

Recovering Context After Interruption

Jerry L. Franke (jfranke@atl.lmco.com)
Jody J. Daniels (jdaniels@atl.lmco.com)
Daniel C. McFarlane (dmcfarla@atl.lmco.com)

Lockheed Martin Advanced Technology Laboratories
1 Federal Street, A&E-3W, Camden, NJ 08102 USA

Abstract

As information systems become more complex and present an increasingly rich amount of information to users, interruptions present an ever larger hurdle to operational efficiency. User interface technologies intended to support increased user-control of the transitions between computer-based tasks can help mitigate that obstacle. We present a three-pronged approach that combines dynamic interruption coordination support with context review mechanisms to aid user navigation through interruptions. These mechanisms are implemented within a spoken dialogue interface system for a radio-based human-software agent military logistics task.

Introduction

Modern information technologies continue to successfully deliver ever more powerful information products. This increase in power, however, can support the user in performing tasks quickly and accurately only if users can exploit this increased information flow for their own needs. People have a limited capacity for information management that directly affects the quality of their decision-making in stressful real-world tasks. If the ever-increasing information stream is not properly managed, these human capacities could become overloaded. The net result is that the adoption of a new information technology can actually cause an overall decrease in mission performance.

New, more powerful information technologies may increase the volume of important information delivered to decision-makers, but at the same time increase the frequency of interruptions of those decision-makers, degrading their information management capacity. The number of alerts that interrupt users affects how they manage their limited attentional cognitive resources. An interrupting alert causes users to switch from their current task to the new alert task. After completing the alert task, users must switch tasks again to resume what they had been doing prior to the interruption. The cognitive demands of these context switches increase the effective workload of users, which in turn increases the probability of mental mistakes.

For example, Foushee and Helmreich (1988) found that the disruptive effects of interruptions have

caused flight errors in commercial airline flights, sometimes resulting in fatal crashes. Human interruption is also a recognized problem in domains such as Navy command and control systems (Osga, 2000) and flightdeck or cockpit systems (Barnes, 1990; Adams and Pew, 1990; Adams et al., 1995).

The literature is rich with descriptions of the cognitive limitations people have relative to resuming tasks after being interrupted. Miyata and Norman (1986) give a general overview of the topic, discussing foregrounded and backgrounded activities and how interruptions are the standard way people switch between tasks in multitasking. Liu and Wickens (1988) discuss task interference and the effect of task type in human multitasking. McFarlane (2002) provides an in-depth review of the published relevant theory and proposes both a definition of human interruption and a taxonomy for classifying human interruptions.

Other studies investigate the causes of the disruptive nature of interruptions. McLeod and Mierop (1979) examine the effect of task similarity for manual tasks. Zijlstra and Roe (1999) found that the frequency of interruptions in an office environment affects the length of delay for people resuming the main task. Latorella (1998) found a modality interaction effect between how interruptions are presented (aurally or graphically) and the type of task that cockpit crew members perform (aural or graphical); different combinations of interface solution and task type resulted in different kinds of adverse effects on crew behavior. Linde and Goguen (1987) found that differences in how cockpit crews interact with each other affect their ability to successfully handle interruptions.

The objective of human alerting technology is to cancel the negative effects of human interruption and allow users to exploit the benefits of greater information volume for making better decisions. Human alerting mechanisms are being integrated within a broad range of commercial and military applications. These include announcement mechanisms for relatively less important systems like email, telephone, voicemail, internet instant messaging, chat rooms, automated help systems (like Microsoft's "Clippy"), computer-based tutor-

ing, and shopping agents. These applications also include many mission-critical systems including military command and control (C2), aircraft flightdeck control, power plant operations, spacecraft control centers, and real-time targeting sentinel-agent systems. McFarlane and Latorella (2002) present an in-depth discussion of the scope and importance of human interruption for HCI design.

Approach

There are three basic strategies for improving human performance on an interruption-laden multitask: (1) training (Hess and Detweiler, 1994); (2) selection of users (Joslyn and Hunt, 1998; Joslyn, 1995); and (3) user interface design. Due to the constraints of our real-world applications, we have focused our approach on the last option.

Our objective is to support efficient task recovery after interruption. It is useful to divide many user interface design approaches for human interruption into three phases. The pre-interruption phase prepares the user to transition from the main task to the interrupting task. The mid-interruption phase generally focuses on the user's transition to the interrupting task and includes the user's efforts and ability to maintain situational awareness of the main task while working on the interrupting task. The post-interruption phase sees the user return and reorient to the context of the original task that was interrupted.

