Supporting the Commander's Information Requirements: Automated Support for Battle Drill Processes Using R-CAST

Jeffrey From [2], Patrick Perrin [3], Daniel O'Neill [3], and John Yen [1]

[1] College of Information Sciences and Technology
Pennsylvania State University
University Park, Pennsylvania

[2] Mission Command Battle Lab
U.S. Army Combined Arms Center
Fort Leavenworth, Kansas

[3] Command and Control Cognition Branch
U.S. Army RDECOM - CERDEC
Aberdeen Proving Ground, Maryland

Abstract—**This paper discusses a novel approach that addresses the problem of supporting the Commander's dynamic information requirements through automation of the Military Decision-Making Process (MDMP) for time-constrained environments and training purposes, as part of the Tactical Human Integration of Networked Knowledge (THINK) Army Technology Objective – Research (ATO-R) initiative. We demonstrate this capability with automated user support for the execution of battle drills. Our approach is based on adapting the R-CAST cognitively-inspired agent architecture towards a context-aware anticipation of information requirements. R-CAST is a computational model of the Recognition-Primed Decision (RPD) model, which models human decision making under time stress. R-CAST agents support and collaborate with human decision making teams as both "smart aids" and "effective teammates" by anticipating, investigating, seeking, and interpreting information relevant to decision making. A key feature of R-CAST is that the proactive sharing of information relevant to decision making is automatically generated by the computational RPD model. The fundamental research question being addressed is whether the inclusion of R-CAST in Army staff processes improves said staff understanding and execution of battle tasks. We adapted R-CAST to Battle Drill #26 (i.e., responding to an IED event) as a proof of concept for team decision making under stress and constant switching of modalities. We demonstrate that the use of R-CAST cognitive agents effectively assists the Battle Manager in the S3 cell with auto-filling certain forms required by doctrine in response to the dynamism of the current state of the environment, improving cognitive performance in this task. Our novel approach integrates relevant context in communication, information, and socio-cognitive networks, coupled with cognitive modeling. We report initial findings that we can use the R-CAST cognitive framework to effectively and efficiently develop individual intelligent training tools that understand and support the dynamic information requirements of Commanders.**

Keywords- MDMP automation; Battle Drill; R-CAST; Recognition-Primed Decision; Cognitive Agents; Information Requirements; Cognitive Performance

I. Introduction and Motivations

In this research, we study a subset of the class of problems aiming at improving Commanders' cognitive performance when engaged in complex Mission Command activities, defined in Field Manual 3-0, ch1 [15], as: (1) driving the operations process; (2) understanding, visualizing, describing, directing, leading, and assessing; (3) developing teams among modular formations and joint, interagency, intergovernmental, and multinational partners; and (4) leading inform and influence activities. This research addresses the improvement of the activities under (2), where Commanders are routinely faced with the highly demanding cognitive tasks of understanding the current situation and making decisions based on an intricate combination of factors, including their current understanding, their cognitive and emotional states, and their motivations and beliefs, as well as the mission's goals & objectives. In addition, rapidly changing and time-constrained environments, situation shifts, and extremely stressful conditions, where teams are exhausted and frequently under high cognitive load, requires frequent switching of modalities. In order to accomplish these cognitive tasks, they rely on collaboration and sharing of information/knowledge with various other groups and individuals in their formal (and informal) organization, based on the available information they can find, what is known, or what is provided to them by their Command and Control information systems from the net-centric information layer. They also rely on proper battle drill training as well as following doctrine, such as the Military Decision-Making Process (MDMP) in Field Manual 5-0 [1], a doctrinal basis that gives structure to Commanders and their staff on how to make effective decisions.

The present study describes a methodology to improve cognitive performance of decision makers through the use of R-CAST intelligent software agents learning to collaborate with human warfighters in the net-centric environment. In this framework, human and software agents are combined to deal with non-routine situations. We demonstrate this capability in the particular case of decision makers following battle drill directives, where the software agents support the decision makers by proactively and reactively seeking knowledge to satisfy the information requirements dictated by the current dynamic situation and the battle drill recommendations. The immediate benefit is first, to ensure that battle drills are correctly followed, and second, that cognitive overload of warfighters is reduced by anticipating and fulfilling information requirements on their behalf.

