Support for Intelligent Interruption and Augmented Context Recovery

Jody J. Daniels, Susan Harkness Regli, and Jerry L. Franke
Lockheed Martin Advanced Technology Laboratories
1 Federal Street
Camden, NJ 08003, USA
Telephone: (856) 338-4242
Fax: (856) 338-4122
Email: {jdaniels,sregli,jfranke}@atl.lmco.com

Abstract—**We describe work to support task alertness, mitigating the negative effects of interruption of one task by others competing for attention. These effects are ameliorated in two ways. First, by comparing the relative priorities of tasks from two domains, we decide whether to interrupt immediately, negotiate to interrupt completely now or later, or to schedule the interruption at the next cognitive break. Second, after interruption we support context recovery of the interrupted task, done to various depths to speed recovery from interruption. Priorities of tasks (and as a result the interruption and recovery strategies) will be adjusted according to the cognitive workload of subjects as measured through biological/physiological sensors and the accuracy and speed of task completion. One task domain involves monitoring and assessing ship mechanical system status while the second task domain involves tracking logistics requests on behalf of forward deployed ground troops. The interruption and context recovery system will be exercised on the US Naval ship Sea Shadow (IX 529).**

I. Introduction

As information systems become more complex and present an increasingly rich amount of information to users, interruptions present an ever larger hurdle to operational and cognitive efficiency. 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 context to the new alert task. After completing the alert task, users must switch contexts 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.

Many systems present alerts as soon as they are generated, without regard to situational context such as the time-sensitivity of the information or the user's current task or cognitive workload. An immediate interruption may not be the best solution in many situations.

In avoiding the pitfalls of immediate interruption, there are a variety of alternate strategies that may be employed instead. These strategies include Negotiated, Mediated, and Scheduled [1]. The Negotiation strategy allows the user to work with the interrupter to determine the timing of the interruption: either right now or at some future mutually agreed to time. The Mediated strategy uses an agent (either human or automated) to determine when the interruption should take place (e.g., a secretary fielding telephone calls). The Scheduled strategy allows the user to define specific moments when she will accept interruptions, such as at every half hour or every two minutes.

We are examining how to automate the reasoning that determines which strategy to use for each alert in a way that optimizes human cognitive efficiency in interruption. Some of the means of doing this are to review the relative priorities of each task, to monitor the state of completion of the current task (just beginning, almost done, etc.), and to assess the cognitive state of the user.

In addition to facilitating an effective transfer to an alert task context, we are also examining methods to optimize the transfer back to the interrupted task context. After the user has completed the task associated with an interruption, it would be helpful to assist the user in regaining the context and prior state of the interrupted task. We are investigating means of providing users with context review so that they may rapidly reacquire context and move forward with the interrupted task.

II. Approach

The objective of our alerting work is to cancel the negative cognitive effects of interruption and allow users to exploit the benefits of greater information

0-7803-7450-9/$17.00©2002 IEEE

volume for making better decisions. Alerting mechanisms have been 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 tutoring, and shopping agents. Applications that rely on alerting mechanisms 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.

User interface technologies intended to support increased user-system interactions can help mitigate the cognitive disruption caused by interruptions by ensuring that they are timed and executed as best possible and by facilitating transfer between tasks. Our approach is to automatically select an interruption strategy suited to the user's context and to provide context recovery support.

A. Interruption Support

When an 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 [2] conducted a theory-based experiment that compared four basic alternative strategies to the problem of how to coordinate human interruption in computer user interfaces. These four strategies 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 are based on a dynamic, context-sensitive 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 I presents the full interruption strategy selection process for a two-valued priority system.

B. Context Recovery Support

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. [3] 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 metadialogue, with possible queries like "Where was I?" or "What was I last working on?" The user can ask questions specific to the task, such as inquiring which supplies have been ordered so far in a requisition application.

The user can also 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.

III. IMPLEMENTATIONS

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 [4]. LCS systems integrate mixed-initiative spoken dialogue interaction with mobile intelligent agents to provide a natural, robust interface to information systems.

