RPD-Enabled Agents Teaming with Humans for Multi-Context Decision Making

Xiaocong Fan
College of Information
Sciences and Technology
The Pennsylvania State
University
University Park, PA 16802
zfan@ist.psu.edu

Bingjun Sun
Department of Computer
Science and Engineering
The Pennsylvania State
University
University Park, PA 16802
bsun@cse.psu.edu

Shuang Sun,
Michael McNeese,
John Yen
College of Information
Sciences and Technology
The Penn State University
University Park, PA 16802
{ssun,mmcneese,jyen}
@ist.psu.edu

ABSTRACT

Team decision making under stress involving multiple contexts is an extremely challenging issue faced by various real world application domains. This research is targeted at coupling cognitive agent technology and human-centered teamwork to address the informational challenges associated with Command and Control (C2) teams in contemporary military operations. Two sets of experiments, each with various settings of context switching frequencies and tasking complexities, were conducted. To ensure that the human subjects were familiar with the C2 context, they were selected from US Army ROTC (Reserve Officer Training Corps) students. Experiments on C2 teams that involve human subjects only were conducted first. We observed the decision making behavior of human subjects and incorporated expert behaviors into R-CAST—an agent architecture built upon a naturalistic decision making model that captures how domain experts make decisions based on experiences and situational similarity recognition. We then conducted another set of experiments with R-CAST agents as teammates and decision aids for human subjects. The results show that RPD-enabled agents can significantly improve the tasking capacity of C2 teams in multi-context decision making under stress. It also suggests that higher demand situations require more competent teammates.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Design, Experimentation, Human Factors

Keywords

Cognitive agent, Human-centered teamwork, Naturalistic decision making, Multiple contexts, Attention switching

1. INTRODUCTION

Team decision making involving multiple contexts is an extremely challenging issue faced by various real world application domains. Command and Control (C2) in complex urban terrain is one of such domains, where C2 teams have to frequently confront the so called three-block challenge [5]—conducting humanitarian, peacemaking, and combat missions in close proximity (i.e., involving three contexts that overlap in time). Multi-context team decision making is challenging because it requires effective team collaborations in rapidly gathering dynamic information from multiple sources (collateral space), in proactively sharing relevant information for establishing global situation awareness, in managing and reasoning across multiple decision spaces for different contexts (areas of interest), and in choosing optimal corridors for movement.

In the Multi-Agent Systems field (MAS), there has been an increasing interest in empowering agents with naturalistic decision making models to better support *human-agent collaboration* in making decisions under time stress [4, 12, 13, 14, 16]. One of such models is Klein's Recognition-Primed Decision framework (RPD) [8]. The RPD model claims that in complex situations human experts usually make decisions based on the recognition of similarities between the current decision situation and previous decision experiences [9]. Cognitive studies have shown that over 95% of human decisions conform to the RPD model in various time-stressed situations [9]. Norling, Sonenberg, and Ronnquist [12] have examined the integration of RPD model into BDI agent framework. Fan, Sun, McNeese, and Yen [4] have implemented an RPD-enabled cognitive agent architecture (R-CAST) and evaluated the performance gains when human decision makers were assisted by R-CAST agents, operating under stress in a real-time simulation environment.

However, multi-context decision making, especially from the human-centered teamwork perspective, has attracted little attention from the MAS field. Norling et al.'s work [12] only explored ways of using reinforcement learning to en-

hance situational recognition. While Fan et al.'s attempt [4] centered on human-agent collaboration in the decision making process, they only investigated the impact of an adaptive decision making mechanism, with the issue of multi-context decision making left open. Although RPD-enabled agents were employed to start to address the issue of multiple contexts [5], we still need experimental studies for fully understanding the nature of multi-context decision making, which is critical to further developing agent technologies for enhancing the performance of human-centered teamwork.

In this paper, we investigate how and to what extent R-CAST agents, as teammates and decision aids, might help in human decision making under stress and multiple contexts. This research is conducted within the C2 domain and is targeted at coupling cognitive agent technology and human-centered teamwork to address the informational challenges associated with multi-context team decision makings. The remainder of this paper is organized as follows. Background on human-centered teamwork and the R-CAST agent architecture are given in Section 2. The problem domain, simulation environment, and scenario designs are described in Section 3. Experiments on human-only C2 teams and human-agent C2 teams are presented in detail in Section 4 and Section 5, respectively. Section 6 presents conclusions.

2. BACKGROUND

2.1 Human-centered teamwork

Multi-agent teamwork [3] is concerned with joint commitments and joint responsibility. In particular, human-centered teamwork, where human and software agents interact as peers, is about establishing situation awareness collaboratively, developing shared mental models as situation evolves, adapting to mixed-initiative activities, etc. It is claimed that agents, if they could collaborate with human peers effectively, can allow humans to pay attention to more important activities [1], and make better decisions using information at a greater accuracy and finer granularity[11].

Bradshaw et al. developed KAoS, a collection of agent services in the form of adjustable policies, which allow the renegotiation of roles and tasks among humans and agents when new opportunities arise or when breakdowns occur [1]. MokSAF is a computer-based simulation system developed to evaluate how humans can interact and obtain assistance from agents within a team environment [11].

