

Results of Mousemap-based Usability Evaluations – Towards Automating Analyses of Behavioral Aspects

Michael Gellner, Peter Forbrig, Manja Nelius

University of Rostock
Software Engineering Group
Albert-Einstein-Str. 21
18051 Rostock, Germany
{mgellner|pforbrig|manel}@informatik.uni-rostock.de
++49 381 498 34-33|++49 381 498 34-34|++49 381 498 34-33

ABSTRACT

In this paper, we present selected results from 23 test sessions that were recorded and analyzed with our usability evaluation environment *ObSys*. The central topic is the question what information we do obtain from *MouseMaps* (see [6] and [7]). *MouseMap* is the working name for the visualization *ObSys* offers. For analyzing this, several usability tests with contrasting situations (fluid workflows vs. the occurrence of constructed failures) were created and executed with testing persons. Then, we analyzed which errors lead to behaviors that could be recognized reliably in the Mouse Map visualization.

The analysis of the test data identified two classes of patterns: behavioral patterns in failure situations and general behavioral patterns. Both data were sampled to understand how ordinary software usage and exceptional acting look alike in *MouseMaps*. A view discusses in which ways this information will be used to automate the recognition of usability problems.

AUTHOR KEYWORDS

Interaction patterns, user interfaces, ergonomics, usability evaluation, usability testing, video recording, screen capturing, event logging, event recording, eye tracking.

INTRODUCTION AND OVERVIEW

Today there is not much doubt about the necessity of usability efforts. Nevertheless, it is very important to decrease costs for efforts and increase efficiency for such activities. Further more, new technologies as well as new target groups for certain products ask for high demands on this working area. Possible approaches to reach these goals are efficient mechanisms for automation of usability tests with little efforts for modeling preconditions.

Headlines like *automated usability evaluations* can sometimes be read (see e.g. [1] or [3]). This sounds as if all problems for usability evaluation were already solved. In contrast to that impression, nearly every usability department tries to decrease time for editing and analyzing recorded video material. Concerning to our experiences, hardly any company or usability group has an automated or even semi-automated solution at his disposal.

There is a wide gap between proposed concepts and the actual workflows in usability labs. Normally, the procedures require results of the complete analysis and specification phases to be modeled with certain tools and special notations (similar to some GOMS approaches).

The workflow with the proposed automating evaluation tools consists of comparisons between the specified models and diverging steps during test sessions. Actions from test persons that are unnecessary or missing to fulfill a task, signalise errors. A further indicator is the time test persons need to perform tasks. Without very detailed formal models this kind of »automation« cannot work.

Since the conceptual tools are one-way tools, a lot of development work had to be done twice: with the ordinary used environment *and* with the experimental or research tools – *only for enlightening the usability evaluation*. In general, extra work without a strong integration in the development artifacts progresses get lost. This even happens with fundamental materials like documentations. For that reason, it is not realistic to expect the maintenance of time-consuming artifacts that do not offer any direct benefit for the project. Consequently, numerous developers refuse such work from the very beginning. Further on maintaining such a de facto »mirror« project in a one-way system will cause costs and efforts not less than analyzing recorded materials.

Automating with the need of such intensive preparations possibly works but seems to be widely denied in practice. Without this strong preconditions there are hardly any automation approaches available up to now.

APPROACH AND GOALS

Deriving from the actual situation there are the following requirements.

- It must be avoided to demand greater modeling efforts.
- Maintenance of models and data through project stages without direct use to the project has to be avoided.

This paper proposes a new evaluation approach based on MouseMaps (see [6] or more generally [2]). MouseMaps visualize recorded input signals captured from input devices (primary mouse and keyboard) by representing events with different metaphors, e.g. lines for movements, dots for clicks, line thickness for speed and some further attributes. The central thesis is that the captured data offer sufficient information to find indicators for errors in test sessions. We do not suggest trusting such an analysis completely. Indicated points in time are *candidates* for further manual in-depth analysis. The problem with ordinary video analysis is to find even these candidates. Most labs calculate 6 to 12 hours watching recording per hour session. Thus, getting a list with a dozen candidates (technical: timestamps) for usability errors and problems would decrease this phase to a minimum. Our analysis environment *ObSys* allows to jump directly to the certain point in the error protocol and to watch a short sequence (some seconds before the occurring of the error until some seconds after this moment) around it. This enables a fast and easy verification or rejection of an error candidate.