Using the LCS interface, users can request notification of particular events. When the event occurs, an agent returns to the system to present the results. If the user is currently engaged in another task, this agent-

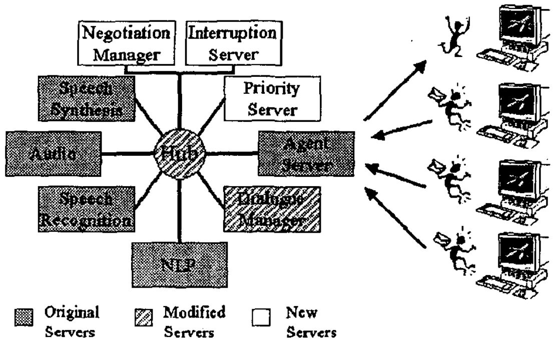

Fig. 1. The ability to task multiple agents to perform persistent tasks (such as monitoring information systems) is a strength of LCS systems. However, since agents may return results at unknown times, advanced methods for handling alerts and interruptions are required. Some enhancements to the standard LCS architecture were required to implement these new alerting mechanisms.

initiated alert results in an interruption. Using our interruption management capabilities, the system examines the priority of the current task, the priority of the alert, and in future work, an assessment of the user's cognitive state. After comparing the priorities, an interruption strategy is selected and executed.

We have prototyped and are refining three interruption strategies: Negotiated, Scheduled, and Immediate. Each of these interacts with the user in a different fashion. Negotiated allows the user to either accept or defer the alert. If the user opts to defer, then the alert is presented again at the next break in conversation. Scheduled alerts take place at a conversation break, which is considered to be equivalent to a cognitive break. Immediate alerts present more information to the user and do not offer the opportunity to defer. When either an immediate alert occurs or the user accepts an alert as a result of the negotiation strategy, the system facilitates the user querying for additional information to explain the alert. For example, it may not be sufficient to inform the user that a request has been cancelled without conveying additional information about the specific content of the request.

A. System Enhancements

The spoken dialogue portion of an LCS system is built upon the Galaxy architecture developed at MIT [5]. 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 alert the user, they would communicate to the dialogue manager directly through the agent server. The dialogue manager, which contained limited control mechanisms for alerting, would interject the alert at the next available moment in the dialogue. This would ensure that the user would not be interrupted midutterance, but does not take into account the effects of interruption on the user's cognitive state.

To integrate our new alert management 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 alerting 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 with the user about an possible interruption (rather than about the alert 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 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 to support context recovery. We implemented the meta-dialogue for post-alert context recovery by adding logic to control the domain-specific dialogue managers.

IV. APPLICATION IN AN OPERATIONAL SCENARIO

Reduced manning often results in warfighters working multiple unrelated tasks. In addition, the use of automated intelligent assistants to ease their burden increases the number of interruptions that they must deal with. The combination of the two means that warfighters are being hit with many interruptions from unrelated sources, which can cause them to lose track of important details.

We present an approach that combines dynamic interruption coordination support with context recovery mechanisms to aid the user in navigating between interruptions. The following example illustrates how we restore the effectiveness of warfighters by managing