Our research reported here differs from the above efforts in that we attempt to deepen the understanding of human-centered teamwork by examining multi-context decision making in time-stressed situations. In particular, we try to uncover the potential impacts of RPD-enabled (or experience-guided) cognitive agents, when acting as human's teammates and decision aids, on the performance of C2 teams.

2.2 The R-CAST Agent Architecture

The R-CAST agent architecture [4] is built on top of the concept of shared mental models [2], the theory of proactive information delivery [6], and Klein's Recognition-Primed Decision (RPD) Model [8]. The R-CAST agent architecture has implemented a "collaborative-RPD" decision process, which supports close human-agent collaborations in relevant information sharing, decision progress monitoring, and expectancy-based decision adaptation.

The major components of R-CAST are shown in Figure

Figure 1: Major components of R-CAST.

1, where the "RPD-based Decision Making" module uses domain knowledge, past experiences, and the current situation awareness to produce a new or adapt an existing decision. The fulfillment of a decision can involve inter-agent and intra-agent activities, which are coordinated by the "Teamwork manager" and "Taskwork manager", respectively. Any unexpected conditions resulting either from situational changes or from action failures will trigger another round of decision making. Also, any behavior (plan) adaptations that have been tested as workable will be used to adapt the relevant experiences.

R-CAST is a configurable architecture. Many of the components can be replaced (e.g., use JESS as the knowledge base) to serve the purposes of various studies. If the default R-CAST configuration is kept, using R-CAST to develop a multi-agent system involves two major steps: (1) elicit domain knowledge and implement a domain adapter for the problem to be studied, and (2) elicit and represent domain expertise about decision making in terms of experiences (cf. Section 5.1.1 for an example and refer to [4] for detail).

Two important features that are relevant to this study are the "expectancy-based adaptive decision making" and "situation-guided switching of decision spaces."

First, R-CAST agents adapt their decisions by detecting and responding to situations that are anomalies to the original expectancy (i.e., what is observed/informed conflicts with what was expected). Anomalous situations can happen frequently in a dynamic environment. For example, suppose under the current situation it is expected that two crowds, G_1 and G_2, should be active in two isolated regions. Anomaly occurs if G_1 and G_2 start to move closer and closer.

Experiences in an experience base are organized hierarchically: experiences at a higher level are refined by those at a lower level (with more relevant cues and expectancies considered). The RPD-based decision making is a process of finding an experience that worked in a previous situation that is closest to the current situation according to certain similarity metrics. This is an iterative and adaptive process in R-CAST. Start with the most abstract experience in the current decision space, as more and more situational information becomes available, agent's recognition of the past experience closest to the current situation becomes more and more concrete (at the lowest possible level of the hierarchy). During the process, the agent simultaneously monitors the expectancies associated with the recognized experience. The recognition is enforced as new events are emerging as expected. It is challenged when some expectancy becomes

false as the situation evolves; in such a case, the agent has to backtrack along the experience hierarchy to seek a better recognition. In the study reported here, this feature is used such that an R-CAST agent (S2) can determine what information the other teammates (S3/S4 agents) need, and then proactively provide the information.

Second, R-CAST allows the specification of multiple experience bases, and the management of multiple decision spaces at run time. The nature of a situation determines what the current working-context is, and situational changes can cause the switch of one context to another. Consequently, an agent working in one decision space can switch into another decision space, and return to a previous space when the current context switched back. This feature allows us to implement agents that can dynamically switch between humanitarian, peacemaking, and combat contexts.

3. SETTING STAGES FOR EXPERIMENTS

In this section we describe the problem domain, scenario design, and the simulation engine used in the experiments.

3.1 Problem Domain

The problem domain involves C2 teams reacting to potential threats that emerge unexpectedly in a metropolis. It imposes challenging information demands associated with the command and control of urban operations, including humanitarian, peacekeeping, and combat operations.

In the environment, of special interest are main supply routes (MSRs) and three kinds of key buildings (religious buildings, schools, and hospitals). A C2 team consists of S2 (intelligence cell), S3 (operations cell), and S4 (logistics cell); they, as a team, need to work collaboratively to handle incoming threats. Their roles have been simplified and defined as follows: S2 is responsible for processing incoming reports, called Spot reports, collecting relevant information from other sources, and alerting S3 and/or S4 of threats appropriately. S3 needs to process alerts from S2 and decide on the courses of action (COA) to handle threats, given varying timing constraints and limited resources. Two types of friendly units are under S3's charge: squad units and EOD (Explosive Ordnance Disposal) teams. S4 needs to process alerts from S2 and minimize damages to MSRs.

Potential threats come from three types of targets: Crowds, Insurgents, and IEDs (Improvised Explosive Device). A *crowd* represents a group of people that may contain key friends or foes. A crowd can be of medium (M) or large (L) size, and the group size of a crowd can change over time. A crowd can move around and it may get merged with another crowd. Table 1 shows under various situations the credit value (the points a C2 team can get if a target is handled successfully) of a crowd, the resources required to handle a crowd, and what action should be taken on a crowd, where 'U' refers to "squad unit", and the value of a crowd has additional 10 points when the crowd is near an MSR. For example, the second entry says that to disperse a medium size crowd with a foe needs two squad units, and 40 points can be credited if the crowd is dispersed successfully.

