UML for automotive multi–modal HCI

Marcel Dausend, André Berton, Jonas Zilles
Daimler Research & Development
Wilhelm–Runge–Strasse 11
89081 Ulm, Germany
{marcel.dausend, andre.berton, jonas.zilles}@daimler.com

ABSTRACT

Multi–modality can help to ease operation and avoid driver distraction, but is hard to specify and in turn makes specification more complex. Existing HCI modeling approaches lack compatibility to the Unified Modeling Language standard. They neither follow formal semantics nor provide methods addressing multi–modality. We base our approach on a formal semantics resolving ambiguities of the UML. We introduce new stereotypes for multi–modality and present a modeling method and tool support, both following our semantics. Proof of concept is given by an automotive multi–modal dialogue specification.

1. BACKGROUND

Up to now, no standard method for modeling complex multi–modal Human–Computer Interactions (HCIs) in the car exists . There are two major fields of studies addressing multi–modal dialogue specifications. First, the research community provides transformation–based approaches using Task Trees for HCI modeling, usually for multi–modal web applications. This method is not adequate for task–specific HCI models, where task–specific dialogues differ for each modality depending on the given context. Second, important contributions deal with modeling dialogue flows in Unified Modeling Language (UML)[3]. Existing methods integrate multi–modal aspects by non–UML extensions and lack of formal described semantics as requested by Broy[1].

2. MODELING APPROACH

Our work is based on a UML state machine semantics using Abstract State Machines (ASM)[2]. This semantics description resolves semantic ambiguities of state machines and gives a formal description of their behavior execution. The modeling method comprises three main steps: modeling the system components, operations and data using class diagrams, describing the task model in a hierarchical manner and modeling the multi-modal HCI behavior using state machines. The class diagrams define the context for the behavior models and assure a consistent design process. We decompose the HCI using task models and derive user operations and user triggered events from those models. These operations and events are added to the class diagrams to extend the behavior execution context. The integration of modality–specific content in a UML conform way requires enriching the UML. Our modeling method uses stereotype definitions for specific semantic constructs: speech recognition grammar, prompt and description of the graphical user interface. These stereotypes collect specific attributes for each semantic construct. While modeling, they can be assigned to any state in any combination, e.g. prompt and speech grammar.

We provide tool support for graphical modeling, verification and simulation of multi–modal state machine models. We specified multi–modal in–car dialogues to prove our approach. We conclude that a UML–conform specification of in–car HCI is possible and adequate regarding multi–modality.

3. CONTRIBUTIONS

Our work is based on a formal semantics so that we provide a clearly defined base for modeling as well as support automatic processing of our models, e.g. code generation. We ensure UML compatibility also for our multi–modal extensions. This encourages reuse of UML–based methods for verification and validation of HCI models. Our tool follows the ASM semantics and supports simulation of multi–modal UML models, which enables validation and comparisons of different HCI concepts already in an early stage of design. We proved our modeling concept by specification of multi–modal in–car HCIs including tool–based verification and validation by simulation.

4. FUTURE WORK

We will further validate our method by case studies. We plan to introduce enhancements like model–based dialogue quality measurement using model checking to offer support for development and test of multi–modal in–car HCIs.

5. REFERENCES

[1] M. Broy. Challenges in Automotive Software Engineering. *Proceedings of the 28th ICSE*, pages 33–42, 2006.

[2] M. Dausend. Entwicklung einer ASM-Spezifikation der Semantik der Zustandsautomaten der UML 2.0. Master's thesis, Universität Ulm, 2007.

[3] OMG. Superstructure (Version 2.1.2), 2007.