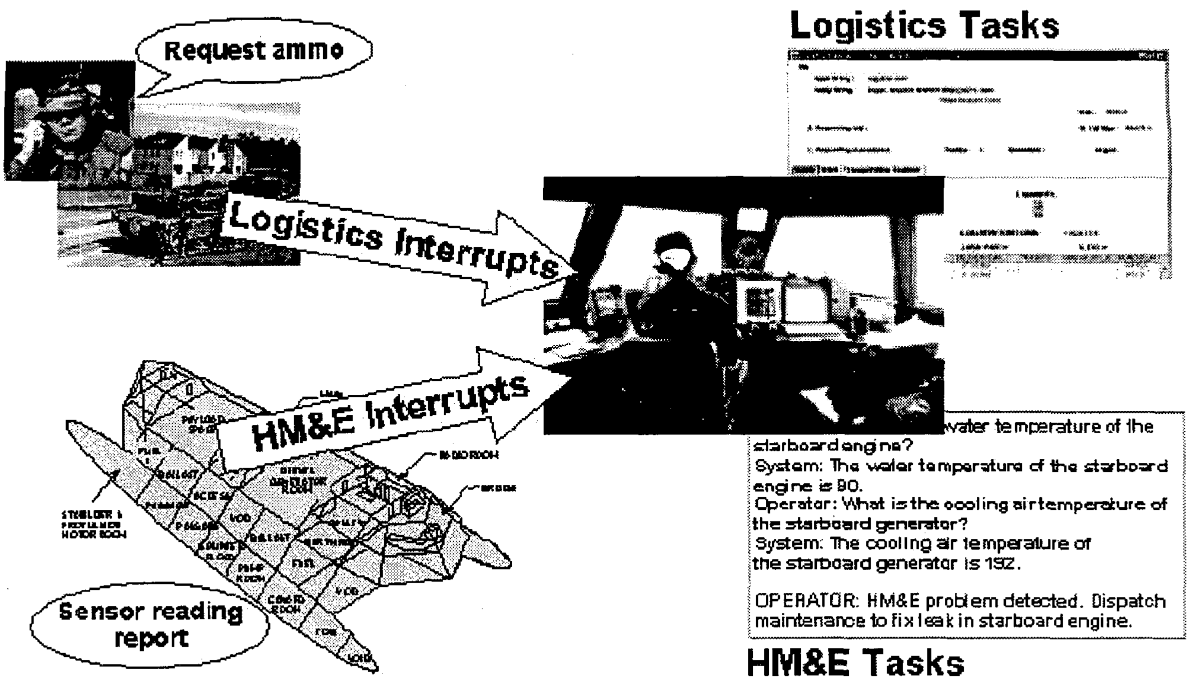

Fig. 2. Alerts from both the Logistics and HM&E domains affect a user attempting to complete tasks in each of these domains.

interruptions and providing assistance in recovering the context of what they were working on before interruption. Our mechanisms are implemented within two spoken dialogue interface systems, one for monitoring and assessing ship mechanical system status
and the second for tracking logistics requests on behalf of forward deployed ground troops through a radio-based interface.

The first domain supports Navy personnel in monitoring Hull, Mechanical, and Electrical (HM&E) sensor data using the Shipboard Ubiquitous Speech Interface Environment (SUSIE). The SUSIE system supports shipboard status monitoring, critical-event alerting, and notification requests for status change events. In our reduced manning environment a warfighter is given two unrelated tasks that require constant attention. For this task the warfighter is responsible for monitoring the status of ship systems to ensure proper operation.

The second domain 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 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 alert the Marine 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 this second task, the warfighter is responsible for placing logistics requests for ground units, some of which are actively involved in combat.

In our scenario the warfighter is interrupted multiple times while trying to keep up with both assigned tasks. In our scenario, while monitoring the ship's systems, several sensor readings creep dangerously outside of acceptable ranges. The warfighter has to watch the ship systems' behavior over time, however, to figure out what the problem is and how it can be fixed. This requires maintaining awareness of all the status alerts that have occurred in the last few minutes. At the same time, however, she receives logistics requests from ground units at multiple locations, some of whom are engaged in active combat. Figure 2 depicts the operational environment in which this user is working.

In their original implementations, the HM&E and logistics systems would accommodate returning alert and notification activities by waiting until a break in the current conversation before providing any alerts or notification results to the user, regardless of the priority of either the alert activity or the current task.
By allowing the systems to break into an ongoing conversation with important news, we can create a spoken dialogue interface that more realistically

Table I. Interruption strategy selection based on relative priorities

Interruption Task	Current Task	
	High Priority	Low Priority
High Priority	Negotiate, default accept	Immediate
Low Priority	Schedule – next cognitive break	Negotiate, default defer

emulates radio protocols. However, this feature brings with it all the challenges associated with interruptions that have been discussed throughout this paper.

Providing interruption support enhances the warfighter's performance in two ways. First, by estimating the relative priorities of what she is currently doing and that of the alert task, the amount of time she has to spend doing the initial priority assessment is reduced dramatically. Second, alerts of high priority are presented differently than those of lower priority alerts. Therefore, she can make quicker and more accurate judgments about whether or not she can safely defer an alert to a more convenient time to avoid losing track of what she's doing.

To support interruption strategy selection, we established a priority comparison scheme based on the relative importance of the operator's current task and the interrupting alert. The comparison scheme is shown in Table I. The designation of high and low priorities is determined by the specific incidents that occur in the two LCS domains. In the HM&E monitoring domain, high-priority tasks involve sensor alerts that indicate imminent system failure; low-priority tasks involve sensor alerts indicate minor equipment problems. In the logistics domain, high-priority tasks are those coming from ground units involved in active combat; low-priority tasks come from non- combat ground units.

We used the interruption strategy selection method described in this paper to govern delivery of agent alerts. Figure 3 shows examples of how interruptions would be presented to the user for each strategy.

By implementing context recovery support we enhance the warfighter's performance by allowing her to review where she was prior to the alert. In addition, she can control how much detail is reviewed. She is able to request increasing amounts of information, starting with the most recent piece of the task she had accomplished. The belief is that this protocol will only need to provide as much context as is necessary to bring her back up to speed on the interrupted task and will reduce the total amount of time spent recovering from the disruption.

