Model-based Measurement of Human-Computer Interaction in Mobile Multimodal Environments

Pedro Luis Mateo Navarro  
Cátedra SAES Laboratories  
University of Murcia  
Ed. Rector Soler, 30.071  
Murcia, Spain  
pedromateo@um.es

Stefan Hillmann  
Quality and Usability Lab,  
Telekom Innovation Laboratories, Technische Universität Berlin  
Ernst-Reuter-Platz 7, 10587 Berlin, Germany  
stefan.hillmann@tu-berlin.de

ABSTRACT
The complexity in developing and evaluating user interfaces has been extremely increased in the last few years, because more and more devices offer capabilities for multimodal interaction. This applies in particular to mobile devices like smartphones and tablet computers. An existing parameter set, aimed at describing aspects of various modalities, was extended and modified to obtain a formal, seamless and generic model of multimodal interaction. This new model is used for run-time and offline analysis of multimodal human-computer interaction. As proof of concept we also developed the Android HCI Extractor. This tool is used to quantify multimodal interaction within Android devices, and to create instances of the proposed model for further analysis and live decision. An example of this tool running on a real application is also described.

Author Keywords
human-computer interaction; multimodal systems; usability evaluation; interaction modeling; run-time models.

ACM Classification Keywords
H.5.2 User Interfaces: Evaluation/methodology; D.2.2 Design Tools and Techniques: User interfaces

General Terms
Design; Experimentation; Human Factors; Measurement; Standardization.

INTRODUCTION
The complexity of interaction with mobile devices has been extremely increased in the last few years mainly due to two chief causes. First, the nature of interactions with such devices has changed from a simple task-directed one (navigation and command) to interactions previously reserved for personal computers (non-task-directed). Second, unimodal input and output (e.g. haptic and GUI) have changed to multimodal input (speech, touch and haptic) and output (speech and GUI) of current state-of-the-art mobile devices. Popular examples are Android and iOS (especially with Siri) based devices. This “new” interaction requires new methods and techniques to implement analysis and evaluation processes from a more dynamic perspective, and regardless the modalities used. These needs, as well as the success [2, 3] in sales of such devices, lead to new challenges in usability engineering and in usability evaluation. Thus, the main question of our work related to this paper was: What is the design of an dynamic model that formally describes multimodal interaction?

In this context we designed and implemented a technique for modeling and analyzing multimodal human-computer interaction to support the usability lifecycle [8] when developing multimodal systems. In order to evaluate such systems, a new formal interaction model, which based on a set of parameters describing multimodal interaction, was designed. Instances of this model are used to dynamically represent the interaction between a user and a system, and are used as the basis for usability evaluation, comparison, transformation, and real-time decision processes. In the following section we will introduce the ideas behind the parameter set and the designed model. A new instrumentation tool called Android HCI Extractor and its usage are presented afterwards. Finally, we give a conclusion and remarks concerning future work.

MODELING MULTIMODAL HUMAN-COMPUTER INTERACTION
The proposed model design (depicted in Figure 1) holds multimodal interaction data (turn content, metacommunication, I/O information, and modality description). The design is centered around the concepts of turn and dialog. Whereby, “A dialogue consist of a sequence of turns produced alternatively by each party” of an interaction, and represents “One of several possible paths through the dialogue structure” of an application[7, p. 359 et sqq.]. Thus, data are collected by turn in order to provide a dynamic description of multimodal interaction.

The proposed model is based on a set of parameters aimed at quantifying multimodal interaction. We decided to base this new set of parameters on the well validated parameters and
concepts described in [6, 5]. Parameters which are conceptual equal among all modalities were generalized, in order to describe multimodal interaction on a more abstract level than previous work; e.g. spoken words in Spoken Dialog Systems and buttons, labels, etc. in GUI are generalized to elements. The resulting generic parameters’ set was extended by new parameters defining additional aspects of multimodal interaction; e.g. noise in a turn. On the other hand, more specific parameters were used to describe peculiarities of each modality; e.g. number of inserted words in a speech recognition error. Moreover, new parameters were defined to improve the quantification of interaction within GUIs; e.g. percentage of use of text and pointing devices.

The new parameter set $P$ contains three distinct classes of parameters: only observable ($P_o$), only computable ($P_c$), and those observable as well as computable ($P_{oc}$). Therefore, it is given that $P = P_o \cup P_c \cup P_{oc}$. In a second step we developed a data model containing only the elements of $P_o$, and such data which are necessary to compute $P_c$ and $P_{oc}$. As a consequence, the data model $P_{dm}$ was as small as possible, and thus a good basis for concrete implementation. For the implementation, we transferred the data model into a design in the Eclipse Modeling Framework ([111]), which provides model transformation and automatic code generation functionality.

