Requirements for Applying Simulation-Based Automated Usability Evaluation to Model-Based Adaptive User Interfaces for Smart Environments

Michael Quade, Andreas Rieger, and Sahin Albayrak

Technische Universität Berlin, DAI-Labor
Ernst-Reuter-Platz 7, 10587, Berlin
{michael.quade,andreas.rieger,sahin.albayrak}@dai-labor.de

Abstract. Users in smart environments benefit from context-aware applications that are able to adapt their user interfaces (UI) to specific situations. In the same way as the development of adaptive applications poses high demands on the designers, the evaluation of their usability also becomes more complex and time consuming because the context of use and different adaptation variants need to be considered. While automated usability evaluations cannot fully replace user tests in this domain, they can be applied to multiple adaptation variants at an early stage of development and thus reduce time and complexity. This paper presents general requirements for applying automated model-based usability evaluations that apply simulated user interaction as an approach to evaluate UIs of adaptive applications based on the underlying development models.

Keywords: automated usability evaluation, adaptive user interfaces, model-based UI development, smart environments.

1 Automated Usability Evaluation of Adaptive User Interfaces for Smart Environments - Benefits and Challenges

The main goal of smart environments is to assist users within their daily routines whether at work or at home. Smart environments are characterized by networked applications capable of coping with different situations that can be captured via integrated sensor systems. Usually, this is achieved with the help of adaptive applications that provide user interfaces (UI) which adapt to (predefined) situations within the observed context of use [7]. As a main challenge, adaptive applications need to present required information properly and tailored to the current users' needs and (dis-) abilities which is a complex task when dealing with many potential adaptations. Further, this high complexity also leads to problems in fully evaluating the usability of adaptive applications with user tests due to the state explosion problem [18]. Even though this would usually provide the best evaluation results, the required costs and time tend to become limiting factors for comprehensive user testing.

Understanding the formalization of interaction means and concepts of adaptive applications remains a main issue when evaluating usability. One way to address this

N. Streitz and C. Stephanidis (Eds.): DAPI/HCII 2013, LNCS 8028, pp. 235–244, 2013.

issue during development is integrating models of the application and models of the user as proposed by model-driven engineering and model-based usability evaluation. On the one hand, models of the application are able to formalize the design, express the underlying concepts and make them interpretable by machines [17]. On the other hand, user models are commonly used to describe users' physical and cognitive abilities [9] and to formalize different groups of users based on these attributes. Above all, the interconnection of both approaches can be utilized for providing adaptations to the UI and to evaluate the usability and accessibility for different groups of users. Especially for this purpose, automated usability evaluation (AUE) emerges as a paradigm allowing detailed and at the same time cheap testing of usability [6]. Most AUE approaches are using predictive analytical modeling and predictive simulation methods. Based on underlying psychological theories, concepts and models, these approaches have proved to correctly predict criteria relevant for judging an application's usability [6]; e.g. automated simulation of interaction paths, execution time predictions, cognitive load and learning time estimations.

However, there still exist main barriers to the adoption of AUE by the interaction design industry and specifically within the domain of smart environments. On the one hand, current AUE approaches require additional specific descriptions of the user, the UI and the tasks. In most cases, such descriptions of the UI and tasks do not exist or cannot be automatically derived from the final UI. For this reason, the required input (e.g. models) needs to be provided by the designers themselves, which is a time consuming and potentially error-prone task. On the other hand, most AUE approaches are hard to apply for complex tasks and more general usability evaluations. Further complicating is the fact that the context of use needs to be determined for an evaluation of adaptive applications, especially as it may change during interaction.

Initial work in the field of applying AUE to model-based UI development and adaptive user interfaces has been demonstrated. While [1] describes how usability evaluations in general can be applied to UIs stemming from a model-driven engineering process, an AUE was solely done on the code level and thereby lacking the benefits of using the development models which would reduce the effort. In [4] a model-based runtime framework for user interfaces is combined with a semi-automated workbench on the level of the final UI but does not involve the underlying development models and adaptation capabilities. Both approaches already address parts of combining model-based development with AUE but do not fully take advantage of the potential benefits or lack adaptivity capabilities.

