CZECH TECHNICAL UNIVERSITY IN PRAGUE

Doctoral Thesis Statement

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Ivo Malý

Analysis of Usability Tests with Context Model

Doctoral thesis statement for obtaining the academic title of "Doctor", abbreviated to "Ph.D."

PhD programme: Electrical Engineering and Information Technology
Branch of study: Computer Science and Engineering

Prague, August 2012

The doctoral thesis was produced in combinedmanner Ph.D. study at the Department of Computer Graphics and Interaction of the Faculty of Electrical Engineering of the CTU in Prague.

Candidate:

Ivo Malý
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering of Czech Technical University in Prague
Karlovo nám. 13,
121 35 Prague 2, Czech Republic

Thesis Supervisor:

prof. Ing. Pavel Slavík, CSc.
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering of Czech Technical University in Prague
Karlovo nám. 13,
121 35 Prague 2, Czech Republic

Opponents:

....

....

....

The doctoral thesis statement was distributed on: ....

The defence of the doctoral thesis will be held on .... at .... a.m./p.m. before the Board for the Defence of the Doctoral Thesis in the branch of study (to be specified) in the meeting room No. .... of the Faculty of Electrical Engineering of the CTU in Prague.

Those interested may get acquainted with the doctoral thesis concerned at the Dean Office of the Faculty of Electrical Engineering of the CTU in Prague, at the Department for Science and Research, Technická 2, Praha 6.

....
Chairman of the Board for the Defence of the Doctoral Thesis
in the branch of study Computer Science and Engineering
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6.

Contents

1 Aims of the Doctoral Thesis

Applications are no longer designed only for desktop computers used in simple context environments, such as an office environment. New category of applications are developed for smartphones, tablets and TVs that are used in ubiquitous computing, mobile environment and ambient intelligence environment. Such environments introduce complex context that has high impact on the usability of applications and their user interfaces. The complex context also brings new challenges to designers and developers.

Goal of every designer is creation of usable application, but the usability of the application is rarely achieved without proper usability testing. In order to get ecologically valid results from the usability test that show real usability problems, the environment of the test should be as realistic as possible.Usability practitioners prefer testing in controlled environments, like usability laboratory. Such environment are preferred among usability practitioners because it allows execution of the test in controlled environment while using known usability measures and analytical techniques. However, such environments are suitable for testing of applications that are not context-sensitive, e.g. finding usability problems in an Internet browser application on mobile phone. In such cases the context of use is simple and most of the usability problems may be found in the usability test in a laboratory [9, 8].

In case of usability testing of context-sensitive and context-aware applications for mobile, ubiquitous computing and ambient intelligence environments, it is difficult to built such realistic environment in controlled laboratory setup. Therefore such tests must be performed in real environment (also called *in the field*) [15, 18]. Tests in the field are usually labor-intensive and more costly to run than tests in a usability laboratory, however, the tests are being increasingly accepted within communities that are developing context-sensitive and context-aware applications.

When testing in an environment with a complex context, it is difficult to use traditional data sets like audio/video recordings and observer notes. There is no ideal place where to put the camera to record whole environment and possibilities of observer notes recording during the test may be limited, e.g. to not disturb the test participant during a task execution. As a consequence, usability practitioners may observe occurrences of events that are hard to explain, because they are not correctly recorded.

Usability practitioners face this issue by recording data sets from additional types of data sources, e.g. user movement in the environment. On the one hand, by increasing the number of data sources, usability practitioners can reduce the possibility to misunderstand events that are potential usability problems. On the other hand, recording and analysis of data sets from multiple data sources brings a different kind of complexity to the analysis of data sources.

Aim of this thesis is improvement of the usability testing workflow in order to allow better analysis of test data sets. This can be done by formal description of the context of the test and by analysis of relations between recorded data sets, context of the tested application and context of the test. In the thesis, such formal description is proposed as a *usability test context model* (UTCM).

The usability test context model is incorporated into the usability test workflow during the test preparation step. Parts of the usability test context model may be adopted from task models and dialog models used for model based application design. When such models are not available, the usability test context model must be created. Aim of this thesis is reduction of overhead required by creation of full context model. Instead, it aims at creation of subset of the model that is adequate to full model during the usability test data analysis.

When relations between usability test context model and recorded data sets are defined, they can be used for sophisticated interactive visual analysis. By means of them, usability practitioners will be able to come up with more usability problems and more accurate explanations of their causes. Aim of this thesis is to develop such advanced visualization and evaluate them in usability test in complex context environment.

2 Current Situation of the Studied Problem

There are several techniques used by analytical tools for analysis of data recorded during the usability test. First technique is analysis of data sets based on time coherence. This technique is typical for interconnection of data sets, e.g. observer notes or user interface events, with audio/video recordings. Based on this type of interconnection, efficient analysis of audio/video recordings can be done. The data sets are visualized in a form of interactive timeline, which is used for identification of usability problems and navigation to the spot in audio/video recording for usability problem description.

Alternative to timeline visualization and data set time coherence is visualization of the recorded data sets in relation to the structure of the tested application. Web Quilt [7] combines UI events logs with the application snapshot to visualize data sets in the form of a graph, where nodes are individual states of the application and edges represent UI events. Multiple data sets can be visualized to show, how the users are using application and in which states there are usability problems. Similar approach can be used for applications developed using model based UI design. For such applications task model and dialog model is available and matched with the application UI events and application state changes. During the evaluation, logs are connected with models and visualized as statistics [10, 16] or as a table [5].

Last technique, used mainly for usability test in mobile environment, analyzes data sets combined with the description of test environment. Chitaro et al. [3] visualize the data in 2D visualization. They use user movement and field of view data sets for analysis of user movement styles, navigation problems and traveled areas. There are also several projects that visualize data sets in 3D environment in order to visualize human interaction in the environment [19], evaluate user reactions [6], and show tasks distribution in the ubiquitous environment [13].

Advantage of the techniques is that they individually allow useful data sets processing and provide interesting visualizations. The disadvantage is that they usually offer limited ways of the data presentation. This can be factor seriously limiting usability test size, scope, and duration, especially for context-sensitive and context-aware applications tested in complex context environment.

3 Results

In this chapter, three main results of this thesis are presented. The first result is formalization of the usability test context model in connection with data sets and incorporation of the model into the usability test workflow. The second result is technique for creation of the subset of the usability test context model in preparation phase of usability test. The third result of this thesis is usage of usability test context model for sophisticated visual analysis of data from usability tests.

3.1 Usability Test Context Model

This section describes structure of the usability test context model, its interconnection with data sets and its integration into the usability test workflow.

3.1.1 Structure of Usability Test Context Model

The usability test context model describes formally context of the test and its relations to recorded data sets. Usability test context model is based on generic description of context given by Dey [4], which states that any information is part of the context.

Proposed usability test context model is divided into five areas that cover main aspect of usability test – see Figure 3.1. Application, User, Environment are areas, which are derived from models used for model based application design [11, 14, 17]. Test and Observations area are derived from usability test context. Each area contains one or more models and these models may be interconnected with each other (e.g. task model with dialog model and task model with task list).

