Middleware-based System Support for Proactive Adaptation in Pervasive Environments

Sebastian VanSyckel
University of Mannheim
Schloss, 68161 Mannheim, Germany
sebastian.vansyckel@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. In this paper, we present our work in progress on middleware-based system support for proactive adaptation. It offers context information, prediction, and influence via a uniform abstraction, update notifications for subscribed context or predictions, and an application model to determine adaptation alternatives.**

Keywords*—*Proactive Adaptation; Context-aware Computing; Pervasive Environments

I. Introduction

In pervasive environments, context-aware applications adapt their behavior or change their composition as a reaction to changes in their context, i.e. their physical environment, the social environment of the user and the technical environment. Reactive adaptation is only able to react on changes. If the context changes frequently, adaptation can occur frequently as well. Further, the adaptation decision is singular in the sense that it only regards the current information. Context prediction has been used to plan adaptation ahead before it actually happens. Based on prediction, applications can choose if they adapt their behavior or adapt the context, as well as optimize the next series of adaptations based on a certain strategy. In the setting of a smart environment, an application, for example, could bind external resources, such as I/O devices, before the user actually enters the room. Regarding optimization, the application could decide to pause a service, as a necessary adaptation is too costly – e.g. battery life – for the short time the user will be passing through the context forcing the application to adapt.

Proactive adaptation in pervasive environments requires several steps. First, application developers need to specify functional configurations of the application with regard to its usability. This includes environment conditions, such as lighting levels, as well as services in the environment, such as a visual output – in the following referred to as their context. Second, applications have to gain knowledge of when and how their context will change. Based on this information, applications have to determine possible violations of their active configuration. If their active configuration will become unfunctional, they have to find at least one possible adaptation. Such adaptations can be limited to the applications, i.e. switching to a different functional configuration, and/ or include adapting the applications environment via actuators. Ideally, however, such adaptations are optimized for a series of upcoming context events, e.g. to avoid frequent switches of configurations. As a result, applications should perform the letter steps for all predicted events as well as compute all possible adaptation alternatives for these. This leads to a constant effort of monitoring predictions, finding all solutions, and deciding on a strategy.

In our approach, we support application developers by offering a uniform abstraction to context interaction, context prediction including update notifications, and configuration management. Hereby, our work builds on existing work on context frameworks and context prediction.

II. Middleware-based System Support

In the following, we present our approach to the process of pervasive adaptation described above.

A. Context Interaction

Context is defined by its identity, location, and the point in time. In the well-researched reactive systems, identity and location are handled by context- [1] and location models [2], respectively. Time, however, is *right now* by default and, hence, not regarded. Further, the systems implementing those models must only provide context. For proactive adaptation, however, we need to be able to define other points in time while requesting predictions, as well as adapt context.

In order to offer a uniform interaction model between applications and their context – i.e. (distributed) sensing-, predicting-, and actuating services – we create an abstraction by the use of *context variables*. The idea behind our approach is to extend location models with variables that represent the different types of context that are present at each location. For interaction, we now link all context services in the environment to their respective variable and location. Further, we formalized interaction in a set of *context queries*. [7]

B. Context Prediction and Notifications

Context prediction is central to proactive adaptation, as it is the information the adaptation decisions are based on. Consequently, we support querying for future context, i.e. location or identity information with a timestamp in the future. We do not focus on context prediction algorithms, but implemented published approaches. These algorithms learn during their lifetime, i.e. their output depends on their knowledge base, which, therefore, has to be updated.

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

We do, however, support learning mechanisms specifically by offering so called *context subscriptions*. All information requesting queries – including predictions themselves – can be subscribed to. With this, *predictors* can monitor relevant information and recompute on an event. Successively, an application that subscribes to a prediction is notified automatically, as soon as the predictor changes its forecast, and can adjust accordingly.

C. Configuration Management

From the before mentioned functional requirements of an application, the configuration management constructs a constraint satisfaction problem (CSP). The adaptation alternatives for an application then are all possible solutions. For this, we implemented several approaches for extensive evaluation and comparison. Even though CSPs in general are NP-complete, our initial tests suggest that they are quickly solvable (in a few *ms*) for the typical size defined by a pervasive environment, i.e. the amount of locations, context variables and their respective set of possible states.

Further, we developed a model to rate the possible adaptation alternatives regarding their cost and benefit over time. Here, the duration of a given context, which again is a prediction, is a main factor.

