TaskObserver: A Tool for Computer Aided Observations in Complex Mobile Situations

Tobias Klug[†‡]
t.klug@sap.com

SAP Research CEC Darmstadt[†]
Darmstadt, Germany

Max Mühlhäuser[‡]
max@informatik.tu-darmstadt.de

Telecooperation Group[‡]
Department of Computer Science
Technische Universität Darmstadt
Darmstadt, Germany

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.

A user study was performed to estimate the accuracy of the real-time annotations and the amount of time that can be saved by using the system. Additionally the usability of the existing tool was evaluated.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Theory and Methods

Keywords

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

1. 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. 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 [10]. 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 [7]. 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 [5]. It was developed as part of the wearIT@work project [8] 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. The accuracy of the proposed method for real-time annotations has been evaluated in contrast to the amount of time that can be saved by using such a system compared to traditional offline video annotations. To this end a user study was conducted.

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 diagrams, are presented. The next section covers the evaluation of the TaskObserver software, followed by its results and conclusions.

2. 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 [7] 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 unavailable. Also, video recordings are sometimes 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 [9]. Both tools can be used to annotate video recordings in real-time. However, these 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 [4] or The Observer [6] 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 hectic scenarios.

3. 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 fractions of a second. 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. cramped spaces).

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 jotting notes takes too much time if several events occur within seconds of each other there is a high chance of missing something important.

Post observational analysis of video material results in extremely accurate data. However, this accuracy is not always necessary. 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 accurately keeping the time of each task performance. We call the concept of using a computer system to support the observation process itself computer

Figure 1: TaskObserver software running on a TabletPC.

aided observations.

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. An estimate of how accurate the solution is has been evaluated in a users study (see Section 6).

4. 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 (see Figure 1). 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 below 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.

5. TASK TRACE DIAGRAMS

After an observation, the data is compiled into a task trace diagram, which presents the raw timestamps in a graphical format. The observer is immediately able to add comments to the diagram, 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 2 shows an example task trace diagram of a gastroscopy examination we observed during our studies. It represents only one instance of such an examination. In practice, several such task trace diagrams 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 diagram are ordered from top to bottom by their first appearance to reflect the progression of time. The parts of the diagr are numbered and explained in Figure 2. The basic structure of the diagram (1-3) is generated automatically from the observational data gathered using the TaskObserver tool. It is available immediately after the observation. Parts 4 and 5 of the diagram were added later by the observer 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 diagrams representing distinct observations, temporal patterns can be identified. Annotations can help point out interesting situations that are not obvious from the diagram alone.

6. EVALUATION

Real-time annotation offers a trade-off between data accuracy and time taken for analysis. However, a decision for one of the options cannot be made if neither accuracy loss nor time gain can be quantified. To gain insights into these issues, a user study was designed to compare the two approaches. The evaluation was conducted with two major goals in mind. The first major goal was to find out how much time can be saved by real-time annotation compared to offline video annotation. The second major goal was to identify the types of errors made using the TaskObserver and to quantify these errors. The effects of different types of events/activities on the accuracy should be investigated. Together these results would allow an informed decision for a given setting. Apart from these goals, the study was used to gather usability feedback about the TaskObserver software from the participants.

The study was split into two parts, each with its own set of participants. The first group used TaskObserver for real-time observation, the second group used a video annotation tool [3] instead of the TaskObserver to perform an offline analysis. The first group's results were used to compute the accuracy as well as to identify usability issues with the TaskObserver while the offline annotations were used to quantify the extra time necessary to perform a video analysis compared to the real-time annotation.

Next the setting used for both study parts will be explained, followed by descriptions for each part of the study. Afterwards the study's results are presented and implications discussed.

6.1 Observation Setting

The task of preparing a meal was chosen as the basic setting for both parts of the study. A person could be seen doing various chores in a kitchen. The overall structure of the scene was as follows:

First the cook consulted the recipe on the kitchen counter. Then he put water on the stove to boil. Afterwards the ingredients were slowly added to the pan while regularly stirring its contents. Once the water was boiling, noodles were put to boil. When done, the noodles were splat cooled and added to the pan. When the meal was done, the contents of the pan were put into a bowl and the pan was washed and dried. In parallel to these tasks, activities like adjusting the stove heat or consulting the recipe were performed.

