Specifying Organizational Policies and Individual Preferences for Human-Software Interaction

Debra Schreckenghost, Cheryl Martin, and Carroll Thronesbery

NASA Johnson Space Center
TRACLabs
1012 Hercules
Houston, TX 77058
ghost@ieee.org

Abstract

Current manned space operations employ policies and protocols to specify an etiquette for human-human interaction. This space operations etiquette presumes that operations are conducted synchronously from a few centralized locations - a control center or space vehicle. The availability of automated control software, portable computing platforms, and wireless communication makes it possible for distributed groups of humans and software agents to conduct some space operations remotely and asynchronously. The space operations etiquette must be adapted to address the challenges that arise when changing to teams of distributed agents. We are investigating intelligent support agents for crew and ground personnel that implement an etiquette for notification and alerting, which is an essential part of the space operations etiquette. Our approach differs from previous work on notification and alerting because it focuses on meeting the *needs of an organization* rather than the *needs of an individual*. In this paper we describe our etiquette for remote space operations and our approach for implementing context sensitive notification.

Introduction

Manned space operations today are a team effort among groups of trained personnel. These groups are organized around system or functional boundaries (e.g., crew life support systems). The complexity of coordinating this large team effort is managed using a hierarchical organization with a single line of command between levels, and a succinct vocabulary for communicating between levels. Roles determine who is authorized to manage a system or function. Protocols for each role define what information about a system should be monitored, and when system changes should be communicated to the level above. These protocols also define what commands can be passed down the hierarchy.

Taken as a whole, these protocols define an etiquette for conducting manned space operations. This etiquette establishes clear expectations about the behaviors of team members based on their role. It uses a sparse, well-defined commanding language, which improves the efficiency and clarity of communication and makes behaviors predictable and reliable.

This space operations etiquette, however, presumes that operations are conducted synchronously from a few centralized locations - a control center or within a space vehicle. The availability of automated control software, portable computing platforms, and wireless communication makes it possible for distributed groups of humans and software agents to conduct some space operations remotely and asynchronously. Remote ground support can enable flight controllers to be away from the Mission Control Center and still be notified when important situations arise. Remote crew interaction permits crew to maintain awareness of automated control operations from anywhere in the vehicle with minimal impact to the performance of other crew activities.

The space operations etiquette must be adapted to address the challenges that arise when changing to teams of distributed agents. These challenges in defining this distributed etiquette include the following:

- *Ensure that team members are notified of significant operational events.*
  This includes both events that orient team members about ongoing operations and events that indicate a need to take action. Such event notification is necessary because the increased use of control automation for routine operations results in humans performing tasks other than control most of the time. Providing for remote notification of operational events requires defining organizational policies to guarantee that the right information gets to the right people in a timely manner. These organizational policies are managed by the organization to ensure consistent treatment of all members holding the same role within that organization. Organizational policies for event notification should specify the conditions determining *who* should be informed of an event. These conditions should consider assigned roles as

well as the type of event. Organizational policies for event notification should also describe *how* to inform interested personnel of events. This description should include information about information saliency and user location that can be used to determine the appropriate notification modalities (e.g., pager, display).

- *Enforce consistent, reliable remote commanding.* Organizational policies for coordinating distributed, remote commanding implement a hierarchical command structure by specifying the allocation and authentication of control authority according to assigned roles and situational needs. Adverse interaction among concurrent commands from different agents should be prevented. The representation of standard operating procedures for crew and automated agents should comply with these policies. For example, these procedures should include reconfiguration of the automated control system to temporarily suspend automated tasks that conflict with manual tasks. Tracking the remote execution commands, including command completion, is a necessary capability to encode these policies.

Etiquette for Remote Space Operations

We are investigating the use of intelligent personalized agents to help implement an etiquette for remote, distributed space operations ([a]Schreckenghost, et al., 2002). We provide a proxy agent that supports this etiquette for each team member by providing services that encode organization policies for event notification and information management, as well as remote commanding. We base our approach for implementing this new etiquette on our experience in fielding automated control software for crew life support systems at the Johnson Space Center ([b]Schreckenghost, et al., 2002).