D. Architecture and Interaction Model

Figure 1 shows the high level architecture of our prototype system as well as how the various components in the environment interact with each other. The prototype is build on top of BASE [3], a lightweight and extendible middleware specifically designed for pervasive computing.

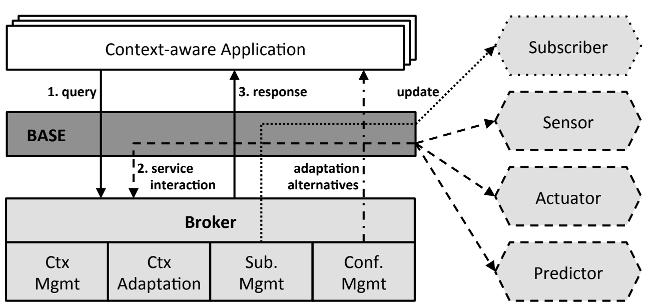

Fig. 1. The system's architecture and interaction model.

Central to the system is a broker that implements the set of context queries as its interface and mediates all context interaction. This way, the context-aware applications have one communication partner and, thus, application developers have to handle only one proxy. Further, the broker manages context- and prediction subscriptions, and notifies the subscriber in case of an update. In the figure, the broker also contains the configuration management, which informs applications about their adaptation alternatives. Please note that this is not final.

III. Related Work

Some work on proactive adaptation frameworks exists, such as the framework by Mayrhofer [6], the CALCHAS system [4], and the MavHome project [5]. The first two systems provide predictions to applications, as well as how to decide actions upon these predictions. However, the key task of context adaptation via actuators are not addressed. The third example covers the entire process, but is a fixed application in a closed environment. In our work, we aim at a more generic approach for an open environment.

IV. Current and Future Work

Currently, we are both finalizing the development of the configuration management as well as designing smart office scenarios for extensive evaluation of the system, e.g. the necessary prediction accuracy as well as the effects of malicious adaptations due to false predictions regarding recovery, etc., and the benefits of proactive- over reactive adaptation in general. One scenario, for example, will feature *public* monitors throughout an office building that are bound by applications based on the predicted locations of the users. Further, *private* monitors in offices will hide confidential information before somebody enters the room. With such scenarios, we also plan to evaluate different adaptation strategies, such as maximizing the quality of service for the user, or minimizing battery usage, respectively.

In future work, we would like to integrate adaptation coordination, i.e. expand the systems functionality to supporting application ensembles that share context and, therefore, influence each other.

V. Conclusion

We presented a middleware-based system that supports applications to adapt proactively in an application-transparent fashion. With our approach, application developers specify their requirements towards their context and, through context prediction subscriptions and configuration management, are notified about the current forecasts as well as the application's adaptation alternatives.

Acknowledgment

This work is being advised by Christian Becker and supported by the German Research Foundation (DFG).

References

[1] Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-aware Systems. *International Journal of Ad Hoc and Ubiquitous Computing, Inderscience*, 2(4), pp.263-277, 2007.

[2] Becker, C., Dürr, F.: On Location Models for Ubiquitous Computing. *Personal and Ubiquitous Computing, Springer*, 9(1), pp.20-31, 2005.

[3] Becker, C., Schiele, G., Gubbels, H., Rothermel, K.: Base - A Micro-broker-based Middleware for Pervasive Computing. In *Proc. Pervasive Computing and Communications (PerCom)*, 2003.

[4] Boytsov, A., Zaslavsky, A.: Extending Context Spaces Theory by Proactive Adaptation. *Smart Spaces and Next Generation Wired/Wireless Networking, LNCS, Springer*, 6294, pp.1-12, 2010.

[5] Cook, D. J., Youngblood, M., Heierman, E., Gopalratnam, K., Rao, S., Litvin, A., Khawaja, F.: MavHome: An Agent-Based Smart Home In *Proc. Pervasive Computing and Communications (PerCom)*, 2003.

[6] Mayrhofer, R.: An Architecture for Context Prediction. PhD thesis, Johannes Kepler University of Linz, Austria, 2004.

[7] VanSyckel, S., Schiele, G., Becker, C.: Extending Context Management for Proactive Adaptation in Pervasive Environments. in *Proc. Ubiquitous Information Technologies and Applications (CUTE), 2012.*