The observation setting was carefully selected and individual instances of the scene were created with the study's goals in mind. Four distinct instances of this basic structure were filmed. Each instance was different in which ingredients were used in which order and also contained at least one event not present in the other instances. These additional events ranged from tasks like cutting vegetables to operating a mobile phone. This variance was created to reflect the circumstances in real field observations where two instances of the same task are rarely the same. Especially during the first observations, new activities are frequently observed.

This setting also possesses the variety of events required earlier. It contained hectic parts with many things happening in rapid succession or even in parallel, but also quiet parts where events happened only occasionally. Participants were asked to capture short (length < 4 seconds, e.g. adjusting the stove heat), medium and long (length > 60 seconds, e.g. pot standing on the stove) events to find out if this had any effect on the capturing accuracy.

Events also varied with how easily they would be noticeable by an observer. They included tasks that were the main activities of the actor, secondary activities and events only implicitly related to the actor's actions. For example stirring the pan's contents was one of the actor's main tasks, whereas adjusting the stove heat was frequently performed in parallel, sometimes without even looking at the controls. Environmental conditions like the presence of the pan on the stove were the other end of the spectrum as they were not direct actions of the actor.

Last the existence of new tasks in each instance forced the observers to use the function of adding additional buttons which allowed testing this particular feature.

6.2 Part 1: Real-Time Annotation

In the first part of the study, participants were asked to use the TaskObserver system to annotate the four cooking scenes. In order to create a realistic observation experience, the videos were projected almost life-size onto a wall. TaskObserver requires and initial list of activities that is normally created after a first observation session. In this study, however, a master actionlist was created and used

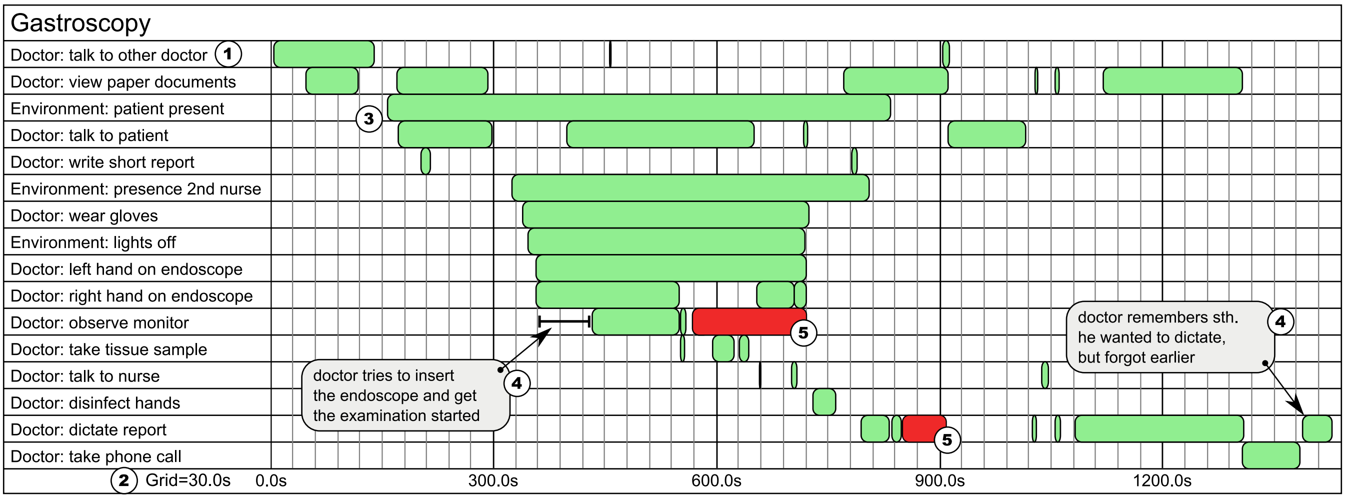

Figure 2: Task trace diagram of a single gastroscopy examination, with a focus on the doctor. (1) list of activities being observed, (2) timeline, (3) bars show when and how long an activity was observed, (4) manually added explanations and (5) bars edited manually, because of erroneous data.

throughout the whole study for comparability of results. As a consequence, in contrast to the tool's normal use, the participants were neither familiar with this list nor with the setting of the cooking scene.