Another type of movable targets is *insurgents*, each is associated with a threat level that can be L(low), M(medium), or H(high). As shown in Table 1, how many squad units are required to capture an insurgent depends on its threat level. *IEDs* are motionless targets. An IED, if exploded, can cause damage to the nearby buildings and MSRs. One squad unit

Table 1: Requirements on handling targets

Targets			Value	Res. req.	Action
Crowd	M	w/o foe	20	1U	monitor
	M	w/ foe	40 (+10)*	2U	disperse
	L	w/o foe	40 (+10)*	2U	disperse
	L	w/ foe	50 (+10)*	3U	disperse
Insurgent (3 threat levels: L, M, H)			50+50n	(n+1)U	capture
			n=1,2,3 for L,M,H		
IED			60 (80)*	1U + 1E	remove
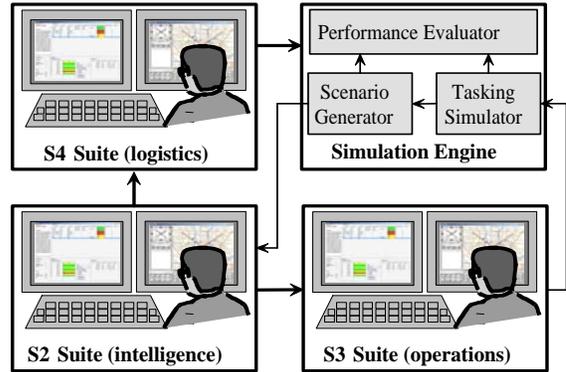
Figure 2: Environment settings of the experiments.

and one EOD team (E) are required to remove an IED. If successful, 60 points can be credited if the IED is close to buildings only or MSRs only, 80 points if it is close to both.

3.2 Simulation Engine

Figure 2 depicts the general settings of the experimental deployment, where each C2 cell (S2, S3, S4) has equipment with a map display and a human user interface (HUI). Human operators can monitor the current situation through the map displays, and interact with other peers through their HUIs.

The development of situations is controlled by the Simulation Engine module. Spot reports about situational changes are dynamically generated and fed to the S2 suite following certain controllable patterns (e.g., every n seconds; interleaving reports of different target types in fixed order or randomly). The S2 suite recognizes potential threats and decides whether and when to alert the S3 and S4 suites. Thus, although each suite has a map display, the S3 and S4 suites may only have partial views of the current situation.

The Simulation Engine module has three components: scenario generator, tasking simulator, and performance evaluator. At every cycle, the *scenario generator* creates a Spot report for each active target based on the script of its dynamics. Here is an example script of a crowd:

```
(  (crowd C121 40.04 -82.96 0.8 10)
   (report_key A12 friend 7) (successor C112)
   (m 40.04 -84.995 1.0) (m 42.05 -82.975 0.5)
   (s 27 20) (s 39 90))
```

This defines a crowd named C121, that appears at location (40.04, -82.96) 10 seconds after the system starts, and can move at full speed of 0.8 u/s. A friend named A12 shows up 7 seconds later. The crowd moves to (40.04, -84.995) at

Table 2: Scenario settings

Active targets (3 types in total)	Macro context switching frequency		
	L: 15 sec.	M: 10 sec.	H: 5 sec.
TC = 2 (× 3)	Scenario 1	Scenario 4	Scenario 7
TC = 3 (× 3)	Scenario 2	Scenario 5	Scenario 8
TC = 4 (× 3)	Scenario 3	Scenario 6	Scenario 9

its full speed, then moves to (42.05, -82.975) at 0.5 times of its full speed. The crowd size changes to 27 after 20 seconds and to 39 after 90 seconds. The disappearance of this crowd will trigger another target named C112 to show up.

Whether a generated Spot report is sent to S2 suite depends on the communication pattern set before each run, which is flexible, so that it allows, for example, the setting where reports about crowds being sent every other cycle.

The *tasking simulator* manages the tasks issued by the S3 suite by monitoring the engagement (e.g., taskload) of each squad/EOD unit, tracking the progress of each ongoing task, and reporting to scenario generator about any situational changes effected by the completion of a task.

The *performance evaluator* records the number of targets successfully handled; whether the S3 suite paid attention to the nearby buildings when issuing tasks; how many times all the resources are engaged after issuing a task; how many times S3 canceled a task that was no longer correct relative to the current situation; how many times resources were wasted on non-threatening targets (i.e., small crowds, IEDs far away from key buildings or MSRs); and how many times resources were re-allocated to another target that could offer a better chance to succeed.

3.3 Scenario Design

To study multi-context decision making, we first clarify what "context-switching" means in the problem domain. Humanitarian context refers to activities related to IEDs; peacekeeping context refers to activities related to crowds; and combat context refers to activities related to insurgents. It is a *macro* context-switching when Spot reports from two consecutive cycles are about different types of targets.

When the S2 (or S3) suite gets Spot reports (or alerts) about multiple targets simultaneously, even though they are of the same type, human operators have to switch their attention back and forth to handle the targets. We refer to this as *micro* context-switching. Because task difficulty correlates with time demands [17], varying the number of active targets (i.e., controlling *micro* context-switching) changes how much time a human operator can take on each target, thus varying task complexity.

