# On Shared Situation Awareness for Supporting Human Decision-Making Teams

**Xiaocong Fan** and **Shuang Sun** and **John Yen**
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
{zfan,ssun,jyen}@ist.psu.edu

**Abstract**

One of the challenging issues in homeland security area is the early detection and successful processing of potential terrorist threats, which demands effective team collaboration. In this research, we investigate the way of incorporating naturalistic decision making models for supporting distributed team decision making. By extending Klein's Recognition-Primed Decision model, we propose a Collaborative RPD model ($C^2$RPD), which encourages proactive information seeking, linking, and sharing in distributed teamwork settings. This model establishes a basis for developing agent architectures that support both agent-agent and agent-human collaborations in developing shared situation awareness and in making decisions based on progressively refined recognitions.

## Motivation

The demand for team collaboration arises in various sectors of homeland security (HS). For instance, to enable early detection and successful processing of potential terrorist threats, anti-terrorist analysts often need to work in teams to quickly gather and make sense of information from multiple sources. However, teamwork in this area is often threatened by the fact that team members need to process voluminous amount of dynamically changing information under time pressure. Moreover, the information and knowledge resident within the broad scope of homeland security area are typically distributed across people, organizations, tools, and environments due to security concerns often associated with their roles and responsibilities. These unique and complex challenges in homeland security area can significantly hamper the quality and the timeliness of decision making, which can have extraordinary and possibly catastrophic consequences. Therefore, the objective of this research is to investigate ways to enhance the effectiveness of human decision-making teams in achieving shared situation awareness, and in making better decisions adaptive to the changing situations.

Both 'optimizing' (Nash 1950) and 'satisficing' (Simon 1955) strategies have been extensively studied and applied in solving practical decision making problems, and it is widely recognized that which strategy works better depends on the nature of the problems. More recently, there has been significant interest in applying decision-theoretic principles to build intelligent systems. For instance, work on Markov Decision Process (e.g., DEC-MDP, POMDP) has gained increasing attention in recent AI conferences (Shen, Lesser, & Carver 2003; Nair *et al.* 2004). There is no doubt that agents with sufficient computational resources can use the MDP approaches to help people make decisions on well-defined problems. On the other hand, researchers in the camp of naturalistic decision making take the opinion that when making decisions, people usually do not know the probabilities of all the choices; they even do not know all the possible options. It is argued that communities dealing with time stress tasks demand simulation systems with realistic (human-like) decision representation (Sokolowski 2002).

Aiming to support humans to make team decisions in dynamic complex domains, we focus on a specific naturalistic decision-making model—the Recognition-primed decision model (RPD)(Klein 1989; 1997)—for two major reasons. First, the RPD model has a clean decision making process that is particularly applicable for ill-structured problems with shifting or ill-defined goals in time stress domains. The RPD process can be extended to support multi-agent collaborations; this enables us to further investigate dynamic information sharing problems and distributed team cognition problems. Second, the RPD model focuses on recognizing the similarity between the current decision situation and previous experiences, which is claimed as the way how human experts make decisions in complex, dynamic environment. Implementing intelligent agents with a computational RPD can encourage close *agent-human collaboration* in the decision-making process (supporting adjustable autonomy). This advocates the view of human-centered teamwork (Sierhuis *et al.* 2003), where from humans' perspective, agents are no longer black-boxes providing decision making supports, but rather active peers whom humans can directly interact with. In short, the RPD model is chosen because it allows us to investigate agent architectures that support both agent-agent collaborations and agent-human collaborations while making decisions.

The remainder of this paper is organized as follows. In Section 2, after a brief review of the RPD decision-making model, we explore the potential opportunities in the RPD

process where team collaboration can be naturally introduced. In Section 3, we describe a collaborative RPD model–$C^2$RPD, focusing on how distributed agents work together to develop shared situation awareness, to collaboratively build stories through hypothesis exploration and experience synthesization, and to support adaptive decision making through expectancy monitoring. Section 4 explains how a human partner may collaborate with an RPD-agent in the RPD process, and Section 5 summarizes the paper.

## RPD And Collaboration Opportunities

In this section, after a brief review of Klein's RPD model, we argue for an RPD-based collaborative approach, and examine where agent-agent collaborations and human-agent collaborations can be incorporated into the RPD process.

### The RPD Model

The RPD model (Klein 1989) (see Figure 1) captures how domain experts make decisions based on the recognition of similarity between the current situation and past experiences. RPD has two phases: recognition and evaluation. In recognition phase, a decision maker needs to develop situation awareness and recognize what course of actions worked before in similar situations. In evaluation phase, a decision maker needs to carry out 'singular evaluation' by imaging how a course of actions will evolve. In case that a course of actions does not work for the current situation, the decision maker can either adjust the course of actions, or find and examine another one until a workable solution is obtained.