This is the first step in a larger study that will use this method to create a Commander's Tool that has the potential to support the Commander's Situational Awareness / Situational Understanding (SA/SU) needs across a broad spectrum of information in a "tailorable" and "scalable" manner. As an example, a Commander can have his interface tailored by varying the level of visibility over subordinate unit actions, depending on their leadership capabilities. In this example, scalable refers to different form factors for different tastes/likes/dislikes.

A Brigade (BDE) Commander might use a tablet, while a Company (CO) Commander might use a smart phone. Commanders need to monitor unit activities while away from the command post. A Commander's tool, given the network, can support the Commander's informational needs, providing tools designed to reduce cognitive workload and therefore, improve cognitive performance.

This study addresses the fundamental research question of whether the inclusion of R-CAST intelligent agents in Army staff processes improves staff understanding and execution of battle-drill tasks We report the methodology and initial results in testing that hypothesis.

A. Needs of Support for Battle Drill Processes

Current Army operations are complex and demanding on commanders and their staffs. They are required to perform numerous detailed cognitive processes (e.g. battle drills), simultaneously in the intense environment of activities and functions associated with an Army Command Post (CP). By reducing this cognitive burden, the unit's leadership and staff performance could potentially be improved.

A battle drill (BD) is an example of a process used by all Army units to execute CP operations. Battle drills provide structure to sometimes chaotic situations and are instructions to ensure all components of the CP and unit leadership are aware of events and actions, while aiding their ability to properly manage unit activities. Often times, BDs include many data elements (e.g. location) that are passed to and shared by a number of staff personnel and leadership, interrupting other tasks or activities. As an example, a BD used by the majority of Army combat units is: react to an IED attack. It is triggered by some message or verbal communications from the unit that encountered the IED, and the staff must then take appropriate actions to ensure the safety of the Soldiers and civilians near the event. From the initiation to the sharing of data, this BD process is largely Soldier intensive and analog. As such it will often add to the staff officer's cognitive burden.

Exacerbating the effects of intense staff processes is the complementary issue of overly complex mission command (MC) system interfaces. The brigade combat team (BCT) command post includes as many as ten different MC systems interfaces, along with numerous commercial software applications. At any time during an operation, a single staff officer could be using three to five separate interfaces/applications. In a less demanding environment, this level of interface complexity might be acceptable; however, in chaotic and hugely complex combat operations, it can be detrimental to individual, staff and unit performance.

An interface is basically an eye into the world that a user or set of analysts is trying to understand - to comprehend the states of various data streams, states of entities or objects, and measures that indicate change, and to gather feedback to make corrective courses of action in the command and control of various activities according to a stated or emerging set of objectives. An interface can also represent a medium by which information is input or transposed, represented, stored and retrieved, sorted or categorized, communicated, and shared across multiple entities. As such it becomes a tool to both see and manipulate the environment one is trying manage. An interface can also be distributed across multiple users (e.g., a group interface, a collaborative system), wherein groups of users can operate on information jointly with the intent of establishing team situation awareness [12,13] or group actions.

Many interfaces are designed and placed in a system in a static way. That is, they are what they are and do not conform to the dynamics that transpond through them. Interfaces can be stabile but may not be responsive to dynamic, non-routine, and changing conditions. Traditional user or human/computer interfaces reflect the given states of the environment to enable the user to see and then manipulate states, but they do not have the ability to adapt on the fly to the state of user. More recently, a new approach to interfaces has been to design adaptivity into their sense and response patterns, creating a portal into the application context that is responsive directly to change of state, therein providing on-the-fly revamping to what an interface can do, which helps the users manage their workload, direct their attention appropriately, switch modalities for improving human performance, and reduce debilitating time pressure and stressful situations. The areas of user-adapted interfaces and affective computing represent two current streams representing the research and design of dynamic display-response systems. The promise of adaptive interfaces is predicated on using feedback from various sources to change on the fly in a way that improves the user's ability to respond to complex, nonlinear, non routine conditions that are typically found in naturalistic work environments [5].