This approach works even more successful, the better the recognition modules match each relevant behavioral aspect. For developing such modules a high amount of usability error scenarios are necessary to learn how exactly errors look alike. There are different goals we want to reach:

- I. Getting a good empirical basis with a wide range of software usage behavior for finding as much usability errors as possible.
- II. Getting expressive behavioral data for testing an implemented recognition module (to do).
- III. Getting realistic data for showing the efficiency of working with the *ObSys Evaluation Environment* (also: further works)

One point that should not be neglected: Of course we cannot find errors that do not occur during executing scenarios. Hence, this approach will never be able to give absolute reliability. However, even humans cannot interpret human behavior exactly.

We would consider this approach already as successful, if the recognition quote would be around 50% or 60% of human observers. The work described in this paper mainly serves goal I from the list above. We started to work on goal II in April 2004.

TEST ENVIRONMENT – THE OBSYS USABILITY EVALUATION SUITE

All tests were performed with an early prototype of *ObSys* [7]. This event recorder captures the messages from input devices, which *MS Windows* operation systems store into message queues, and saves them in a database. These data can be visualized in different ways. It is also possible to playback the messages and watch what users did. For using this *correctly* it is necessary to reconstruct the scenario environment exactly. Otherwise, the click and drag operations might fail or lead to undesired executions of processes.

Test sessions were conducted with every test person at the same place. It was paid attention to an authentic working atmosphere (one test series varied only this parameter). The used hardware consisted of a laptop (1.4 GHz, 240 MB RAM, 14.1" display), an optical three-button mouse with a scroll wheel. The recording tools were started and stopped with assistance of macros to automate the test progress and minimize data falsification.

TEST PERSONS

23 usability tests were accomplished with 11 test persons. The test persons had various backgrounds and different experiences with computers. See [Figure 1](#) for further details.

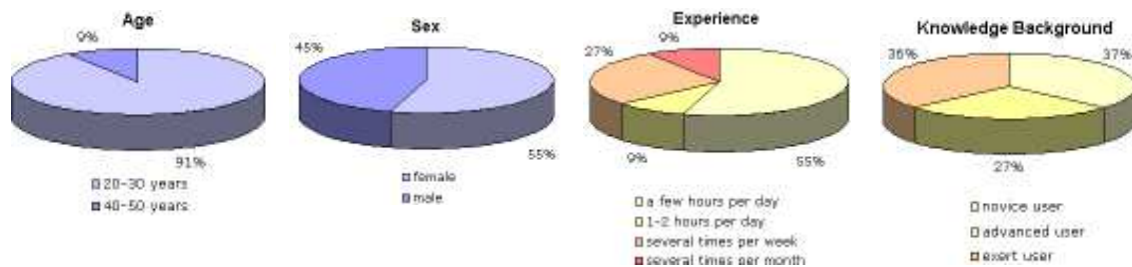


Figure 1, information about test persons concerning sex, age, experience and knowledge background

TEST SCENARIOS

Generally

In this first testing series, different aspects of usability problems were stressed. Scenarios became preferred, which were assumed to generate outputs as expressive as possible in the MouseMaps. Most of the scenarios placed especially constructed error situations beneath expected workflows.

Phase of Familiarization

In respect to the fact that all tests were conducted with an unfamiliar system, the tests of this category allowed the test persons to familiarize with the mouse. First they played a little game and after that they had to follow the contours of some figures with the mouse pointer to accustom to the mouse speed and the exact movement.

Search Strategy

These scenarios were intended to analyze how far MouseMaps are able to make statements about the search strategies and the user's problems while searching for an element in the work area. Therefore the test persons had to select certain buttons first in an unsorted and then in an alphabetically sorted alignment (see fundamental works in [8], [9]).

Supportive Function of the Mouse Pointer

This category was concerned about using the mouse as pointer. It was watched whether and how the mouse pointer was used for orientation on the screen. The effects of different formatted text on the input behavior were of interest here. Thus, the test persons had to read some texts.

Ability of Self-Description

The effects of refusing the principle of self-describing interaction elements [4] were analyzed in these scenarios. Test persons had to choose the right icons from a toolbar. In the first test, the icons were represented by common and self-describing pictograms. In the second test scenario unclear icons were used.

Efficiency

Efficiency is an important criterion for evaluating software usability. That means programs must not lead to unnecessary effort of concentration or force complicated proceeding [5]. These scenarios concerned about the effects of low efficiency of forms on the user's input behavior. Therefore, the test persons had to fill out two forms with differently aligned input fields.