We thus designed 9 scenarios as listed in Table 2, each with a different combination of context-switching frequencies and task complexities. For example, Scenario 5 switches context every 10 seconds, and has 3 active targets for each type (9 active targets in total). The scenarios also differ in settings of initial locations (targets, MSRs, key buildings, IEDs), itineraries and velocities of movable targets, sizing of crowds, threat levels of insurgents, targets' appearance time, etc. These are specified as scripts to the Scenario Engine.

Each target has a lifespan, which depends on whether a movable target moves to the end of its pre-specified itinerary, whether it comes to the expiration time of a motionless target, or whether the target has been successfully handled by the S3 suite, whichever comes first.

Other settings common to the 9 scenarios include: (a) each scenario lasts 10 minutes; (b) under the control of the S3 suite are 10 friendly units (1 EOD team and 9 squad units); they all move at the same speeds; (c) crowds with size between 10 and 35 are of M size, crowds with more than 35 people are of L size; (d) the Scenario Engine is set such that Spot reports of different target types are sent interleavingly in a random pattern.

4. EXPERIMENTS ON HUMAN C2 TEAMS

The aim of the experiments on Human C2 teams is two-fold. First, we want to gain a better understanding of how cognitive limitations affect C2 team performance and how to minimize such impact. Second, we need to gather experiences for RPD-agents to use in experiments on human-agent C2 teams. We thus chose US Army ROTC (Reserve Officer Training Corps) students as the human subjects. At the current stage, we only used a dummy agent simulating the S4 suite, focusing mainly on S2 and S3 suites.

4.1 Team Organization

4.1.1 S2 Human

The left screenshot in Figure 3 is the S2 human user interface, the design of which was largely influenced by the domain experts we consulted. Displayed on the left are incoming Spot reports and the details. The upper-left area shows the current *macro* context. In the experiments, S2 human subjects were asked to do communication actions by following the current macro context as far as they could.

Shown in the list on the upper-right pane are fused information about targets, which are categorized into target type, name, status, crowd size, persons associated with crowds, nearby buildings, and whether S3/S4 suites have been alerted once. Except for a target's location information, which will be sent automatically to S3/S4 suites once they are aware of the target, S2 human has to update S3/S4 suites' knowledge about the situational changes in a timely manner. S2 humans can perform four kinds of communication actions. After selecting an active target from the target list, an S2 human can (a) press the button "MIDB Query" (MIDB stands for Military Integrated DataBase). Depending on the type of the selected target, from MIDB S2 humans can gather threat level information for an insurgent, recognize whether a key person is a friend or foe, and update information about buildings nearby an active target; (b) press the button "Alert to S3" to refresh S3 suite's view of the selected target; (c) press the button "Alert to S4" to alert S4 when the selected target is moving close to a MSR; and (c) press the button "Cancel Alert to S4", if the selected target is moving away from MSRs. The bottom portion of the interface shows S2's communication behaviors.

4.1.2 S3 Human

Figure 4 shows a screenshot of the S3 user interface used in *human-agent* C2 experiments. The interface used in human-only experiments is the same except that (1) there is no cue for resource requirements (i.e., the "Expected Units" column is blank), and (2) there is no color coding for target priorities (cf. Section 5.1.2 for improvements and the reasons).

Shown on the left are context indicator (S3's context switching depends on S2's situated decision making on information
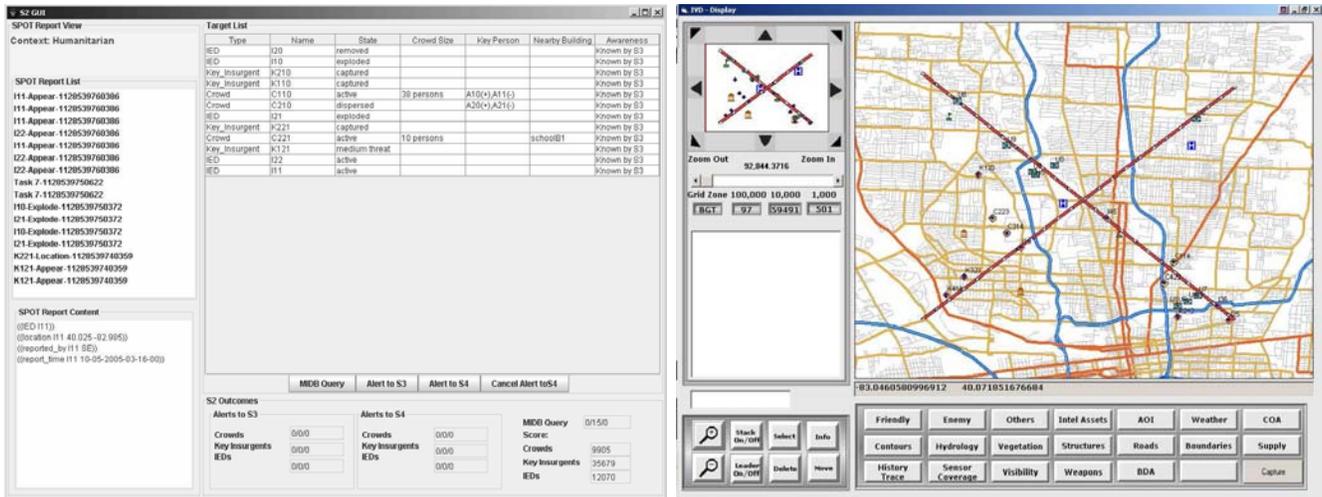
Figure 3: Left: S2 human user interface; Right: Map Display (one for S2, one for S3).
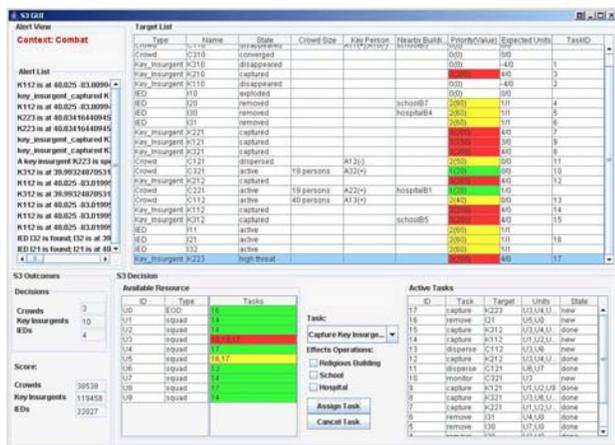