In this paper, we examine the underlying basic requirements that have to be fulfilled for applying simulation-based automated usability evaluations on the same models that are already designed and implemented during the development process of adaptive applications within smart environments. At first, we start by narrowing the scope of applicable AUE methods to simulation-based approaches and lead over to specific requirements for such approaches during development of adaptive UIs in smart environments. Finally, we conclude this paper with a summary and give an outlook on our current and future work within this domain.

2 Why Apply Simulation-Based Automated Usability Evaluation to Adaptive User Interfaces?

As a matter of fact, different usability evaluation methods are suitable to predict and uncover different types of usability attributes. Hassenzahl et al. [5] distinguish these usability attributes into pragmatic and hedonic attributes. While the latter are mainly related to aspects of User Experience (UX); e.g. novelty and beauty of a design; they can usually only be provided with the help of extensive user tests and questionnaires. It is hard to predict hedonic attributes using AUE as these methods allow reasoning about human performance measurements mainly [6], which fall into the category of pragmatic attributes. Consequently, hedonic attributes should be out of scope when applying current AUE methods. However, quantitative and qualitative usability criteria can be applied when predicting pragmatic usability attributes; such as interaction execution time, number of required interaction steps and uncovering interaction errors by tracing the interaction path.

A simulation-based AUE method is essential in order to automate the interaction process and thereby gather a variety of different interaction paths by minimizing the effort involved. Especially designers of adaptive applications profit from such an approach, because they do not need to provide the interaction paths by hand for each possible adaptation of the UI and thereby tackle the state explosion problem. However, this does not exclude the possibility to provide a predefined interaction path in case a specific solution needs to be evaluated in more detail.

Simulation-based AUE approaches require specific input for conducting the evaluation process. This relies on the fact, that each targeted evaluation criteria, which defines how the outcome of the simulated interaction process is evaluated, can only be applied if the input information is available to the appropriate AUE method (Fig.1). Hence, this dependency between the chosen criteria for evaluation, the applicable simulation-based AUE approach and the required input-information is of high importance for the development and evaluation process.

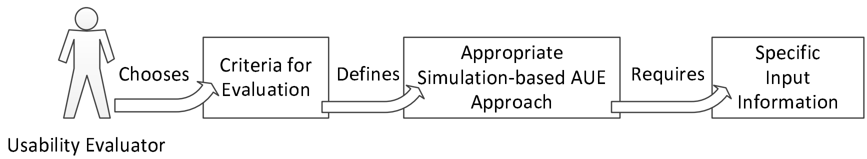

Fig. 1. Appropriate input-information for the applied AUE method is required

3 Requirements and Benefits of Simulation-Based Automated Usability Evaluation for Adaptive User Interfaces

In this section we derive basic requirements for a simulation-based approach of AUE during development of adaptive user interfaces for smart environments. For this purpose we identify and provide detailed information about four basic factors:

- **Application factors**, such as UI information and interaction logic,
- **Context of use factors** that influence the interaction and the adaption process,
- **User factors** that are relevant for simulating user behavior, and
- **Task factors** (or a set of goals) which the simulated user tries to fulfill.

3.1 Application Factors

A formal description of the application needs to be available because simulation-based AUE rely on abstract interactions between user and application (models). As depicted in Fig. 2, information about the UI and its interaction logic is required and needs to be provided in a computer-processable way. For example, when applying GOMS-based usability evaluations [6] to graphical user interfaces, this usually comprehends all visible UI elements along the path of interaction and their specific attributes; e.g. type of the UI elements and their size and position on screen. Thus, the first basic requirement is:

- **(Req. 1)** Simulation-based AUE need to represent application-specific UI information for simulating their effect on the interaction process.

Further, for a simulation-based AUE it is essential to establish a connection between the UI elements and the task the user is currently performing in order to automatically simulate user interactions and the according system behavior. This implies that it has to be traceable what happens next if a specific UI element is activated; e.g. by clicking a button the next UI mask gets activated. Hence, this interconnection allows reasoning about the effects of using UI elements for a specific purpose within the current task. Current approaches for model-based UI development make use of executable UI and task models and thus are well-suited to provide this functionality; e.g. [14, 3]. This interconnection between UI and interaction concept builds the basis for the next requirements:

- **(Req. 2)** Simulation-based AUE for adaptive user interfaces requires access to interaction capabilities of the UI elements and their purpose for specific tasks.
- **(Req. 3)** Simulation-based AUE for adaptive user interfaces needs access to the application's follow-up states after simulated user interaction or changes in the context of use occurred.