Effects of Different Influences

During conducting tests with test persons all distraction were usually avoided. At real workplaces this is not possible. Some users listen to music while working. Time pressure can have a negative mental impact. The scenarios of this category evaluated the effects of different influences like aggressive and relaxing music, stand-up comedy and time pressure on the user's input behavior.

Selection of Elements

This category concerned about the user's behavior during the most fundamental action with the mouse pointer: the selection of elements on the work area. Therefore, the test persons had to click on some interaction components during several tests, whereas some of these elements were very small or closely aligned by each other respectively a not useful mouse pointer was used.

RESULTS (EXCERPTS)

Pattern Classification

The patterns that were recognized recurrently are separated in two classes *behavior in failure situations* and *general behavior*. The first class deals with such behavioral patterns that were observed during the failures of test scenarios. As some of the patterns show great similarity and differentiate only in their cause, these were divided into categories and summarized under a generic term:

- Extreme seeking movements
- Roaming search movements
- Intensive assistance while reading

In contrast, the general behavioral patterns cannot necessarily be attributed to failure situations, but describe general results about the use of the mouse and the various strategies that can be derived from the user's input behavior.

- Curved movement
- Two phase search strategy

In the following, a more detailed description of the patterns will be given. A schema of context, characteristics, causes, description and examples describes mainly the patterns.

Example Patterns for Behavior in Failure Situations

These patterns were observed in the behavior of all 11 participants. The examples demonstrate a typical behavior of a specific person. Very similar behavior was observed for all the other persons as well.

Extreme Seeking Movements

Context: Selecting an element with the mouse pointer respectively triggering an event at a certain point

Characteristics: Repeated regional jerky movements in coherence with a longer pause of the mouse pointer at a position

Causes: Too small interaction elements, not useful mouse pointer, absence of the system response to an action (e.g. appearance of a tooltip)

Description: Having problems with placing the mouse pointer to select an element or activating an event, the user tries to correct the position of the cursor with many little movements.

Examples:

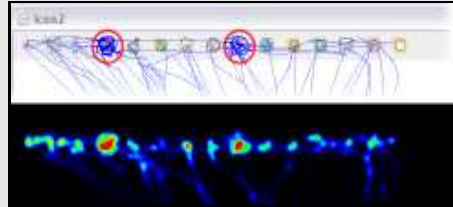


Figure 2, Tool tip text does not appear

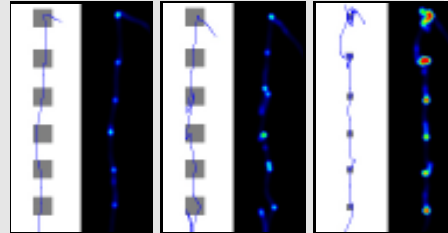


Figure 3, a) ideal movement b) not useful mouse pointer c) to small elements and not useful mouse pointer

These results are derived from two test scenarios where the test persons were expected to act with the mouse. In the first scenario two different symbol lists were presented. The first symbol list consisted of well-known symbols whereas the second symbol list presented unknown symbols. In this way, the test persons were forced in some way to wait for the appearing of the tool tip pop up windows. If they did not appear, a behavior like in [Figure 2](#) was shown. [Figure 3](#) shows the results of marking big and small elements with two different types of mouse pointers. All test persons showed precise positioning efforts to meet the snapping points if e.g. the mouse pointer was constructed badly ([Figure 3b](#)) and if the elements become smaller ([Figure 3c](#)).

Roaming Search Movements

Context: Looking for elements like icons, buttons, or menu entries on widgets

Characteristics: Roaming search movements (vertical, horizontal or circulative) in coherence with decreased work speed.

Causes: Unsorted group of elements (e.g. buttons); unclear or unknown elements (e.g. icons).

Description: Having problems to find an element on the desktop – e.g. because of unsorted elements – an increased use of the mouse and a decreased work speed can be observed.