Figure 4: S3 human user interface.

sharing), a list of alerts from the S2 suite, and the S3 suite's current performance. The upper-right list shows target information, similar to S2 human interface. Allocatable resources and their taskloads, command area, and task list are displayed on the bottom-right. S3 subjects can issue 4 types of operations: monitor/disperse a crowd, capture an insurgent, and remove an IED. S3 subjects also need to pay special attention to the effects of their operations; here we simply require S3 subjects to check the type of buildings nearby the selected target, if applicable, when they issue a task. S3 subjects were asked to follow the target values and resource requirements listed in Table 1 when issuing tasks.

The S3 suite's main task is to decide which target to handle first, and how to allocate resources appropriately to multiple active targets, which is a constraint-satisfaction problem. In the multi-tasking environment, S3 subjects need to consider (a) resource constraints–balancing requirements (resource type, amount) among multiple targets; (b) utility constraints–maximizing the total values of targets successfully handled; (c) spacial constraints–others being equal,

considered first resources closest to a target; and (d) timing constraints–earlier tasking has a better chance to succeed, due to the uncertain nature of a target's lifespan.

4.2 Result Analysis

We recruited 10 C2 teams (20 human subjects): 8 teams composed of ROTC students, and 2 teams composed of ROTC officers. Each team was tested using the 9 scenarios described in Table 2. The average performance in terms of the total value of targets handled successfully is plotted using a dotted line in Fig. 5(a). It clearly indicates that there is a tasking capacity (limit); the human teams approached the limit roughly in the medium context-switching settings.

This conforms to cognitive studies [17, 10] in general, because both S2 and S3 subjects are limited by their cognitive capacities. While monitoring the situation development from the Map Display, at the same time S2 subjects need to process Spot reports under time stress, to recognize/ gather missing information, and to constantly make decisions on *when* to share information with *whom* about *what*. Similarly, S3 subjects, while being limited by resources and multi-tasking capacity, need to constantly monitor the status of active targets, to decide *when* and *how* to handle *which* target (prioritization, resource (re-)allocation), and to judge whether to cancel an ongoing task if situation changes.

In addition to gaining expert experiences that can be applied to the human-agent studies, critical issues were also revealed from the experiments. For instance, in low-stressed situations some S2 subjects tended to frequently query MIDB much more than necessary, while in high-stressed situations, some S2 subjects simply ignored the spacial information from the Map Display when they should. It was also observed that some S2 subjects had difficulties in switching to the new context, tending to process targets belonging to the last context until completion. Similarly, in high-stressed situations, having no time to check the Map Display, some S3 subjects tended to allocate resources in bundles (e.g., U1-U4), rather than those closest to the target.

Such human "errors" rest on their fundamental cognitive capacities. The performance cannot be improved by simply providing a better human user interface without employing

appropriate technologies to appropriately support or match human cognitive capacities and capabilities.

5. HUMAN-AGENT C2 TEAMS

In this set of experiments, we (1) used an R-CAST agent (S2 agent) to play the role of S2 suite, and (2) used another R-CAST (S3 agent) to act as S3 human operators' decision aid. The purpose is to investigate whether and to what extent cognitive agents, being empowered with expert experiences, could help in improving C2 teams' performance.

5.1 Team Organization

5.1.1 S2 Agent

An S2 agent, playing the role of the S2, acted as the S3's teammate. The S2 agent uses a knowledge base to track the current world state (including targets, S3/S4's information needs, status of their information awareness). Whenever receiving a Spot report, the S2 agent will update the current state of the target under concern, query relevant information from the MIDB, then check whether the target's state has changed significantly (i.e., the change can affect the S3/S4's decisions and decision adaptations). If so, send alerts to whoever affected.

