Computer Aided Observations of Complex Mobile Situations

Tobias Klug
SAP Research
Bleichstraße 8
Darmstadt, 64283
Germany
t.klug@sap.com

Telecooperation Group
Darmstadt Univ. of Technology
Hochschulstraße 10
64283 Darmstadt
Germany

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.
ACM 978-1-59593-642-4/07/0004.

Abstract
Designing mobile and wearable applications is a challenge. The context of use is more important than ever and traditional methodologies for elicitation and specification reach their limits. This paper investigates the challenge of creating and communicating information about the user's primary task with regards to its fine grained temporal structure. TaskObserver is a TabletPC software that allows real-time logging of events during observations of complex mobile scenarios. The results are communicated to other team members using task trace graphs of the events observed.

Keywords
observations, field coding, multi modality, wearable computing, mobile computing, context of use

ACM Classification Keywords
H5.2. User Interfaces: Theory and Methods.

Introduction
The field of wearable computing is at a point where some of the more mature concepts are slowly being turned into applications and products. However, it is not well understood how to develop a usable wearable computing solution efficiently. Traditional software engineering processes reach their limits, because a wearable user's primary task is in the real world (e.g.

Figure 1. Tablet PC with TaskObserver software running.

examining a patient, maintaining an engine). Therefore, using the wearable system is only a secondary concern and needs to blend into the more important primary task. To achieve this goal, the context of use requires special attention to avoid disturbing the user's primary task. Especially the selection of in- and output modalities plays an important role, as certain combinations of modalities perform better than others when used in parallel to the primary task[9]. For example the development team needs to know at which points during the task the user is able to use his hands for interaction with the system. Therefore, to design usable systems, the context of use needs to be communicated in detail to the designers and developers.

Context of use analyzes are common practice in user centered development. However, the methodologies used and data typically gathered are tailored towards desktop environments and applications as Pedell et al. point out[6]. Information about the task's temporal structure and modalities used is missing or not represented in sufficient detail with existing methodologies. Especially parallel performances of tasks are not well captured.

The method described in this paper addresses three challenges. Capturing information in sufficient detail, communicating it to other team members and reducing the total time needed for post field study analysis[4]. It was developed as part of a large project aiming to build wearable computing prototypes in industrial scenarios. The method consists of two parts. The TaskObserver software is used for real-time logging of parallel tasks during field observations. This data is then converted into task trace graphs to communicate the temporal structure to other team members. Internal use of the method has been promising so far, but an extended evaluation of its usefulness is still pending.

The remainder of this paper is structured as follows. First related work in the area of mobile context of use and video analysis is covered briefly. Next, the problem of observing complex situations is further described and our solution to this problem, the TaskObserver software and Task Trace Graphs, are presented. The next section covers our experience with using the software and an explanation of how the resulting task trace graphs benefit the communication process. The last section concludes the paper.

Related Work

Several approaches have been proposed to properly integrate the context of use into the requirements and software development processes of mobile design. These approaches use different media to visualize the context of use. For example the Picture Scenarios[6] approach uses annotated pictures to explain the context of use. The technique Software Cinema[1] uses video clips to communicate scenarios between end users and developers. However, producing such a video takes a lot of resources that are often not available. Also, video recordings are often not possible, especially in sensitive settings, such as hospitals.

Our solution is loosely based on the Timelines approach of Harrison et al[2] and the Marquee tool of Weber et al[7]. Both tools can be used to annotate video recordings in real-time. However, both tools focus on scenarios in office type settings that normally happen at a much slower pace than mobile activities. They are not designed for a mobile observer and their interfaces are not appropriate for multiple parallel events happening within short periods of time.

Available commercial software like INTERACT[3] or The Observer[5] offers similar features, but relies on automatic data gathering or video recordings for fast paced scenarios. Manual real-time annotations are possible, but optimized towards less frequent events.

Observing Complex Situations

Observing and describing the temporal structure of a complex work environment accurately is not an easy task. In addition to the possibly concurrent task performances of the user, the surrounding environmental conditions also need to be recorded. A brief study of existing observational techniques shows, that there is a conflict between precision of results and time needed for analysis, the two extremes being manual note taking and video analysis.

Video analysis provides the most accurate results, because annotations can be made with a precision of milliseconds. However, the video needs to be viewed at least once after the observation, to insert annotations, which takes a considerable amount of time. Additionally, there are scenarios where video recordings are not an option, either because of privacy issues (e.g. healthcare) or in tight spaces, where video cameras cannot easily capture the whole scene (e.g. aircraft maintenance).

The other extreme is traditional observation, taking notes on paper. This approach leads to an understanding which tasks are performed in a specific situation, but it is hard to capture its temporal structure accurately, because writing down notes takes too much time if several events occur within seconds of each other and one might miss something important.

