Task Switching in Audio Based Systems

Melanie Hartmann and Dirk Schnelle

Telecooperation Group
Darmstadt University of Technology
Hochschulstraße 10
D-64283 Darmstadt, Germany
{melanie, dirk}@tk.informatik.tu-darmstadt.de

Abstract. The worker on the move has an ever-increasing need to access information, such as instructions on how to process with a task. The use of audio to convey that information and for interaction has many advantages over traditional hands&eyes devices, especially if the user needs his hands to perform a task. In this paper, we focus on a task model stored in a workflow engine. The execution of a task is often interrupted by external events or by the user who wants to suspend a task or switch to another one. If the user wants to resume the task he has to be aware of his current position in the workflow. Due to the transient nature of speech, he does not have the possibility to review what he has done before in audio-only systems. In this paper, we present a novel approach, based on psychological theories, to assist the user to get back into the context of an interrupted task. The usability of this recovery concept was successfully tested in a user study.

1 Introduction

The future workplace will get more and more complex and lead to an ever-increasing need to deliver information to workers, such as manuals or instructions on how to proceed with a task. The use of audio to convey that information and for interaction has many advantages over traditional devices like a PDA, especially in those environments, where the user needs his hands and eyes to perform the actual task.

In this paper, we focus on tasks that can be described in a workflow definition language. The execution of a task is often interrupted by external events or by the user who wants to suspend the task or switch to another task. If the user wants to resume the task, he has to be aware of his current position in the workflow. Due to the transient nature of speech, he does not have the possibility to review what he has done before if he is dealing with audio-only systems. The user has to know how he can ask for information about the current state of the process, which leads to a decreased usability especially for novice users. In this paper we present a novel approach to assist the user to get back into the context of an interrupted task.

2 Why Is Task Switching a Problem?

Before answering that question, we give a short overview of the terminology being used in this paper.

Petr Sojka, Ivan Kopeček and Karel Pala (Eds.): TSD 2006, LNAI 4188, pp. 597–604, 2006.

We regard a **task** as an arbitrary sequence of actions which belong together and share a common goal. A **task switch** is the process of stopping or pausing the execution of a task for the benefit of a second task. A task switch decomposes in its components *start*, *interrupt*, *resume* and *end*. An **interruption** of a task is defined by Corragio [1] as a discrete event that breaks continuity of cognitive focus on the primary task. An interruption can be caused by an external source (e.g. another person), by the user himself or by our system in the following referred to as task-handler (e.g. by informing the user about an incoming email).

If the user wants to resume a task, he has to restore its context. At this point several challenges are faced:

- How much information is needed to help the user resuming the task?
- Which information is necessary for the resumption?
- How should the user interact with the task-handler considering the transient and invisible nature of speech?

In order to determine the relevant information for getting back into the context of a task, we refer to some basic psychological assumptions stating that goals are crucial for the problem solving process [2]. In order to achieve a goal it often has to be decomposed into subgoals. Having achieved the subgoals, the cognitive system has to evaluate if it can now continue with the corresponding supergoal. Therefore, this supergoal has to be readily available in memory. Goals are assumed to be in a stack-like cognitive structure, ensuring that the items belonging to the control flow reside in memory. The newest goal on the stack directs the behavior [3,4]. In this cognitive architecture, goals are sources of activation without requiring active maintenance or rehearsal and they are linked associatively to other goals via task constraints. Our approach bases on the cognitive architecture ACT-R (*Adaptive Control of Thought, Rational*, [3]), which consists of a set of inter-related mechanisms to simulate and explain human cognition. Its basic processing assumption states that, when the central cognition queries the memory, it returns the most active item in memory (corresponding to the newest goal), which then directs the behavior. Hence, our system has to assist the user to activate all relevant items needed to perform the current task.

3 Related Work

To our knowledge there is hardly any research on how to help the user resuming a task, especially in the audio domain. In this chapter, we present an interface developed by Franke, Daniels and McFarlane [5] and discuss its relation to our approach.

They developed a "spoken dialog interface system for a radio-based human-software agent military logistics task" that contains context review mechanisms. For that purpose, the user is provided with special commands to query the interface about aspects of the previous task. This can be general queries like *"Where was I?"* and also specific questions, e.g. *"Which supplies were ordered?"*. Additionally, the user can request a full progress review of the interrupted task.