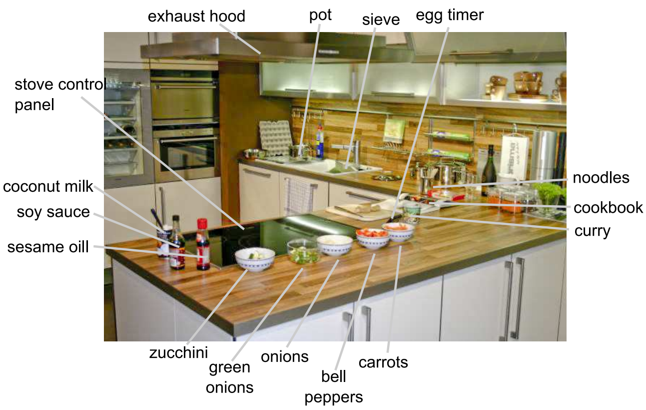

Figure 3: Picture used to introducte the cooking scene to participants. Buttons for the row of bowls for example were arranged according to their spatial layout.

To reduce the unfamiliarity of the participants with that list, it was engineered to be easily memorable and unambiguous during several iterations before the study. Memorability was achieved by clustering actions of similar content like ingredients, equipment and environmental conditions. Where possible the buttons were also laid out corresponding to the spatial location of that activity within the video.

The study itself was organized as follows. First, the users were introduced to the TaskObserver software and the cooking setting, Figure 3 shows the introductory picture shown to participants. After this brief introduction, they were shown the first video that was meant solely for practicing the use of the software and for clarifying the semantics of certain actions and events as well as getting to know the cooking scene. Then the participants were asked to annotate the remaining three videos without further assistance. In the end the users were asked questions concerning the tool's usability.

Often observations in the field require the observer to remain standing or move around. In these cases it is important to know weather carrying a TabletPC is feasible. For this reason, participants were asked to remain standing as long as they felt comfortable, but were given the choice to sit down.

6.3 Part 2: Offline Annotation

The second part of the study was carried out in order to estimate the time needed for an offline annotation of the same scenes used in the real-time annotation. Here a software package called ANVIL [3] was used. It is a mature tool that features multitrack annotation of videos and a clean and efficient user interface (see Figure 4). The initial actionlist and introduction were the same as for the participants of part one of the study. After the introduction the first video from the set was used for explaining the use of ANVIL and clarifying the semantics of activities. The participants were given as much time as they felt necessary for this training. Then participants were asked to annotate two additional videos without interruptions. The time taken for each of these videos was measured and compared to the duration of the source video.

7. RESULTS

The study was conducted with staff members of the computer science department of our university. A total of 8 participants performed the real-time annotation. Only one participant had done user observations before and three had experiences with using a TabletPC. Another 5 participants

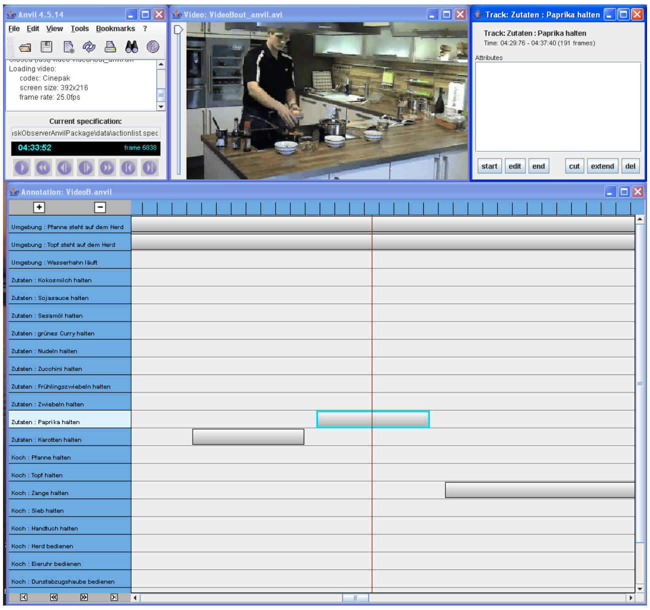

Figure 4: ANVIL offline video annotation tool used.

took part in the offline annotation study. None of them had used a video annotation tool before or had any experience with user observations.