Data sets recorded during any phase of the usability test, like user interaction logs or observer notes, can be interconnected with particular submodels. Main goal of this interconnection is provision of novel relations between multiple data sets and submodels during analytical phase of the usability test. Special case is audio/video recording. This data set contains complex information about the test but can be interconnected with other data sets only through timestamps. Therefore it stays outside of any particular area.

Application area of the usability test context model represents application structure, logic and user interface through task model, dialog model and user interface (UI) model.

Task model represents description of user tasks supported by the tested application. In analytical phase of usability test workflow the task model may provide information about currently executed subtask and about next possible subtasks. The task model may also provide information to detect execution of incorrect subtask.

Dialog model is used for description of application dialog states. States of dialog model are connected by transitions that represent user interactions (e.g. submitting the form) or application actions (e.g. timeout running on background). In the analytical phase of the usability test workflow the dialog model can be used to provide information about

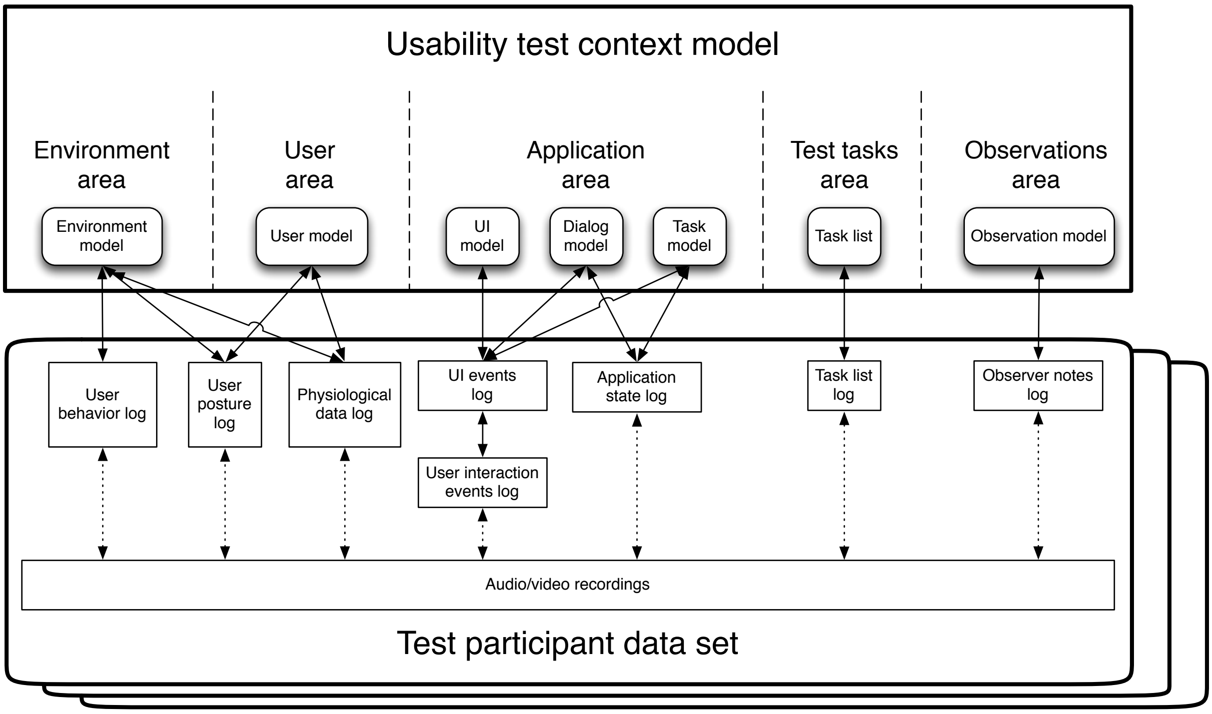


Figure 3.1: Overview of usability test context model.

the dialog model states and available actions. When both task model and dialog model are used, their states can be interconnected [12]. This relation can be used in analytical phase to highlight states of the dialog model that are reachable from particular subtask of task model.

Each application dialog model state can be associated with one or more states of the UI model. Each state of the UI model represents hierarchical structure of UI elements. In analytical phase of usability test workflow the UI model can be used to visualize user interface of particular state of dialog model. This can be useful when recording of display of device running the application is difficult or there is no such interface, e.g. for voice operated applications.

User area of the usability test context model focuses on modeling of the test participant by means of user model. User model represents parameters of users from application target user group that are being observed during the test. Observed parameters are usually related to physiological data recorded during the usability test and also to possible user posture parameters. In analytical phase, the user model can be used for advanced description of particular test participant and also for visualization of posture log in form of user's avatar, either alone or in the test environment, when environment model is provided.

Environment area of the usability test context model represents environment of the test. Such environment can be real or virtual and is described by one environment model. Main purpose of environment model is providing description of the environment and analysis of interaction between user and environment (e.g. light switch) that may influence course of the usability test. Environment model can be represented as 2D graph of segments, 2D map or as 3D model. In the analytical phase the environment model can be used for visualization of the environment, typically together with user's avatar. Comparison and analysis of movement strategies can be analyzed as well as interaction with the environment.

Test tasks area of the usability test context model represents test tasks performed by test participants during the test. Tasks are typically described in the form of a task list, where each task contains instructions for the test participant and expected steps for usability practitioner. The task list tasks can be connected with the tasks from the task model. The task list is used mainly during the usability test execution as a set of instructions for the test participant. In the usability test analysis phase the task list can be used to provide instruction for particular task and it can be also used for segmentation and filtering of logs based on task duration.

Observations area of the usability test context model is represented by observation model that characterizes categories of observer notes. This approach is derived from categories for observer notes in Techsmith Morae tool and from coding scheme from Noldus Observer tool. In analytical phase of the usability test workflow, the observations model is used mainly for provision of additional information about each observer note and for filtering of observer note log.

3.1.2 Usability Test Context Model and Data Sets Interconnection

Main feature of the usability test context model is possibility to interconnect with various data sets, see lower part of Figure 3.1. Supported data sets are therefore also formally described.

Observer note in the observer notes log contains textual description and one or multiple categories, through which they are connected to the observation model. Task list log records contains identification of the task from the task list and indication about start or end of the task.

In case of user interaction events, these are firstly integrated with UI events and connected with dialog model and consequently with UI model and task model. Application state log is also connected with dialog model and consequently with task model. Application state log records consist of identification of states of the application during the usability test execution. Through this identification the log is connected with the dialog model or task model. The application state log is usually combined with the UI events log in order to identify connection used to change the application state.

User behavior log contains primarily information about user position during the course of the usability test. The position can be represented as an absolute user position or as incremental updates to the previous user position. Moreover, user behavior log also contains data about interactions with other objects in the environment. In such a case identification of the object and type of action is logged.

User posture log contains information about the posture of the user. Currently, the user model of usability test context model works with body and head heading and three body postures: lying, sitting, standing. Usability test context model can be also connected with physiological data log record, that consists of Heart Rate Variability (HRV) values, subjective stress assessment and Galvanic Skin Response (GSR) values.

3.1.3 Integration of the Usability Test Context Model into the Test Workflow

Integration of the usability test context model into the usability test workflow covers steps from preparation of usability test (Step 2) to reporting of findings from the test (Step 5) – see Figure 3.2. In planning the test (Step 1) the usability test context model is not directly used. At the end of the preparation of the test (Step 2), the usability test context model is available. During the execution of the test (Step 3) the usability test context model can be used by logging and recording tools for test participant data set recording. In the data analysis step (Step 4), the usability test context model and test participant data sets are interconnected and analyzed. Result of the analysis is list of usability problems that are compiled from individual usability problems of each test participant.