From one perspective, the proxies implement an etiquette that has not been previously encoded and applied for human interaction with automated control systems. This etiquette resembles the interaction between a busy executive and her personal assistant. The assistant collects information and manages interruptions so the executive can work more effectively while maintaining the ability to respond when special skills or authority are required. The proxy agent encodes an etiquette for event notification that filters and annotates events received from other agents, such as automated control agents or the crew (via their proxy agents). This event processing ensures that the proper team members are notified of important changes in operational situation and that the means by which team members are notified are well-suited to their location, activities, and assigned roles. It is possible for proxy users to customize event processing according to their individual preferences, but these preferences are not permitted to override or compromise organizational policies. For example, a user can ask to be informed of additional events not required to support their assigned roles, but can not prevent receiving events required by these roles.

The proxy agent also encodes an etiquette for remote commanding. Before the user can issue a command to the automated control agent, his proxy determines if he is authorized to issue the command, based on his assigned role. If so, the proxy agents manage the allocation of control authority to avoid conflicts with other agents who are commanding. When commands issued by humans potentially conflict or interfere with automated control actions, a proxy can reconfigure the automated control system from a fully autonomous mode to a partially autonomous mode where conflicting automated tasks are temporarily suspended. When human commanding is complete, it can return the automated system to a fully autonomous mode. Because commanding is mediated through the proxy, the activities of team members can be tracked and coordinated.

From a second perspective, the proxies change the existing operations etiquette among humans by enabling remote, supervisory operations for a team of distributed personnel. For some tasks, the proxy can *represent its user* in team interaction by performing a task in place of its user. Such tasks include providing current information about its user's activities and location to other team members. For other tasks, the proxy can *assist its user* in performing team interactions to ensure these interactions comply with the operational etiquette (e.g., interaction with automated control agent). The proxy also assists its user in making timely responses by reminding her of pending activities and deadlines as well as notifying her of changes in role or schedule when problems arise (e.g., a person with special skills is called in). Finally, the proxy can *critique its user* by implementing safeguards that detect and alert the user of unsafe control conditions resulting from crew error.

Although etiquette for human-to-human interaction is often understood implicitly, etiquette must be explicitly established and encoded for processing by computers in human-computer interaction. In the remainder of this paper we describe our approach for establishing the rules of etiquette for event notification. The design of protocols for remote commanding is in progress and thus is not described in detail.

Related Work

The work presented in this paper differs from previous work on notification and alerting primarily because it focuses on meeting the notification needs of an organization (perhaps overlaid with personal preferences) rather than on meeting the notification needs of an individual.

Organizational notification needs arise from requirements to ensure that a given individual has the information awareness he or she needs to perform his or her function in the organization. Organizational notification needs can be defined independently of a particular individual's preferences and are related instead to the current organizational context of any individual. This organizational context can be as simple as a role assignment or as complex as a description of an individual's current activity in support of organizational goals. Organizational notification needs relate only to the domain of the organization's function and do not change as different individuals take on a particular context, for example, a role.

In contrast, individual notification needs must be defined with respect to a specific individual and may span various domains or contexts that the individual participates in, including but not limited to his or her organizational functions. Individual notification needs arise from the information awareness requirements of a specific person to be effective in his or her day-to-day activities. Examples of research focusing on individual notification needs include the *comMotion* system, which exhibits capabilities including reminding a person of his or her grocery list when he or she is near a grocery store (Marmasse, 2000); email filtering and filing/viewing systems, which can be highly personalized and span many subject-matter domains (Horvitz et al., 1999; Schmandt et al., 2000; Venolia et al., 2002; Whittaker and Sidner, 1996); and document alerting services, which notify registered users when publications of interest become available (Hinze and Faensen, 1999). Although the subject-matter domain and the content of notifications studied in many previous projects may have organizational relevance (for example, a document alerting service may help a research scientist perform his or her job more effectively), the vast majority of previous work focuses on defining and meeting the notification needs of a particular individual rather than defining and meeting the notification needs related to a particular function or job in an organization.

However, many of the same principles that have previously been examined to assist individual users can be applied at the organizational level. Because notifications are ultimately being delivered to individuals, regardless of how the need for notification arises, previous work concerning the aesthetics and mechanics of user interfaces and notice delivery is very relevant. Several previously implemented notification systems exhibit the capability to deliver notices using multiple modalities (display, pager, email, or voicemail, etc.) (Horvitz et al., 1999; Schmandt et al., 2000). In particular, the Active Messenger system described by (Schmandt et al., 2000) monitors a user's incoming email and forwards messages, based on their priority, to the most appropriate available communication channel (pagers, fax machines, or phones). After sending the message to this channel, it monitors for user reactions. If the system determines that the user has not seen the message within a given time (configurable), it sends the message to the next appropriate channel that is available until the user sees the message or the possible channels have been exhausted. The capability to manage multiple modalities is important to ensure delivery of important notifications in organizations when people are distributed or working remotely and possibly asynchronously.