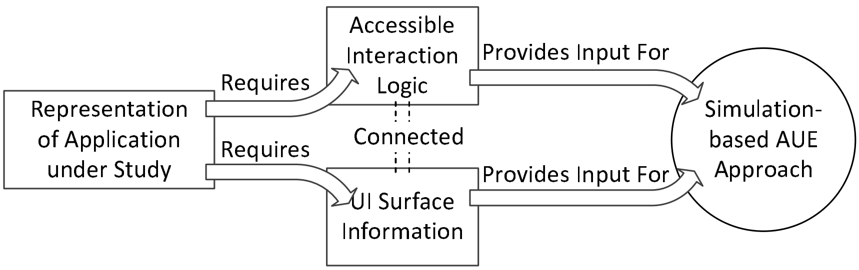

Fig. 2. The application's UI information and interaction logic serve as input for AUE methods

3.2 Context of Use Factors

In case of adaptive applications for smart environments, the system may actively take the initiative or respond to user input depending on the observed context of use. This is leading to a multitude of unique situations for which an appropriate system response needs to be ensured. Usually, an adaptation engine handles the analysis of these situations and applies appropriate adaptations. Thus, if an adaptation engine and a representation of the context of use, that can be edited and simulated, are available, simulation-based evaluation can be applied to extensively test possible interaction and adaptation paths in context-aware systems.

In general, the context of use [13] is distinguished into information about the *user environment*, the *computing environment* and the *physical environment* (see Fig. 3).

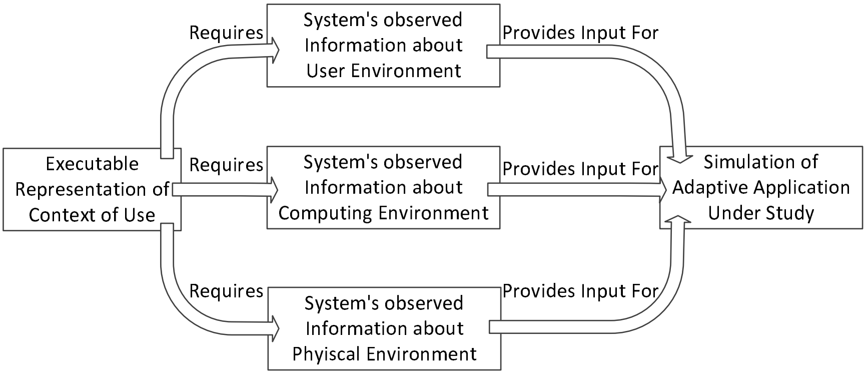

Fig. 3. The representation of the context of use needs to include the sensed information about the user, the computing environment and the physical environment in order to simulate unique situations to which the application under study can adapt to.

The information about the *user environment* includes all relevant information that can be sensed by the context-aware system via its sensor systems and internal representations via a user profile. Depending on the integrated sensor data, this may include the current location of the user and other people as well as historical and social data from a user profile. All of this information is required for the simulation-based interaction process in order to trigger required adaptations.

Information about the *computing environment* needs to include at least the relevant information about available interaction devices (for input and output); e.g. keyboard, mouse, touchscreen or display. Some AUE approaches require this information to be integrated into the information about the application under study because in some cases no sharp distinction can be made between software and hardware components. By providing this information further results can be expected from the AUE process, as e.g. GOMS-based approaches include extra time for switching between different interaction devices which might provide additional insights for the designer.

Information about the *physical environment* may include surrounding factors; e.g. acoustic or lighting conditions. In case of adaptive applications, the modeled surrounding factors need to include at least those which are sensed by the context-aware system and are source for potential adaptations; e.g. simulated movements and noise. Further side effects can be included if this data from the outside world is available to the usability evaluation. Consequently, we state the next requirement as following:

- **(Req. 4)** Simulation-based AUE for adaptive applications requires a representation of the context of use that can be edited and simulated to evaluation needs in order to create different situations to which the application can adapt to.