Most importantly, expertise about the S2's operations has to be encoded as experiences in the form acceptable to R-CAST agents. Since the problem domain involves three contexts, the S2 agent also needs to maintain three experience bases. The following is an example experience format used by an R-CAST agent:

```
(Experience      e26
    (Cue         (Close_enough ?crowd1 ?crowd2))
    (Expectancy  (Move_closer ?crowd1 ?crowd2))
    (Anomaly     (Move_away ?crowd1 ?crowd2))
    (Goal        (peacekeeping yes))
    (Action      (Alert_converged_threat)))
```

It says that experience "e26" applies to the cases where two crowds are coming close enough, and it is expected that they are moving closer; if this is the case, the action "Alert_converged _threat" is triggered; if it turned out that the crowds moved away, the action would be terminated.

5.1.2 S3 Agent and S3 Human

For the S3 suite, an S3 agent acted as the S3 human's decision aid. To offer highly acceptable aids for decisions and decision adaptations, we incorporated a recommendation model into the S3 agent based on the observations during and confirmations after the experiments on human-only C2 teams. Such an approach of employing heuristics from human experts to further help them is significant in practice.

The recommendation model has four components. First, the S3 agent can recommend *target priorities* to the S3 human. Based on the advice collected from the S3 subjects in the human-only C2 experiments, the S3 agent was implemented such that it can use colors to show target priorities: red for targets with a high priority, yellow for targets with a medium priority, and green for targets with a low priority. However, there may exist multiple targets with a high priority at the same time. Also, due to the uncertainty of targets' lifespans and the current locations of targets and friendly units, the best target selection may not be the one with the highest priority. Thus, which target to handle first

still depends on the S3 human's decision, which can largely impact how the situation evolves and the team performance in the long run. Second, the S3 agent can remind the S3 human about the current resource requirements of each active target (i.e., how many more units are needed). Third, the S3 agent can automate the selection of operation types and the "effects of operation". This releases the S3 human's attention for other activities.

Lastly, based on the following heuristic model, the S3 agent can recommend resources to the currently selected target to support S3 human's *resource allocation behavior*.

We use $1, 2, \cdots, n$ to denote resource types; R_i^o to denote the set of resources of type i; R_i to denote the set of *free* resources of type i; and R_i' to denote the set of *re-assignable* resources of type i. *Re-assignable* resources are those that have already engaged in a task but can be preempted by another task. The current domain has 2 resource types: squads and EOD.

Each target places a resource requirement (cf. Table 1). The resource requirement of target t is denoted by $\overline{r_t} = (r_1(t), r_2(t), \cdots, r_n(t))$, where $r_i(t)$ is the number of needs on resources of type i. For example, $\overline{r_{c121}} = (2, 0)$.

Given a target t, if $r_i(t) > |R_i|$, we need to find re-assignable resources engaged in the on-going tasks. The on-going tasks can be ordered ascendingly by *priorities* up to tasks with priorities no more than the task being considered. Let $target(T)$ be the target of task T, $R(T)$ be the set of resources engaged in T. The priority of a task T with respect to a target t, is determined by its *TValue*:

$$TValue(t, T) = \frac{TargetValue(t)}{\sum_{r_i \in \overline{r_t}} w_i \cdot r_i} \times \frac{TargetRemainTime(t)}{max\{distance(t, \gamma) | \gamma \in R(T)\}}$$

where w_i is the weight of resources of type i, representing the cost of using a resource with type i, and the second factor on the right represents the chance of success when using resources engaged in T to handle target t. Here, $TargetRemainTime(t)$ represents the *estimated* value of how long t will remain before being removed from the simulation.

Resources engaged in task T will be considered re-assignable for t if $TValue(t, T) > \delta * TValue(target(T), T)$, where a relatively good value for the adjustable variable δ can be obtained by running pre-experiments.

Let $(T_1, T_2, \cdots, T_k)$ be a list of tasks determined in the above way relative to target t, and $(y_i^{T_1}, y_i^{T_2}, \cdots, y_i^{T_k})$ be a list, where $y_i^{T_j}$ is the set of resources of type i from task T_j. Then, resources of type i re-assignable for t is $R_i'(t) =$

$$\begin{cases} y_i^{T_1} & (r_i(t) - |R_i|) \leq |y_i^{T_1}| \\ \bigcup_{j:1..m} y_i^{T_j} & |\bigcup_{j:1..m-1} y_i^{T_j}| < (r_i(t) - |R_i|) \leq |\bigcup_{j:1..m} y_i^{T_j}| \end{cases}$$

Next, given the set $R_i(t)$ of resources of type i available for target t, resources closest to the target t ought to be selected first. Let $D = (D_1, D_2, \cdots, D_x)$, where $\{D_i : (1 \leq i \leq x)\}$ is the partition of $R_i(t)$ by the distance of resources to target t, and D_i in D are ordered by distance ascendingly. That is, resources in D_i have the same distance to t and they are closer to t than those in D_{i+1}. Then, $closest(R_i(t), x) =$

$$\begin{cases} select(D_1, x) & x \leq |D_1| \\ select(\bigcup_{j:1..m} D_j, x) & |\bigcup_{j:1..m-1} D_j| < x \leq |\bigcup_{j:1..m} D_j| \end{cases}$$