A key enabling technology for user-adaptive interface is the capability to anticipate information needs of the leader and commander, and adapt the interface regarding these needs based on the role, the stress level, and other information about the user and the context. In this paper, we describe an approach to the design of user-adaptive interface using an agent technology (R-CAST) that anticipates information needs using declarative knowledge about the battle drill process relevant to the situation at hand. We will illustrate the approach using the battle drill process for responding to IED events (i.e., BD-26). Because the approach uses a declarative representation of a battle drill, it is flexible and is easy to scale for handing other types of events.

B. Improving Cognitive Processes

The vision of this research is improving cognitive processes of battle staff officers through an iteration of improved understanding and improved performance, which is enabled and facilitated by experiments and training. Toward this vision, we have conducted research to develop a user-adaptive interface to investigate whether a user-adaptive interface based on R-CAST for Army staff processes can improve staff understanding and execution of battle-drill tasks. Adding R-CAST intelligent agents in the net-centric environment improves cognitive performance in terms of reducing the time human decision makers take to respond to something like an IED event, and to reduce their cognitive overload. This is due to the fact that the software agents generate the information requirements in stressful conditions and correctly provide the necessary and sufficient information necessary to satisfy these battle drill requirements for the particular situation.

II. R-CAST: RPD-ENABLED COLLABORATING AGENTS SIMULATING TEAMWORK

R-CAST, RPD-enabled agents for simulating teamwork, is a cognitive agent architecture that is designed to be able to anticipate relevant information needs for teammates, like a human team member of a high performance team [9,10]. The capability for R-CAST agents to anticipate information needs came from a computational realization of the recognition-primed decision (RPD) model [11]. The RPD model, which is a naturalistic decision making model that claims that human experts usually make decisions in time-stressed situations based on the recognition of similarities between the current decision situation and previous decision experiences [11]. RPD provides the context for different types of information needs related to decision makings. For example, missing information can be identified from cues of decision making experience and from information requirements needed in executing a course of action (COA) after it is chosen. In addition to RPD, the R-CAST agent architecture is also built on top of the concept of shared mental models [7], the theory of proactive information delivery [10] that builds on the SharedPlan theory, and agent communication theories.

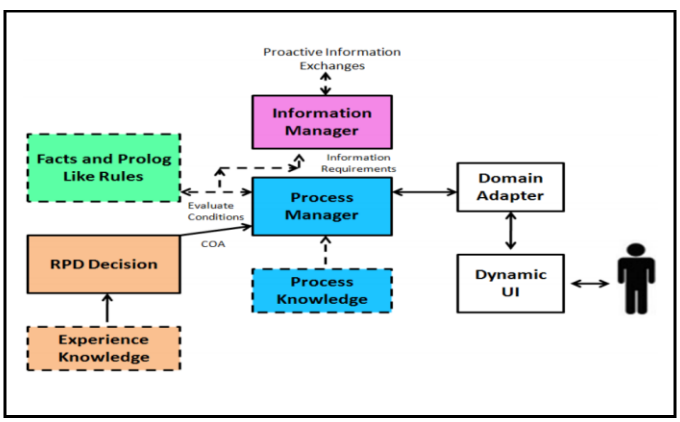

Figure 1. The high-level architecture of R-CAST

The use of context is of growing importance in developing computational systems that are more responsive to human needs. R-CAST distinguishes experience context, inference context, and process context. These three types of context representation enable R-CAST to identify information needs in these different contexts. The R-CAST agent architecture, shown in Fig. 1, uses three types of declarative knowledge:

(1) the experience knowledge used by the RPD Decision module

(2) the facts and Prolog-like rules

(3) the process knowledge used by the process manager.

The information manager generates information requirements, which enables proactive information exchanges between agents.

The language to describe a process in R-CAST is flexible, thereby allowing a wide range of Battle Drill processes and associated information requirements to be represented. In the next section, we will use battle drill 26 to illustrate how information needs are anticipated using R-CAST agents.