3.3 User Factors

Information about users, which is relevant for the simulated interaction and thus also for the AUE, needs to be considered, because different users may interact differently with the same application. This representation of the user may be partially similar, but usually differs from the information that is sensed by the adaptive application for adaptation purposes (compare Fig. 3 and Fig. 4). More specifically, the user factors described here focus on information required for the interaction and the evaluation process from the user's perspective.

In order to provide a wide basis for potentially applicable AUE methods two more requirements have to be addressed, which can be divided into modeling of the user's expertise and representing the user's abilities for interaction.

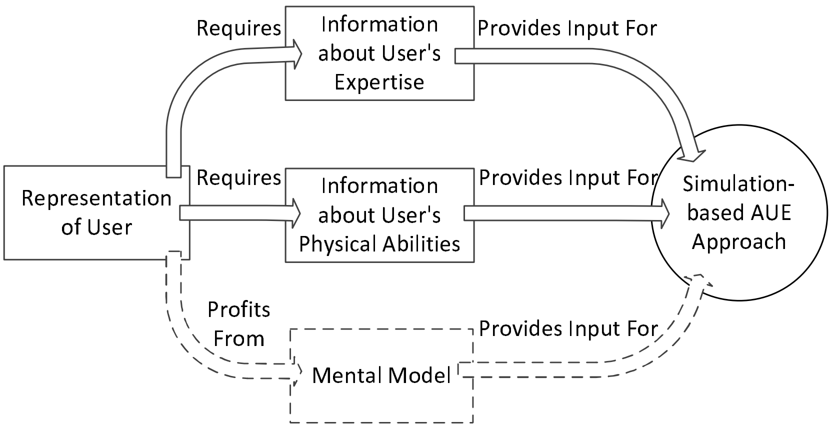

Fig. 4. The representation of the user requires information about the expertise and physical abilities for the evaluation process and profits from a mental model to reason about performed actions and their consequences to the application

On the one hand, users may differ in their expertise regarding the application and its UI, but also regarding the domain of the task. On the other hand, there exist adaptive applications that provide different user interfaces depending on the availability of this information. So basically, user expertise should be simulated in order to evaluate its effect on the interaction process on the user side and on the system side. However, most AUE methods expect that the user is an expert in the domain of the application and therefore optimal interaction paths are evaluated only. Just a few approaches model novice users, but then expect that the user has no knowledge of the domain at all; e.g. [15]. A main difference between modeling expert users and novice users is that in the latter case interaction problems due to a wrong understanding of the application can be accounted for. Unfortunately, errors due to a lack of experience with the application or domain are hard to predict. An exception is the simulation of browsing behavior, where category labels are evaluated based on semantic similarity between goal concepts and UI element labels; e.g. as in [15, 2]. On the other hand, known errors can also be modeled and their consequences can be evaluated using a simulation. This applies to cases, when e.g. the error type and its preconditions are known, but the design cannot easily be inspected manually for all adaptations due to complex adaptation rules; see e.g. [13]. A practical application of an AUE method is then to develop knowledge about error types and their precondition, either in a general way, or during the task analysis phase preceding the design of the actual application.

Another facade of the user's expertise is a mental model [10] which reflects beliefs about the application's behavior and the outside world. Such a mental model is especially useful when comparing the actual outcome of actions to the intended outcome and thus helps a user model to notice that an error occurred and may affect the following behavior; e.g. recovery strategies or canceling the interaction.

Finally, different users may have different preferences regarding interaction devices and techniques (e.g. using shortcuts) which should be accounted for as well when using simulation-based evaluation. These preferences could be included into the representation of the user (expertise) and then have an influence on the chosen actions during simulation. Thus, the fifth requirement reflects the user's expertise:

- **(Req. 5)** Simulation-based AUE for adaptive applications requires information about the user's expertise to account for individual behavior and profits from a mental model to account for expected and perceived results of actions.

