



HAL
open science

ViSE - A Virtual Smart Environment for Usability Evaluation

Stefan Propp, Peter Forbrig

► **To cite this version:**

Stefan Propp, Peter Forbrig. ViSE - A Virtual Smart Environment for Usability Evaluation. Third IFIP WG 13.2 International Conference on Human-Centred Software Engineering (HCSE), Oct 2010, Reykjavik, Iceland. pp.38-45, 10.1007/978-3-642-16488-0_4 . hal-01055197

HAL Id: hal-01055197

<https://inria.hal.science/hal-01055197>

Submitted on 11 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

ViSE – A Virtual Smart Environment for Usability Evaluation

Stefan Propp, Peter Forbrig

University of Rostock, Institute of Computer Science,
Albert Einstein Str. 21, 18059 Rostock, Germany
{stefan.propp, peter.forbrig} @ uni-rostock.de

Abstract. Within the research field of HCI task models are widely used for model-based development of interactive systems. Recently introduced approaches applied task models further to model cooperative behaviour of people interacting in smart environments. However there is a lack of usability methods to support the needs of evaluations during a model-based development process for smart environments. Particularly during early stages of development building a prototypical environment for user evaluations is resource consuming. To overcome the challenges we present a process model and according tool support. We provide the virtual smart environment ViSE to conduct expert evaluations and user studies during a user-centred design process, supporting iterative evaluations.

Keywords. Model-based Usability Evaluation, Task Models, Smart Environment

1 Introduction

According to Weiser's vision [9] of ubiquitous computing, devices are weaving themselves into everyday life, allowing people to fully concentrate on performing their tasks, while hiding complexity of necessary devices. A smart environment (SE) recognizes user behavior and provides assistance to achieve the users' objectives. For instance within a meeting scenario the presenter should concentrate on the talk, while the SE assists by adjusting the projector and capturing audiovisual data for meeting documentation if needed.

While offering a higher degree of comfort, also new challenges are introduced. In contrast to desktop computing, where a user focuses the attention on a single device, in a SE multiple devices influence user behavior. Reflected from a task-oriented point of view, a certain task can be started at one device, while being finished at another device. For instance during a discussion some comments are quickly typed into a PDA, being revised later back in the office at a notebook. Different user interactions lead to a wide variety of options [7] for the design of the SE, comprising modalities (like speech or gestures), device selection (like direct access by a user or dynamic

selection by the SE) and the initiative (explicit or implicit). An example for explicit initiative is a user manually turning the light on via switch, whereas an example for implicit initiative describes a scenario where the SE senses that it is too dark and decides autonomously to turn the light on. As we can see, an interaction needs to be interpreted within a certain context. Contextual conditions like available devices, currently present users, light or temperature, may influence task performance for each user. Furthermore cooperation is an important aspect. Within SEs a certain task may be performed cooperatively by several users, who reflect certain roles.

These characteristics of SEs impose challenges for usability evaluation. An evaluation is particularly resource consuming. To conduct a user study within a real SE a completely functional prototype has to be set up, while evaluating a simulated environment needs effort for instance to prepare a “Wizard of Oz”-experiment.

For model-based development of SEs some approaches apply task models [1, 2, 3, 8, 10]. We suggest reusing these task models also for usability evaluation to integrate evaluation into development. Advantages are (a) the reduced effort to prepare evaluations rapidly at all stages of development, (b) an easier interpretation of task-based log files than of low level events and (c) the direct link back to the development models facilitates their improvement.

2 Related Work

Several recently published approaches apply task models for model-based development of SEs. Trapp et al. [8] define each device’s capabilities with a task model chunk. When a new device connects to the room infrastructure, the corresponding model chunk is added to the room task model. As a result combined functionalities may be offered. For instance a scanner and a printer may offer an additional copying functionality. Wurdel et al. [10] model behaviour of persons with task models and describe their collaboration. For instance, that person “A” has to finish the talk first to give person “B” the floor. Each task is described by a set of preconditions and effects on the environment. Feuerstack et al. [1] enhance the task modelling notation CTT to serve as model for runtime interpretation. Domain concepts are annotated and the object flow is modeled. Different users’ task models are synchronized with domain objects. Luyten et al. [3] focus on modelling distributed user interfaces for dynamic environments. Task and environment models are visualized to reduce complexity.