The RPD model states that 'feature-matching' and 'story-building' are two typical strategies used by experts to develop situation awareness. In feature-matching, a decision maker tries to find whether he/she has ever experienced situations similar to the current one by matching the set of observed cues (synthesized from information describing the current situation) with the pattern of cues considered in past experiences. In case that feature-matching cannot produce an adequate account for the current situation due to lack of experience, story-building will be used to construct an explanation, with the collection of observed information being coherently linked. A story gives an explanation of how the current situation might have been emerging. In story-building, a decision maker can explore potential hypotheses in his/her mind and evaluate how well each of them may fit the observed events.

The recognition phase has four products: relevant cues (what to pay attention to), plausible goals (which goals make sense), expectancy (what will happen next), and course of actions (what actions worked in this type of situation). An expectancy serves as a gate-condition for continuing working on the current recognition. Due to the dynamic and uncertain nature of the environment, it is very important to monitor the status of expectancy because a decision maker may have misinterpreted the current situation but he/she cannot recognize it until some expectancy is invalidated as the situation further evolves. In such cases, the decision maker needs to further diagnose the current situation (e.g., to gather more information).

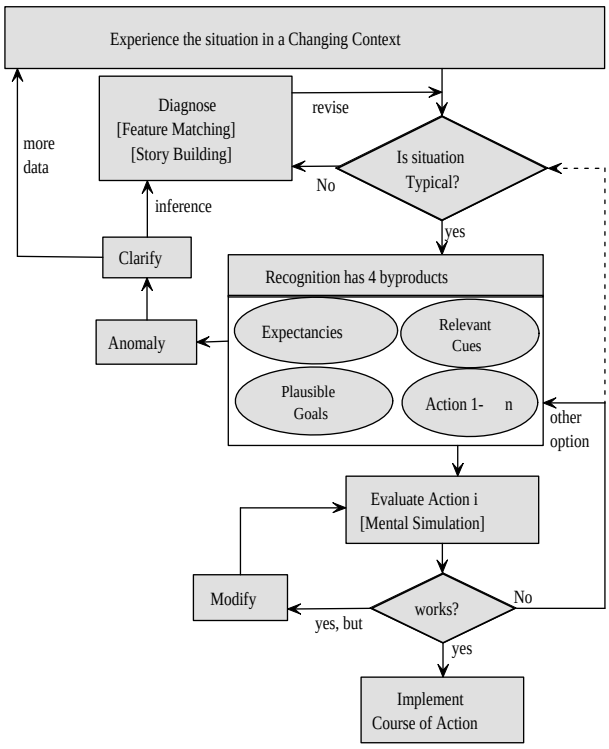


Figure 1: The RPD Model

### Collaboration Opportunities

The RPD model captures the cognitive activity undergoing in the mind of a decision maker when he/she faces a decision task. In essence, RPD is an individual process because it is within a decision maker's mental state. However, it becomes more interesting when a team of human experts, each making decisions using RPD, needs to work together in distributed dynamic environments. Intuitively, team performance can be considerably enhanced if the team can establish a shared mental model about the dynamic progress of the RPD process being pursued by the decision maker. Emphasizing the individual nature of the RPD process may weaken or ignore the active roles played by the other teammates in the process, expecially from the information seeking and sharing perspective.

On the other hand, as domain complexity increases, decision making often involves various kinds of expertise and experiences, which are typically distributed among a group of decision makers (Hollenbeck *et al.* 1997). In such cases, the timeliness and quality of decision making highly depend on the effectiveness of team wide collaboration (e.g., anticipating others' information needs, proactive sharing information and expertise).

Thus, in our study, we consider the situations where a group of people who are experts in different areas, each assisted by one RPD-enabled agent (i.e., agents capable of making decisions using RPD model), face the pressure to make better and faster decisions in an environment with high domain complexities. In such a setting, collaboration may

exhibit among RPD-agents, between an RPD-agent and its human partner, and among the human experts. Here, we focus on agent-agent and human-agent collaborations.

A careful scrutiny of the RPD model under teamwork settings reveals that potential agent-agent collaboration opportunities include:

- Situation Awareness: Each agent may only know (or be sensitive to) certain kinds of information. To develop a complete view of the global state, team members need to share their information;
- Feature Matching: In feature matching, an agent can monitor the information needs of the decision making agent, and proactively deliver relevant information;
- Story Building: When building a story, the agents can collaboratively explore potential hypotheses and progressively anticipate other' information needs;
- Expectancy Monitoring: All the agents keep an eye on the active expectancies and report anomalies to others whenever applicable.