In comparison to our work, Franke et al. focus on rather simple and short tasks, in a specific application area with very formal standardized military communication, whereas we want to develop a more general approach, which can be used for every workflow description in order to be independent of the application domain. With Franke's approach, the user has to be trained which questions he can use to get the information he wants. We cannot act on this assumption in our work, surely the user has to know some basic commands, but he has to be able to quickly find his way through the task without having much experience. Additionally, Franke et al. do not vary the amount of repeated information, e.g. depending on the elapsed time, because they only cope with short tasks and short interruptions, whereat only few information accumulate. In contrast, we most likely deal with a larger amount of data which would be too annoying to listen to, if the interruption was not really disruptive. Thus we have to adapt the presented amount to the current situation.

4 Concept

We assume that a workflow has one main goal (e.g. *Repairing a car*) which can be split into several subgoals (*Error diagnosis* and *Fix part*). These can in turn consist of some subgoals (*disassemble*, *clean* and *reassemble*). Each subgoal can thereby be associated with a subtask of the workflow or a workflow step.

Further, we suppose that it is sufficient for short interruptions to restore the context of the current subtask, because the higher level goals are better encoded and are therefore not affected that much by retention loss. If the interruption takes longer, the user also needs help to restore the higher level goals, thus more information has to be presented.

In this section, we present how we determine the amount of required information, how the knowledge about the current task is obtained and stored and how the user can interact with our system.

4.1 Determining Amount of Required Information

In order to determine how much information the user needs for resuming a task, we need to estimate the adverse effect of the interruption. Therefore, we identified several factors divided in four categories that have an impact on the disruptive effect of an interruption. These factors are rated on a uniform scale and its weighted sum estimates the adverse effect of the interruption.

- Characteristics of the primary task: cognitive load at the point of interruption [6,1,7]
- Characteristics of the interruption and its announcement: e.g. interruption frequency and predictability [1,8]
- Characteristics of the interrupting task: duration and interference with the interrupted task [1,7,9,10]
- Characteristics of the user: involvement in ongoing task and expertise [10]

4.2 Knowledge Representation

As mentioned before, a task consists of several subtasks and thus subgoals. If a goal of a task is activated, it activates all individual propositions of the task. In contrast, a single proposition (e.g. a workflow step) mainly activates the propositions it is directly linked to. Thus the priming effect of a goal is superior to the one of a workflow step. An experiment conducted by Trafton et al. [11] also indicated that being aware of the current goals is more important than remembering the current position. This shows the importance of being aware of the current goals and subgoals. Additionally, the user needs to know what he has done at last (position information). However, the goals are sometimes not explicitly stated. In this case we can only help the user with the repetition of the shared history (position information), which induces the priming of the user's goals. The goal and position information is stored hierarchically. Thereby, the goal-items describe the aims of the task in different granularities and the position-items what the user has already done. Figure 1a illustrates such a goal-hierarchy in form of a tree which contains all possible goals during a shopping task. The steps of the task are represented by its leaves. The relevant information for a given step consists of all goals located between the root of the goal-hierarchy (the task's goal) and the node corresponding to the step. The position items are specified by all already achieved goals that are directly linked to a goal on this path. Figure 1a points out the relevant goal and position items for the step *enter credit card number*. In order to aid the user in resuming a task it is sufficient to store these relevant information items (adapted to the user's situation) in a goal-stack as can be found in Figure 1b. The order of elements in the stack is determined by the breadth-first search of the corresponding goal-tree.

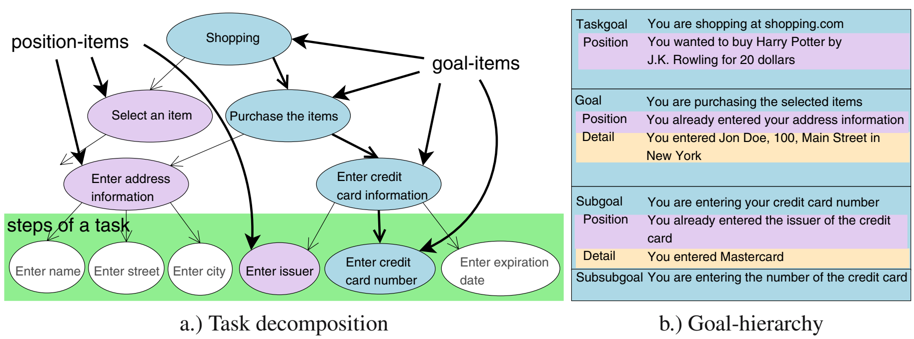


Fig. 1. Example of a goal-hierarchy using the example of a shopping process with highlighted goal and position items for the point of entering the credit card number

How much information (goal and position information) is really repeated is determined by the factors listed in the previous section. If the user needs very few information, the system only provides him with the last position information, because we can assume that higher goals are still sufficiently activated or that at least the presentation of some cues (like the current position) suffices to reactivate them. If the user lacks little more information, the system starts at a higher level goal and repeats all underlying goal and position information. Every item can also feature some detail information, which are read with the corresponding item and must never be repeated alone. Every goal and subgoal marks a starting point in the goal-stack at which the system can start reading the information. However, only the last position item can act as a starting point, because we think that the last performed action is sufficient to prime the higher goals for short interruptions.