Our approach has three parts, matching each of the three interruption phases.

Pre-interruption

Before the actual interruption takes place, the interface should give the user support for quick rehearsal of the current task before switching context to handle the interruption. Gillie and Broadbent (1989) noted that rehearsal may have potential for aiding human interruption in user interface design. Storch (1992) suggests that rehearsal may be useful in diminishing the negative effects of interruption after obtaining unexpected results in experiments unrelated to rehearsal. Detweiler et al. (1994) describes two experiments related to early warnings of interruptions that indicate that providing warnings is only marginally useful if the interruption task has a low memory load and is dissimilar to the main task while providing warnings can be extremely useful if the interruption task has a high memory load and is similar to the main task.

To allow the user to rehearse before interruption, some cue must precede or accompany the incoming alert. This cue helps to differentiate between the main task and interrupting task contexts and can take many forms, such as a visual flash, an audible beep, or a vibration. Because our particular applications involve a spoken dialogue interface, we have differentiated incoming alerts from the current task

by having the interface use a different voice. For example, the interface may use a female voice while participating in dialogue related to the user's current task, then switch to a male voice when introducing the interruption to the user. This cue gives the user the opportunity to register the alert, allowing the user to rehearse the context of the main task before continuing into the interruption.

Mid-interruption

When the interruption occurs, the interface should support user control of context switching and help the user maintain situational awareness of backgrounded tasks. This switch can take many forms. McFarlane (2002) conducted a theory-based experiment that compared four basic alternative solutions to the problem of how to coordinate human interruption in computer user interfaces. These four solutions are: (1) interrupt immediately and get it over with; (2) provide negotiation support so that the user controls the timing and exact context of switching between tasks; (3) provide intelligent mediation that brokers the onset of interruption tasks on behalf of the user; and (4) the use of scheduled interruption time cycles so that interruptions only occur during set times or contexts. Of these four solutions, negotiation was measurably the best approach for all kinds of user performance, except in cases where even small differences in the timeliness of handling interruption tasks are critical (either the current task is too important to allow distraction by the negotiation process, or the interrupting task is too important to wait for the negotiation to be completed).

Our approach involves the intelligent, automated selection of interruption strategy on a case-by-case basis. Our selection criteria is based on a dynamic automated assessment of the relative importance between the current task and the interrupting task. If the interrupting task is mission critical compared to the current task, the user is interrupted immediately. If the current task is critically important compared to the interrupting task, the alert is held until the user is finished with the current task (that is, it's scheduled for the next cognitive break). In all other cases, the interruption is negotiated.

To further aid the user in assessing relative task importance, we vary the default option in negotiation. That is, if the interrupting task appears to be slightly more important than the current task, the default option for the user is to accept the interruption. If the interrupting task is not deemed to be of higher importance, the default option for the user is to defer the interruption until the next cognitive break. Table 1 presents the full interruption strategy selection process for a three-valued priority system.

Interruption Task	Current Task		
	high	medium	low
high	Negotiated, default defer	Negotiated, default interrupt	Interrupt immediately
medium	Negotiated, default defer	Negotiated, default defer	Negotiated, default interrupt
low	Defer interruption until cognitive break	Negotiated, default defer	Negotiated, default defer

Table 1: Interruption Strategy Selection based on relative priority

Post-interruption

After the interruption is complete and the user transitions back to the original, interrupted task, the interface should provide recovery support to the user. That is, it should provide mechanisms to aid the user in recalling the context of the interrupted task, helping the user return more quickly to that previous task. Malin et al. (1991) state that user interfaces should be designed to reorient users to previously interrupted activities when they try to resume them. In their work, a simple log of relevant recent decisions is made easily available to the user for reference.

Our approach to context recovery involves providing the user commands that query the interface about aspects of the previous task. In a spoken dialogue system, this takes the form of meta-dialogue, with possible queries like “Where was I?” or “What was I last working on?” The user can also ask questions specific to the task, such as inquiring which supplies have been ordered so far in a requisition application.

Finally, the user can request a full progress review of the interrupted task. This provides a complete replay mechanism to the user, catching the user up to previous task context quickly and in detail. In a spoken dialogue system, this takes the form of requests for a summary of the task progress to-date.