Potential agent-human collaborations include:

- Situation Awareness: An agent shows its human partner the cue patterns being considered. The human partner can suggest the agent to consider new cues for the current decision situation;
- Feature Matching: An agent can show its human partner the result of feature-matching, e.g., the degree of similarity between each of the matched experience and the current situation. The human partner can adjust the matching strategies (e.g., thresholds), provide information that is still missing, suggest expectancies to be considered, adjust the pressure (the stress level of the current decision situation);
- Story Building: A human partner can suggest the agent to explore a specific hypothesis, help the agent to link the available information together. The agent can show its human partner the supporting or negative evidences regarding the explored hypotheses, show the stories being built, etc;
- Expectancy Monitoring: A human partner can input insights on how to handle the exceptions resulted from expectancy anomalies. The agent can show its human partner the potential side-effects caused by an expectancy anomaly.

## Collaborative RPD Model

In this section, we describe C$^2$RPD—a computational collaborative RPD model (as illustrated in Figure 2), which captures both agent-agent collaborations and agent-human collaborations. We mainly focus on how agents take the aforementioned opportunities to support iterative recognition and adaptive decision makings.

### Types of Decision-Making Tasks

Making decisions on what to do next is a task at an abstract level above domain tasks. Each decision-making task has certain knowledge (expertise) requirements on the decision maker. Two decision-making tasks belong to the same *type* if they place the same knowledge requirements. For example, Hollenbeck et al. (1997) described a scenario involving a four-person naval command and control team. The team needs to monitor the airspace in the vicinity of an aircraft carrier battle group, and to decide how to respond to the incoming air targets. This type of decision making requires two forms of expertise: (a) the ability to measure and translate raw attributes (e.g., speed, altitude, angle, IFF, etc) of an aircraft into internal opinions (cues) regarding how threatening the aircraft is, and (b) the rules specifying how to combine cue values to determine the level of threat.

In addition, decision making in human society is typically connected with societal hierarchies; a higher-level decision-making task usually depends on the results of lower-level decision-making tasks. In order to support multiple-level decision makings (Hollenbeck *et al.* 1995), in our model, we use the concept of "decision spaces" to organize experiences hierarchically according to decision types: experiences related to one decision type are maintained in one decision space or experience knowledge base (EKB).

Treated as such, decision making and information fusion can be tightly coupled at multiple levels, if we view the process of decision making as a process of fusing input information into decision results. This is also flexible for supporting teams where knowledge and expertise are distributed among team members. For instance, members of an anti-terrorist analyst team may have different access to various information sources due to security concerns or due to their roles and responsibility in the team. The team members can make decisions at different levels relative to their past experiences.

What is also important is that an agent can be involved in multiple decision tasks, which may or may not have relations. It is thus necessary for an RPD-agent to be able to effectively manage multiple attentions (RPD processes). For the flexibility of the C$^2$RPD model, we leave the issue of attention management open to the designers of agent architectures.

Now we come to the experience representation. Each experience has four parts: cues, goals, course of actions, and expectancies. An experience can be formally denoted as $e_i = \langle C_i, G_i, E_i, A_i \rangle$, where $C_i$, $G_i$, and $E_i$ are collections of predicates, and $A_i$ is a set of plan names, referring to predefined courses of actions. Let the collection of cues consider by a decision space $EKB$ be $C_{EKB} = \bigcup_{e_i \in EBK} C_i$.

In our model, no restriction is placed on the sources of decision tasks [1]. Responding to a decision task $t$, an RPD-agent will first follow the procedure below to initialize the RPD process:

1. From the current situation, determine the collection $F_t$ of features that are potentially relevant to the task;
2. Explore the decision space hierarchy to pick one that is applicable to the decision task. A decision space $EKB_j$ is chosen and fixed in the rest of the RPD process if $F_t$ can

[1] It could be a pre-specified step in a plan, could be a request from a human partner or other teammates, or could be dynamically identified by an agent itself.