While several approaches for the development of SEs are available, there are only a few appropriate usability methods. Scholtz et al. [6] suggest a general framework with evaluation metrics to give a starting point for structuring the evolving evaluation techniques. Maly et al. [4] describe an approach of visualizing the user’s behaviour as 3D representation and allow experts to create artificial scenarios when sensor data is not available yet. However, task models are not used. In [5] we suggest a task model-based process for evaluating smart environments. Performed tasks are visualized within animated task models and serve for analysis. In this paper we extend the approach to build a virtual smart environment, which visualizes user interactions as a 2D representation of a physical SE.

3 Usability Evaluation Method

Software development is structured into several stages, comprising requirements analysis, design and implementation of the final product. Accordingly we describe a structured process of development and evaluation of SEs. This section presents the process model (section 3.1) and subsequently discusses the usage of ViSE to conduct expert evaluations (section 3.2) and user studies (section 3.3).

3.1 Process Model

The process model (fig.1) describes an approach for the iterative development and evaluation of SEs. During each development cycle three stages are considered.

(1) During planning stage the development methods and evaluation techniques are chosen. If any developed artifacts of a previous cycle exist, their evaluation results are taken into account.

(2) During development stage existing artifacts are refined and new artifacts are developed. Examples are requirements documents, models and source code.

(3) Iterative evaluations comprise direct feedback for the developed artifacts based on feedback of user studies and expert evaluations.

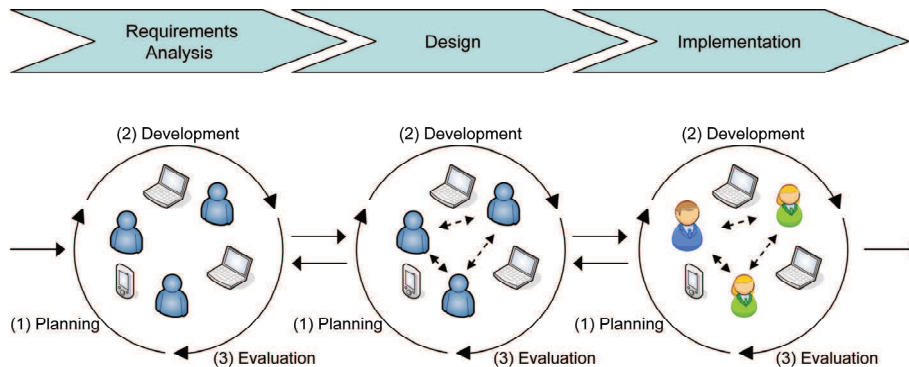


Fig. 1. Process model for design and evaluation of Smart Environments

Depending on progress within development different activities are carried out during each cycle, which should be subsequently outlined.

Requirements analysis. To elicit requirements a dialog with customers and prospective users is conducted. We identify user roles and create a task model for each role. A task model captures task hierarchies and their temporal relations. To provide additional insights (e.g. context dependencies) textual scenarios are captured, which may serve as test cases during evaluations.

During evaluation the consistency of the designed task models has to be analyzed. A task model animation tool allows interactively walking through the models.

Another challenge is to evaluate the users' stated assistance functionalities, which the smart environment should provide, like automatically switching lights on or showing presentation slides of a presenters' laptop. We provide a virtual smart environment (ViSE), which allows conducting "Wizard of Oz"-Experiments for SEs: An expert hides in another room, observing user interactions via video camera. The expert operates the devices within the physical environment remotely to mimic the envisioned assistance. Changing devices' states in ViSE allows changing the devices' states in the physical SE accordingly. Involved users get a feeling of the envisioned assistance and reflect their requirements

Design. Within the design stage task models have to be enriched with further details to more accurately reflect user behavior and to serve as initial models for implementation. Textual scenarios elicited during requirements stage contain additional details about cooperative dependencies between different users, like a chairman is required to give a presenter the floor. Further details comprise contextual dependencies for performing tasks, like having a laptop containing slides to give a presentation. These aspects are modeled in CTML ("collaborative task modeling language") [10]. Each task is annotated with preconditions and effects. A precondition formalizes the environments' state before a particular task can be performed. An effect describes the environments' state after a task is performed.