Another critical area of development in related research is the design and implementation of user interfaces that promote background awareness (Cadiz et al., 2001; MacIntyre et al., 2001). These systems take advantage of a human's peripheral perception and reduce the need for interruption when an important (but not critical) notice should be delivered. The Kimura project described by (MacIntyre et al., 2001) allows a person to manage multiple activities by bringing one into focus on a high-resolution desktop environment and allowing others to be collapsed and represented as montages projected on a wall-display. Notices of events such as the completion of a printing job pertaining to a background activity can be displayed on the wall-projected montage representing the activity. The Sideshow project described by (Cadiz et al., 2001) also provides peripheral awareness using a small strip of screen area on the edge of the primary display. The development of user interfaces that maximize the effectiveness of peripheral information is critical to reducing the negative impact that notice presentation can have as a result of interruption. Previous studies have documented the negative impacts of interruption (Cutrell et al., 2001; Czerwinski et al., 2000). Remaining cognizant of this impact, our future work will attempt to minimize both the number of interruptions, using peripheral awareness interfaces to deliver notices, as well as help users manage interruptions that cannot be avoided by providing tools for task and context switching.

Our work also pursues goals similar to previous research seeking to build capabilities to filter and direct messages or notices in the most appropriate manner (Horvitz et al., 1999; Schmandt et al., 2000). However, we require more flexibility and control in specifying notice filters and routing than exhibited by many previously developed systems. This requirement for flexibility and control stems from the need to encode existing or emerging organizational etiquettes for notification. The norms and expectations that make up an organizational etiquette for notification are often implicit, but any deviations from this implicit understanding would become readily apparent in an automated notification system. Therefore, the encoding mechanism for organizational notification etiquette must allow a great deal of flexibility as well as a fine degree of control over the handling of individual notices when needed.

Most previous systems use a two-stage approach for notice categorization and then routing. First an incoming notice, event, or message is categorized or assigned a numerical priority. Then, based on this result, the notice is routed to the user. As mentioned above, many systems can route notices through multiple modalities in this second stage. The first-stage categorization and prioritization processes developed by some previous research are quite sophisticated. The Clues system described by (Schmandt et al., 2000) uses dynamic filtering rules based on information in a person's calendar, to-do list, or mail logs to prioritize messages based on "timeliness," or relevance to recent events, tasks, or messages. The PRIORITIES email system described by (Horvitz et al., 1999) learns message classifiers from examples drawn from a user's email and applies the classifiers in real time to assign expected criticalities to incoming email messages. This system then incorporates an element of dynamic interruption management by calculating the cost of alerting the user of the email message based on its criticality versus the cost of delaying the alert. Although these systems are very powerful, they may not allow the degree of control over if, when, and how notices get delivered that a particular organization may desire for encoding their notification etiquette.

The following sections describe how we have developed automated support that encodes an etiquette addressing the notification needs of an organization.

Context Sensitive Notification

Our approach to encoding the rules of etiquette for organizationally relevant notifications allows a set of *notice specifications* to be defined for use during a given *organizational context*. A set of notice specifications describes which notices should be presented to the user and gives guidance to the user interface software about when and how a notice should be presented to the user. Multiple sets of notice specifications can be defined, each for a given organizational context. We currently use organizational role assignments as the context differentiator for choosing the set of notice specifications to apply for any incoming notice. Organizations often rely on role assignments to coordinate activities among distributed agents (Singh, 1991; So and Durfee, 1998). As a user's roles in an organization changes, the set of notice specifications currently in effect for that user is re-evaluated and updated to reflect the role changes.

Currently, the sets of notice specifications we use are all statically pre-defined, each for a given organizational context. These specifications may be redefined by the organization as it changes over time (e.g., creates new organizational contexts). Knowledge of the organization's existing notification etiquette is used to define these notice specifications. Examples of this knowledge include policies allocating authority/responsibility and standard operating procedures. If no explicit knowledge is available, the notice specifications can be built incrementally using exemplar notices and experience with how those notices should be filtered and routed. Previous work has noted that static rules are difficult to manage if the burden to maintain which static rules should be in effect for a given time is placed on the user (Schmandt et al., 2000). Our work removes this burden from the user by monitoring the user's role changes and automatically updating which notice specifications are in effect to reflect the new operations context. Future work will incorporate more complex representations of organizational context, including current activity descriptions. A user's current activity will be monitored along with his or her role, which will make notice specifications even more context sensitive at no additional burden to the user.