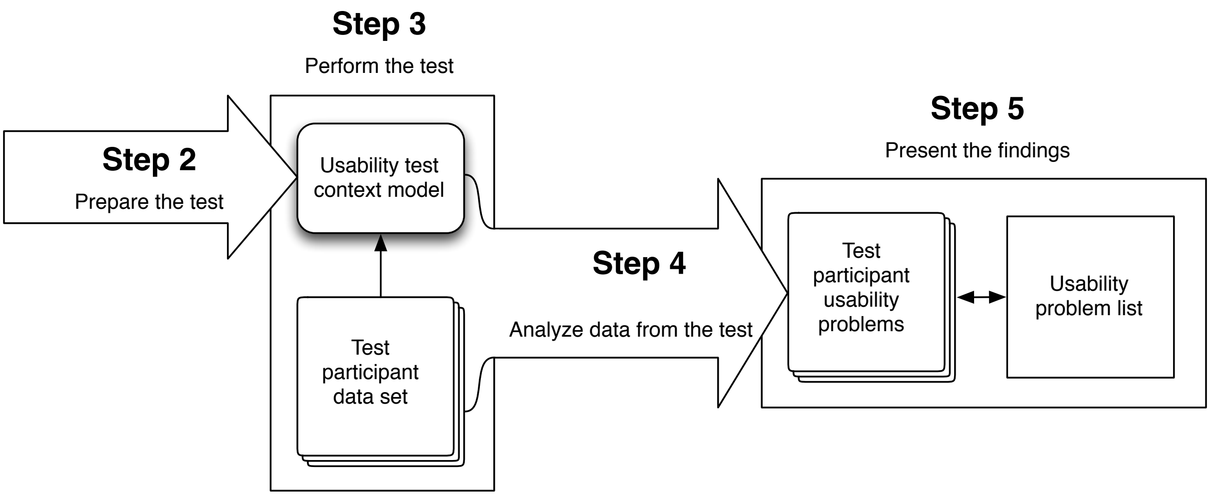


Figure 3.2: Usability test workflow extended by usability test context model.

3.2 Usability Test Context Model Creation in Pre-test Phase

Usability test context model should be prepared during the test preparation step. Parts of the model can be already created during application development phase. When this is not the case, usability practitioner must create them. However, the process of reconstruction of all submodels of usability test context model is very demanding. Next result of this thesis is approach that creates subset of the usability test context model. The idea of this approach is based on three assumptions.

1. During the preparation phase the task list is created.
2. During the usability test, participants are using only subset of application dialog model and task model. Therefore, only this subset of task and dialog model may be used for analysis.
3. There is relation of the task list with the task model, dialog model and with UI events log. Such relations may be used for creation of subset of all three submodels in parallel.

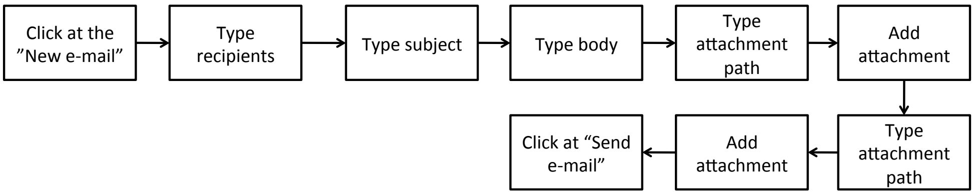


Figure 3.3: Visual representation of task model skeleton for task Send e-mail with attachment.

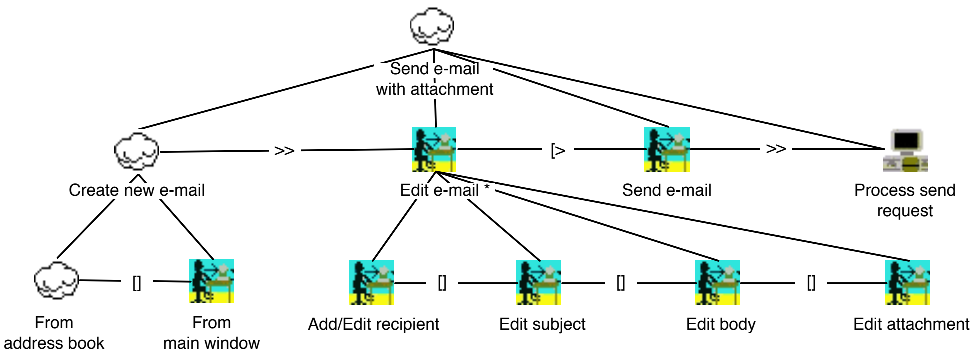


Figure 3.4: Visual representation of full task model for task Send e-mail with attachment.

Because this approach creates only subset of the models and the task list necessary for the usability test, these subsets are labeled as skeletons. Even though they do not cover entire application, skeletons should have similar properties like the full submodels created during the design of application in order to be used in execution and analytical steps of usability test workflow.

Skeletons are created from the usability expert walkthrough through the application. This walkthrough is recorded in a form of UI events logs – see Figure 3.5. Initially, task model skeleton is created. Each subtask of the task model skeleton is represented by one UI event record, see Figure 3.3. When compared with full task model (e.g. in CTT notation, see Figure 3.4) there are some differences like missing subtasks relations (only the sequential relation is defined) and missing subtask properties as the optional subtask or subtask iteration. Moreover, some subtasks that were atomic in the full task model, they consist of several subtasks (UI event records) in the task model skeleton. Also, alternatives subtasks, that represents alternative execution of the task in the full task model are missing in the skeleton.

Despite the differences between the task model and the task model skeleton, the task model skeleton covers all the subtasks for particular realization of the task model. Such information should be sufficient for correct association of the user's interactions with the appropriate task. Missing relations can be taken into account during the data analysis step, when the user's interactions are compared with the task model skeleton. Because of the nature of task model skeleton generation, the task model skeleton may be application dependent.

From the task model skeleton, the task list skeleton is derived. The task list skeleton contains list of individual tasks that have predefined structure consisting of the task

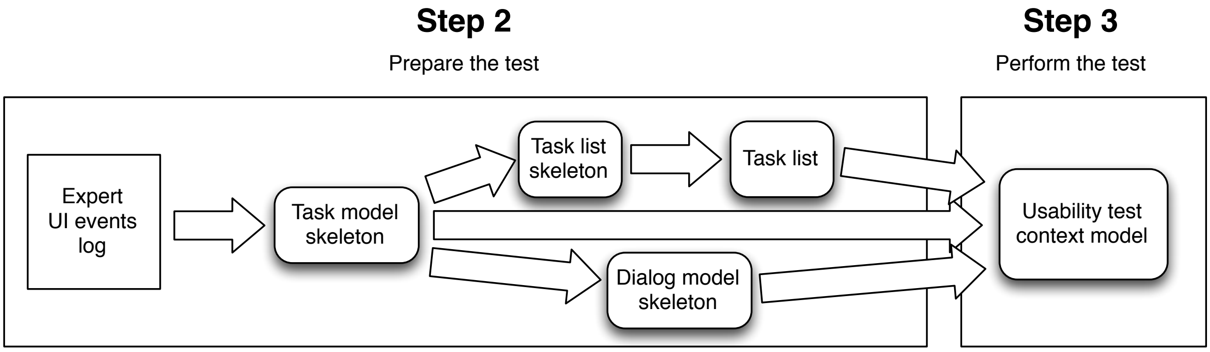


Figure 3.5: Update to usability test workflow when testing with task model, dialog model and task list skeletons.

description and list of expected steps. Unlike task model skeleton, usability practitioner can transform task list skeleton into full task list by entering the missing task descriptions to each task.