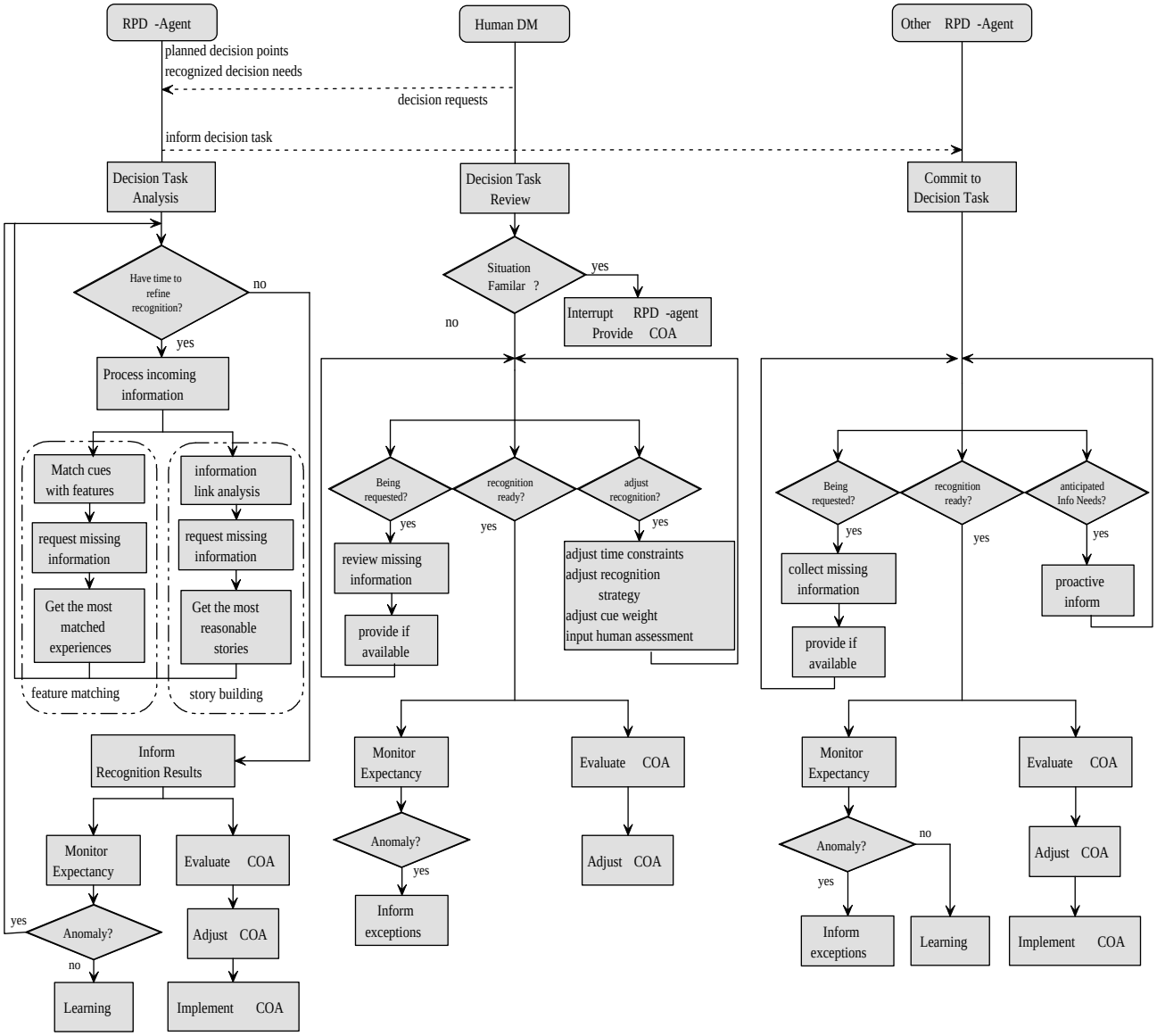


Figure 2: The C$^2$RPD Model

be covered by $C_{EKB_j}$. However, the determination of $F_t$ itself is a high-demanding cognitive task, and the selected decision space might not be the desired one. Thus, an internal mechanism of decision-space adaptation is highly needed so that the RPD-agent can recover from wrong selections of the decision space.

3. If there is no decision space applicable to $t$, i.e., the RPD-agent cannot handle the task due to lack of expertise, this agent has to find one competent RPD-agent and transfer the task to that agent.
4. The agent whoever makes an commitment to a decision task will then let others know so that the others can help in the rest of the decision making process. Those agents who also have the required decision-making knowledge (they thus know what is needed in making that decision) can proactively help on task $t$ and lower-level decision tasks, if any. Although those agents who do not have the required knowledge cannot easily anticipate the decision maker's needs, they can also provide help if they receive information requests for developing situation awareness.

Now we look at the organization of experiences within a decision space. We can formally define a refinement relation ($\sqsubseteq$) among experiences (recognitions). For any experiences $e_i = \langle C_i, G_i, E_i, A_i \rangle$ and $e_j = \langle C_j, G_j, E_j, A_j \rangle$ in a decision space, $e_i \sqsubseteq e_j$ iff $C_i \subseteq C_j$. This is a rather weaker requirement: an experience is a refinement of another if it considered more information (cues) in the recognition. A stronger refinement relation can be defined in terms of addi-

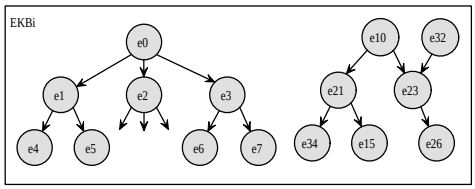


Figure 3: The structure of a decision space

tional relations between the other components. For instance, for some types of decisions, it may make sense to define: $e_i \sqsubseteq' e_j$ iff $C_i \subseteq C_j$, $G_i = G_{i+1}$, and $A_i \prec A_{i+1}$, where $\prec$ is sequence prefix relation: an experience is a refinement of another if it considered more information (cues) in the recognition, both share the same goals, and the course of actions associated with $e_i$ is simply the prefix of what is associated with $e_j$. From such a perspective, experiences in an EKB can be virtually viewed as being partitioned by some experience refinement relations. For the example illustrated in Figure 3, after experience $e_{10}$ being considered, experience $e_{23}$ (where both $e_{10} \sqsubseteq e_{23}$ and $e_{32} \sqsubseteq e_{23}$ hold) may be selected in the next round of recognition as more information becomes available. However, to elicit a meaningful experience-refinement relation from a decision space is domain-dependent, and the detailed discussion of it is out of the scope of this paper.