The stored data cannot only be used to help the user to get back into the context of the interrupted task, it can also aid the user orienting himself during the performance of the task. For that purpose, we simply have to provide him with the possibility to ask for this information. Further, we can use the same storage structure for some additional information which might be useful during the task execution, e.g. the content of the shopping cart. This allows for an easy and uniform way of providing the user with additional information about the current task.

We specified an XML-structure (*RecoveryXML*) to store the goal-stack. It contains several so-called information containers. The most important one is the position-container. This container consists of the goal and the position information. Further containers can be used to provide the user with additional information as mentioned before. The information which is stored in the RecoveryXML file is derived from the task's workflow description (specified in XPDL [12]) during the task execution. The goal and position information has to be added as extended attributes of the workflow activities and their transitions.

4.3 Interaction

We provide the user with the possibility to skip the repetition and to ask for more information if he is not satisfied with the represented amount. The same interaction is supported when he requests data about his current position or any other available information during the task execution. This provides a consistent way of interacting with the system during the whole performance of the task.

5 User Study

We conducted a user study to test the usability of our recovery concept. We measured the performance of users when they resume an interrupted task and when they are thereby supported by the repetition of different amounts of information. We divided the participants into three groups that were supported by different amounts of repeated information. We intended to find out whether the amount of information has an influence on the user's performance and on the user experience. Further, we aimed at determining how much information should be repeated for being helpful and without being annoying. In this section, we present the setup of our user study and its results.

5.1 Test Setup

For the user study we used a shopping task consisting of two parts: searching and purchasing. In order to search an item the user had to navigate in a category tree where the books and CDs, which can be bought, represent the leaves. To purchase the selected items, the user had to enter his credit card information, a billing and a shipping address.

Each user got interrupted once in every subtask (search, purchase) and had to perform a memorization task. For the task resumption some context information about the task was read to the user before the workflow continued at the point where it was interrupted. Thereby, the amount of information depended on the recovery strategy the user was assigned to. The recovery strategies used were:

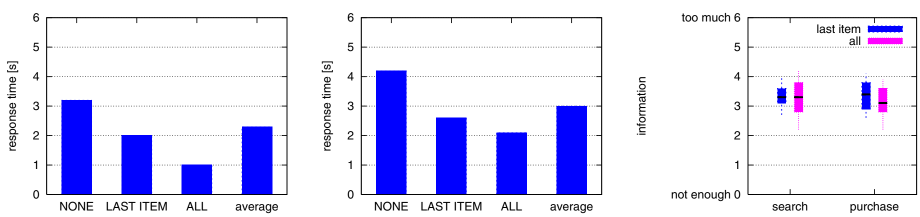
ALL Repeat all information collected from the shopping task so far
LAST ITEM Repeat only the last stored information item
NONE The shopping task continues without giving any additional information (control group)

Finally, the users filled out a questionnaire regarding their subjective feedback on the system.

5.2 Results

We found that the repetition of information really led to a decreased users' response time after the interruption and that most users rated the repetition as very helpful (the mean rating was 4.65 with 0 meaning "not helpful at all" and 6 "very helpful"). Further, we noted that the users who received the maximum amount of information were surprisingly not annoyed by its repetition. In this section, we present these results in more detail.

Response Times. We compared the response times (resumption lags) of the various groups using a one-way analyze of variance (ANOVA) to test whether the response time after the interruption was shorter for the participants in the ALL- and LAST ITEM-group than for members of the NONE-group. The results are illustrated in Figure 2. The differences for the various groups were significant for the search-part ($F(2, 25) = 5.1, MSE = 12.7, p = 0.014 (< 0.05)$) and marginal significant for the purchase-part ($F(2, 24) = 3.1, MSE = 12, p = 0.064 (< 0.1)$). The result cannot be due to the duration of the attention-switch, because we played a jingle indicating the restart of the shopping task to compensate this effect.



a.) Search b.) Purchase c.) Amount of information.