Notice specifications from several pre-defined contexts may be combined into a larger set to form the overall set of notice specifications currently applicable for a given user. There are two major reasons why this may occur. First, a user may play more than one role concurrently in an organization. Second, the user may wish to overlay personal notification preferences on the organizational notification requirements to increase his or her efficiency or operational awareness. Any user-defined preferences must also be defined to apply for some specified organizational context.

Although our primary goal concerns encoding the organizational requirements for notification, we recognize that individuals who fill organizational roles will have different preferences with respect to notice frequency and information bandwidth. We require the organization to define the minimal requirements for notification based on organizational policy. We also allow users to specify notification preferences that do not compromise any of the organizationally specified requirements. To ensure that organizational requirements are not compromised, we do not allow notice specifications from the organization to be overridden. Our current overall approach allows only those notices for which notice specifications exist to be passed to the user. The user-defined notice specifications are added to those defined by the organization. Therefore the user can only add to the set of notices being passed or add supplementary directives for how notices should be routed. The user interface software combines duplicate directives for the same notice if they exist. Therefore, the user always receives the information he or she needs as required by the organization but may choose to receive more information in additional ways as desired.

Notice Specifications

The set of *notice specifications* in effect for a user at a given time informs automated notification software about the user's notification requirements and preferences. Notice specifications allow an organization or an individual user to identify *what* a user should be notified about and constrain *when* and *how* the notification should occur.

In general, notice specifications associate a particular notice or set of notices with a notification directive indicating when and how to inform the user about that notice or set of notices. Therefore, each notice specification has this form:

Notice Filter Conditions → Notification Directive.

The *notice filter conditions* in the antecedent of this statement identify a set of conditions to match against an incoming notice. A match implies that the notification directive in the consequent of the statement should be applied to this notice.

The form of the *notification directives* in the consequent of the specification statement allows a great degree of flexibility in identifying when and how a notice should be presented to the user. There are several dimensions of presentation that an organization or a user may wish to specify for a given notice:

- **Pass/Prevent:** Some notices need to be brought to a user's attention, others should be filtered out to avoid overloading or distracting a user.
- **Notification Saliency:** Some notices should be brought to the attention of the user sooner than others or with more emphasis. The notice specification should be able to associate this type of saliency information with a notice.
- **Modality:** Certain modalities (e.g., pager, email, display) may be required or preferred. More urgent notices may be presented through more aggressive channels.
- **Modality Conditions:** A required or preferred modality may depend on the user's current state in addition to the content of the notice. Because users change state (for example between being online or offline and nearby or remote) the notice specification should be able to identify desired modalities based on user state information.

A notice specification associates a directive in one or more of these categories with a particular notice filter condition. This allows a great deal of control over how an incoming notice is handled. A notice filter condition can be as detailed as necessary and can range from matching only a single possible notice to matching all possible notices. As discussed in the Related Work section, this approach (as opposed to a categorization approach for incoming notices) gives us added control and flexibility to encode very complex notification etiquettes that may exist in an organization.

Notice Filter Conditions

Our approach for *notice filter conditions* is based on content filtering. We represent notice filter conditions as a set of triplets identifying conditions to match against incoming notices. Each triplet indicates (1) a property of a notice, (2) a matching condition, and (3) the value to match against. An example of a notice filter condition is shown below:

<(event-category) (exact-string-match) (life-support)>

We have implemented several forms of matching conditions including string comparisons, ordinal comparisons, integer comparisons, and ontological comparisons that allow us to consider hierarchies of abstraction. Consider the abstraction hierarchy of categories for life support notifications pictured in Figure 1 and the following two possibilities for notice filter conditions:

<(notification-category) (ontology-subclass-or-equal) (life-support-notification)>
<(notification-category) (ontology-subclass-or-equal) (warning)>

The first notice filter condition would match any incoming notice labeled with any notification category classification pictured in Figure 1. The second notice filter condition would match any incoming notice labeled with the notification category of "warning," "alarm," "alert," or "notice". It would fail to match notices labeled with the notification category of "activity tracking," for example.