## Collaborative Situation Awareness

The term *feature* is used to refer to those environmental variables that describe elementary physical or mental parameters (e.g., velocity, moving direction, etc.) (Sokolowski 2003). These environmental variables together characterize the decision situation, and circumscribe the range of information to which the decision-making team should pay attention. Each team member may have limited sensing capabilities (e.g., types of sensors, sensing ranges) regarding the environmental variables, therefore the whole team has to exchange information in order to develop a shared awareness of the global situation. We assume each agent use an individual knowledge base (KB) to maintain all the information (acquired either from the environment or from other agents) regarding the current situation.

The term "cue" refers to an agent's internal representation of the decision situation. Cues are higher-level abstractions of the elementary data or synthesization of lower-level information. For example, internally an agent may only care the fuzzy category (e.g., high, medium, low) rather than the real value of an object's velocity; the "moving pattern" of an approaching unit can be synthesized (fused) from the information regarding the moving directions of all the individuals in the unit. Generally, a cue can be the root of several tree-like information-dependence structures, which describe the ways how the cue is abstracted from low-level information. We use $\{\tau(A)^i_c\}$ to denote the collection of information-dependence trees with cue $c$ as the root, relative to agent $A$'s expertise; use $T(A)'_c$ to denote the instantiation of $\tau(A)'_c$, where marked are the nodes of which the agent $A$ is lacking information, relative to $A$'s KB.

**Situation Awareness Driven By Relevant Cue Analysis** An agent can derive information requirements regarding the current decision task from the cues under its consideration. The following process can be used to collect missing information:

**Procedure** Investigate($task$, Team)
1. if ($self$ is the decision maker for $task$)
2. EKB = getDecisionSpace($task$);
3. Cues = getRelevantCues(EKB);
4. Repeat
5. $\{\langle c_k, T(self)_{c_k}\rangle\}$ = getCuesUnknown(Cues);
6. for each $\langle c_k, T(self)_{c_k}\rangle$
7. Peers = getProviders($\langle c_k, T(self)_{c_k}\rangle$);
8. Ask(Peers, $\langle c_k, T(self)_{c_k}\rangle$);
9. if ($\langle c_n, T(sender)_{c_n}\rangle$ received)
10. SynthesizeInfo(self.KB,$\langle c_n, T(sender)_{c_n}\rangle$);
11. Learning($\langle c_n, T(sender)_{c_n}\rangle$);
12. Until (Terminated)
13. else
14. if ($self$ has expertise for $task$)
15. EKB' = getDecisionSpace($task$);
16. Cues' = getRelevantCues(EKB');
17. Repeat
18. $\{\langle c_j, T(self)_{c_j}\rangle\}$ = getCuesAvailable(Cues');
19. Tell(task.decisionMaker(), $\{\langle c_j, T(self)_{c_j}\rangle\}$);
20. Until (Terminated)
21. else
22. Repeat
23. if (Being asked of $\langle c_k, T(sender)_{c_k}\rangle$)
24. m = getAvailableInfo($\langle c_k, T(sender)_{c_k}\rangle$);
25. Reply(sender, m);
26. Until (Terminated)

Figure 4: The algorithm for shared awareness

As is encoded in the algorithm in Figure 4, agents in a team may play different roles when investigating a situation.

- The decision maker agent (DM), while trying to synthesize the available information into appropriate cues, may not be able to attain a sufficient level of situation awareness due to lack of critical information. For every unknown cue, the DM can ask for help from some potential information providers. Upon receiving information from teammates, as well as synthesize the new acquired information to develop better situation awareness, the DM could also learn new information-dependence trees from teammates, and revise experiences in its EKB to incorporate additional cues being considered by other agents.
- Teammates who have the experiences pertinent to the current task can proactively tell the decision maker (DM) about the information relevant to the cues that need to be considered. Since an agent and the DM agent may have different EKBs and information-dependence structures, the teammate can (1) inform the DM agent about

the cues that have been synthesized in ways beyond the DM's existing expertise, (2) share experiences by informing the cues that are considered in its EKB, but may not in the DM agent's.

- Teammates who have no experience for the current task can help the DM agent upon being requested of certain information. Here, an teammate could reply the DM agent with information that is synthesized in a way different from the structure being considered by the DM agent.