Fig. 2. Results of the user study: a.) and b.) show the response times after an interruption (average: average response time over all groups during the whole search or purchase part), c) shows the users' rating of the amount of repeated information

Amount of Repeated Information. We asked the participants to judge the amount of repeated information in order to test whether the members of the ALL-group rated the amount of repeated information as more annoying than the members of the LAST ITEM-group, whereby 0 meant "not enough information" and 6 "too much information". The result is shown in Figure 2 (group ALL: $M = 3.20, SD = 1.31$; group LAST ITEM: $M = 3.33, SD = 0.95$). The results are counter to our expectations, both groups found the amount of repeated information very suitable and we could not detect any differences between the groups.

This result is rather surprising because the users of the ALL-group perceived about 23s of repeated information for the search-part and 38s for the purchase-part (for comparison 4s and 3s for the members in the LAST ITEM-group). It is doubtful that these ratings persist, when the participants use the shopping process more frequently, but this has to be proved in further user studies

6 Summary and Outlook

In this paper we presented a novel approach to manage knowledge about performed tasks in order to help the user getting back into the context of an interrupted task. We assume that goals are the most important information to be remembered for resuming a task. We specified an XML-structure called RecoveryXML that stores all relevant information about the state of a task focusing on information about the current goals and about the current position in the workflow. The RecoveryXML structure is retrieved from the task's workflow description. In order to determine how much information the user needs in which situation, we took a closer look at the factors that make an interruption disruptive and used them to estimate the deleterious effect of the interruption.

Finally, we conducted a user study to prove the usability of our recovery concept and found that the repetition really helped users to resume a task.

In order to provide the user with the right amount of information, we have to test and refine our estimation of the disruptive effect of an interruption by conducting further user studies.

Our recovery concept is not limited to the usage in task-oriented workflows. It can also be helpful for the orientation in structured texts. In order to transfer our concept to the reading of text, we have to re-define the meanings of goal- and position items. Goals correspond to headings or the currently read sentence. The position items comprise the last paragraph, a text-summary for the currently read section and the already read headings (like for workflows only those items are of interest that are directly linked to the "goal-path"). A detail-item can optionally be appended to each heading. It can contain a summary of the corresponding chapter. In order to retrieve the recovery file automatically from the text's structure, we have to be able to summarize the texts automatically or find another way to speed up the repetition, for example using techniques like SpeechSkimmer [13]. This expansion to structured text can be useful for many application areas, for example for museum guides or for searching and jumping in audio-documents. In the latter case, the user is not aware of the context of the search term or the position he jumped to. Our system can help him by reading some relevant information to him, like headings or summaries, to understand the context.

Acknowledgments

Thanks a lot to SAP Research Palo Alto for supporting our research. Special thanks to Frankie James, her comments were very useful and helped to conduct the user study.

References

1. Corragio, L.: Deleterious Effects of Intermittent Interruptions on the Task Performance of Knowledge Workers. In: 18th international conference on information systems. (1990).
2. Anderson, J.: Cognitive Psychology and Its Implications. Worth Publishers (2004).
3. Anderson, J., Lebiere, C.: The atomic components of thought. Erlbaum (1998).

4. Newell, A.: Unified theories of cognition. Harvard University Press (1990).
5. Franke, J., Daniels, J., McFarlane, D.: Recovering Context after Interruption. In: 24th Annual Meeting of the Cognitive Science Society. (2002).
6. Miyata, Y., Norman, D.: Psychological issues in support of multiple activities. In: User Centered System Design. (1986).
7. Latorella, K.: Effects of the modality on interrupted flight deck performance: Implications for data link. In: Human Factors and Ergonomics Society. (1998).
8. Cohen, S.: Aftereffects of stress on human performance and social behavior: a review of research and theory. In: Psychological Bulletin, 88. (1980).
9. Altmann, E., Trafton, J.: Memory for goals: an activation-based model. In: Cognitive Science 26. (2002).
10. Edwards, M., Gronlund, S.: Task interruption and its effects on memory. In: Psychology Press Ltd. (1998).
11. Trafton, J., Altmann, E.: Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. In: International Journal of Human-Computer Studies, 58. (2003).
12. WfMC: Process Definition Interface - XML Process Definition Language. In: WfMC Specification. (2005).
13. Arons, B.: A system for interactively skimming recorded speech. In: ToCHI. (1997).