Compared with tasks from task list created manually without relation to the task model skeletons, the expected steps describe sequence of UI events, which may be too detailed description. Expected steps are generated automatically from the particular task model skeleton tasks. Therefore, each task from task list skeleton relates to one task model skeleton. Usability practitioner may use such detailed informations for step-by-step checking of the course of the test. However, when alternative interaction is taken, it may cause problems in analysis.

Dialog models are also derived from task model skeleton. Structure of the dialog model skeleton follows the concept of transformation of task models into dialog models [12]. Connections between states correspond with the subtasks from task model skeleton. State description is generated as a constant "After" and name of the task model subtask, e.g. "After Type body". When screenshot is taken by the UI event logging tool, this screenshot may be added to the state as substitute to the UI model.

The usability study of the web e-mail client Roundcube was conducted to evaluate possibilities of the usage of skeletons in real tests. In the study, creation of the task model skeleton and the task list skeleton generated only a small time overhead over the standalone task list generation. The analysis of the data showed that skeletons can be successfully used as substitute to full task models.

3.3 Data Analysis with Usability Test Context Model

Third result of this thesis is usage of the usability test context model during analysis of data sets. This thesis focus on formative usability tests in which interactive visual analysis is mainly used.

Usability test context model was incorporated into Information Visualization Reference Model [2], see Figure 3.6. This model represents framework suitable for implementation of various information visualization applications. Data sets recorded from from data sources during the usability test are connected with usability test context model.

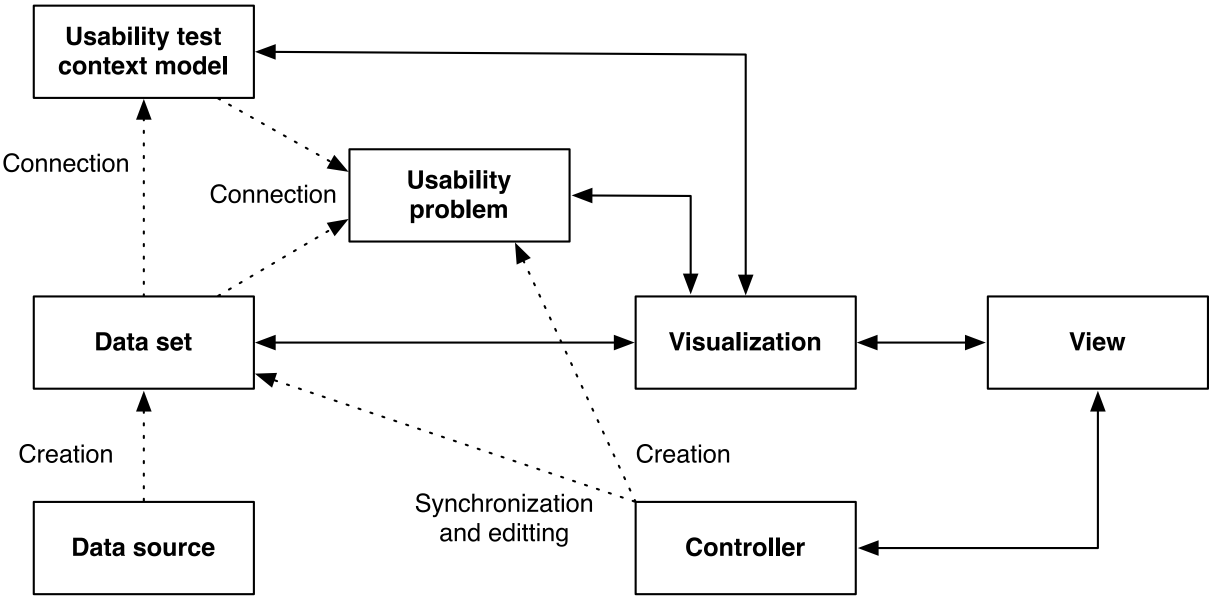


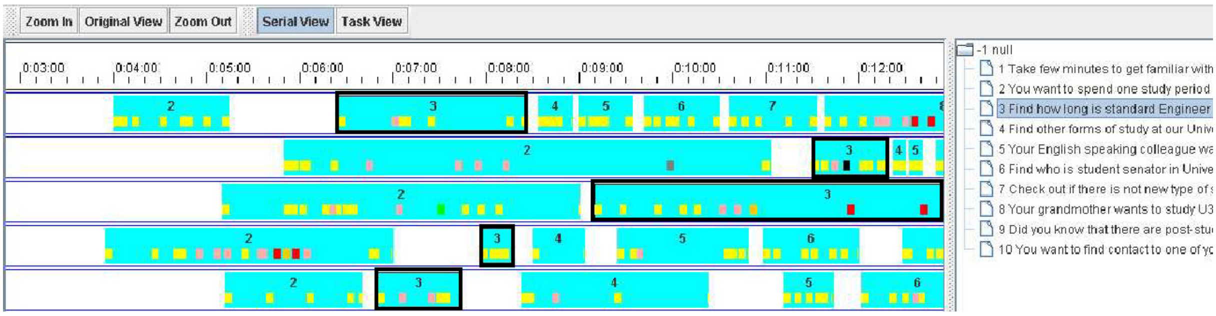
Figure 3.6: Information Visualization Reference Model derived from [2] taking into account usability test context model

Visualizations uses entire or part of usability test context model and particular data sets and visualize them through one or more views. Each view can be controlled by one or more controller components. Controller components can also edit the data sets, in order to add new observer notes or to synchronize the data sets. Usability problems, which may be connected to data sets and usability test context model, are created using controllers in the views. Usability problems can be also used in visualizations to allow backtracking of sources of usability problems.

This concept was implemented in IVE tool in a form of convertor plug-ins that stores data sets into database and interconnects them with usability test context model. Several multiview synchronized visualization plug-ins were implemented to shows data sets and usability test context model. In following subsections, three types of visualizations are described that allow novel data analysis.

3.3.1 Timeline Segmentation

Visualization of data sets in timeline can be improved by segmentation of the timeline into blocks according to submodels of usability test context model. Typically, this segmentation can be done for task list log, application state log, user behavior log for environment model in form of 2D graph of segments and their models. Example of such segmentation for 5 data sets is presented in Figure 3.7a. The timeline is divided into several blocks, each representing one task, either from task list or from task model. Tasks are identified by a number which stands in the middle of the task rectangle above the log records. Visualization of the model is also available to the usability practitioner, in this case the task model is visualized in the form of a tree on the right. Timelines separated into such blocks are easier to analyze, while usability practitioner can focus on specific part of the



(a) Serial view of 5 timelines.

(b) Task view of 5 timelines.

Figure 3.7: Comparison of two alignments of the timeline.

test.