Notices with relevance to an organization often contain jargon or organization-specific terms with pre-existing relationships (such as those shown in the life-support notification category ontology in Figure 1). Being able to define filters that incorporate this specialized language is important for encoding an organizational etiquette for notification. Our approach for specifying notice filter conditions is extensible for any domain terminology. New matching conditions can be easily encoded, and the triplets representing the notice filter conditions are written using XML format. Although each notice filter condition given as an example in this section contains only one triplet, our representation and matching algorithms support conjunctions of triplets to specify more complex matches. Disjunctions are currently handled by adding additional notice specifications to a given set.

Notification Directives

The *notification directive* part of the notice specification allows an organization or user to guide how and when a notice is presented to the user. If the notice filter condition is met, an empty notification directive implies that the notice is simply passed to the user via user interface software. The *notification directive* can optionally specify a saliency, a modality, or both, as an output of processing notice specifications. Notification directives can also consider the user's state (for example, location - both physical and "cyber") in selecting a required or preferred modality. The upcoming paragraphs describe the overall processing of notification specifications, which indicates how notification directives are currently used in our system. Following this description, the details of representing a notification directive are given.

Figure 2 shows the inputs and outputs for processing an incoming notice based on the set of notice specifications for that user as well as the user's current state. Currently only location is used as state information as shown in Figure 2. An incoming notice is passed or prevented based on the notice specifications currently in effect for that user. If no notice filter condition matches the incoming notice then the notice will be "prevented" and not passed to the user. If one or more notice filter conditions matches the notice, it will be passed to the user's interface management software and annotated with saliency and modality information arising from the notification directive. Since multiple notice specifications may apply to a given incoming notice, the sets of directives from all matching specifications are collected and delivered to the user interface software together.

The user interface software uses these annotations to determine how to actually present the notice to the user. Modalities are represented as a member of a predefined set. Currently we use the following set of modalities: pager, display, email, and archive. Saliency information is passed to the user's interface management software as two pieces of information (1) latency and (2) focus-of-attention. *Latency* indicates how soon the user's attention should be drawn to the notice. *Focus-of-attention* indicates how forcefully the user's attention should be drawn. Currently, we use three levels to represent each of these saliency dimensions, as shown in Figure 3. Examples of how the user interface management software can use these saliency annotations include determining the degree of emphasis when displaying a notice (interrupting or peripheral) or assigning the urgency codes to pager messages.

Our approach to notice specification allows the content of an incoming notice to determine its saliency. Modality is determined by both the content of the notice and the current state of the user (for example, his or her location). Therefore, the notification directive expands as shown here:

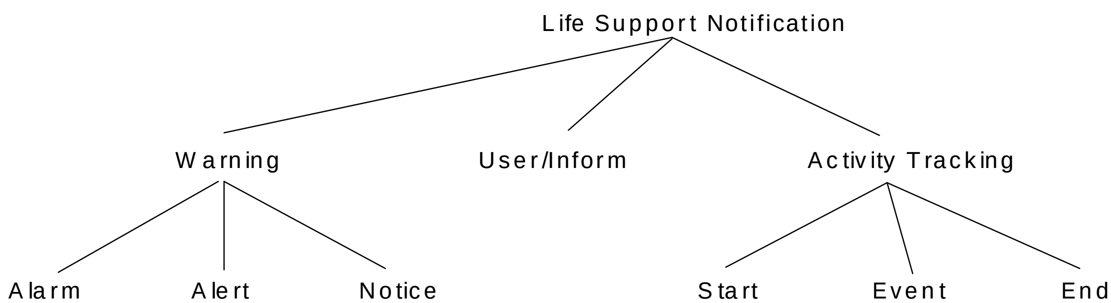

Figure 1. Notification Category Ontology

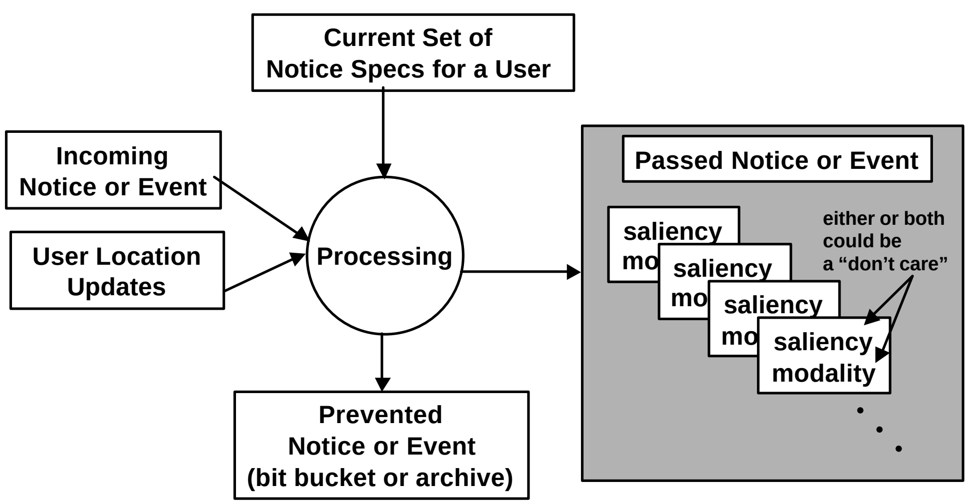

Figure 2. Processing for Notice Specifications.

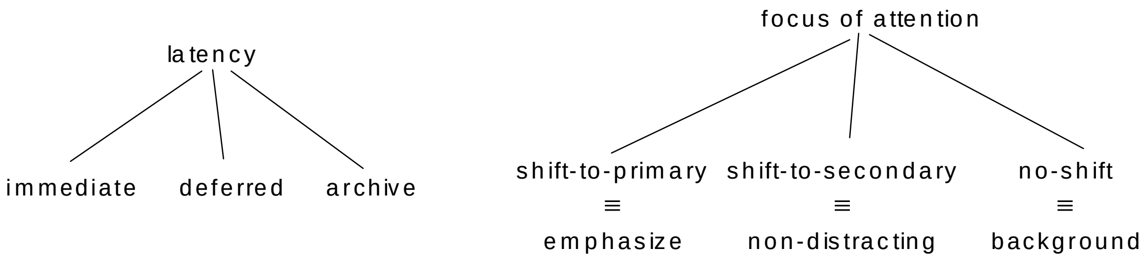

Figure 3. Degrees of Latency and Focus-of-Attention.

Notice Filter Conditions →
(saliency AND modality-condition).

Both or either saliency and the modality-condition may be left unspecified as a don't-care condition. The user interface management software makes decisions about the most appropriate saliency and/or modality if the notice is passed without one or both of these annotations. However, these annotations give organizations and users the ability to represent notification policies and preferences, establishing clear rules of etiquette for notification interactions to the degree most appropriate for each type of notice.

The *modality-condition* can be represented in either of these two forms:

{M} or
if (user state) then {M1} else {M2}.

In these forms, {M}, {M1}, and {M2} refer to sets of modalities chosen for delivery of the notification. For example M could be {email, pager}. If a set of modalities alone is specified, as in the first form, then the user state condition is effectively a don't-care. The indicated modality set is desired regardless of whether the user is, for example, currently online or offline. The second form indicates that the required or preferred modality is dependent on the user's state. For example, a display notification may be desired if the user is currently online, but a pager notification for the same notice may be desired if the user is currently offline. Overall, our representation for notification directives gives us the needed flexibility and control to specify notification requirements reflecting emerging standard operating procedure in the domain of NASA crew interaction with automated life support control systems.

Conclusions and Future Work

An etiquette for interaction among human and automated software agents is needed for teams of these agents to effectively perform remote, distributed space operations. This etiquette should address both the new rules and policies required for humans and automated software agents to work together, as well as changes to the existing etiquette for human-human interaction. We are investigating intelligent personalized agents that implement an etiquette for human-software interaction by providing services for event notification and information management, as well as remote commanding. These services are customized based on organizational policies defined for the roles each agent fulfills.

