Using a Model-based Interface Construction Mechanism for Adaptable Agent User Interfaces

Lars Braubach
braubach@informatik.uni-hamburg.de
Distributed Systems and Information Systems

Alexander Pokahr
pokahr@informatik.uni-hamburg.de
Distributed Systems and Information Systems

Daniel Moldt
moldt@informatik.uni-hamburg.de
Theoretical Foundations of Computer Science

Winfried Lamersdorf
lamersd@informatik.uni-hamburg.de
Distributed Systems and Information Systems

Computer Science Department, University of Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

ABSTRACT

Agent systems in ubiquitous environments possess properties, which make the construction of user interfaces more complicated than for other systems. Under consideration of these properties, it becomes clear that an especially flexible, dynamically adaptive mechanism is required with the construction of user interfaces for such systems. In this paper, for the user interface construction significant properties of agent systems are discussed and a mechanism to the solution of these problems is proposed.

1. INTRODUCTION

There are properties of agent systems in ubiquitous environments, which can complicate the construction of user interfaces. These properties will be discussed next.

When agents are mobile and therefore can change their site dynamically, the user interface modality may change because of device heterogeneity. For example, when an agent moves from a computer to a WAP-phone it changes from GUI to WML. Besides the interface modality change, further adaptation of the presented information is needed, e.g., the space to display information on the phone is more restricted than on the computer. Furthermore, the users of agents can act in different roles and the actual role must be considered by the agent, when an interaction is started. Therefore, different user interfaces are needed with respect to the actual role of the user. Agents act on behalf of a user and the user should have the possibility to instruct, observe, and to interact with the agent. This means that the user should have a configuration management tool to instruct the agent what to do and how to achieve this goal. Further on she should be able to gather status information about the agents state and in this way be able to observe what it is doing. Interaction is for example necessary, when submitting a request to an agent. These different activities need to be supported by user interfaces. There are agents in an agent system, that are influenced by the functionality of other agents close to it. So the agent should provide some kind of composition mechanism which allows it to alter dynamically its appearance.

The paper is organized as follows: In Section 2 an example application context as a real world scenario is described. Thereafter, Section 3 proposes a solution for the agent user interface problems. Finally, the conclusion in Section 4 summarizes the main ideas.

2. SCENARIO

In this section a scenario from the MedPAge project [3] will be presented which takes advantage of the above mentioned techniques and enhances the status quo. Banavar et al. [1] point out the importance of a service-oriented application design when developing in a ubiquitous computing environment. This design of cooperating services fits well within the agent metaphor.

Considering a hospital setting one can observe a multitude of heterogeneous devices that should be connected in one coherent network to unclose broader communication channels for the hospital personnel and the patients to the hospital management system. It should be one objective to enable as many hospital devices as possible to be part of the system-infrastructure, to offer information retrieval and planning services. As all users of the system are mobile throughout the hospital, the information has to be mobile, too. That is, every information should be presentable on any device.

Furthermore, the system has to take into account the current context and user settings with respect to interface selection and adaptation. For example a doctor who uses a medical device should be offered other capabilities than a nurse or a patient. Additionally, users of the system may have different roles (e.g. some of the hospital staff may have administrative roles), therefore, the information presented to a user has to match her current role.

The different activities with respect to the agents have to be supported. The doctor may instruct the agent, to respect her personal preferences, e.g., when scheduling operations. An example for the interaction with an agent is to confirm the date of a scheduled operation. Additionally, the doctor may want to observe the agent, while it is trying to assemble