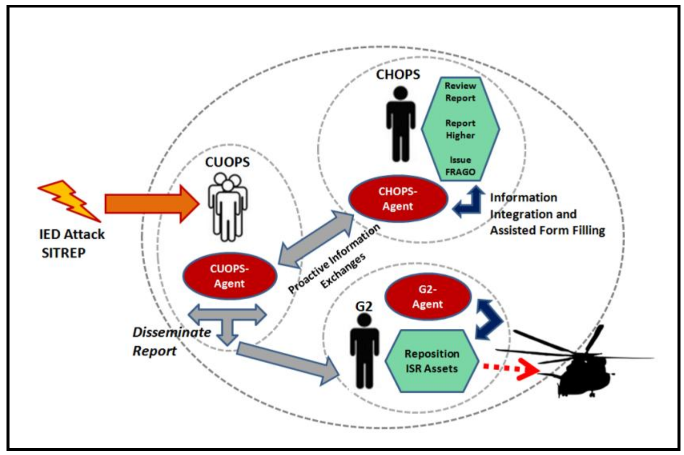

Figure 2. A Team of R-CAST Agents for Supporting Battle Staff in the Context of BD26

III. AGENT-ASSISTED ANTICIPATION OF INFORMATION NEEDS IN BD-26

The Experience Base, the Process Manager, and the Information Manager components play particularly important roles in assisting the Army staffs with the use of networked knowledge in the context of the Battle Drill 26. The experience base is to model decision making processes at a reasonable abstraction level and identify information requirements based on current conditions and previous experiences that may affect the course of actions of interest. The process manager component provides the agent architecture with the abilities to perform a variety of behaviors. The Information Manager component deals with information needs of the R-CAST agents.

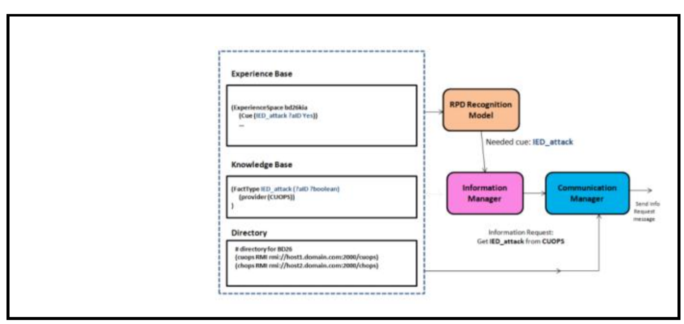

Figure 3. An Example of Anticipating Information Needs in BD26

R-CAST can be configured to assist many different Army units with different roles such as CHOPS (Chief of Operations), CUOPS (Current Operations), and G2 using different knowledge, experiences, and processes. For instance, a team of agents supporting the Division CP could include CUOPS agent, CHOPS agent, G2 agent and so on. The agents can collaborate with each other based on the battle drills encoded in their knowledge base. As an example, once the CUOPS agent receives a SITREP about an IED attack as a cue, it invokes the BD26 experience and related processes for initial assessment of the event. The CUOPS agent can communicate with other agents including the CHOPS and G2 for proactive information exchanges when necessary. The IED_attack SITREP used in the experience base and the knowledge base in Figure 3 is part of the cues for activating BD-26. This example illustrates that the provider of needed information (e.g., CUOP for IED_attack SITREP) is captured in the knowledge base of CHOPS agent, which enables the agent, if needed, to contact the provider to obtain the information in a proactive way.