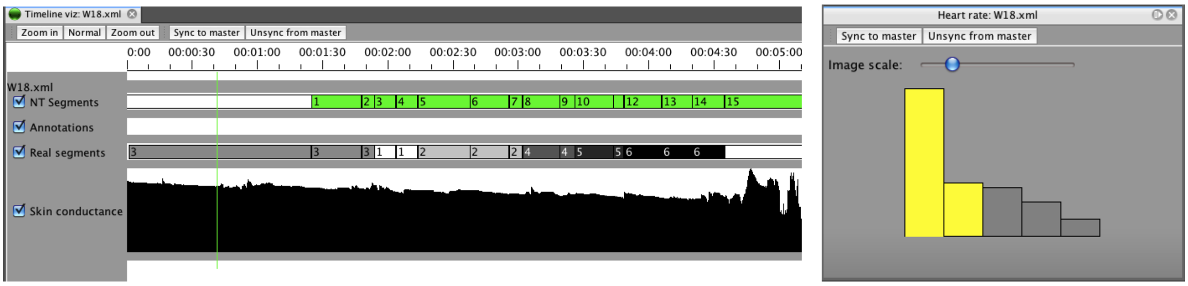
When the segmented blocks should be aligned by the task start time, the aligned mode presented in the Figure 3.7b can be used. This visualization mode aligns a segmentation blocks, in this case tasks, so that for all participants this selected task starts at the same time (e.g. see task 3 in Figure 3.7b). This feature allows the usability expert to analyze each task for all participants more in depth, because he/she do not need to search for the task by each participant. Typically, the test participants spend a different time with the tasks (as seen in Figure 3.7a) and thus the tasks are naturally not aligned in the timeline.

3.3.2 Physiological Data Analysis

Two specific visualizations were developed for analysis of the three physiological data sets, see Figure 3.8. These data sets were analyzed based on time coherence, however, it was necessary to combine data sets with models.

Timeline was used for visualization of skin conductance data set (GSR), see particular timeline in Figure 3.8a. The higher stress level is indicated as a sudden significant deviation of the skin conductance values.

Timeline visualization of user behavior log combined with environment model (2D graph of segments) was used for analysis of post-test subjective stress assessment. The position of the participant in the segment is visualized in the timeline *Real segments*. Each rectangle represents one segment. Subjective stress values were mapped to the timeline visualization by the hue and by the number in particular segment. White color (1) is



(a) Timeline visualization with real segments and subjective stress assessment.

(b) HRV data visualization plug-in view.

Figure 3.8: Comparison of two alignments of the timeline.

lowest stress; black color (6) is highest stress.

HRV data sets are visualized in a form of bar graph, see Figure 3.8b. HRV values represent ratio between low frequency heart rate and high frequency heart rate (LF/HF ratio) calculated by means of Fast Fourier transform (FFT) [1]. So each ratio of values refers to the 4 minute of the test (due to the 4 minutes sliding window used in FFT). Stress level from HRV data set is estimated from active bar values highlighted in yellow, relatively to each other.

Using these visualizations it was possible to evaluate correlation of the stress level using all three parameters, despite the fact that the data sets values could not be directly used for such analysis.

3.3.3 Application State Log Visualization

Application state log visualization combines data from application state log and UI events log with dialog model and task model. This visualization shows the data sets and models as combination of timeline-like visualization and graph based visualization of models.

An example of the application log visualization is in the Figure 3.9. The application state log records are presented as a states in the diagram (solid line circles). The states are connected by black solid line arrows that represent sequential character of the application state log records. The time goes from top down. Because of the connection with the dialog model, alternative states (dashed line circles) can be added into the visualization. Alternative states are those dialog model states, which can be reached from each particular state. These alternative states are connected with the log record states with dashed line arrows and are aligned horizontally. To preserve simplicity of the diagram any other states of the dialog model are not visualized. UI event log records are associated with each state and they provide description of action, that was performed by the test participant and that caused movement between states in dialog model.

When the application state log contains a large number of records, it is necessary to do some filtering and clustering. Therefore the task model is incorporated into the visualization. In the Figure 3.9, the CTT task model notation is used, but HTA diagrams can be used as well. Task model diagrams are visualized originally rotated 90 degrees counterclockwise compared to the visualization. Therefore the top task is the rightmost

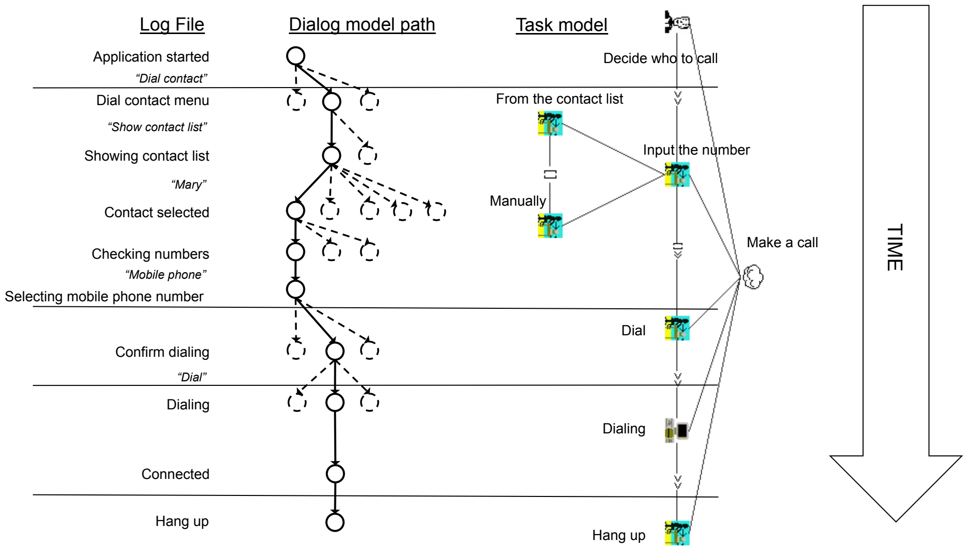


Figure 3.9: Example of the log visualization.

(originally upmost) rectangle and atomic tasks are the leftmost (originally downmost). Visualizations of the application state log connected with dialog model diagram and task model visualization are synchronized. All states, which are part of individual atomic task model tasks, are grouped and separated by horizontal lines. This way the usability expert can navigate easily in the longer and complicated logs, moreover, he/she can do a direct comparison of several logs side by side solving the same task.

3.4 Evaluation

Concept of usability test analysis with usability test context model was evaluated in three studies of NaviTerier, a "mobile indoor navigation application for blind users." These studies were executed in the real environment, and the real data sets were recorded and analyzed.

The first study focused on data organization, like interconnection of usability test context model and data sets, time synchronization of data sets and initial annotation of data sets with observer notes. The results showed that IVE tool and developed visualizations allow faster and more usable data synchronization compared to spreadsheet application and video player. Average time for log video synchronization was approximately 2 times faster and video annotation was approximately 1,5 times faster. During the analysis, the dialog and environment models were used to reveal approximate test participant's position, when quality of audio/video recording was low, e.g. during night hours. Dialog model allowed identification of potentially problematic situations in the timeline, mainly when the test participant navigated to previous segment.