The results are structured according to the goals formulated earlier. First the estimated time needed for the offline annotation is presented, followed by the real-time observation accuracy results and the usability findings.

7.1 Time saved

The five participants of the offline annotation study were free to decide how much training they needed. Therefore the time taken for training ranged from 8 minutes to over an hour. How much time it took each participant to annotate the videos can be seen in Table 1. Because of technical problems, the performance measure for participant 5, video A is missing. In average (considering both videos) the video annotation took about 3.2 times as long as the source video (std.dev.=0.4).

If we assume that the time needed for setting up and operating the video equipment is roughly equal to the amount of time necessary to prepare the initial actionlist, we can ignore the cost for setting up observations. When doing an offline annotation, the time for taking the video adds to the time for the annotation leading to a total time of 4.2 times the scenario's duration. In contrast, the time necessary to do a real-time annotation equals the duration of the scenario observed plus the time needed to correct errors made during the observation. Typically the time needed for corrections is about 20% of the scenario's duration. However, for the purpose of this estimation we assume an upper bound of 50% required for corrections. This leads to a total time of about 1.5 times the scenario's duration. Therefore, using the TaskObserver equals time savings of roughly 65%.

Table 1: Time needed for offline annotation and multiple of video length.

	Video A		Video B	
Participant	time	factor	time	factor
1	32:42	3.17	32:29	3.05
2	28:55	2.81	27:53	2.62
3	32:19	3.14	38:55	3.66
4	40:09	3.90	34:02	3.20
5			36:53	3.47
Video Length	10:18		10:38	
Average	33:31	3.25	34:02	3.20

Example: If the scene observed was 10 minutes long, the time necessary for an offline annotation would be 42 minutes, 10 for shooting the video and $10 * 3.2$ for the annotation. The real-time annotation would take only 15 minutes for the observation and corrections, saving 27 minutes for this particular scenario.

7.2 Accuracy

The participant's annotation were compared to a reference annotation created with the offline annotation tool to compute various accuracy measures. The annotations of the first video were excluded, because this video was used mainly for training. Events from buttons that were manually added by the participants were also excluded, because they were different for each participant. With between 45 and 65 events remaining for each video, a total number of 1123 events was recorded by the participants. These events were manually mapped to the reference annotations for comparison.

While mapping user annotations to the reference dataset, four basic classes of errors were discovered: missing events, extra events, missing splits and extra splits.

Missing events(10.52%) were present in the reference dataset, but non-existent in the real-time annotation. This kind of error happened mostly, when the participant was too distracted to notice a relevant event had happened. Some of these errors were caused by distractions through the use of the TabletPC for example, when the participant was looking for a particular button which he could not find. Others happened, when the participant was concentrating on some other event happening. For example the "adjust stove heat" task was frequently missed when happening in parallel to the "stir pan contents" task.

Another frequent mistake were extra events not present in the reference set. This case frequently happened, when the observer pressed the wrong button on the TabletPC, sometimes correcting his mistake, sometimes not. This error also occurred when the observer falsely anticipated an event that didn't actually happen or when the observer had a different understanding of what counted as a certain event and what didn't (e.g. "Does lifting the pot's lid count as handling the pot?").

Less frequent mistakes were missing (0.8%) and extra splits (4.22%). A missing split was counted, when two successive events of the same action were annotated by only one event in the observer's data, i.e. the gap in between was