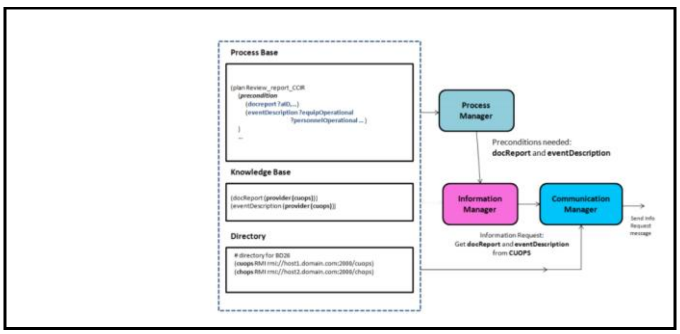

Figure 4. An Example of a plan execution having preconditions to be met

Fig. 4 illustrates R-CAST agents' capability of executing a certain plan associated with information anticipation. In the example, the plan Review_report_CCIR waits for two information objects, docReport and eventDescription, to be available. The precondition is used to specify what preconditions (e.g., availability of further information in this case) must be satisfied before making the actions defined in the plan. As the sources of the two information objects is CUOPS, an information exchange between the two agents is made through the information manager and the communication manager when the information objects become available; the information manager identifies the sources of the two information objects and the communication manager establishes physical communication between the CHOPS and the CUOPS agents by referring to the directory. The R-CAST-assisted anticipation of information needs leads not only to the BD automation but also to user centric adaptive user interface.

IV. Agent-Assisted Adaptive User Interface

A. *User Assessment*

In order to perform user-centered interface adaptations, user assessment must be carried out. This appraisal will determine the user's state and map to adaptation strategies. Affect assessment has been noted to be important with regards to cognitive performance in previous adaptation literature [5]. There are several methods for user state assessment including physiological sensing, conversation monitoring [4], expert knowledge elicitation [5], user self-reporting [5], appraising time of day and work schedule [3], evaluating the density and timing of incoming information, and gathering and evaluating user keyboard and mouse input data [6]. User self-reporting, time of day, density and timing of incoming information and user keyboard and mouse input will be used to inform current user states and thus possible adaptation strategies.

User self-reporting provides the sole static assessment method utilized to assess the user [5]. This method will inform the system whether the user is hands-on or hands-off with regard to accessing information within the system; the user's preference. A hands-on user is defined as a user who prefers to search for the information located within the event history database with little system interference. A hands-off user is defined as a user who prefers the system to automatically present applicable event history information to the user. It has previously been shown that user interruption can potentially affect user task performance [2]. This method will allow the system to adapt to a user's system interaction preference.

The dynamic assessment measurements utilized are time of day and work schedule, incoming information density, and keyboard and mouse input [5]. These assessments will be utilized to appraise the user's current affective state. The dimension of affect used in this assessment will be user stress level. The user stress level will be categorized as high stress or low stress. Examples of dynamic actions leading to high stress are a high density of incoming information in a certain time interval, a high amount of key strokes and mouse movements in a certain time interval, and a major event occurring towards the end of the user's work shift. An example of low stress assessment would be a lack of events incoming to the user. A user is assumed to be in a low stress state before interacting with the system.

The combined use of static and dynamic assessment factors is a key to accurately evaluating the user's current state. Providing more general categorizations of user states from user inputs allow for further specification if found to be practical after experimentation. Assessments of high stress or low stress and hands-on or hands-off will help determine the adaptation strategies enacted in the user interface.

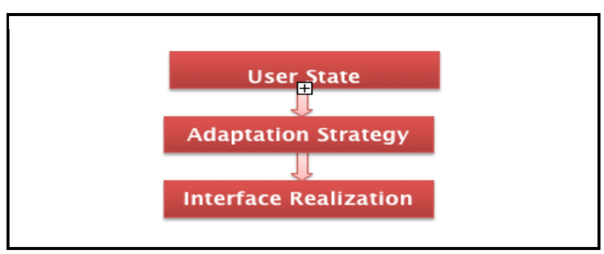

Figure 5. A High-level Overview of the User-Adaptation System

B. *Adaptation Strategies*

In this subsection, the paper presents three levels of adaptation strategies for the adaptive user interface. First, we select adaptation strategies following general rules. Then we define the mappings between adaptation strategies and the user's affective state and preferences in specific tasks. Finally, we define the graphic user interface adaptation. By means of anticipating and gathering information, adaptive user interface meets individuals' information needs. A customized display based on task-specific context and user state assessment reduces attention demands and has a positive impact on improving users' cognitive performance.