Examples:

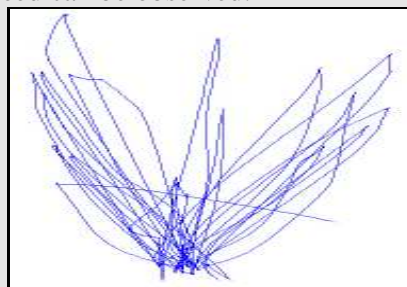


Figure 4, search movements in alphabetically ordered buttons



Figure 5, search movements in unordered buttons

For identifying this pattern, two scenarios were adducted. One of them was the symbol list scenario mentioned above. The other one consisted of two forms filled with buttons. The test persons were instructed to press a certain button. The difference between both forms consisted in the order of the buttons: First the buttons were unordered. The buttons on the form shown afterwards were ordered alphabetically. In the second case, most of the moves went directly to buttons with the searched terms. Strong differences between both behaviors were observed by 9 from 11 test persons. The results from 2 participants did not allow a clear differentiation.

Intensive Assistance while Reading

Context: Text reading

Characteristics: Increased horizontal and vertical mouse movements and hesitating with the cursor over certain regions of the text.

Causes: Poor readable text (e.g. because of too small fonts, unsuitable colors).

Description: Increasing horizontal respectively vertical movements during reading a text can be a sign of poorly readable texts, so that the user has to support the eyes with the mouse pointer.

Examples:

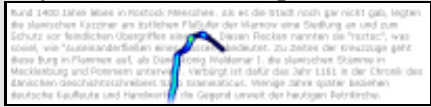


Figure 6, readable text




Figure 7, too small font

It is well known that different text attributes (like fat, italics or capitals) are only suited well for short passages. Since reading is based normally on matching word or phrase outlines (instead of capturing single letters) the reading speed decreases significantly if the outlines become too similar. Based on this information the test for reading assistance was constructed: 4 texts about history were given to the test persons. The first paragraph was set normally, i.e. good to read, the second one was set extremely small, the third paragraph was set in capital letters and the last one had red script on an intensively green background. Each paragraph had 6 lines up to 10 lines. In a second series, the test persons were expected to answer a question concerning the text per paragraph. The answer was used to check whether the persons understood the text or whether they did not (all participants did).

The usage of the mouse during reading showed different behaviors. Within the first scenario there were only a few mouse movements. The following behaviors occurred: *no use of the mouse*, *accompanying vertically* (see Figure 6), *supporting reading intensively* and *removing the mouse from the reading area*. This changed significantly in the scenario with questions. 6 persons showed a lot more mouse activity than before (see Figure 7), 3 persons behaved in the same way as before and 2 persons used the mouse a little fewer than before. Increasing activities occurred in the bad to read paragraphs always.

Example Patterns for General Behavior

Curved Movements

Context: Moving the mouse pointer from starting point to target.

Characteristics: Curved wandering from the correct path between two points.

Causes: Probably physiologic reasons in coherence with the user's mental constitution.

Description: In most cases, the test persons did not move the mouse pointer on the direct path to the target, but made curved movements – even in a situation with time pressure. In other cases, almost exact movements could be observed. Physiological and mental factors can be reasons for this.

Examples:

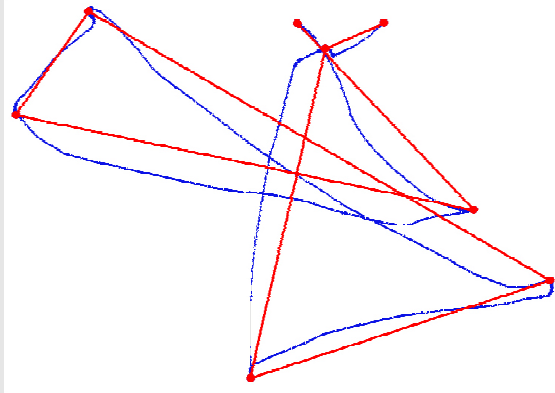


Figure 8, Curved movements between points (red/solid line - ideal path, blue/broken line - actual movement)

This pattern occurred in different test series. It was already reported ([6] and [7]). At the moment there are several explanation approaches for the phenomenon. The true reason for this behavior is not really clear, but it is significant that most movements with the mouse are conducted this way. Small and edgy movements like e.g. in Figure 3b and Figure 3c are exceptions and often signal problems (like in these figures).

Two-Phase Search Strategy

Context: Searching for elements on widgets

Characteristics: – (not general describable)

Description: When searching for an element, in the first phase the user searches only with the eyes. After a few unsuccessful moments, the mouse pointer is used for assistance to find the wanted element. This is considered as the second phase of this pattern.