Besides the user's expertise, further factors are required for a more beneficial combination of AUE and adaptive applications (see Fig. 4). An example gives the evaluation of GUIs for users with special needs and abilities, such as visual or motor impairments [8, 16]. This additional information allows to simulate different user groups and to respectively evaluate the effects of:

– different layout variants or adaptations in combination with information from the application's UI surface and interaction logic ; or
– different interaction devices and surrounding effects in combination with information from the context of use.

Consequently, AUE profits from a clear modeling of different user groups based on abilities, as it allows to reason about the effects of different adaptations based on the modeled abilities. We therefore state the sixth requirement as following:

- **(Req. 6)** Simulation-based AUE for adaptive applications requires a representation of the user's abilities to allow reasoning about their effects on the interaction process and the application's capabilities to cope with different users.

3.4 Task Factors

User interaction usually serves one or more specific goals which users try to achieve during interaction. Consequently, a description of the users' goals is also required in order to evaluate an application's usability.

Goals for interaction can be specified in form of a task which the user wants to perform. When conducting usability tests with real users, goals are usually predefined and the participants receive a description of the task to perform and (in most cases) a clear description when the goal is achieved. Like real user tests, automated usability evaluations based on simulated interaction require such predefined tasks.

If the task is to be used in simulations, the actions performed by the user to reach the goal state can be provided within the task description (Fig. 5). This would be similar to a step-by-step walkthrough which is applied by most predictive analytical modeling approaches. However, in case of some automated usability evaluations, these steps are not contained in the task description or it is not desired to have this information in advance; e.g. when evaluating novice users and their browsing behavior when looking for specific information (see e.g. [15]). Instead, the required steps have to be determined on the fly based on information describing how users would try to proceed; e.g. with the help of rules describing user behavior [12], available knowledge or further semantic information required to fulfill a task. Thus, as a final requirement regarding the task for simulation and evaluation we state:

- **(Req. 7)** Simulation-based AUE for adaptive applications requires a task description with information relevant for fulfilling the task, whether as a list of actions to perform or via an integrated solution approach.

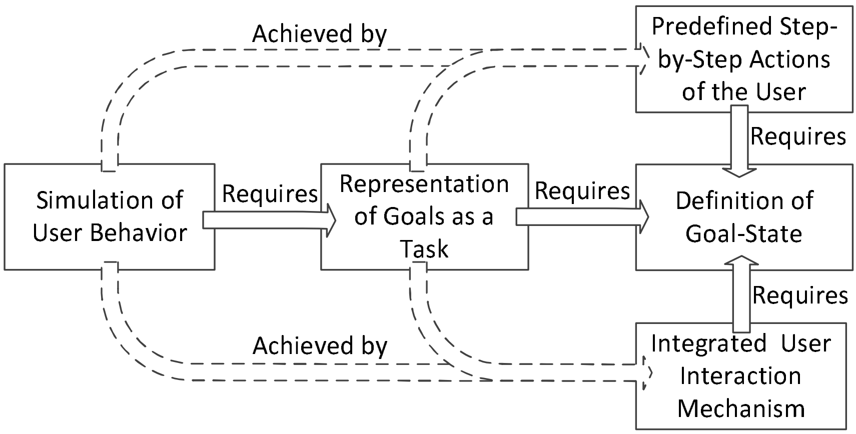

Fig. 5. The representation of the task for automated usability evaluations based on simulated user interaction requires a clear definition of the goal-state and all actions to perform or, if available, an integrated solution approach for simulating user interactions

4 Conclusion

In this paper we introduced seven generic requirements for applying simulation-based automated usability evaluations to adaptive user interfaces within the domain of smart environments. We have explained the necessity for each requirement and expected results if each of these requirements can be fulfilled.

A current state of implementation and exemplary evaluation results for such an approach that uses executable UI models stemming from a model-based runtime framework and a semi-automated usability workbench is described in [11]. Further, we are investigating the benefits of applying AUE to executable development models of adaptive user interfaces in more detail by conducting user tests within a testbed that provides a smart environment. By monitoring the effort involved and the gathered evaluation results from these user tests and comparing them to the effort and the results of the AUE conducted with the help of the development models we intend to give more insights on the benefits of the described approach in terms of saved time and ratio of uncovered usability issues.