It is worth noting that the information exchanged among RPD-agents is of form ⟨*cue, dependence-structure*⟩. This has two benefits. First, in doing so, the agents are not only requesting/sending information relevant to the cue, they also explicitly requesting/sending information that is *indirectly* related to the cue according to the accompanying dependence-structure. In essence, the dependence-structure establishes an information use context; an agent can help the information needer with all the information matched with the nodes of the dependence-structure that are marked as 'unknown' by the needer. Second, an agent may respond to an information request using a dependence structure different from the one being used by the requesting agent. Accordingly, the agents in a team can exchange and learn expertise about information-dependence structures from each other. This can progressively enable the agents to better anticipate others' information needs.

The investigation process is the key to evolving recognitions. It is an anytime algorithm; the decision making agent can trigger the feature-matching function at any point of the investigation process, as long as the agent has attained a satisfactory level of awareness of the current situation. Since the information regarding the current situation is recorded in the DM agent's KB, the feature matching process simply iterates over the experiences in the active EKB and casts queries to the DM agent's KB with the cues to be evaluated. The experiences with the most number of cues satisfied wrt. the DM agent's KB are returned as the recognition results.

## Collaborative Story Building

While feature matching is used to find past experiences that most match the current situation, story building is used when agents are experiencing unfamiliar situations. From another perspective, feature matching is to build a story for the current situation using 'templates' worked before, while story building is to construct a story from scratch.

Story building also involves information gathering, but it is more than cue-driven information investigation (Fig.4), because the agents are still unclear about what cues to investigate. Therefore, the key is to identify a collection of cues which the team needs to pay attention to. C$^2$RPD adopts a combination of two mechanisms: *hypothesis exploration* and *experience synthesization*.

A hypothesis could be an abstract story (e.g., there is a chance that City Y is facing some terroristic attack within 2 weeks), or a plausible intention (e.g., the object moving toward Building Z is probably hostile). To support collaborative hypothesis exploration, we employ the idea of multi-party conversation (Dignum & Vreeswijk 2004), where all

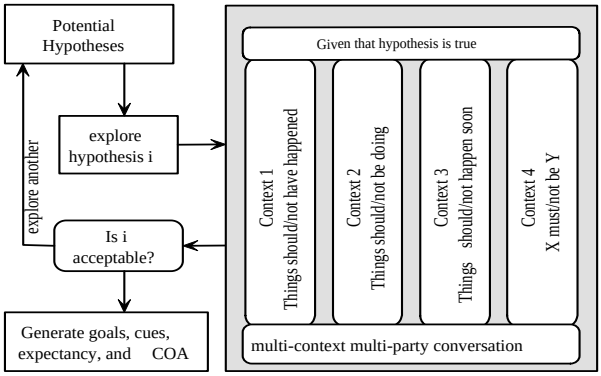


Figure 5: Hypothesis exploration in story building

the agents can hear the messages exchanged in a team, even though some are not necessarily the intended audiences. For instance, agent B can comment on agent C's messages directed to agent A. Figure 5 illustrates the process of hypothesis exploration. First, the decision maker agent (DM) can initiate a conversation to collect potential hypotheses that the team needs to further investigate. The team can use different strategies (e.g., agreed by most, proposed by a creditable agent, suggested by human partners, etc.) to determine the hypotheses that reflect the current situation best. To explore a hypothesis $h$, the team members are involved in conversations under four contexts: (1) given that $h$ is true, each agent, based on its expertise, infers what things should/not have happened, and proclaim to others for confirmation. A thing that should have happened but actually not, or the other way around, shows a negative evidence to $h$; (2) given that $h$ is true, each agent infers what things should/not be doing now, and proclaim to others for confirmation. A thing that should be doing but actually not, or the other way around, shows a negative evidence to $h$; (3) given that $h$ is true, each agent infers what things should/not happen later, and proclaim to others for validation. A thing that should happen within a minute but actually not as expected, or the other way around, shows a negative evidence to $h$; (4) given that $h$ is true, each agent draws reasonable conclusions of form "X must/not be Y" (e.g., the unidentified object must be a neutral force; City P must have an airport) and proclaim to others for verification. In case that X must be Y but actually not, or the other way around, shows a negative evidence to $h$. Note that all the above four kinds of claims could be associated with a value that indicates the claiming agent's confidence degree.

To evaluate a hypothesis, the DM agent needs to consider both the positive and negative evidences collected in the multi-party conversation, and to reconcile conflicting conclusions drawn by different team members, if any. Certain human-adjustable thresholds can be used to judge whether the hypothesis applicable to the current situation. If not, the agents continue to explore another hypothesis until find an acceptable one. If yes, the DM agent can then generate a *pseudo-experience* in the following way: (1) use the proven

hypothesis to produce goals, the fulfillment of which could prevent the situation from evolving toward undesired direction (e.g., to invalidate a hostile attempt, to protect a vulnerable place); (2) use the information gathered in context 1, 2, and 4 to produce cues; (3) use the information gathered in context 3 and 4 to produce expectancies; and (4) use the information gathered in context 2 and 3 to construct a course of actions (COA). Once it is proved that the COA works, the pseudo-experience will be stored as a valid experience for dealing with ensuing similar situations. Agents can also learn from unworkable pseudo-experiences so that they would not make the same mistakes when similar situations occur.

