is key for any context-aware system, as it provides the data on which the system adapts. In the case of proactive adaptation, this data includes context predictions. There exist a multitude of context prediction approaches. However, none is suitable for all forms of context, e.g. numerical vs. symbolic. For a generic framework, this results in the challenge of integrating several prediction methods into a single *context-aware prediction* approach.

Predictions are never certain. Research questions posed by this are: (i) how to efficiently manage chains of interdependent predictions, especially in the case of a wrong prediction, (ii) how to support a range of application-specific prediction requirements (QoS), such as false positive/ negative tolerance, and (iii) how can the system in general handle wrong predictions, especially in the case of a malicious adaptation due to false information. A basic recovery approach for malicious adaptations is a fallback to the reactive scheme. Therefore, the context management must also provide current context information to the application.

In summary, the context management provides both current and predicted context, following QoS parameters, to the application model, which we describe next.

2.2 Application Model

The application model is responsible for calculating and rating all possible adaptation alternatives given one context, current or future, and the applications requirements towards its context. In contrast to reactive systems, it is not sufficient to find one functioning adaptation, as the greatest benefits of proactive adaptation lie in the optimization of an adaptation series. The optimization is then done by the adaptation strategies (Section 2.3). This subdivision enables the framework to support reactive adaptation by simply omitting the last step.

Challenges, in addition to developing a feasible approach to the above mentioned search problem, are: (i) how to specify the application's requirements and model their importance, e.g. environment conditions and remote resources, as well as *must-have* and *nice to have*, (ii) how to calculate ratings of possible adaptations, e.g. goodness, reconfiguration costs and duration, and cost to benefit ratio, and (iii) the implications of enabling applications to alter their context via actuators. Expanding adaptation from an application to the entire environment dramatically increases the search space of adaptation alternatives and puts forth further questions, such as adaptation coordination.

2.3 Adaptation Strategies

The adaptation strategies calculate the best chain of adaptations, considering an application's goals, situation and current condition, the alternatives provided by the application model, and its predefined policy. Exemplary strategies may be energy efficiency or service consistency. A mobile application, for instance, should be cautious about its resources, whereas a stationary service should perform at its highest level.

These goals, situations and conditions may also interfere, increasing the complexity of finding a viable solution. A further challenge is how to factor in the uncertainty of predictions. As mentioned before, false predictions may lead to malicious adaptations. The risk of this to happen may outweigh an inferior configuration, leading to the question of how to calculate risk and model risk aversion.

3 Related Work

Various adaptive frameworks, such as Gaia [4], iROS [3], and PCOM [1], have been developed over the past decade. They enable mobile applications to adapt their behavior at runtime, as well as incorporate remote resources and services. Hereby, the underlying adaptation strategies, especially the distribution of responsibilities throughout the system, reside on a spectrum between *laissez-faire*, i.e. application-initiated, and *application-transparent*, i.e. system-initiated, adaptation [5]. In any case, adaptation in the above mentioned systems is reactive, i.e. executed after the triggering event. The framework Aura [2] is an exception in the sense that it anticipates proactive adaptation for some aspects of the system, such as network load and data distribution. Further work on this topic, however, has not yet been published.

4 Current and Future Work

The presented framework constitutes our approach to a proactive system on the level of the components' tasks. Currently, we are designing the system architecture and developing a middleware-based prototype. Within, the context management is realized as a centralized service that – next to providing access to current and predicted context – acts as a mediator between the applications and the pervasive environment. To achieve this, we created an additional abstraction using context variables. This allows an application to request any context service without knowledge of the providing device.

The next step will be the development of the application model, before we implement a small set of adaptation strategies. With the completion of the prototype, we expect to show significant improvements over reactive adaptation, such as increased energy efficiency and service quality.

5 Conclusion

We have presented a framework that enables context-aware applications to prepare for their future context based on prediction, requirement satisfaction, and strategic adaptation. This not only reduces delays caused by the reactive nature of traditional systems, but also allows applications to run in an optimal configuration with regard to a series of context events. Following one of several strategies, a proactively adapting application can, amongst others, save resources and provide a consistent level of service.

References

1. Becker, C., Handte, M., Schiele, G., Rothermel, K.: PCOM - A Component System for Pervasive Computing. In: IEEE Proc. Pervasive Computing and Communications (PerCom), pp. 67–76 (2004)

2. Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P.: Project Aura: Toward Distraction-free Pervasive Computing. IEEE Pervasive Computing 1(2), 22–31 (2002)
3. Johanson, B., Fox, A., Winograd, T.: The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. IEEE Pervasive Computing 1(2), 67–74 (2002)
4. Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., Nahrstedt, K.: A Middleware Infrastructure for Active Spaces. IEEE Pervasive Computing 1(4), 74–83 (2002)
5. Satyanarayanan, M.: Mobile Computing: Where's the Tofu? ACM SIGMOBILE Mobile Computing and Communications Review 1, 17–21 (1997)