References

1. Abrahão, S., Iborra, E., Vanderdonckt, J.: Usability evaluation of user interfaces generated with a model-driven architecture tool. In: Maturing Usability. Human-Computer Interaction Series, pp. 3–32. Springer, London (2008), http://www.springerlink.com/content/t8861113t2764657/
2. Blackmon, M.H., Kitajima, M., Polson, P.G.: Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. In: CHI 2005: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 31–40. ACM, New York (2005)
3. Blumendorf, M., Lehmann, G., Roscher, D., Albayrak, S.: Ubiquitous User Interfaces: Multimodal Adaptive Interaction for Smart Environments. In: Multimodality in Mobile Computing and Mobile Devices: Methods for Adaptable Usability, pp. 24–52. IGI-Global (2009)
4. Feuerstack, S., Blumendorf, M., Kern, M., Kruppa, M., Quade, M., Runge, M., Albayrak, S.: Automated usability evaluation during model-based interactive system development. In: Forbrig, P., Paternò, F. (eds.) HCSE/TAMODIA 2008. LNCS, vol. 5247, pp. 134–141. Springer, Heidelberg (2008)
5. Hassenzahl, M., Schöbel, M., Trautmann, T.: How motivational orientation influences the evaluation and choice of hedonic and pragmatic interactive products: The role of regulatory focus. Interacting with Computers 20(4-5), 473–479 (2008), http://dx.doi.org/10.1016/j.intcom.2008.05.001
6. Ivory, M.Y., Hearst, M.A.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. 33(4), 470–516 (2001)
7. Jameson, A.: Adaptive interfaces and agents. In: Sears, A., Jacko, J.A. (eds.) The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, 2nd edn., pp. 433–458. CRC Press, Boca Raton (2008)
8. Keates, S., Clarkson, J., Robinson, P.: Investigating the applicability of user models for motion-impaired users. In: Proceedings of the Fourth International ACM Conference on Assistive Technologies, Assets 2000, pp. 129–136. ACM, New York (2000)

9. Kobsa, A.: Generic user modeling systems. User Modeling and User-Adapted Interaction 11(1-2), 49–63 (2001)
10. Norman, D.A.: Some Observations on Mental Models. In: Mental Models, pp. 7–14. Erlbaum, Hillsdale (1983)
11. Quade, M., Lehmann, G., Engelbrecht, K.P., Roscher, D., Albayrak, S.: Automated usability evaluation of model-based adaptive user interfaces for users with special and specific needs by simulating user interaction. In: Martín, E., Haya, P.A., Carro, R.M. (eds.) User Modeling and Adaptation for Daily Routines. Human-Computer Interaction Series, vol. 9, pp. 219–247. Springer, London (2013)
12. Ruß, A., Quade, M., Kruppa, M., Runge, M.: Rule-based approach for simulating age-related usability problems. In: Wichert, R., Eberhardt, B. (eds.) Ambient Assisted Living. Advanced Technologies and Societal Change, vol. 2, pp. 149–166. Springer, Heidelberg (2012)
13. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US (1994)
14. Sottet, J.S., Calvary, G., Coutaz, J., Favre, J.M.: A Model-Driven Engineering Approach for the Usability of Plastic User Interfaces. In: Gulliksen, J., Harning, M.B., van der Veer, G.C., Wesson, J. (eds.) EIS 2007. LNCS, vol. 4940, pp. 140–157. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-92698-6_9
15. Teo, L., John, B.E.: The evolution of a goal-directed exploration model: Effects of information scent and goback utility on successful exploration. Topics in Cognitive Science 3(1), 154–165 (2011)
16. Trewin, S., Pain, H.: Keyboard and mouse errors due to motor disabilities. Int. J. Hum.-Comput. Stud. 50(2), 109–144 (1999), http://dx.doi.org/10.1006/ijhc.1998.0238
17. Vanderdonckt, J., Guerrero-Garcia, J., González-Calleros, J.M.: A model-based approach for developing vectorial user interfaces. In: Proceedings of the LA-WEB 2009 (2009)
18. Goldsby, H.J., Cheng, B.H.C., Zhang, J.: AMOEBA-RT: Run-time verification of adaptive software. In: Giese, H. (ed.) MODELS 2008. LNCS, vol. 5002, pp. 212–224. Springer, Heidelberg (2008)