Experience synthesization is used when two or more past experiences can be combined together to deal with an unusual situation. Synthesization could be based on experiences from single or multiple agents, and the four components of a synthesized experience are typically the coupling of the corresponding parts of the input experiences. From another perspective, novel situations offer opportunities for agents to share and reflect on their experiences.

### Collaborative Expectancy Monitoring

After an RPD-agent makes a recognition, it will continuously monitor the associated expectancies until the completion of the selected course of actions. Expectancy monitoring is one of the key features to support adaptive decision makings (Serfaty, Entin, & Johnston 1998). A DM agent can subscribe information relevant to the expectancies that need to be continuously monitored. Such collaborative expectancy monitoring can take full advantage of the team's distributed cognition, so that the DM agent can terminate the actions resulted from a wrong recognition at the earliest opportunity.

First, expectancies can be used to initiate a complete new decision. An expectancy states what will happen, serving as a gate-condition for keeping following the current recognition. Some expectancies may be so *crucial* that whenever they conflict with the new observed facts, it indicates the decision maker has heavily misinterpreted the current situation. In such cases, the RPD-agent has to diagnose the current recognition, re-considering the *whole* space of the active EKB for another time.

Second, RPD-agents can use expectancies to *refine* a decision, leveraging some structures within the active EKB. The invalidation of some expectancies may indicate that the once workable recognition is no longer applicable to the changing situation. The already executed part of the selected course of actions may still make sense, but the rest part has to be adjusted. In such cases, the RPD-agent can start another round of recognition, using the experience refinement relation described earlier to develop a better solution.

Therefore, the computational $C^2$RPD model is an iterative model. It explicitly incorporates the idea of "recognition refinement", and supports situation reconsideration during action execution phase. In addition, the computational model is much flexible for studying the time pressure issue in decision making. Since RPD-agent can make a sequence of decisions (the length of the sequence is restricted by the external time pressure), with one decision refining the preceding ones, it can always return an acceptable decision relative to the timing constraints. This guarantees no critical decision is missed under stress situations.

## Human-Agent Collaboration

As shown in Figure 2, a human may collaborate with his/her partner agent in various ways along the RPD process. On the one hand, as an assistant to its human decision maker, an agent needs to help the human partner develop situation awareness, considering the limitation of human's cognitive capacity; needs to help the human understand the progress of its internal RPD process (e.g., displaying the progress in a user-friendly flow chart); needs to request heuristic information from the human without imposing too much interruption; needs to recommend workable solutions to the human, clarifying what's the impact to other teammates if one option is chosen.

On the other hand, RPD-agents, leveraging humans' meta-level reasoning, are reflective and self-adaptable. First, a human can direct the process of recognition refinement by suggesting/revising the experience refinement relations in the current decision space. Agents can thus take humans' meta-cognition into consideration in the next round of recognition. This can also stepwisely help people elicit the tacit expertise that may reside solely in their minds, and transform it into explicit knowledge that can be effectively used by agents in later decision situations.

Second, a human can guide an RPD-agent in its information gathering and sharing by suggesting new or revising the existing information dependence structures. The new or revised dependence structures, which reflect the human's dynamic adaptation, will largely affect the patterns of information exchange among team members.

Third, human's input is critical for effective story building. (a) An RPD-agent's exploration strategy is governed by its human partner. A human can initiate, suspend, or terminate the exploration of a specific hypothesis, considering all the supporting and negative evidences so far collected by the RPD-agent. (b) In order to orchestrate the information from distributed sources into a coherent form, the key is to discover the tacit information linkages, which can be gracefully handled by human. The information linkages suggested by human can be further generalized to create new, or incorporate into the existing, information dependence structures. (c) To finalize the story being built, an RPD-agent has to ask its human partner for approval regarding the four components of the pseudo-experience. Once a proven pseudo-experience is appended into an RPD-agent's experience KB, human's insights have actually been encoded as reusable knowledge for handling similar situations in the future.

Fourth, human can also help in monitoring the development of certain expectancies, in adjusting the expectancies that are no longer applicable, in proposing crucial expectancies that have been neglected before, and in suggesting ways to handle exceptions accompanied by the violation of an expectancy. An RPD-agent can benefit from such human-agent interactions in two ways. (a) Based on human's input during expectancy monitoring, the experience (say, $e_i$) selected for

handling the current situation can be evolved into a better format ($e_i'$). Alternatively, the agent can alter the structure of the active decision space by creating a new experience $e_{i+1}$ that refines $e_i$. (b) An agent can learn from human's response to exceptions, and incrementally gain experiences for handling similar exceptions.