Implementation

As a testbed for our approach to intelligent alerting and interruption management, we applied our techniques to a spoken dialogue interface. We have implemented a number of speech applications following the Listen, Communicate, Show (LCS) paradigm (Daniels, 2000). LCS systems integrate mixed-initiative spoken dialogue interaction with mobile intelligent agents to provide a natural, robust interface to information systems.

In most domains, users can use LCS to command agents to persistently monitor information sources for specific information events. When these events occur, the agents inform the LCS interface, which calls and alerts the user. If the user is currently

engaged in another task, this agent-initiated conversation can result in an interruption.

The spoken dialogue portion of an LCS system is built upon the Galaxy architecture developed at MIT (Seneff et al., 1999). Galaxy supports distributed, plug-and-play systems in which specialized servers are coordinated through a centralized communication hub. LCS systems contain servers specialized for speech and natural language processing, a dialogue manager to direct the system’s side of the conversation with the user, and an agent server for communicating and coordinating with the agent system.

Originally, when LCS monitor agents would notify the user, they would communicate to the dialogue manager directly through the agent server. The dialogue manager, which contained limited control mechanisms for interruption, would interject the interrupting alert at the next available moment in the dialogue. This would ensure that the user would not be interrupted mid-utterance, but does not take into account the effects of interruption on the user’s cognitive state.

To integrate our new interruption techniques, we added several new servers to the LCS architecture (see Figure 1 for illustration). The priority server ascertains the relative priorities of the current and interrupting tasks. The dialogue manager keeps the priority server informed of the task in which the user is engaged, while the agent server communicates the priority of incoming alerts.

The interruption server selects the interruption strategy most appropriate for the relative priority determined by the priority server. Once the interruption strategy is determined, the interruption server supervises as the system enacts the strategy. If the interruption is deferred, the interruption server tracks it to make sure that the alert is eventually delivered.

Because negotiated interruptions require interaction about the interruption (rather than about the interrupting task itself), we implemented a dialogue manager to drive this interaction in a domain-independent manner. The negotiation manager controls the system’s part of the negotiation process and

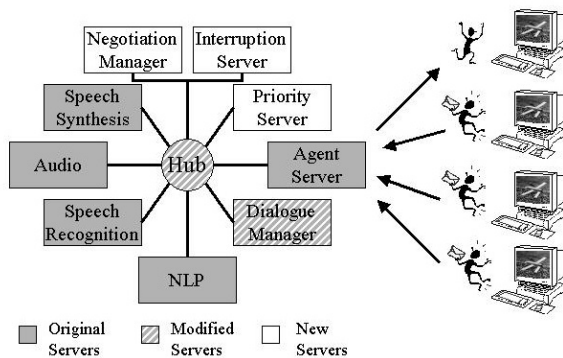


Figure 1: The ability to task multiple agents to perform persistent tasks (such as monitoring information systems) is a strength of LCS systems. However, since multiple agents may return results at the same time, advanced methods for handling interruptions were required. Some enhancements to the standard LCS architecture were required to implement these new interruption mechanisms.

coordinates with domain-specific dialogue managers to ensure that the system speaks to the user in a reasonable, focused manner.

In addition to constructing the new servers, we made several enhancements to the already existing LCS infrastructure. We implemented the meta-dialogue for post-interruption context recovery by adding logic for meta-dialogue control to the domain-specific dialogue managers. We also added an intuitive pre-interruption cue, programming the hub to have the system use voices for interruptions that are different from the voice used for the interrupted task.

Application

We applied the new interruption techniques to an LCS domain that supports Marines in managing requests for supplies using regular military radio protocols. This application was originally developed as part of the Small Unit Logistics (SUL) Advanced Concept Technology Demonstration (ACTD) program. The spoken language interface assists a Marine user in placing, modifying, deleting, or checking the status of a supply request. The SUL system also supports the creation of monitor agents to track requests and notify the user when either the status of the request changes or if the agent observes that the request hasn't been given attention over a set period of time.

In its original implementation, the SUL system would accommodate these returning notification activities by waiting until a break in the current conversation before providing any notification results to the Marine, regardless of the priority of either the no-

IMMEDIATE: *System:* Break! Break! MAGTF-5, this is BSSG1. Alert! Urgent Rapid Request 1738 has changed status to be canceled. over
NEGOTIATED - INTERRUPT: *System:* Break! Break! MAGTF-5, This is BSSG1. Alert about Immediate Rapid Request 1738...Accept now? over
NEGOTIATED - DEFER: *System:* Break! Break! MAGTF-5, This is BSSG1, Alert about Routine Rapid Request 1738...Defer now?
SCHEDULED: [No interaction until the user ends the current conversation. Then the system contacts the user.] *System:* MAGTF-5, MAGTF-5, this is BSSG1, over.