We have described our approach for specifying and enforcing an etiquette for notification and alerting. Our approach differs from previous work on notification and alerting because it focuses on meeting the notification needs of an organization rather than on meeting the notification needs of an individual. We encode the rules of etiquette for organizationally relevant notifications by defining a set of notice specifications that describe which notices should be presented to the user and give guidance to the user interface software about when and how a notice should be presented to the user. Instead of simple categorization of incoming messages, we determine which notices should be presented by filtering over logical combinations of domain-specific conditions defined by the organization. This approach is extensible to new terminologies or jargons by defining, for example, domain-specific ontologies and using them to encode new conditions. Notification directives are applied to all notices that match filtering conditions. These directives can optionally specify a presentation saliency, user interface modalities, or both. Notification directives can use the content of an incoming notice to determine its saliency. Modality is determined by both the content of the notice and the current state of the user. To meet the needs of a dynamic organization, we monitor for changes in the user's roles and automatically update which notice specifications are in effect to reflect the new operations context. We also have encoded a strategy by which user preferences can be applied without compromising required organizational policies, ensuring the user always receives the information he or she needs as required by the organization while permitting the user to receive more information in additional ways as desired.

We have done preliminary work on designing an etiquette for coordinating distributed, remote commanding. The basis of this design is the dynamic allocation and authentication of control authority according to organizational roles and situational needs.

Once we have implemented both designs, we will investigate different models of human-software interaction for remote space operations and encoding the rules of etiquette needed to support these different models.

Acknowledgements

We want to acknowledge the support of Dr Michael Shafto, the manager of the Human-Centered Computing topic in NASA's Intelligent Systems Program, under which this work was done. We also want to recognize the support Dr. Jane Malin/JSC has given the authors in doing ground work for this research while working on her projects.

References

Cadiz, J., Venolia, G. D., Jancke, G., and Gupta, A. 2001. *Sideshow: Providing Peripheral Awareness of Important Information*, Technical Report, MSR-TR-2001-83, Microsoft Research, Redmond, WA.

Cutrell, E., Czerwinski, M., and Horvitz, E. 2001. Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance. In *Proceedings of the Human-Computer Interaction - INTERACT'2001*.

Czerwinski, M., Cutrell, E., and Horvitz, E. 2000. *Instant Messaging: Effects of Relevance and Timing*.

Hinze, A. and Faensen, D. 1999. A Unified Model of Internet Scale Alerting Services. In *Proceedings of the ICSC,* 284-293.

Horvitz, E., Jacobs, A., and Hovel, D. 1999. Attention-Sensitive Alerting. In *Proceedings of UAI '99, Conference on Uncertainty and Artificial Intelligence, Stockholm, Sweden, July 1999*. San Francisco: Morgan Kaufmann, 305-313.

MacIntyre, B., Mynatt, E. D., Voida, S., Hansen, K. M., Tullio, J., and Corso, G. M. 2001. *Support for Multitasking and Background Awareness Using Interactive Peripheral Displays*.

Marmasse, N. 2000. Location-aware Information Delivery with comMotion. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC),* 157-171. Bristol, UK: Springer-Verlag.

Schmandt, C., Marmasse, N., Marti, S., Sawhney, N., and Wheeler, S. 2000. Everywhere Messaging. *IBM Systems Journal* 39(3&4): 660-677.

[a]Schreckenghost, D., Martin, C. E., Bonasso, R. P., Kortenkamp, D., Milam, T., and Thronesbery, C. 2002. Supporting Group Interaction Among Humans and Autonomous Agents. In *Proceedings of the AAAI-02 Workshop on Autonomy, Delegation, and Control: From Inter-agent to Groups*. Edmonton, Canada.

[b]Schreckenghost, D., C. Thronesbery, P. Bonasso, D. Kortenkamp, and C. Martin. 2002. "Applying Human-Centered Computing to Intelligent Control of Life Support for Space Missions". *IEEE Intelligent Systems*, special issue on Human-Centered Computing at NASA. Sept/Oct.

Singh, M. P. 1991. Group Ability and Structure. In *Decentralized A.I. 2 : Proceedings of the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Saint Quentin en Yvelines, France, August 13-16, 1990,* Demazeau, Y. and Müller, J.-P., Eds. Amsterdam: Elsevier Science, 127-145.

So, Y.-p. and Durfee, E. H. 1998. Designing Organizations for Computational Agents. In *Simulating Organizations*, Prietula, M. J., Carley, K. M., and Gasser, L., Eds. Menlo Park, CA: AAAI Press / The MIT Press, 47-66.

Venolia, G. D., Dabbish, L., Cadiz, J., and Gupta, A. 2002. *Supporting Email Workflow*, Microsoft Research, Collaboration & Multimedia Group, Redmond, WA. <citeseer.nj.nec.com/489980.html>.

Whittaker, S. and Sidner, C. 1996. Email Overload: Exploring Personal Information Management of Email. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1996).* 276-283: ACM Press.