Fifth, the recognition phase may result in several solutions (COAs). To find a workable COA, the DM agent needs to consider multiple factors such as potential resource conflicts, timing conflicts, and side effects upon failure. In such a case, human's intuition about the priorities of the factors can facilitate the agent to make a better and faster selection.

## Summary

There has been much theory and research presented that investigates team cognition, naturalistic decision making, and collaborative technology as it relates to real world, complex domains of practice. However, there has been very little work in combining intelligent agent technology and naturalistic decision-making models to support distributed team decision making. In this paper, we described the collaborative RPD model ($C^2$RPD), which extends Klein's Recognition-Primed Decision model, leveraging both agent-agent collaborations and agent-human collaborations during the decision-making process. This model encourages proactive information seeking and sharing in distributed teamwork settings, thus can be incorporated into cognitive agent architectures to support distributed team cognition and decision making. R-CAST (Yen *et al.* 2004), which extends the CAST agent architecture (Yen *et al.* 2001), is one system that has realized the $C^2$RPD model.

$C^2$RPD is an abstract model that naturally combines naturalistic decision making with computational intelligence. It can be used to develop agent systems for enhancing the capabilities of anti-terrorist analysts in early detection of potential terrorist threats. $C^2$RPD supports adjustable autonomy: agents, together with human partners, can collaboratively monitor expectancies and progressively refine recognitions. $C^2$RPD is also a self-evolving model, encouraging the learning of novel experiences and strategies for handling exceptions from human partners.

## Acknowledgement

This research has been supported by AFOSR MURI grant No. F49620-00-1-0326.

## References

Dignum, F., and Vreeswijk, G. 2004. Towards a testbed for multi-party dialogues. In Dignum, F., ed., *LNCS-2922: Workshop on Agent Communication Languages*, 212–230.

Hollenbeck, J. R.; Ilgen, D. R.; Sego, D. J.; Hedlund, J.; Major, D. A.; and Phillips, J. 1995. Multilevel theory of team decision making: Decision performance in teams incorporating distributed expertise. *J. Appl. Psychol.* 80(2):292–316.

Hollenbeck, J. R.; Major, D. A.; Sego, D. J.; Hedlund, J.; Ilgen, D. R.; and Phillips, J. 1997. Team decision-making accuracy under difficult conditions: construct validation of potential manipulations using the TIDE$^2$ simulation. In Brannick, M. T.; Salas, E.; and Prince, C., eds., *Team performancs assessment and measurement*. 111–136.

Klein, G. A. 1989. Recognition-primed decisions. *Advances in man-machine systems research (Ed: W. B. Rouse)* 5:47–92.

Klein, G. A. 1997. The recognition-primed decision (RPD) model: Looking back, looking forward. *Naturalistic decision making (Eds: C. E. Zsambok and G. Klein)* 285–292.

Nair, R.; Roth, M.; Yokoo, M.; and Tambe, M. 2004. Communication for Improving Policy Computation in Distributed POMDPs. In *Proceedings of The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 1098–1105. ACM Press.

Nash, J. F. 1950. The bargaining problem. *Econometrica* 18(2):155–162.

Serfaty, D.; Entin, E.; and Johnston, J. 1998. Team coordination training. In Cannon-Bowers, J., and Salas, E., eds., *Making decisions under stress: implications for training and simulation*. APA Press. 221–245.

Shen, J.; Lesser, V.; and Carver, N. 2003. Minimizing Communication Cost in a Distributed Bayesian Network using a Decentralized MDP. In *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, 678–685. Melbourne, AUS: ACM Press.

Sierhuis, M.; Bradshaw, J. M.; Acquisti, A.; van Hoof, R.; Jeffers, R.; and Uszok, A. 2003. Human-agent teamwork and adjustable autonomy in practice. In *7th International Symposium on Artificial Intelligence (I-SAIRAS)*.

Simon, H. 1955. A behavioral model of rational choice. *Quarterly Journal of Economics* 69:99–118.

Sokolowski, J. 2002. Can a composite agent be used to implement a recognition-primed decision model. In *Proceedings of the Eleventh Conference on Computer Generated Forces and Behavioral Representation*, 431–436.

Sokolowski, J. 2003. Representing knowledge and experience in RPDAgent. In *Proceedings of the Twelfth Conference on Behavior Representation in Modeling and Simulation*, 419–422.

Yen, J.; Yin, J.; Ioerger, T.; Miller, M.; Xu, D.; and Volz, R. 2001. CAST: Collaborative agents for simulating teamworks. In *Proceedings of IJCAI'2001*, 1135–1142.

Yen, J.; Fan, X.; Sun, S.; McNeese, M.; and Hall, D. 2004. Supporting anti-terrorist analyst teams using agents with shared RPD process. In *IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, 53–60.