In the second study, analysis of data sets with usability test context model (IVE tool) was compared with analysis using only audio/video recordings and observer annotations (Morae). Twelve usability practitioners were analyzing data sets from four test participants using one of the tools. Results showed that usability experts who were using tool with usability test context model, were able to better observe the course of the test, reveal nature of usability problem and navigate easily in data sets. Subjectively, all user evaluated their tools as very good (4 on scale 1 to 5, where 1 is worst and 5 is best), even though that especially usability practitioners from Morae group had problems finishing some tasks. This can be attributed to two main reasons. Firstly, Morae is commercial and works seamlessly and even though it does not support all the tasks from this study. Secondly, IVE tool is still in development, e.g. in some plug-ins not user friendly format of times or attribute values was be observed.

In this third study, the usability test context model was applied to analysis of stress impact on navigation of blind person. During the study, data sets from 22 test participants were recorded. Thanks to the usage of usability test context model it was possible to analyze the stress measurement data in the context of uncontrolled situational variables that potentially cause stress and distinguished them from true indications of stress. When analyzing both HRV data and subjective stress assessment of the test participants it was found that the unsystematic and uncontrolled situational variables on the route (e.g. random social encounters, environmental features such as distant noises that could work as navigation cues) were causing significant stress levels in both experimental and control group.

4 Conclusions

In this chapter, results of the thesis and main contributions are summarized.

4.1 Usability Test Context Model

Motivation of this thesis was to address the difficulty of analysis of data from usability tests of applications that are context-sensitive and context-aware. Such applications should be tested in ecologically valid environment, e.g. in the field, in order to reveal real usability issues in real context of use. However, such a test setups require a complex data set recording so that usability practitioners are able to understand the reasons of the usability problems, and they can suggest appropriate changes for the application user interface design.

In this thesis, the usability test context model was proposed. It describes formally context of the test and its relations to recorded data sets. Proposed usability test context model is divided into five areas with submodels that cover main aspect of usability test – Application, User, Environment, Test area and Observations area. Each area contains one or more models and these models may be interconnected with each other and with data sets recorded during the usability test.

The main contribution is possibility to incorporate the usability test context model into the usability test workflow. Based on that, the model can be used for deeper understanding of interrelations between all steps of the usability test workflow. This can help usability practitioners to prepare better test setups, which generate usable data sets and allow them run tests more efficiently. In the analytical step, it allows to perform more sophisticated analysis of data sets that lead to identification of more usability problems and explanations of their causes.

4.2 Usability Test Context Model Creation in Pre-test Phase

The usability test context model incorporates into the usability test workflow during the test preparation step. Parts of the usability test context model may be adopted from task models and dialog models used for model based application design. When such models are not available, the usability test context model must be created.

In this thesis, task model, dialog model and task list skeleton are proposed. They represent subset of the full models and task list necessary for the usability test. Creation of the skeletons is based on three assumptions: task list is created in this step, only subset of the application is used during the test execution, and there is relation between the task list with the task model, the dialog model and UI events log.

Main contribution of this approach is possibility to reduce complexity and overhead when creating usability test context model in the test preparation phase, compared to the creation of full models (task, dialog, task list). Even though the skeletons do not cover entire application, they have similar properties like full submodels in order to be used in test execution and data analysis.

During the test data analysis, the created skeletons were successfully used in an analytical application and they allowed observation of a set of usability problems comparable to the usage of full task and dialog models.

4.3 Data Analysis with Usability Test Context Model

In this thesis, the usability test context model was developed mainly for analysis of data sets recorded during the test. As a first step, the usability test context model was incorporated into information visualization reference framework. Based on this, new analytical tool (IVE tool) was implemented together with several novel visualizations that take advantage of usability test context model.

Main contribution of usage of usability test context model in analysis is possibility to visualize data sets with different characteristics that are hard to be visualized and analyzed side by side directly, e.g. data recorded during test execution and data recorded in post-test session.

The concept of data analysis with usability test context model was successfully evaluated in three studies with the use case “Mobile indoor navigation application for blind users.”

The course of the future work may be the usage of the usability context model for other types of usability evaluations, like heuristic evaluation or simulation of virtual participants. Combination of a usability evaluation methods is a recommended approach that allows the usability practitioners to effectively spread resources available for usability evaluation.

List of literature used in the thesis statement

[1] R. M. Bayevsky, G. G. Ivanova, L. V. Chireykin, A. P. Gavrilushkin, P. Y. Dovgalevsky, U. A. Kukushkin, T. F. Mironova, D. A. Priluzkiy, U. N. Semenov, V. F. Fedorov, A. N. Fleishmann, and M. M. Medvedev. HRV Analysis under the usage of different electrocardiography systems (Methodical recommendations), 2002. Available at `http://www.drkucera.eu/upload_doc/hrv_analysis_(methodical_recommendations).pdf`.

[2] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think (Interactive Technologies)*. Morgan Kaufmann, San Francisco, CA, USA, 1999.

[3] L. Chittaro and L. Ieronutti. A Visual Tool for Tracing Users' Behavior in Virtual Environments. In *Proceedings of the working conference on Advanced visual interfaces AVI '04*, pages 1–8, 2004.

[4] A. K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, Feb. 2001.

[5] P. Forbrig, G. Buchholz, A. Dittmar, A. Wolff, and D. Reichart. Model-Based Software Development and Usability Testing. In *INTERACT 2005 Workshop*, Sept. 2005.

[6] R. Holm, M. Priglinger, E. Stauder, J. Volkert, and R. Wagner. Automatic data acquisition and visulatization for usability evaluation of virtual reality systems. *Eurographics Short Presentations*, 2002.

[7] J. I. Hong and J. A. Landay. WebQuilt: a framework for capturing and visualizing the web experience. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 717–724, New York, NY, USA, 2001. ACM Press.

[8] A. Kaikkonen, T. Kallio, A. Kekäläinen, A. Kankainen, and M. Cankar. Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *Journal of Usability Studies*, 1(1):4–16, Nov. 2005.

[9] J. Kjeldskov, M. B. Skov, B. S. Als, and R. T. Høegh. Is It Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field. In *Mobile Human-Computer Interaction – MobileHCI 2004*, pages 529–535, 2004.

[10] A. Lecerof and F. Paternò. Automatic Support for Usability Evaluation. *IEEE Trans. Softw. Eng.*, 24(10):863–888, 1998.

[11] Q. Limbourg and J. Vanderdonckt. Usixml: A user interface description language for context-sensitive user interfaces. In *ACM AVI'2004 Workshop: Developing User Interfaces With XML: Advances On User Interface Description Languages*, 2004.

[12] K. Luyten, T. Clerckx, K. Coninx, and J. Vanderdonckt. Derivation of a Dialog Model from a Task Model by Activity Chain Extraction. In *DSV-IS: 10th International Workshop - Interactive Systems - Design, Specification, and Verification*, pages 203–217, 2003.

[13] K. Luyten, C. Vandervelpen, and K. Coninx. Task modeling for ambient intelligent environments: design support for situated task executions. In *TAMODIA '05: Proceedings of the 4th international workshop on Task models and diagrams*. ACM, Sept. 2005.

[14] G. Mori, F. Paternò, and C. Santoro. Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. *IEEE Trans. Softw. Eng.*, 30:507–520, 2004.