To ensure the consistency of the designed models, the virtual SE visualizes the modeled SE and allows an interactive walk-through as collaborative animation. Scenarios from requirements stage are animated by a usability expert and identified inconsistencies can be corrected.

Implementation. Based on the designed task models the SE's software has to be implemented. The envisioned SE assists users while performing their tasks and proceeds in three steps: firstly it senses user movements and further context information (e.g. location sensors to separate presenter and audience), secondly it infers the task currently performed (e.g. a person is beginning to present) and finally it triggers devices accordingly (e.g. a projector performs a system task "show agenda"). Within our physical SE this recognition of performed tasks is accomplished by a probabilistic behavior model, which is derived from previously specified task models as discussed in [2]. Finally the software is deployed at the devices within the environment.

To ensure an adequate integration of all software and hardware components, user studies have to be conducted. During evaluation a vast amount of sensor data within the SE is captured, leading to the challenge of a suitable interpretation of user behavior to discover usability issues. Our virtual SE allows playing captured data forth and back to interactively explore issues.

After having introduced the process model and provided tool support we take a closer look at expert evaluations during design stage and user evaluations during implementation stage.

3.2 Expert Evaluation of cooperative task models

After having specified a CTML design model its consistency according to the elicited requirements has to be evaluated. Fig. 2 depicts ViSE within an expert evaluation.

The left part of ViSE contains a birds' view of the environment, comprising users, devices (e.g. laptops and projectors) and other objects (e.g. pens and whiteboard erasers). Interactions with these entities allow changing the SEs' state and performing tasks. The following interactions are supported:

- Changing the persons' location
(e.g. moving to a an empty seat)
- Changing the persons' corpus orientation
(e.g. to look to the front, where the presenter is speaking)
- Attaching items to persons
(e.g. a presenter is taking up a laptop)
- Changing devices' states
(e.g. switching a laptop on)
- Establishing connections between devices
(e.g. connecting a laptop to a VGA port to connect it to the room infrastructure)

The upper right part contains the persons within the current scenario with their associated roles. For each role a task model is animated. A task can be performed if both the temporal relations and the preconditions are fulfilled.

The lower right part contains the preconditions for the current task and indicates which conditions are currently "true" and "false". For the task "present with projector" three preconditions are specified (see fig.2, lower right part):

- The person with the role "presenter" has to be located at the "PresentationZone".
- The person has to carry a device of type laptop or PDA.
- The persons' device has to be connected to a VGA port.

In the scenario in fig.2 requirements engineer Stefan fulfills all three preconditions and the temporal relation that he has already performed task "enters room". Hence he is allowed to perform the task "presents with projector".

If any preconditions are differing from the requirements, they can be edited textually to directly improve the evaluated artifacts. As an alternative preconditions can be edited graphically. A designer may move persons to a certain location and equip them with devices. A subsequent option "generate preconditions" allows updating the preconditions accordingly.

3.3 User Evaluation of a physical Smart Environment

After all components are set up, a user study can be conducted to evaluate the physical SE. Users are invited to perform tasks specified in a test plan as individual tasks or within a cooperative scenario. At the same time user interactions are captured: user movements via UbiSense location sensors, handled devices or other items via RFID tags, performed tasks via task recognition algorithms and video streams via cameras. Most data is already captured for normal operation of the SE. Hence it is available for evaluation.