Examples:

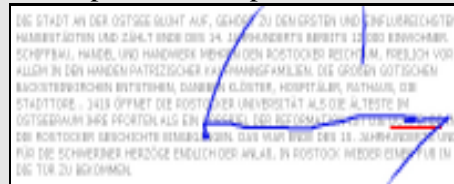


Figure 9, Locating an answer within a text in the second phase

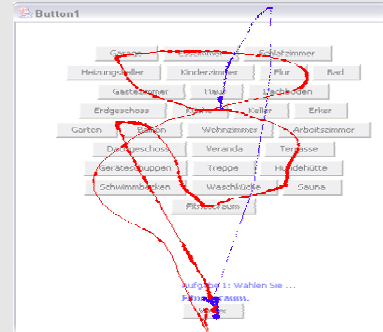


Figure 10, An existing element can be selected almost directly in the first phase (short/broken line); searching for a not existing element requires the help of the mouse pointer in the second phase (swallowed/solid line)

This pattern is based mainly on the mentioned test series with a form containing buttons without ordering and in alphabetical order. Another crucial source was the mentioned symbol list scenario since both scenarios require searching operations from the test persons. In other scenarios the phenomenon could also be observed at all test persons.

CONCLUSION

Similar to the phenomenon of body language, humans show recurrently certain behavior. But similar to everyday life body language, a certain behavior does not indicate always and universally the same inner state. Hence, we find the wide range in the recognized behavioral patterns. A bow can be a line if it is the MouseMap from one person. A line with a small angle should be interpreted as a line if it is part of a MouseMap from another person. This behavioral ranges complicate automatic detection. At the moment we consider it as impossible to decide about the inner state of a user without further »feedback« information. However, certain user actions can be recognized automatically. They give hints like: "It looks like searching." At this moment it is a question if a testing person is expected to search for something or not. This is a limitation our recognition shares with human observers.

OUTLOOK

Our current work focuses on further feedbacks that confirm the indicators MouseMaps offer. An important resource seems to be e.g. the galvanic skin response (GSR). Dissatisfaction, growing impatience, excitement and other emotions alter the GSR values. If the timely run from GSR values would correlate with MouseMap indicators our theses would increase strongly on reliability. The current ObSys Evaluation Environment becomes extended to be able to measure GSR values and to analyze correlations between other data like the input frequency of the keyboard or the speed with that the mouse is moved.

REFERENCES

- [1] Balbo, S., Coutaz, J. and Salber, D., *Towards automatic evaluation of multimodal user interfaces*. In: Gray, W. D., Hefley, W. E. and Murray, D. (ed.), Proceedings of the 1st international conference on Intelligent user interfaces, pp. 201-208, Orlando, USA, ACM Press, 1993.
- [2] Hilbert, D.M. and Redmiles, D.F. Extracting Usability Information from User Interface Events, *Technical Report UCI-ICS-99-40*, University of California, 1999.
- [3] Ivory, M.Y. and Hearst, M.A., *The state of the art in automating usability evaluation of user interfaces*. In: ACM Computing Surveys, Vol. 33, Iss. 4, pp. 470-516, ACM Press, 2001.
- [4] DIN EN ISO 9241, *Ergonomic Requirements for Office Work with Visual Display Terminals, Part 10: Dialogue principles*. International Organisation for Standardisation, Genf (1996).
- [5] DIN EN ISO 13407, *User-centred design process for interactive systems*. International Organisation for Standardisation, Genf (1998).
- [6] Gellner, M., *MouseMaps: Ein Ansatz für eine Technik zur Visualisierung der Nutzung von Software und zur Automation der Entdeckung von Bedienungsfehlern*. In: Ziegler, J. und Szwillus, G. (ed.), Mensch & Computer 2003: Interaktion in Bewegung, pp. 197-206.
- [7] Gellner, M. and Forbrig, P., *ObSys: a Tool for Visualizing Usability Evaluation Patterns with Mousemaps*. In: Jacko, J. and Stephanidis, C., (Ed.), *Human-Computer Interaction, Vol. I, Theory and Practice, Proceedings of the 10th International Conference on HCI*, Lawrence Erlbaum Associates, Mahwah, New Jersey, London, 2003, pp 469-473.
- [8] Katz, D., *Gestaltpsychologie*. Schwabe, Basel/Stuttgart, 1969.
- [9] Nelius, M., *Mousemap-basierte Erkennung der Problemfelder von Anwendern bei der Bedienung von Software*. Master Thesis, University of Rostock, Department of Computer Science, Rostock, Germany, 2003.