Figure 2: Example opening utterances for each interruption strategy. In all cases, a voice different from the one the user had been listening to was used for the alert. Note that when an interruption occurs, the user is explicitly informed of the new task's priority to support the user's decision to switch tasks.

tification activity or the current task. By allowing the SUL system to break into an ongoing conversation with important news, we can create a spoken dialogue interface that more realistically emulates radio protocols. However, this feature brings with it all the challenges associated with interruptions that have been discussed throughout this paper.

To support interruption strategy selection, we established a priority comparison scheme based on the priority field of each logistics request. For a SUL request, there are three priorities: routine, immediate, and urgent. We mapped these priorities to the low, medium, and high priority scheme described earlier in Table 1. We used the interruption strategy selection method described in this paper to govern delivery of agent alerts. Figure 2 shows examples of how interruptions would be presented to the user for each strategy.

To support post-interruption context review, we implemented two sets of meta-commands, relying on radio protocol to guide us. In the first case, the system repeats just its most recently stated utterance from the prior conversation. For the SUL domain, the proword (that is, a military procedure word) "Read back" is used. In the second case, the system reiterates all information it has been given about the current task. Figure 3 shows an example in which the user has requested more than just the prior system utterance. For this, the proword "Read back my request" is used.

In addition, we implemented dialogue that allows the Marine user to examine specific parts of a supply request by querying the interface specifically about

that part. For example, the user might ask, “Who is the point of contact for this request?” or “How many grenades did I order?” This provides the user complete control in returning to the context of the interrupted supply request. Similar dialogue supports the user in orienting quickly to interrupting alerts about other supply requests as well.

The SUL spoken dialogue system, with alerting enhancements, has been demonstrated successfully multiple times in operational settings. The enhanced LCS alerting infrastructure is being used as the basis for several more applications that will be field tested in the near term.

Future Work

We are working toward several enhancements of the current LCS interruption mechanisms. In each case, the enhancements build upon a core capability present in the current system.

Our current use of overall task priority to select interruption strategy assumes that a coarse-grained decision is sufficient. A more-informed decision would result from a finer-grained knowledge of where the user is in the current task. For example, in the SUL domain, the system is programmed with knowledge of the information that is necessary to fully complete a supply request. With its programmed knowledge of the request process, the system should be able to ascertain how close the user is to the beginning or end of completing the task, or if the user is in the middle of clarifying a particular step in the task process.

We are also investigating adding multiple modalities, where possible, as a method to further differentiate interrupting tasks from the current task, taking advantage of modality effects like those described in (Latorella, 1998). While our use of different voices helps, the use of another modality to cue interruption may prove more helpful to the user in mitigating cognitive disruption. This approach will be implemented and tested in the field.

In addition, we plan to add finer control of context review. Currently, our system provides for review of prior context in one of two forms: either the most recent system utterance or the entire set of known information items that the system has. While this is quite useful, with long, complex tasks, an intermediate level of detail might be preferred. We plan to construct and test dialogue methods for giving the user that finer control.

Conclusions

The disruptive effects of human interruption by information systems is a serious concern, particularly in high stress situations such as military operations. There are technical solutions available at all points during the interruption process to help mitigate this problem. By alleviating this cognitive disruption,

we can help the user move from task to task more quickly, resulting in potentially more efficient, less error-prone work by the user.

Acknowledgements

The authors would like to thank the other members of the Recovering Context After Interruption project, including Dan Davenport, Frank Davis, James Denny, Steve Knott, Dan Miksch, Mike Orr, Kathleen Stibler, Mike Thomas, and Ben van Durme. We would also like to thank Dylan Schmorrow, Program Manager of the DARPA Augmented Cognition program. This work was funded under ONR contract N00014-01-0455.