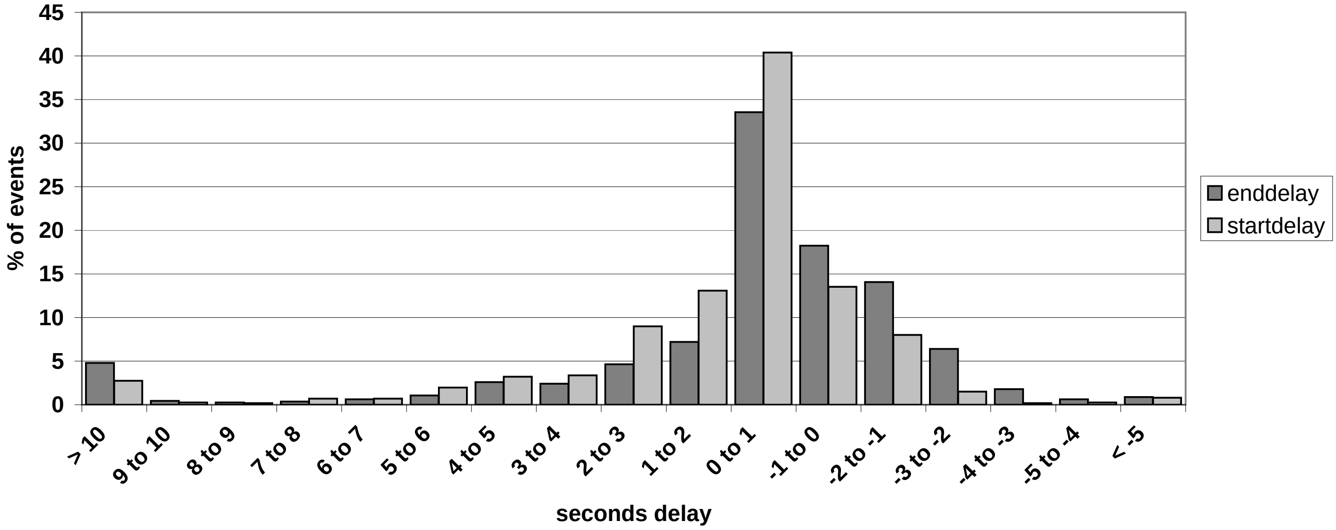

Figure 5: Distribution of event start delays(light bars) and event end delays (dark bars). Positive values indicate the annotation was made after the reference. Negative values indicate annotation made too early.

Table 2: Percentage of annotations that fall into a certain error range.

range [s]	startdelay	enddelay
-4 to 4	89.06%	88.26%
-3 to 3	85.50%	84.07%
-2 to 2	75.00%	73.04%
-1 to 1	53.91%	51.78%

missing. An extra split denotes the opposite case of a single reference event being approximated by two successive events, introducing a gap.

Additionally to these major mistakes, the accuracy of annotations was measured. To do this, the delay of the beginning and end of each event was calculated. Figure 5 shows the distribution of these delays. As we can see both distributions peak in the range between zero and one second delay compared to the reference event. The distribution of enddelays is slightly shifted to the right, meaning the end annotations tend to be too early. This can be accounted to the fact, that observers have already noticed the event is taking place and are now more alert to when it ends. The existence of quite a large amount of annotations being early is due to the fact, that humans start anticipating activities after they have seen them a few times.

Looking at the range of being less than one second late or early compared to the reference event, we find a little over 50% of all annotations made (startdelay=53.9%, enddelay=51.78%) and about 85% of annotations have an error of less than three seconds (see Table 2).

However, the accuracy was not equal for all event types as can be seen in Table 3. As described in Section 6.1, we suspected 3 event features to have an effect on annotation accuracy: event frequency, event duration and noticeability. As can be seen in Table 3, the event frequency does not seem to have a large effect on the annotation accuracy. “Hold spoon” and “hold pan” both occurred frequently. However, “hold spoon” was almost never missed, while about 40% of all “hold pan” events were missed. Their average duration is also not very different, which leaves their noticeability as the discriminating factor. This hypothesis can be confirmed, when looking at the videos. The “hold pan” task was frequently performed in parallel to other tasks and without getting any attention by the actor. “Hold spoon” on the contrary was always the center of attention, when performed. The same observation as for the “hold pan” task holds for the “operate stove” task.

Other tasks that had a low noticeability were “pot on stove” and “pan on stove”. Both events were only indirectly started and ended through the actor, but because of their average duration of over 5 minutes they were never missed. However, these events were the most frequently delayed. This indicates, that extremely long lasting events are a bad choice for an actionlist, as they are less likely to be captured accurately. A better choice would be more noticeable actor tasks that can be associated with the start and end of these events.

Very short tasks like “operate egg timer” and “operate exhaust hood” were also more frequently missed or delayed than other tasks, despite their excellent noticeability.