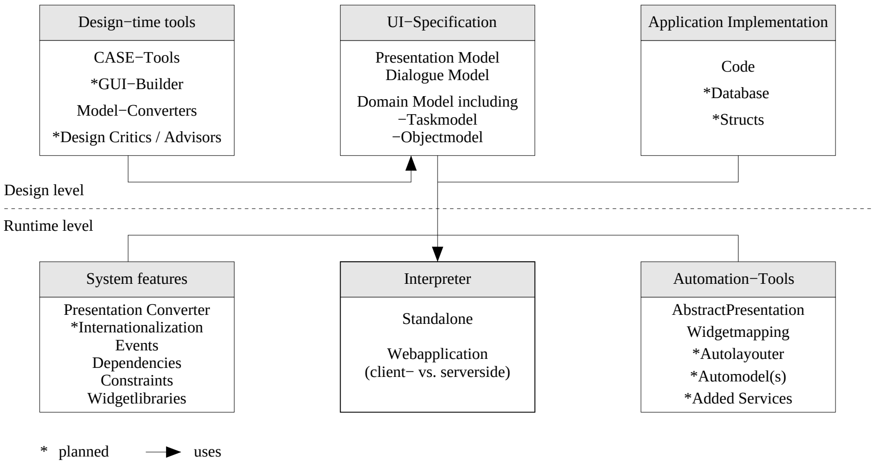

Figure 1: Vesuf system components.

an operation team.

Due to dynamic changes in the environment, the composition of agent interfaces is required. For example, when the head physician presents the latest statistics of her department, using the PDA and a video projector, the agent managing the presentation should include controls related to the settings of the video projector (e.g. brightness and contrast).

3. PROPOSED SOLUTION

An extensive research in the field of user interface architectures, techniques, and tools has revealed the class of model-based user interface development environments (MB-UIDEs) to be an encouraging foundation for building flexible and dynamic systems [4]. As argued in [2] interpreter-based MB-UIDEs are potentially well suited for all ubiquitous computing requirements. The above mentioned requirements for agent systems can be achieved by the special characteristics of model-based systems as follows.

Mobile agents can adapt their interfaces to heterogeneous devices by selecting an appropriate presentation model. Additionally they can change the view sequencing by using different dialogue models. The use of various presentation and dialog models also enables the agents to present user interfaces conformant to the user roles and facilitates the different types of user agent communication. The composition of user interfaces becomes possible, because the models are settled on an abstract level, i.e., they describe what can be done with the interface, and not just how it looks like.

It has been revealed, that there is currently no system available which has the prerequisites to fulfill all requirements, therefore, a new system has been prototypically implemented [2]. The Vesuf system consists of six major components that are depicted in Fig. 1. Central component is an *interpreter*, that dynamically generates user-interfaces from *UI-specifications*, integrating them seamlessly with an *application implementation*. For the specification of the models, *design-time tools* are used. Incomplete UI-models can be completed by the interpreter using *automation tools*. For the integration of user interface and functional core, the interpreter uses framework-like *system features*, implemented at the programming level. The most important property of the environment is the openness with respect to additional design-time tools, automation tools and system features. This allows to integrate any technologies and tools of different abstraction-levels into the environment.

4. CONCLUSION

In this paper a scenario was presented to exemplify the need for adaptation of agent user interfaces to dynamic contexts of use. The presented system makes it possible to create different user interfaces with respect to one agent depending on device, user role and activity. It lays a sound foundation for dynamic composition and adaptation through the use of declarative models and the interpreter style.

5. REFERENCES

[1] G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman, and D. Zukowski. An Application Model for Pervasive Computing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM-00)*, pages 266–274, N. Y., August 6–11 2000. ACM Press.

[2] L. Braubach, A. Pokahr, D. Moldt, A. Bartelt, and W. Lamersdorf. Tool-Supported Interpreter-Based User Interface Architecture for Ubiquitous Computing. In P. Forbrig and J. Vanderdonckt, editors, *Proceedings of DSV-IS'2002*. Springer, 2002. To appear.

[3] MedPAge. Medical Path Agents. DFG Schwerpunktprogramm, `http://vsis-www.informatik.uni-hamburg.de/projects/medpage/`, 2000.

[4] A. Pokahr, L. Braubach, A. Bartelt, D. Moldt, and W. Lamersdorf. Vesuf, eine modellbasierte User Interface Entwicklungsumgebung für das Ubiquitous Computing. In H. Oberquelle, editor, *Mensch & Computer 2002*. Teubner, September 2002. To appear.