1) Selecting the Adaptation Strategy

To begin with, we define the adaptation strategies based on the following two categories of rules: generic strategies rules and context-based strategies rules [5]. Following the generic strategies rules, two aforementioned factors are considered to avoid the bias from different individuals, i.e., the user's affective state and preferences. Following the context-based strategies, we consider the task-related information to fit users' information need. From another perspective, the proposed adaptation strategies employ both the format-based strategies and the content-based strategies. The format-based strategies refer to individualized layout and color differences, etc. The content based strategies refer to displaying task-relevant information. For instance, if users are conducting the MED-EVAC (Medical Evacuation), then the task-relevant information may include presenting the conditions of helicopter nearby or security situations nearby.

2) From Adaptation Strategies to User's Affective State and Preferences

Following the rules, we define the mappings considering the dynamic and static factors that may influence the user's affective state. As we have discussed above, in this paper, we define the mappings based on the user's stress level and the user's preferences (hands-on person or hands-off person). These two factors lead to four combinations, i.e., people of high stress level plus hands-on preference, people of low stress level plus hands-on preference, people of high stress level plus hands-off preference, and people of low stress level plus hands-off preference. For people of high stress, the adaptive interface provides more alert to present the most relevant information of the event. The purpose is to provide the users in high stress level with concise information to fit their needs. For people of low stress, the user interface is proposed to display more comprehensive information based on what we have anticipated using R-CAST agent. Meanwhile, the adaptive user interface supports users of hands-on preferences to gather the relevant information by retrieving the database of anticipated information. People of hands-on preference are also enabled to analyze the information by themselves. For people of hands-off preferences, the adaptive user interface proactively gathers relevant information and making the analysis as recommendations. This guides the people of hands-off preference to conduct the activities.

3) Defining the Graphic User Interface Adaptation

Finally, we define the format of the graphical user interface. The purpose of this step is to select the best information presentation format to improve users' cognitive performance. For the Battle drills events, location is one of the most important pieces of information in describing what happens and where it happens. This motivates the employment of electronic maps as an element of the graphic user interface. To make the user interface adaptive to the user's factors in specific task, we consider the display of user interface from four levels: Icon level, Display level, Notification level, and User Interface level [5]. By changing the display on these levels respectively, the graphical user interface meets the individuals' information need to conduct the task.

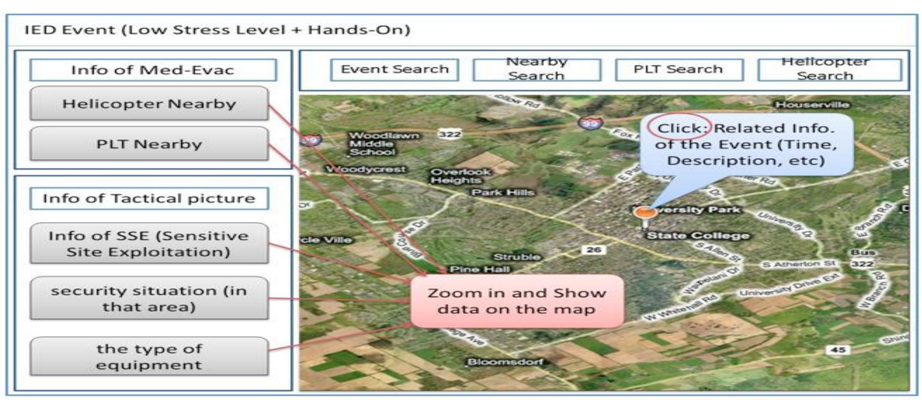

Figure 6. Adaptation User Interface Markup

Fig. 6 is a markup example of the adaptive user interface. The adaptive display is for people of low stress level plus hands-on preference. As shown in the figure, the display proposes to display anticipated information comprehensively and does not employ much alert considering the users' affective state (i.e., low stress level). For example, in the display, it presents the information of the nearby helicopter and PLT, which are important to conduct the Medevac (Medical Evacuation) task. Meanwhile, to facilitate users' of hands-on preference to gather information, the adaptive user interface provides a series of search functions based on the knowledge database. This provides users of hands-on preference with more flexibility to gather information employing the adaptive user interface.