7.3 Usability

After the real-time observation each of the 8 participants was asked to answer a series of usability related questions. All 8 participants found the TaskObserver software easy to use. When asked about the usability of adding new tasks, opinions were mixed. Four found the current implementation easy to use, three had minor problems with the different modes involved and one didn’t like the way new tasks were added. The most frequently mentioned problem was the fact

Table 3: Accuracy statistics by event. Average duration of tasks was measured in the reference dataset. All other values were computed comparing the reference with participants annotations.

event name	events	total missing	startdelay $> 5s$	enddelay $> 5s$	average duration
pot on stove	23	0,0%	4,3%	**52,2%**	**354920**
pan on stove	35	0,0%	**20,0%**	**11,4%**	**276024**
hold spoon (for stirring)	**260**	0,8%	4,6%	3,1%	19384
hold sieve	46	**21,7%**	**8,7%**	0,0%	16786
water tap runs	84	6,0%	4,8%	4,8%	14400
hold pot	46	2,2%	6,5%	4,3%	14060
hold pan	**190**	**39,5%**	**8,4%**	8,4%	12960
hold green curry	15	0,0%	6,7%	0,0%	11780
hold noodles	38	0,0%	5,3%	**10,5%**	11392
hold green onions	22	4,5%	4,5%	9,1%	11013
hold sesame oil	31	0,0%	6,5%	3,2%	8780
hold onions	23	0,0%	**8,7%**	8,7%	8307
hold towel	47	2,1%	2,1%	**10,6%**	8206
hold soy sauce	23	0,0%	4,3%	4,3%	8173
hold carrots	15	0,0%	6,7%	0,0%	7720
hold zucchini	23	4,3%	4,3%	4,3%	6133
hold peppers	23	0,0%	0,0%	0,0%	6026
read recipe	61	3,3%	1,6%	3,3%	4505
operate stove	**181**	**16,0%**	2,2%	6,1%	**3686**
operate egg timer	38	5,3%	2,6%	7,9%	**2528**
operate exhaust hood	31	**9,7%**	0,0%	9,7%	**1510**

that several button presses had to be done in a specific order to complete the task. Also, the current implementation does not allow adding tasks without also activating them. This could be helpful if new tasks already added in a previous observation are likely to happen again and the observer wants to add the button during a time of low activity. Another suggestion made was to activate tasks when starting to write in the scribble field to reduce the number of clicks necessary.

When asked whether the participants were able to concentrate well on the video shown, 5 answered that the ability to concentrate was directly linked to the amount of time spent operating the TabletPC. Especially when the location of an event happening rarely was not known by heart and had to be found first, other events in the video were missed.

To the question if they were able to memorize the locations of specific buttons, the answers were similar. Frequently used buttons and those arranged according to the spatial arrangement in the video were memorized easily. Rarely used buttons and those that were arranged illogically could not be memorized at all and had to be searched for every time.

These answers indicate that special care needs to be taken when arranging the initial buttons. An arrangement that corresponds to the scene being observed and anything else that can serve as a reminder should be taken into account. Also the amount of 21 initial buttons seemed to be near the upper limit of what can easily be memorized.

During the study we also observed how the participants got along with the TabletPC. Only 3 of the 8 participants were able to use the device in a standing position during the whole study of about 60 minutes. However, even those three complained, that the TabletPC was uncomfortable to use after a while, because of its weight and the constant strain put on the arm holding it. Two participants remained standing till after the second video and the remaining three sat down during or after the first video.

When adding new tasks during an observation, participants used a variety of methods to distinguish these new buttons. Some used one or two words describing the button's topic, others used letters or even simple sketches. However for the last video most participants used very simple symbols or letters, because these could be written faster and memorizing a button's meaning was never a problem.

8. 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.

Compared to offline video annotations, the TaskObserver approach saves roughly 65% of time per observation made. However, at the same time an error is introduced. In the study presented, roughly 10% of events were missed and about 85% of events were delayed up to 3 seconds compared to a reference annotation. The remaining 15% had even larger delays. However, the overall temporal structure remains intact, as can be seen in Figure 6 comparing a reference annotation with its real-time counterpart.