References

- Adams, M. J. and Pew, R. W. (1990). Situational awareness in the commercial aircraft cockpit: a cognitive perspective. In *IEEE/AIAA/NASA 9th Digital Avionics Systems Conference*.
- Adams, M. J., Tenney, Y. J., and Pew, R. W. (1995). Situation awareness and the cognitive management of complex systems. *Human Factors*, 37(1):85–104.
- Barnes, V. E. (1990). Cockpit distractions: Precursors to emergencies. In *Proceedings of the Human Factors Society 34th Annual Meeting*.
- Daniels, J. (2000). Integrating a spoken language system with agents for operational information access. In *Proceedings of Innovative Applications of Artificial Intelligence (IAAI-2000)*, Austin, TX.
- Detweiler, M. C., Hess, S. M., and Phelps, M. P. (1994). Interruptions and working memory. Technical report, The Pennsylvania State University, Department of Psychology. Unpublished.
- Foushee, H. C. and Helmreich, R. L. (1988). Group interaction and flight crew performance. In Wiener, E. L. and Nagel, D. C., editors, *Human Factors in Aviation*. Academic Press, Inc., San Diego, CA.
- Gillie, T. and Broadbent, D. E. (1989). What makes interruptions disruptive? a study of length, similarity, and complexity. *Psychological Research*, 50(4):243–250.
- Hess, S. M. and Detweiler, M. C. (1994). Training to reduce the disruptive effects of interruptions. In *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting*.
- Joslyn, S. (1995). *Individual Differences in Time Pressured Decision Making*. PhD thesis, University of Washington.

System: <domain voice> Delivery time is oh four hundred hours. Awaiting delivery location...

System: [alert voice] Break! Break! MAGTF-5, this is BSSG1. Alert about Immediate Rapid Request 1376. Accept now? over.

User: Affirmative BSSG1.

System: [alert voice] Request 1376 has changed status from New to be Cancelled.

User: What was it requesting?

System: [alert voice] The supply for Rapid Request 1376 is 5000 five point five six ball rounds and 400 gallons of water.

User: Roger BSSG1, anything else?

System: [alert voice] Negative MAGTF-5. BSSG1 out.

System: <domain voice> MAGTF-5, This is CSSOC. Back to your request. Delivery time is oh four hundred hours. Awaiting delivery location...

User: Read back my request.

System: <domain voice> Roger, MAGTF-5. I read back new rapid request. Precedence is routine. Supply is 100 dodic alpha five five five and 200 cases of MREs. Delivery time is oh four hundred hours. Awaiting delivery location...

Figure 3: Example of an interruption with recovery back to the prior conversation. The user queries the system about the details of the interrupting task's request for quick orientation to the new task. Returning to the interrupted task, the user can request a full summarization of the request in its current state.

- Joslyn, S. and Hunt, E. (1998). Evaluating individual differences in response to time-pressure situations. *Journal of Experimental Psychology: Applied*, 4(1):16-43.
- Latorella, K. A. (1998). Effects of modality on interrupted flightdeck performance: Implications for data link. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting*.
- Linde, C. and Goguen, J. (1987). Checklist interruptions and resumption: A linguistic study. Technical report, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA. NASA Contractor Report NASA-CR-177460.
- Liu, Y. and Wickens, C. D. (1988). Patterns of task interference when human functions as a controller or a monitor. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pages 864-867.
- Malin, J. T., Schreckenghost, D. L., Woods, D. D., Potter, S. S., Johannesen, L., Holloway, M., and Forbus, K. D. (1991). Making intelligent systems team players: Case studies and design issues, vol. 1, human-computer interaction design. Technical report, National Aeronautics and Space Administration, Washington, DC. NASA Technical Memorandum 104738.
- McFarlane, D. C. (2002). Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *HCI*, 27. to be published.
- McFarlane, D. C. and Latorella, K. A. (2002). The scope and importance of human interruption in HCI design. *HCI*, 27. to be published.
- McLeod, P. and Mierop, J. (1979). How to reduce manual response interference in the multiple task environment. *Ergonomics*, 22(4):469-475.
- Miyata, Y. and Norman, D. A. (1986). Psychological issues in support of multiple activities. In Norman, D. A. and Draper, S. W., editors, *User Centered System Design*, pages 265-284. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Osga, G. A. (2000). 21st century workstations: Active partners in accomplishing task goals. In *Proceedings of the Human Factors and Ergonomics Society 44th Annual Meeting*.
- Seneff, S., Lau, R., and Polifroni, J. (1999). Organization, communication, and control in the galaxy-ii conversational system. In *Proceedings for Eurospeech '98*, Budapest, Hungary.
- Storch, N. A. (1992). Does the user interface make interruptions disruptive? a study of interface style and form of interruption. Technical report, Lawrence Livermore National Laboratory. UCRL-JC-108993.
- Zijlstra, F. R. H. and Roe, R. A. (1999). Temporal factors in mental work: Effects of interrupted activities. *Journal of Occupational and Organizational Psychology*, 72:163-185.