To support context recovery, 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

IMMEDIATE: *System:* `Break! Break! MAGTF-5,this is BSSG1. Alert! Urgent Rapid Request 1738 has changed status to be canceled. over`

NEGOTIATED - ACCEPT: *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 with the user until a break in the conversation, then the system contacts the user.] *System:*`MAGTF-5, MAGTF-5, this is BSSG1, over.`

Fig. 3. Example initial utterances for each interruption strategy. In all cases, a voice different from the one the user had been talking to is used for the alert. Note that when an alert occurs, the user is explicitly informed of the new task's priority to support the user's decision to switch tasks.

conversation. For the logistics domain, the pro-phrase (that is, a military procedure phrase) "Read back" is used. In the second case, the system reiterates all information it has been given about the current task. Figure 4 shows an example in which the user has requested more than just the prior system utterance. For this, the pro-phrase "Read back my request" is used.

In addition, we implemented dialogue that allows the Marine user to verbally examine specific parts of an interrupted 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 and the HM&E task as well.

The logistics 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.

V. EXPERIMENTS

During the summer of 2002 we will collect baseline measurements of users performing logistics

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?`
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? over.`
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.`

Fig. 4. Example of an alert with negotiation and recovery back to the prior conversation. The user can query the system about details of the request associated with the alert before returning to the interrupted task. The user can request additional context, which in this case is a full summarization of the data input thus far into placing a new request.

and HM&E monitoring tasks without the assistance of any alert management support. All alerts will be given in an immediate mode. A single operator will be responsible for monitoring ship status while simulated critical incidents occur; simultaneously that operator will need to place logistics requests for forward deployed ground troops. Both the shipboard critical incidents and the logistics requests will have varying priority levels dictated by the operational scenario.

Performance metrics we will collect include time for completion of each task, accuracy of task solutions, and error rate (i.e., the ratio of accuracy over time, to account for potential tradeoffs between working quickly and working carefully). We will be taking initial biological and physiological measurements of users to record data

on stress and fatigue as indicators of cognitive load. Sensors employed to collect this data will include a combination of heart rate, breathing rate, galvanic skin response, and electrical activity (EMG, EOG, EEG).

In the winter of 2002 we will test our interruption management and context recovery mechanisms within in the same set of baselined tasks. In addition to the performance and biometric measurements collected in the baseline study, we will analyze work patterns of users to understand the strategies they adopt to deal with interruptions using the interruption management facilities of the system. We will administer a subjective questionnaire to gage subjects' perceived degree of workload, stress, success in completing tasks, degree of interruption, ability to focus on tasks, ability to resume work after interruption, and system effect on successful mission completion.

VI. 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 an 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 logistics domain, the system has full knowledge of the information that is necessary to fully complete a supply request (or modify, delete, etc.). With this 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 within the task process. This is information that should be added into the interruption strategy selection and timing decision.

We plan to add finer control of context recovery. Currently, our system provides meta-commands that enable 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 (or multiple levels) of detail might be preferred. We plan to construct and test methods for giving the user that finer control.

Our system architecture includes the ability to tap into information gathered by biological/physiological sensors to better assist the user. We will be analyzing a set of sensors to determine whether enough information can be gleaned from the physio-cognitive states/changes in a user to proactively alter the behavior of the interruption facilities. We will also collect performance data during our experiments, which we will examine for correlations with the biological/physiological data.

VII. CONCLUSIONS

It is imperative that we develop systems that can better facilitate their interactions with a user. We are taking one step forward by adding intelligence to the alert process. By ascertaining the type and timing of an alert and by helping a user regain their context within an interrupted task, we hope to reduce the cognitive disruptiveness of interruptions. We believe that both automated strategy selection and context recovery support will decrease stress level, resulting in an increase in performance and a reduction in cognitive load.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank the other members of the RCAI (Recovering Context After Interruption) project, including Dan Davenport, Steve Knott, Dan McFarlane, 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

[1] D. C. McFarlane. Coordinating the interruption of people in humancomputer interaction. In *Human-Computer Interaction: INTERACT'99*, pages 295–303, 1999.

[2] D. C. McFarlane. Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *HCI*, 27, 2002. to be published.

[3] J. T. Malin, D. L. Schreckenghost, D. D. Woods, S. S. Potter, L. Johannesen, M. Holloway, and K. D. Forbus. 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, 1991. NASA Technical Memorandum 104738.

[4] Jody Daniels. Integrating a spoken language system with agents for operational information access. In *Proceedings of Innovative Applications of Artificial Intelligence (IAAI-2000)*, Austin, TX, August 2000.

[5] Stephanie Seneff, R Lau, and J Polifroni. Organization, communication, and control in the galaxy-ii conversational system. In *Proceedings for Eurospeech '98*, Budapest, Hungary, 1999.