[15] C. M. Nielsen, M. Overgaard, M. B. Pedersen, J. Stage, and S. Stenild. It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*, pages 272–280, New York, NY, USA, 2006. ACM Press.

[16] F. Paternò and G. Ballardin. Model-Aided Remote Usability Evaluation. In *INTERACT'99*, pages 434–442. IOS Press, Sept. 1999.

[17] F. Paternò, C. Santoro, and L. D. Spano. Engineering the authoring of usable service front ends. *Journal of Systems and Software*, 84(10):1806–1822, 2011.

[18] Y. Rogers, K. Connelly, L. Tedesco, W. Hazlewood, A. Kurtz, R. E. Hall, J. Hursey, and T. Toscos. Why it's worth the hassle: the value of in-situ studies when designing Ubicomp. In *UbiComp '07: Proceedings of the 9th international conference on Ubiquitous computing*. Springer-Verlag, Sept. 2007.

[19] A. A. N. Shirehjini and F. Klar. 3DSim: rapid prototyping ambient intelligence. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM, Oct. 2005.

Refereed Publications in Journals with Impact Factor

[20] Ivo Malý, Zdenek Míkovec, Jan Vystrčil, Jakub Franc, and Pavel Slavík. An evaluation tool for research of user behavior in a realistic mobile environment. *Personal and Ubiquitous Computing*, Online first:1–12, 2011. (Ratio: 47%, 16%, 15%, 15%, 7%).

Refereed Publications in WoS

[21] Ivo Malý and Zdenek Míkovec. Web applications usability testing with task model skeletons. In Regina Bernhaupt, Peter Forbrig, Jan Gulliksen, and Marta Lárusdóttir, editors, *Human-Centred Software Engineering*, volume 6409 of *Lecture Notes in Computer Science*, pages 158–165. Springer Berlin / Heidelberg, 2010. (Ratio: 80%, 20%).

[22] I. Malý, J. Cuřín, J. Kleindienst, and P. Slavík. Creation and visualization of user behavior in ambient intelligent environment. In *Information Visualisation, 2008. IV '08. 12th International Conference*, pages 497 –502, july 2008. (Ratio: 70%, 10%, 10%, 10%).

[23] Zdeněk Míkovec, Ivo Malý, Pavel Slavík, and Jan Cuřín. Visualization of users' activities in a specific environment. In *Winter Simulation Conference (WSC) 2007*, pages 738–746, Washington DC, USA, December 2007. WSC. (Ratio: 50%, 20%, 20%, 10%).

[24] Ivo Malý and Pavel Slavík. Towards visual analysis of usability test logs using task models. In Karin Coninx, Kris Luyten, and Kevin Schneider, editors, *Task Models and Diagrams for Users Interface Design*, volume 4385 of *Lecture Notes in Computer Science*, pages 24–38. Springer Berlin / Heidelberg, 2007. (Ratio: 80%, 20%).

Refereed Other Publications

[25] Ivo Malý, Michal Hapala, Jiří Bittner, and Pavel Slavík. On Tools for Game Interaction Analysis. In C Hellmich, M H Hamza, and D Simsik, editors, *2nd IASTED International Conference on Assistive Technologies*, pages 835–841, 2012. (Ratio: 40%, 25%, 25%, 10%).

[26] Ivo Malý, Jiří Bittner, and Pavel Slavík. Using annotated task models for accessibility evaluation. In Klaus Miesenberger, Arthur Karshmer, Petr Penaz, and Wolfgang Zagler, editors, *Computers Helping People with Special Needs*, volume 7382 of *Lecture Notes in Computer Science*, pages 315–322. Springer Berlin / Heidelberg, 2012. (Ratio: 65%, 25%, 10%).

[27] I. Malý, Z. Míkovec, and J. Vystrčil. Interactive analytical tool for usability analysis of mobile indoor navigation application. In *Human System Interactions (HSI), 2010 3rd Conference on*, pages 259 –266, may 2010. (Ratio: 80%, 10%, 10%).

[28] Ivo Malý, Jan Cuřín, Pavel Slavík, and Jan Kleindienst. Framework for Visual Analysis of User Behaviour in Ambient Intelligence Environment. In Felix Jesus Villanueva Molina, editor, *Ambient Intelligence*, pages 85–107. InTechopen.com, March 2010. (Ratio: 60%, 20%, 15%, 5%).

[29] Ivo Malý, Pavel Slavík, and Jan Kleindienst. Combination of models and logs for visual analysis of data from usability evaluation. In *Proceedings of the Third IASTED International Conference on Human Computer Interaction*, HCI '08, pages 279–284, Anaheim, CA, USA, 2008. ACTA Press. (Ratio: 75%, 15%, 10%).

[30] Ivo Malý and David Tichý. Usability Testing and Development of Car Simulator - City Rider. *Workshop CVUT*, 2007. (Ratio: 75%, 25%).

[31] Ivo Malý, Tomáš Macek, and Pavel Slavík. Car simulation environment for specialized research. In *Third Central European Multimedia and Virtual Reality Conference (CEMVRC) 2006*, pages 101–107, Eger, Hungary, November 2006. Pannonian University Press. (Ratio: 70%, 15%, 15%).

Unrefereed Other Publications

[32] Filip Hanzl and Ivo Malý. User Interfaces for Senior Users in iDTV Environment. In *Poster*, pages 1–5, Prague, 2009. Faculty of Electrical Engineering, CTU in Prague. (Ratio: 50%, 50%).

Citations of [20]

[33] V Bures. Interactive digital television and voice interaction: Experimental evaluation and subjective perception by elderly. *Electronics and Electrical Engineering*, 122(6):87–90, 2012.

Citations of [22]

[34] Mario Buchmayr, Werner Kurschl, and Josef Küng. A simulator for generating and visualizing sensor data for ambient intelligence environments. *Procedia Computer Science*, 5(0):90 – 97, 2011. The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011).

[35] Futoshi Naya, Ren Ohmura, Masakazu Miyamae, Haruo Noma, Kiyoshi Kogure, and Michita Imai. Wireless sensor network system for supporting nursing context-awareness. *International Journal of Autonomous and Adaptive Communications Systems*, 4(4):361–382, 01 2011.

[36] J. Synnott, L. Chen, C.D. Nugent, and G. Moore. Assessment and visualization of parkinson's disease tremor. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1 –4, nov. 2010.

[37] Christoph Burghardt, Stefan Propp, Thomas Kirste, and Peter Forbrig. Rapid prototyping and evaluation of intention analysis for smart environments. In Djamshid Tavangarian, Thomas Kirste, Dirk Timmermann, Ulrike Lucke, and Daniel Versick, editors, *Intelligent Interactive Assistance and Mobile Multimedia Computing*, volume 53 of *Communications in Computer and Information Science*, pages 239–250. Springer Berlin Heidelberg, 2009.

[38] Stefan Propp and Peter Forbrig. Vise – a virtual smart environment for usability evaluation. In Regina Bernhaupt, Peter Forbrig, Jan Gulliksen, and Marta Lárusdóttir, editors, *Human-Centred Software Engineering*, volume 6409 of *Lecture Notes in Computer Science*, pages 38–45. Springer Berlin / Heidelberg, 2010.

Citations of [23]

[39] J. Synnott, L. Chen, C.D. Nugent, and G. Moore. Assessment and visualization of parkinson's disease tremor. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1 –4, nov. 2010.

