How to survive Multi-Device User Interface Development with UIML

Dienstag, den 23. 11. 2007 um 15:30 Uhr, RIGZ

The User Interface Markup Language has been around for 10 years now and it is still evolving. The language did not reach a large audience yet and there is a limited set of real publicly known cases where the language is used. Nevertheless, UIML is probably the most generic and flexible user interface markup language available, mainly because it is in fact a meta-language that is not limited to a particular domain, device, platform or widget set.

The design of the language makes it one of the best candidates to support future evolutions in novel user interface techniques and guarantees the survival of user interface designs over time.

On the dawn of the 4.0 version of the UIML specification, which will be released soon, this talk provides an overview of the language, the novelties of the 4.0 specification and some tools that we have build and are available today.

We will also provide an overview of our work to integrate UIML in a model-based interface development process that allows to design multi-device user interfaces without the abstractions often required when targeting multiple platforms in a single design iteration.

Engineering Interactive Systems for Ambient Intelligent Environments

Mittwoch, den 24. 11. 2007, um 9:00 Uhr, FhG, Raum Erdgeschoss

The increasing complexity of ubiquitous computing environments puts the current software development methods to the test. There is a large variation in different types of hardware that need to be addressed. Besides, the environment composition will very likely evolve over time, making the software developed for the initial environment deprecated and updates or reconfiguration actions are required during usage.

Software deployed in such an environment should be sufficiently dynamic to cope with new environment configurations, even while the system is in use. Tools and techniques from the model-based user interface development (MBUID) methodology ensure the interactive software is tailored according to the user needs and goals, but are often not suitable for highly dynamic environments.

With new approaches, such as pro-active/mixed-initiative interfaces and semantic modelling of the user environment we try to overcome the current limitations of MBUID.