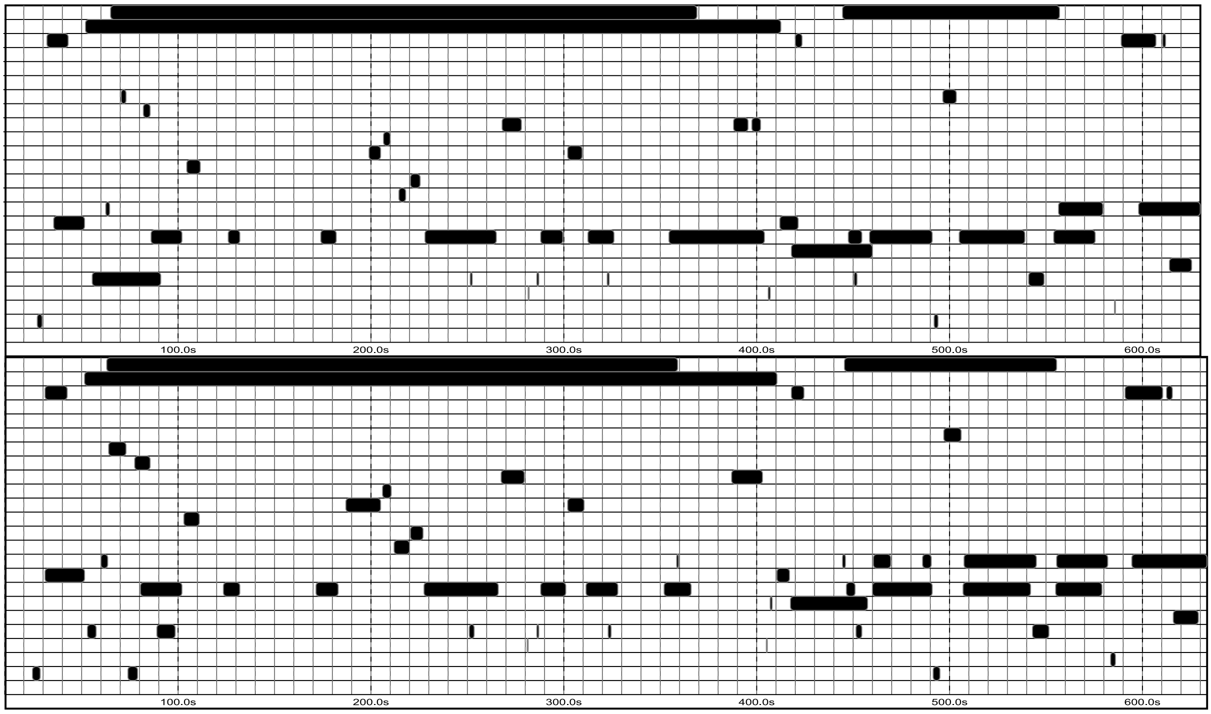

Figure 6: Comparison of a real-time annotation(top) with its reference(bottom).

The study results indicate that the selection of appropriate events for the initial actionlist is important to achieve a good annotation accuracy. First of all selected events need to be very noticeable to an observer. Important events that are only indirectly influenced by the actor should be avoided as well as events with a short average duration. About 4 seconds seems to be the lower limit for easily observable events.

The overall usability of the TaskObserver software was found to be good by all the study's participants. Existing shortcomings can be countered by carefully selecting the initial actionlist and arranging the buttons intelligently. If the user is required to remain standing during the observation, the device is of limited use due to its weight. The amount of time the device can be used in a standing position largely depends on the physique of the observer. Newer more lightweight TabletPCs could reduce this problem.

Future research will focus on additional studies of using the TaskObserver system in the field and further improving the usability of the tool.

9. ACKNOWLEDGMENTS

This research was partly funded by the European Commission as part of the IST project wearIT@work (No. IP004216-2004).

10. REFERENCES

[1] O. Creighton, M. Ott, and B. Bruegge. Software cinema-video-based requirements engineering. In *RE '06: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*, pages 106–115, Washington, DC, USA, 2006. IEEE Computer Society.

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

[3] M. Kipp. *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation.* PhD thesis, Saarland University, Saarbrücken, Germany, December 2004.

[4] Mangold International GmbH. http://www.mangold-international.com/, 2007.

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

[6] Noldus Information Technology. http://www.noldus.com/, 2007.

[7] S. Pedell and F. Vetere. 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*, pages 271–274, New York, NY, USA, 2005. ACM Press.

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

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

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