Model-based Evaluation and Adaptation of Self-adaptive Applications at Runtime

Michael Quade
michael.quade@dai-labor.de
www.dai-labor.de

Motivation

Smart environments are characterized by smart objects and multiple applications which are able to interact with each other and build up a complex system. Moreover, users of such applications expect to be ubiquitously supported in various situations throughout the day. Therefore, all applications in a smart environment need to present required information properly and tailored to the current user’s needs and (dis-)abilities.

These newly arising demands are currently approached by the development of adaptive applications which aim at providing user interfaces that are able to adapt to the user instead of requiring the user to adapt to the user interface. Here it is of crucial importance that these applications are not just self-adaptive but are also able to self-evaluate the usability of their user interface regarding to the current user and context.

Problem Description

Evaluation of self-adaptive User Interfaces becomes difficult due to high complexity

Emerging need for dynamic evaluation methods at runtime

Detailed information about user, context and environment are only available during runtime VS. Custom usability evaluation methods are usually carried out at development time

Evaluation Engine

An Evaluation Engine holds copied instances of system and user models during interaction of the real user. Interaction of the models is evaluated according to well-defined criteria and best fitting adaptations of the UI are chosen.

Predicate

Simulation

Evaluation

Adaptation

Cycle of model interaction simulation, evaluation and adaptation

Outlook

First results show that simple layout routines can be adapted with the help of related evaluation criteria. The integration of more detailed rules for adaptations of specific user interface elements related to the current user’s goals and more dynamic aspects of interaction are also considered. User goals could be inferred from external sources, e.g. schedules and location based data; and mechanisms for maintaining the user model could be introduced.

As next steps, the integration of further available information like the context-of-use and models of the environment are planned. This allows checking for UI distribution over available devices. Finally, the approach could be enhanced to be used for enabling user guidance by highlighting more efficient ways of interaction. This way, the domain expertise of the user can be increased slowly while interacting with the adaptive application.

Approach

The work proposed here develops an approach which utilizes runtime models for reflecting the internal state of an adaptive application and for changing its underlying configuration in order to evaluate adaptations of the user interface. Furthermore, the approach will be extended with a simulation and an evaluation component. In particular, an application model and a user model will be employed. Through interconnection of these models, the user model is performing the process of perception and interaction with a model of the current application at runtime. The main benefit of this approach lies in evaluating future interaction states of the application at runtime and adjusting adaptations of the user interface to specific user needs. Compared to existing static adaptation approaches with predefined user groups and application capacities at development time, this approach promises to be more flexible and dynamic.

Links

SmartSenior
http://www.smart-senior.de

MASP
http://masp.dai-labor.de

DAI-Labor
Distributed Artificial Intelligence Laboratory