which returns x number of resources closest to t, where $select(A, z)$ returns any one of A's subset of size z. Here, it
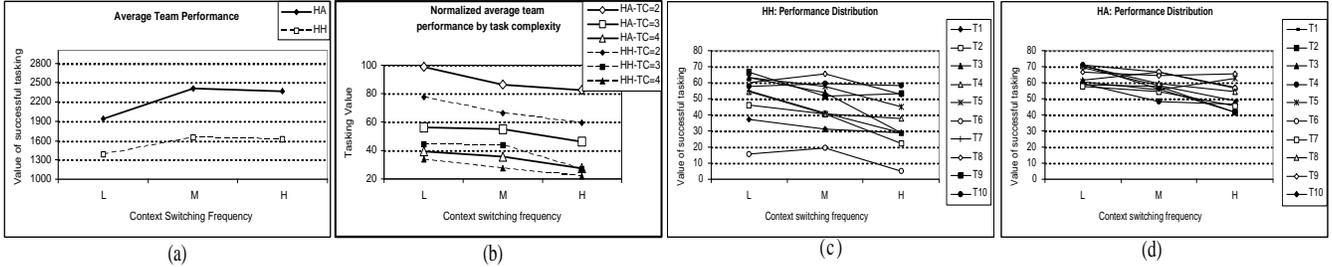
Figure 5: (a) Average performance (in net value); (b) Normalized performance by task complexity, (c) Performance distribution of human C2 teams, (d) Performance distribution of human-agent C2 teams.

doesn't matter whether to consider resources in $\bigcup_{j:1..m-1} D_j$ first, because at least one will come from D_m, which dominates the time it takes for the resources to reach t's location (we assume all the resources move at the same speed).

Now, given a target t and its resource requirement $\overline{r_t} = (r_1(t), r_2(t), \cdots, r_n(t))$, S3 agent follows the criteria below to allocate resources for target t. For each resource type i, (1) if $r_i(t) \leq |R_i|$, return $closest(R_i, r_i(t))$; (2) if $|R_i| < r_i(t) \leq |R_i \cup R'_i(t)|$, return $R_i \cup closest(R'_i(t), r_i(t) - |R_i|)$.

Actually, in cases where $r_i(t) > |R_i \cup R'_i(t)|$, two tasks can be issued separately to meet the resource requirement of t: one task employs resources in $R_i \cup R'_i(t)$, then issue another task later when more resources become available. However, this might mess things up and human subjects may easily forget to return to finish those partially committed operations. We thus left this open: human subjects could choose to split tasks if they could manage their attention.

In experiments, S3 humans could, if they want, override the above four kinds of recommendations from the S3 agent.

5.2 Result Analysis

We recruited ROTC students to form 10 human-agent C2 teams, and each team was tested using the same 9 scenarios. The average performance in terms of total value of targets handled successfully is plotted in Figure 5(a). It suggests that the C2 teams, even assisted by cognitive agents, still have a tasking capacity (limit). However, it also clearly indicates that, compared with human-only C2 teams, the tasking capacity of human-agent C2 teams have been significantly improved (all results discussed were 'significant' at the $p < .05$ level). This suggests that cognitive agents can play a critical role in alleviating the impact of human's cognitive capacity on the performance of decision makings involving multiple contexts.

5.3 Performance Comparisons & Discussion

In order to gain a better understanding of the performance difference, we plotted the *normalized performance* categorized by task complexities (amount of concurrent tasks, cf. Table 2) in Figure 5(b). It shows that the performance of both human C2 teams (dotted lines with filled markers) and human-agent C2 teams (lines with hollow markers) were affected considerably by task complexity. In addition, human-agent C2 teams performed much better when the performance at the same level of task complexity is compared.

Going deeper, we plotted the performance of individual teams in Figure 5(c)-(d). It shows that human-agent C2 teams' performances are relatively stable, while human C2

Table 3: Decision making behaviors in detail

	Tasks	*FullUses*	*Reassigns*	*W. use*	*W. att.*
HA	32.52	3.84	11.65	0.207	0.336
HH	23.73	3.07	4.97	0.443	0.556

W. use: incorrect resource use; W. att.: incorrect attention

teams differed enormously. This may demonstrate that cognitive agents could augment human cognitive capacity and improve the overall C2 performance. Additionally, in both cases, the performance deviation became larger as context switching frequency increased. This may suggest that more stressed situations require more competent C2 operators.

The experiment logs also allowed us to further identify what had contributed to the performance difference. Table 3 gives some descriptive statistics.

With the help of R-CAST agents, S3 subjects could issue more tasks (32.52 vs. 23.73), could use the limited resources more effectively (the number of times where all the resources were engaged: 3.84 vs. 3.07), and were apt to reallocate resources (11.65 vs. 4.97). Such improvement can be attributed to the balance of workload between the humans and agents such that the limits of human cognitive capacity were accommodated. In particular, the impact of S2 subjects' cognitive capacity was eliminated (S2 agent took the role); the impact of S3 subjects' multi-tasking capacity and mental efforts in resource allocation was alleviated.