The proposed design of the model provides the three benefits to practitioners and researchers evaluating the usability of human-machine interfaces:

1. Describing multimodal interaction using generic/abstract parameters to allows analyzing different modalities with the same metrics. More specific parameters are used to describe particular details of a specific modality.

2. Having the same metrics, based on formally well described parameters, to describe interaction in different contexts allow developers and designers to easily compare among different interaction records.

3. The turn-based design provides a “step by step” description of multimodal interaction. A relationship between collected data and time is created, providing evaluators with new opportunities for the dynamic analysis of interaction.

The parameters used in our model to quantify multimodal interaction are classified following the classification in [4], extended by a set of modality-related parameters:

- Dialog- and Communication-related parameters (e.g. turn duration, number of communication elements)
- Modality-related parameters (e.g. modality used by the user or the system in a specific time)
- Meta-communication-related parameters (e.g. number of correction turns, speech or gesture recognition errors)
- Cooperativity-related parameters (e.g. contextual appropriateness of system output)
- Task-related parameters (e.g. task success)
- Input-related parameters (e.g. words in a user utterance, number of pointing actions)
- Output-related parameters (e.g. number of elements which are shown on the screen)
To show the validity of the model described above, we implemented a tool to record multimodal interactions into instances of the model. Despite the model can be used potentially within any platform —the design was made in EMF and the implementation was done in Java— we chose Android because its open-source nature and its availability for different multimodal devices as smartphones, tablets, smart tv, in-vehicle systems, etc. Thus, the Android HCI Extractor (AHE) is aimed at analyzing the interaction between the user and the system within mobile multimodal environments in the Android platform. It collects data during interaction, which are used to create run-time instances of the model. These instances can be accessed both, while the user is interacting with the system, or once the interaction is finished.

The AHE keeps track of user actions and system responses. It incorporates a set of event-filter modules into the target application. Filter deploying and interaction analysis processes are performed automatically, and are completely transparent to the user interacting with the Android application. The valuable data included in such events are thoroughly analyzed and used to quantify the exchange of information between the user and the system. The AHE collects interaction data according to the parameters described in the previous section, and use them to create instances of the model. To ease the creation of run-time model instances, the AHE was used along with the instantiation framework described below.

The instantiation framework (see Figure 2) was designed and implemented to ease the integration of our model into research and production systems. Interaction is notified “atomically” to the framework; e.g. the user made a click, or the system provided five feedback words. Such actions are automatically translated into internal events that are analyzed by a set of event listeners, each one specialized in a particular aspect of multimodal interaction. Then, these listeners use the received information to properly and progressively fill the model instances. This framework also manages the life-cycle of the resulting instances. They can be accessed at run-time during the interaction process (e.g. to implement user monitoring), or once it is finished (e.g. for a usability evaluation).

A further approach is the usage of recorded instances of the model as basis of a model transformation processes. Therefore, recorded instances were transformed in a structure aimed at analyzing users’ peculiarities when using touch modality. The results were used to evaluate the efficiency of touch modality within the experiments we conducted.

The AHE is implemented in Java, as a library that can be easily integrated and deployed into Android 2.1 (or higher) compatible applications. It is provided as open-source, and can be downloaded at [1]. The library provides a set of handler-methods which are called when some interaction event occurs. These methods can be used by developers to implement their own multimodal interaction analysis. A tutorial to integrate and use this tool within Android applications can be found on the Cátedra SAES website [9]. Please, note that the AHE is in an early stage of development, thus its implementation is in constant evolution right now.

**CONCLUSIONS AND FUTURE WORK**

The complexity of current user interfaces poses new challenges in usability engineering and evaluation. This work proposes a model-based approach aimed at describing multimodal interaction dynamically and formally, and treating all
modalities at the same level of abstraction. The model design is based on a set of parameters aimed at quantifying multimodal interaction. Instances of such model can be used to implement further analysis, comparison, transformation, and real-time decision processes. The Android HCI Extractor is able to log automatically many of the model’s parameters, but not all (e.g. errors in the ASR.) The former can be used to create model instances at run-time.

As a statement of direction, we would like to take full advantage of the different possibilities of the model (system comparison, model transformation, run-time decision, etc.) by integrating it into other projects. We would like to refine its design, as well as adding specific parameters to analyze peculiarities of other modalities (e.g. gestures.) Improving the AHE’s compatibility and stability is also among our goals.

Acknowledgments
We thank Christine Kühl for her support and advice concerning parameters describing multimodal interactions. We thank Gregorio Martinez, Diego Sevilla and Sebastian Möller for their comments on the AHE implementation. This work was partially funded by SAES (www.electronica-submarina.com) and Telekom Innovation Laboratories (www.laboratories.telekom.com).

REFERENCES
1. Android HCI Extractor.
5. ITU-T Suppl. 25 to P-Series Rec. Parameters describing the interaction with multimodal dialogue systems, 2011.
   http://www.catedrasaes.org/trac/wiki/MIM.