Citations of [27]

[40] Marta K. Larusdottir. *User Centred Evaluation in Experimental and Practical Settings*. PhD thesis, KTH, Media technology and interaction design, 2012. QC 20120522.

Citations of [24]

[41] Fabio Paternò, Andrea Russino, and Carmen Santoro. Remote evaluation of mobile applications. In *Proceedings of the 6th international conference on Task models and diagrams for user interface design*, TAMODIA'07, pages 155–169, Berlin, Heidelberg, 2007. Springer-Verlag.

[42] Stefan Propp, Gregor Buchholz, and Peter Forbrig. Task model-based usability evaluation for smart environments. In Peter Forbrig and Fabio Paternò, editors, *Engineering Interactive Systems*, volume 5247 of *Lecture Notes in Computer Science*, pages 29–40. Springer Berlin / Heidelberg, 2008.

[43] Maik Wurdel, Stefan Propp, and Peter Forbrig. Hci-task models and smart environments. In Peter Forbrig, Fabio Paternò, and Annelise Pejtersen, editors, *Human-Computer Interaction Symposium*, volume 272 of *IFIP International Federation for Information Processing*, pages 21–32. Springer Boston, 2008.

[44] Gregor Buchholz and Stefan Propp. Towards usability evaluation for smart appliance ensembles. In T. Graham and Philippe Palanque, editors, *Interactive Systems. Design, Specification, and Verification*, volume 5136 of *Lecture Notes in Computer Science*, pages 294–299. Springer Berlin / Heidelberg, 2008.

[45] Stefan Propp and Peter Forbrig. Defining expected behavior for usability testing. In Masaaki Kurosu, editor, *Human Centered Design*, volume 5619 of *Lecture Notes in Computer Science*, pages 110–119. Springer Berlin / Heidelberg, 2009.

[46] Stefan Propp, Gregor Buchholz, and Peter Forbrig. Integrating usability methods into model-based software development. In Victor Lopez Jaquero, Francisco Montero Simarro, Jose Pascual Molina Masso, and Jean Vanderdonckt, editors, *Computer-Aided Design of User Interfaces VI*, pages 125–136. Springer London, 2009.

[47] Stefan Propp, Gregor Buchholz, and Peter Forbrig. Integration of usability evaluation and model-based software development. *Advances in Engineering Software*, 40(12):1223 – 1230, 2009.

[48] Tonio Carta, Fabio Paternò, and Vagner Santana. Support for remote usability evaluation of web mobile applications. In *Proceedings of the 29th ACM international conference on Design of communication*, SIGDOC '11, pages 129–136, New York, NY, USA, 2011. ACM.

[49] Tonio Carta, Fabio Paternò, and Vagner de Santana. Web usability probe: A tool for supporting remote usability evaluation of web sites. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, volume 6949 of *Lecture Notes in Computer Science*, pages 349–357. Springer Berlin / Heidelberg, 2011.

Summary

Each application is designed to be used in specific context of use. In order to reveal real usability problems, context of the test environment should be as close as possible to the designed context of use of the application. In case of context-sensitive and context-aware application such test environment is usually in the field rather then in the usability laboratory. Unfortunately, tests in the field require collection of large data sets to help usability practitioners analyze usability problems and derive correct suggestions for application change. Currently, usability analysis tools do not satisfactorily support analysis of such large and diverse data sets, that makes it difficult and time consuming for usability practitioners to perform usability tests in the field.

As a solution to this problem a usability test context model is presented in this thesis. The usability test context model represents context of the application design, in the form of application models, user model and environment model. It represents also context of the usability test in form of task list and observations model. Data sets recorded in usability tests can be interconnected with the usability test context model in order to allow advanced data analysis. Additionally, recorded data sets and usability test context model can be connected with usability problems to allow easier understanding of usability problem origins.

In order to support adoption of usability test context model by usability practitioners, the update to standard usability test process is proposed. It shows how to include creation of application models, namely task model, dialog model and task list, in the preparation phase of usability test by means of skeletons. These skeletons are subset of the model, however they can substitute full models in test execution and analysis phase.

Main contribution of the thesis is usage of usability test context model in analysis of test data sets. Thanks to interconnection of data sets and usability test context model it is possible to create synchronized multiview visualizations that help usability practitioners to analyze recorded data sets. This concept is implemented by Integrated Interactive Information Visualization Environment (IVE tool) that consists of data set, model import plug-ins and multiview synchronized visualization plug-ins.

This concept was evaluated on data set from use case “Mobile indoor navigation application for blind”. The evaluations showed that provided concept allows for efficient data set organization and analysis of usability problems in field studies using novel visualization methods.

Shrnutí

Každá aplikace je navrhována pro použití ve specifickém kontextu. Pokud chceme při testu použitelnosti odhalit pravé problémy použitelnosti, je nutné použít testovací prostředí s takovým kontextem, který se blíží kontextu, pro který byla aplikace navržena. V případě testování aplikací využívajících informace z kontextu (context-sensitive) nebo aplikací sledujících kontext (context-aware), takové prostředí není možné simulovat v laboratoři, ale je nutné je hledat v reálném světě (in the field). Bohužel, testy v reálném světě vyžadují sběr velkého množství dat, aby byly testující osoby schopny správně analyzovat problémy a vyvodit správné návrhy na změny. V současné době, nástroje pro analýzu dat nepodporují dostatečně analýzu tak velkých množství rozmanitých druhů dat, takže je těžké a časově náročné testy v reálném světě provádět.

Jako řešení tohoto problému je v této práci navržen model kontextu testu použitelnosti (usability test context model). Tento model reprezentuje jak kontext vlastností aplikace ve formě modelů aplikací, modelu uživatele a modelu prostředí, tak i model testu ve formě seznamu úkolů a modelu pozorování. Data, získaná z testu použitelnosti mohou být propojena s modelem kontextu testu použitelnosti za účelem pokročilé analýzy dat. Navíc, získaná data a model kontextu mohou být propojeny se zaznamenanými problémy použitelnosti za účelem lepšího pochopení vzniku těchto problémů.

Za účelem snazšího využití modelu kontextu při testech použitelnosti, byla vytvořena metoda pro vytvoření částí modelu, zejména model úkolů (task model), model dialogů (dialog model) a seznam úkolů. Namísto tvorby kompletních modelů je navržena metoda pro tvorbu podčástí modelu kontextu (skeletons) v přípravné fázi testu. Tyto podčásti je možné vyrobit rychleji než kompletní modely a je možné je také použít v dalších fázích testu použitelnosti.

Jeden z hlavních přínosů práce je použití modelu kontextu pro analýzu dat z testů použitelnosti. Díky propojení dat a modelu je možné vytvořit sadu synchronizovaných víceokenních vizualizací, za účelem analýzy nasbíraných dat. Tento koncept je implementován v aplikaci IVE (IVE tool) ve formě zásuvných modulů (plug-ins). Celkový koncept testování použitelnosti s modelem kontextu byl otestován ve třech testech aplikace NaviTerier, což je mobilní navigační aplikace pro nevidomé pohybující se v budovách. Výsledky testu ukázaly, že prezentovaný koncept umožňuje efektivní zpracování a analýzu dat z testů v reálném prostředí za použití nových vizualizačních metod.