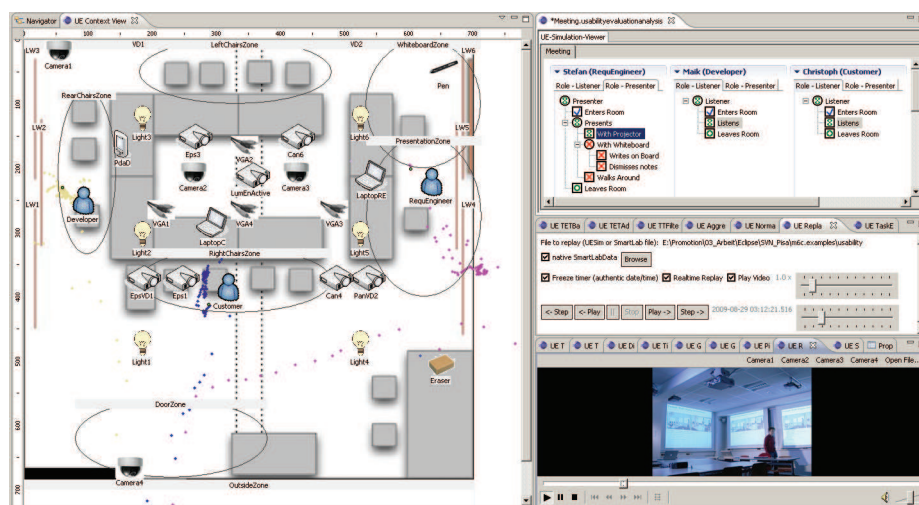


Fig. 3. Replay of sensor data with ViSE

ViSE was enhanced with a replay mode (fig. 3), which allows replaying captured sensor data with different animation speeds to provide insights into user interactions. A birds' view visualizes user movements, depicting locations of users and devices as according icons and way points reflecting past location sensor data with a particular focus on locations where tasks were performed. Graphically selecting users, devices or way points of user movements gives further information as tool tip (e.g. which task was performed at a certain way point) and filters sensor data. For instance spanning a selection rectangle over the presentation zone in front of the audience delivers an overview of all performed tasks within that area. During replay further expert observations can be annotated. For instance currently our physical SE recognizes only a limited set of tasks automatically. Additional tasks can be identified manually during video replay. According replay controls are depicted within a view at the right of fig. 3. Further views provide several logs and visualizations of sensor data on demand, for instance the progress of task performance as animated task model or gantt chart.

5 Conclusion

In this paper we have given an overview of model-based development of SEs and shown how to exploit designed task models for usability evaluation. A process model describes a process of iterative development and evaluation. The virtual smart environment ViSE provides tool support at requirements, design and implementation stage. In early development stages, when a physical SE for user studies is not available, ViSE allows conducting expert evaluations for SEs. After a physical SE is set up ViSE helps to cope with the vast amount of captured sensor data. We emphasize on providing an integrated view on task performance, a graphical presentation of user and device movements and video streams, for both presenting and interactively exploring usability data.

Future research avenues comprise gathering further real life experiences to improve the presented method and tools.

Acknowledgments. The work of the first author was supported by a grant of the German National Research Foundation (DFG), Graduate School 1424.

References

1. Feuerstack, S., Blumendorf, M., Albayrak, S.: Prototyping of Multimodal Interactions for Smart Environments based on Task Models. AMI'07 Workshop on Model-Driven Software Engineering, Darmstadt, Germany, 2007.
2. Giersich, M., Forbrig, P., Fuchs, G., Kirste, T., Reichart, D., Schuhmann, H.: Towards an Integrated Approach for Task Modeling and Human Behavior Recognition. HCII 2007, Beijing, China, 2007.
3. Luyten, K., Vandervelpen, C., Coninx, K.: Task modeling for ambient intelligent environments - design support for situated task. Tamodia 2005, Gdansk, Poland, 2005.
4. Maly, I., Curin, J., Kleindienst, J., Slavik, P.: Creation and Visualization of User Behavior in Ambient Intelligent Environment. Information Visualisation, London, Great Britain, 2008.
5. Propp, S., Buchholz, G., Forbrig, P.: Task Model-based Usability Evaluation for Smart Environments TAMODIA 2008, Pisa, Italy, 2008.
6. Scholtz, J., Consolvo, S.: Towards a Discipline for Evaluating Ubiquitous Computing Applications, Technical Report of Intel Research, 2004.
7. Shirehjini, A.: A Multidimensional Classification Model for the Interaction in Reactive Media Rooms. HCII 2007, Beijing, China, 2007.
8. Trapp, M., Schmettow, M.: Consistency in use through Model based User Interface Development. Workshop at CHI'2006, Montreal, Canada, 2006.
9. Weiser, M.: The Computer for the 21st Century. Scientific American, 265, pp.94-104, 1991.
10. Wurdel, M., Propp, S., Forbrig, P.: HCI-Task Models and Smart Environments. HCIS 2008, Mailand, Italy, 2008.