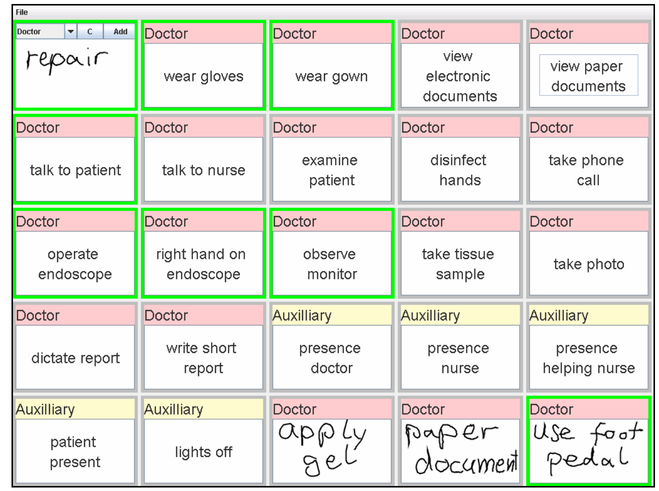

Figure 2. Screenshot of the TaskObserver interface. The blank area in the upper left corner allows adding new tasks, the two rightmost boxes in the bottom row have been added during the observation.

Post observational analysis of video material results in extremely accurate data. However, this accuracy is not always needed. If the relative temporal structure is more important than the exact times, it is possible to reduce the analysis time dramatically by doing a less accurate annotation of the observation in real-time. Such a real-time annotation is made possible by the help of a computer system that handles the tedious task of keeping the time of each task performance. We call the concept of using a computer system to support the observation process itself computer aided observations.

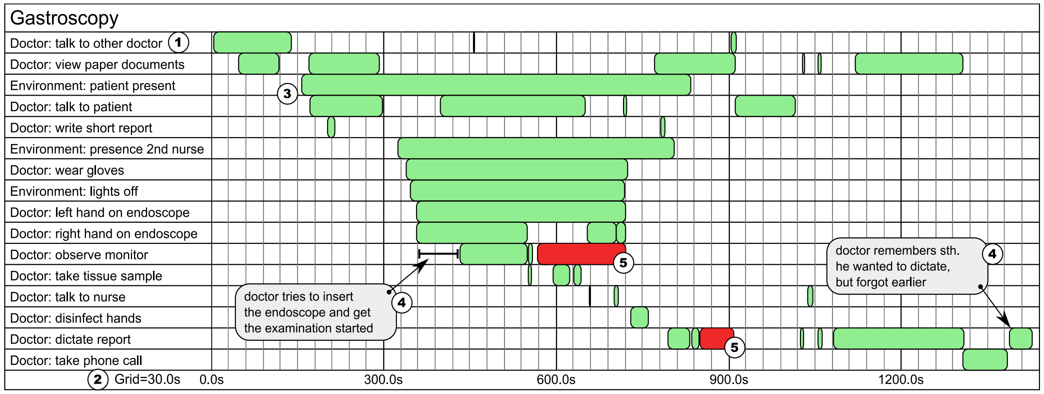

Figure 3. Task trace graph of a single gastroscopy examination, with a focus on the doctor and some environmental factors.

(1) list of activities being observed, (2) timeline, (3) bars show when and how long an activity was observed, (4) manually added explanations and (5) bars edited manually, because of erroneous data.

On this basis we have developed a solution, the TaskObserver software, enabling the observer to capture the dynamic behavior of a situation in real-time. The accuracy is less than what can be achieved with video analysis. But for understanding a situation, relative positions and durations of tasks are more important than their exact position and length.

Task Observer

Our solution uses a Tablet PC with pen input as the interface for the observer. It displays an array of buttons, each of which represents a task, event or environmental condition that occurs in the observed situation (Figures 1,2). When an event is observed, the observer presses the corresponding button on the Tablet PC to indicate its beginning and end. The button's border color signifies the event's current status.

Before the tool can be used for a specific scenario, an initial list of button labels needs to be created. This list can be the result of a traditional observation session, where the focus is on determining relevant tasks, events and environmental conditions. However, there is no need for this list to be complete, as unforeseen events can be easily added.

If an unexpected event occurs during the observation, new buttons can be added on the fly. First, the observer indicates an unknown event has started by tapping a button in the upper left corner. Then he uses the blank space next to it to create a label by scribbling the name of the event or a rough symbol. Finally, a new button is added using the picture as a label. If there are no blank buttons left, a new row is added to the layout. Because writing the label can be delayed, the time error made is reduced. Pictures used for new buttons are stored on disk and can later be converted into plain text.

Task Trace Graphs

After an observation, the data is compiled into a task trace graph, which presents the raw timestamps in a graphical format. The observer is immediately able to add comments to the graph, while the memories are still fresh. This is a major improvement over existing methods that require a video analysis before this step can be taken.