V. FUTURE DIRECTIONS

To evaluate the impact of the agent-assisted support regarding the Commander's dynamic information requirements, we will conduct experiments to investigate (1) how the R-CAST-assisted user-centric adaptive interface can improve the cognitive performance of the Army staffs in different conditions including their stress levels and preferences, and (2) the effectiveness of R-CAST's automated support for Battle Drill processes as a training tool. As a training tool, for example, R-CAST -based coaching agent can measure how much a trainee deviates from the process and produce personalized coaching feedback. These experiments will leverage related previous research, such as the decision space visualization [14].

Additionally, as part of the Tactical Human Integration of Networked Knowledge (THINK) Army Technology Objective – Research (ATO-R) program, a flexible and robust experimentation architecture is being developed, allowing the capture and analysis of cognitive user data. The performance of R-CAST will be evaluated during a series of planned experiments under THINK.

ACKNOWLEDGMENT

This research was supported by participation in the Tactical Human Integration of Networked Knowledge (THINK) Army Technology Objective – Research (ATO-R) program sponsored by the U.S. Army CERDEC and the U.S. Army Research Laboratory under Cooperative Agreement R.ARL.2009.05.

REFERENCES

[1] United States Army Field Manual 5-0, The Operations Process, March 2010.

[2] Bailey, B. P., Konstan, J. A., & Carlis, J. V. (2001). The effects of interruptions on task performance, annoyance, and anxiety in the user interface.

[3] Blatter, K., & Cajochen, C. (2007). Circadian rhythms in cognitive performance: Methodological constraints, protocols, theoretical underpinnings. Physiology & behavior, 90(2-3), 196-208.

[4] D'Mello, S., Jackson, T., Craig, S., Morgan, B., Chipman, P., White, H., . . . Picard, R. (2008). AutoTutor detects and responds to learners affective and cognitive states.

[5] Hudlicka, E., & Mcneese, M. D. (2002). Assessment of user affective and belief states for interface adaptation: Application to an Air Force pilot task. User Modeling and User-Adapted Interaction, 12(1), 1-47.

[6] Khan, I. A., Brinkman, W. P., & Hierons, R. M. Towards a Computer Interaction-Based Mood Measurement Instrument. Proc. PPIG2008, ISBN, 978-971.

[7] Cannon-Bowers, J. A.; Salas, E.; and Converse, S. 1990. Cognitive psychology and team training: Training shared mental models and complex systems. Human Factors Society Bulletin 33:1–4.

[8] Cohen, P. R., and Levesque, H. J. 1991. Teamwork. Nous 25(4):487–512.

[9] Fan, X.; Sun, B.; Sun, S.; McNeese, M.; Yen, J.; Jones, R.; Hanratty, T.; and Allender, L.. 2006. RPD-Enabled Agents Teaming with Humans for Multi-Context Decision Making. In Proc. AAMAS'06, pages 34--41, Japan.

[10] Fan, X.; Yen, J.; and Volz, R. A. 2005. A theoretical framework on proactive information exchange in agent teamwork. Artificial Intelligence 169:23–97.

[11] Klein, G. A. 1997. The recognition-primed decision (rpd) model: Looking back, looking forward. In Naturalistic decision making (Eds: C. E. Zsambok and G. Klein). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc. 285–292.

[12] Wellens, A. R & Ergener, D. (1988). The C.I.T.I.E.S. game: A computer-based situation assessment task for studying distributed decision making. Simulation & Games, 19(3), 304-327.

[13] Cooke, N. J., Gorman, J. C., Duran, J.L., & Taylor, A.R. (2007). Team cognition in experienced command-and-control teams, Journal of Experimental Psychology: Applied, 13(3), 146-157.

[14] Kim, H.W., Oh, S., McNeese, M., Yen, J., Hanratty, T., Strater, L., Cuevas, H., and Colombo, D., Enhancing Situation Awareness with Visual Aids on Cognitively-inspired Agent Systems, in the Proceedings of the 3rd International Conference on Applied Human Factors and Ergonomics (AHFE 2010).

[15] United States Army Field Manual 3-0, Operations, Appendix B, The Military Decisionmaking Process, March 2008.