With respect to the metric of "closer-resource-used-first", S3 subjects in human C2 teams (HH teams) made more incorrect judgments than S3 subjects in human-agent teams (HA teams): 0.443 vs. 0.207. S3 agent's recommendation could be wrong because its world model can be outdated: it only gets updated information periodically from the Simulation Engine. Also, S3 subjects missed most of the attention requirements on the selection of "Effects of Operations", while it was a big change when S3 agent helped S3 subjects: 0.556 vs. 0.336. Again, S3 agents might judge the situation incorrectly due to its outdated world state. Using agents, other kinds of human errors (e.g., wasting resources on non-threatening targets, issuing wrong operations to targets) were significantly diminished or avoided under high-stressed situations.

The results are encouraging. Although the sample space is restricted, the selected human subjects can be taken as representatives of domain experts—they have been trained to do similar operations. It may be argued that ROTC students typically are near the bottom of the expertise hier-

archy. Referring to the subjects as experts might be problematic, because the working memory load for a task differs greatly for experts vs. non-experts. However, as far as this particular experiment is concerned, ROTC students can be taken as domain experts. This was confirmed by our experiment which showed that the performance of the 2 ROTC officer teams were not the best among the 10 teams.

6. CONCLUSION

Multi-context real-time decision making is an extremely challenging problem faced by various real-world application domains. We have started to address this challenge associated with C2 teams operating in complex urban terrain, using the R-CAST cognitive agent architecture as human operators' teammates and decision aids. This paper, focused on the coupling of cognitive agent technology and human-centered teamwork, reported our experimental studies about the impact of RPD-enabled agents on C2 teams' performance in multi-context decision making under stress.

The experiments represent an important step forward in uncovering the nature of real-world problems, because the environment simulated the real domain and the human subjects were recruited from Army ROTC students. The results demonstrated that C2 team performance, while still limited by the human cognitive capacity, could be largely improved when they were assisted by cognitive agents capable of proactive information gathering/sharing and experience-based decision making.

Intelligence analysts need tools and techniques to help protect themselves from avoidable errors [15, 7]. Our experiments demonstrated that RPD-enabled agents can serve as one such tool to achieve reduced cognitive load, enhanced situation awareness, and positive human-agent collaboration. However, in real-world applications where information availability and credibility is concerned, it is expected that C2 teams with both the S2 and the S3 assisted by an R-CAST agent could outperform either of the two team structures considered in this paper, because S2 human's ability to reason about imperfect information can be fully exploited. This is worthwhile to consider in future studies.

7. ACKNOWLEDGMENTS

8. ADDITIONAL AUTHORS

Rashaad Jones (College of Information Sciences and Technology, Penn State University, email: `rjones@ist.psu.edu`); Timothy Hanratty (US Army Research Lab, Aberdeen Proving Ground, email: `hanratty@arl.army.mil`); and Laurel Allender (US Army Research Lab, Aberdeen Proving Ground, email: `lallende@arl.army.mil`).

9. REFERENCES

[1] J. Bradshaw, M. Sierhuis, A. Acquisti, Y. Gawdiak, D. Prescott, R. Jeffers, N. Suri, and R. van Hoof. What we can learn about human-agent teamwork from practice. In *Workshop on Teamwork and Coalition Formation at AAMAS'02*, Bologna, Italy, 2002.

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

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

[4] X. Fan, S. Sun, M. McNeese, and J. Yen. Extending the recognition-primed decision model to support human-agent collaboration. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 945–952. ACM Press, 2005.

[5] X. Fan, S. Sun, B. Sun, G. Airy, M. McNeese, and J. Yen. Collaborative RPD-enabled agents assisting the three-block challenge in C2CUT. In *Proceedings of BRIMS'05*, pages 113–123, 2005.

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

[7] R. J. Heuer. *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, 1999.

[8] G. A. Klein. Recognition-primed decisions. In W. B. Rouse, editor, *Advances in man-machine systems research*, volume 5, pages 47–92. JAI Press, 1989.

[9] G. A. Klein. Recognition-primed decision making. In *Sources of power: How people make decisions*, pages 15–30. MIT Press, 1998.

[10] G. Kuk, M. Arnold, and F. Ritter. Effects of light and heavy workload on air traffic tactical operations. *Ergonomics*, 42(9):1133–1148, 1999.

[11] T. L. Lennox, T. Payne, S. K. Hahn, M. Lewis, and K. Sycara. MokSAF: How should we support teamwork in human-agent teams? Technical Report CMU-RI-TR-99-31, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 1999.

[12] E. Norling. Folk psychology for human modelling: Extending the BDI paradigm. In *AAMAS '04: International Conference on Autonomous Agents and Multi Agent Systems*, pages 202–209, 2004.

[13] E. Norling, L. Sonenberg, and R. Ronnquist. Enhancing multi-agent based simulation with human-like decision making strategies. In S. Moss and P. Davidsson, editors, *Proceedings of the Second International Workshop on Multi-Agent Based Simulation*, pages 214–228, 2000.

[14] J. Sokolowski. Enhanced military decision modeling using a multiagent system approach. In *Proceedings of BRIMS'03*, pages 179–186, 2003.

[15] A. Tversky and D. Kahneman. Judgement under uncertainty: Heuristics and biases. *Science*, 185:1124–1131, September 1974.

[16] W. Warwick, S. McIlwaine, R. Hutton, and P. McDermott. Developing computational models of recognition-primed decision making. In *Proc. of the tenth conference on Computer Generated Forces*, 2001.

[17] C. D. Wickens. Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2):159–177, 2002.