Figure 3 shows an example task trace graph of a gastroscopy examination we observed during our studies. It represents only one instance of such an examination. In practice, several such task trace graphs are used to get a more complete picture of the variety of situations that can be encountered, because each situation observed potentially offers different insights.

The activities in the chart are ordered from top to bottom by their first appearance to reflect the progression of time. The three areas in the graph marked with capital letters are of special interest, because each represents one aspect of the situation that strongly influences the space of suitable designs. The other parts of the graph are numbered and explained in Figure 3. The basic structure of the graph (1-3) is generated automatically from the observational data gathered using the TaskObserver tool. It is available immediately after the observation. The other parts of the graph (4,5) were added later to offer additional insights and explanations.

The spatial arrangement of the different parallel tasks and environmental conditions makes the temporal structure of the user’s primary task instantly accessible. With several graphs representing distinct observations, temporal patterns can be identified. Annotations can help point out interesting situations that are not obvious from the graph alone.

Experiences

This section gives a brief description of the author’s experiences of using the methodology in a wearable computing project.

The scenario observed in this example was an endoscopy examination in the outpatient area of a hospital. The examination’s goal was to inspect the gastrointestinal tract with an endoscope. The endoscope is a long flexible tube with a camera at its tip. It is operated with both hands and the video picture is displayed on an external monitor. Our goal was to give the doctor access to electronic documentation during the examination procedure using a computer system and appropriate interaction methods. Therefore we had to find out which interaction methods were possible and suitable in this specific situation, where the doctor is primarily occupied with the examination itself. Hospital rules allow no video or audio recordings of any kind to protect the privacy of the patients.

The TaskObserver software was used by the authors during eight observations of endoscopy examinations. The software turned out to be a valuable tool, allowing the observer to concentrate on what was actually happening. Besides using the tool it was still possible to take notes and write down questions about situations that were new to the observer on a sheet of paper.

As expected the recordings were not 100% accurate. Some errors occurred due to technical problems (e.g. inaccurate pen input), some could be accounted to human error (e.g. forgetting to start/stop a task). However, many coarse errors were easily corrected when looking at the task trace graph, shortly after the observation. As expected, the timestamps were inaccurate within a few seconds due to delays introduced by the human observer. However, the goal of capturing the temporal structure of the endoscopy examination was achieved. Using the task trace graphs it was possible to communicate the implications of the task for the design of the system to other team members.

Conclusions

The tool TaskObserver provides a way to rapidly gather information about the temporal structure of a situation. It provides information that is less accurate than video analysis, but also takes considerably less time to perform and does not require a video recording. It can therefore be used in sensitive settings where video recordings are impossible.

Graphical charts of these task traces are produced automatically and can be enhanced and corrected, to augment the graph with annotations and explanations. Together with information gathered through other methodologies like interviews and focus groups, these charts improve the process of transferring the knowledge about the observed situation to other the development team members like designers and developers.

Our own work was positively affected by the presence of the charts, but extensive test within other projects are still pending. Another goal is to find out how the granularity of observed tasks affects the annotation accuracy and error rates. Without this information it is not possible to decide whether real-time logging is feasible for a given scenario or not. Another question that needs additional studies is how the users are influenced when the observer uses a Tablet PC instead of pen and paper to take notes.

We will continue to use the method discussed here in our own projects and the experience we gather there will help us further improve the usability of the software. We will also look at how we can assist the annotation and correction of task traces with editing software.

Acknowledgements

We thank the Dr.Adamer for his help in organizing all the field studies at our partner hospital. Furthermore, this research was partly funded by the European Commission as part of the IST project wearIT@work[8] (No. IP004216-2004).

References

[1] Creighton, O., Ott, M., Bruegge, B.: Software Cinema-Video-based Requirements Engineering, In *RE '06: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*, pp. 106-115, 2006

[2] Harrison, B.L., Owen, R., Baecker, R.M.: Timelines:An Interactive System for the Collection and Visualization of Temporal Data. *In Proc. of Graphical Interface'94*, 1994

[3] Mangold International GmbH, http://www.mangold-international.com/

[4] Millen, D.R.: Rapid ethnography: time deepening strategies for HCI field research. *In DIS '00: Proc. of the conference on Designing interactive systems*, 280–286, New York, ACM Press 2000

[5] Noldus Information Technology, http://www.noldus.com/

[6] Pedell, S., Vetere, F.: Visualizing use context with picture scenarios in the design process. *In MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pp.271–274, New York, ACM Press 2005

[7] Weber, K., Poon, A.: Marquee: a tool for real-time video logging, *Conference on Human Factors in Computing Systems*, 1994

[8] wearIT@work EU Project, http://www.wearitatwork.com/

[9] Wickens, C.D.: Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2):159–177, 2002.