

Determining Information Technology Project Status using Recognition-primed Decision-making enabled Collaborative Agents for Simulating Teamwork (R-CAST)

Anthony Barnes
abarne1@towson.edu

Robert J. Hammell, II
rhammell@towson.edu
Computer and Information Sciences Department
Towson University
Towson, MD 21252 USA

Abstract

In the project management environment, it is critical to know how the project is performing from the viewpoint of scope, schedule, cost, and other constraints. One source reports that 70% of IT-related projects do not meet their objective (Lewis, 2008). This paper examines the use of the R-CAST architecture to develop a collaborative agent system that aids in recommending the status of a project using color indicators (red, yellow, green) that are based on the progress and standing of the project-related constraints.

Key Words: Project management, R-CAST, Recognition-primed Decision (RPD) Model, human-centered teamwork (HCT), knowledge base (KB), experience knowledge base (EKB)

1. INTRODUCTION

Project management is the discipline that guides the execution of projects with the goal of producing a quality product to customer satisfaction. A project is defined as a temporary endeavor to produce a unique product, service, or result. Project management is used to control the project's activities by applying knowledge skills, tools, and techniques to meet project requirements. (PMI, 2004) The boundaries of producing a quality product are limited due to its associated project constraints of *scope*, the amount of work that will be completed; *schedule*, the amount of time needed to complete the scope; and *cost*, the amount of money budgeted for the project

including all direct and indirect expenses (Snedaker, 2005).

A Project Manager assumes the lead role in managing a project. Managing a project includes identifying requirements; establishing clear and achievable objectives; balancing the competing demands for quality, scope, time, and cost; and adapting the specifications, plans, and approach to the different concerns and expectations of the various stakeholders. Projects are sometimes divided into lifecycles that help with scheduling, managing and controlling them. (PMI, 2004; Kerzner, 2006)

Even with today's technology and modern processes and practices, negotiation of the

many steps involved with successfully managing a large project is extremely difficult. Information technology projects seem to be particularly difficult to manage; it has been reported that 70% of all IT-related projects do not meet their objectives (Lewis, 2008). Maintaining visibility of the status for large, complex projects is of major importance (Burgess, 2003). When judging the project status of an information technology project, it is important for the project manager to consider input from the project team and stakeholders. The input from these experts is invaluable due to their knowledge and experience in certain situational contexts.

There are many variables that factor into the decision for project status. These include the input from the team experts; information regarding the project's constraints of scope, schedule, cost and performance; and other environmental contexts. Accounting for all of these variables increases the complexity of the decision facing the project manager. According to Felsen (1976), the human mind cannot perform a successful decision analysis in situations that are characterized by a large number of decision variables. It has also been discovered that if a decision maker is given more than four facts then the quality and speed of his decisions are reduced (Hayes, 1962). The project manager can employ decision support system technology to assist in the decision-making process.

A decision support system (DSS) is a computer technology solution that is used as an aid in complex decision-making and problem solving (Shim, 2002). DSS are beneficial to decision makers by providing more information that will enable well educated decision-making.

Over the past three decades, DSS have taken on both a narrower and broader definition through the evolution of research that resulted in the development of additional concepts and views. The evolution of the research has allowed the emergence of DSS to assist specific types of decision-makers faced with specific kinds of problems such as group decision support systems (GDSS) and executive information systems (EIS). Research that is characteristic of DSS

often focuses on how information technology can improve the efficiency with which a user makes a decision and can improve the effectiveness of that decision. (Shim, 2002; Carlsson, 2002) As an illustration of the notion of the emergence of systems developed to assist specific types of decision makers, we introduce the Recognition-primed Decision-making enabled Collaborative Agents for Simulating Teamwork (R-CAST) architecture.

R-CAST, developed at The Pennsylvania State University, is a "team-oriented cognitive agent architecture built on top of the concept of shared mental models (SMM) in team cognition, the theory of proactive information delivery, and Klein's Recognition-Primed naturalistic decision model" (Fan, 2007). According to Prindible (2006), R-CAST was developed "to study (a) high-level cognitive behaviors such as adaptive decision-making and planning, and (b) team behaviors such as collaboration and information sharing."

In this work, we are concerned with using the R-CAST architecture to develop a collaborative agent system that will recommend the status of a project. More specifically, we intend to design, develop and demonstrate a system using the R-CAST architecture that can make a relevant and accurate recommendation of project status (red, yellow, or green) after reasoning over a knowledge and experience knowledge base that is related to the context of a situation in a project management environment. The goals of the work presented in this paper are to: investigate the R-CAST architecture to determine its capability as an aid for recommending the status of a project, perform initial knowledge acquisition activities with subject matter experts, and to evaluate the knowledge acquired from the experts.

The remainder of the paper is outlined as follows: background of R-CAST architecture; an example of the R-CAST architecture with project management knowledge; an explanation of the knowledge acquisition process used in this research to acquire information for the knowledge and experience base; and a conclusion and future work section.

II. R-CAST Architecture

R-CAST originates from research conducted on human-centered teamwork (HCT). HCT, where human and software agents interact as peers, focuses on using a collaborative approach to establish situation awareness, developing shared mental models for evolving situations, and adapting to mixed-initiative activities. The idea is that humans could have the capability to pay more attention to vital activities and make better decisions using information that is more precise and of finer granularity if software agents could effectively collaborate with human peers. (Fan, 2006)

R-CAST is an attempt at producing a successful agent architecture for HCT by incorporating three principles identified as necessary to accomplish such a task. The three principles described in (Fan, 2007) are:

- **Architectural Flexibility:** involves accommodating teamwork in a variety of domains by offering flexibility to conquer domain-specificity. This principle is important when it is required to form human-agent teams under timing and resource constraints within complex team organizations.
- **Teamwork Adaptability:** uses the shared mental model concept as one of the key components of team-oriented agent architecture. The notion of computational SMM is critical to teamwork adaptability both in coordinating planned team activities and in responding to teamwork failures. An agent architecture with a functional SMM that has broader scope and a richer structure requires stronger supports for team adaptability where the development of human-centered collaborative systems, involving both human-agent and agent-agent collaborations, are concerned.
- **Context Reasoning Capability:** the concept of context is becoming more important in building computational systems that are highly responsive to human needs. According to (Fan, 2007), the R-CAST developers assert that there are only weak or no explicit supports for context in the implemented agent architectures for reasons of vague

scope and tight connections with other functional components. They take the position that teamwork context should be at a level higher than SMM as it governs an agent's attention to the very portion of SMM that is relevant to the current activities so they made sure to include this capability in R-CAST.

R-CAST Components

The key components of R-CAST, shown in Figure 1, are explained in (Fan, 2006) as follows. In the "RPD - based Decision Making" module domain knowledge, past experiences, and the current situation awareness are used to produce a new, or adapt to, an existing decision. The "Teamwork manager" and "Taskwork manager" modules are used to coordinate the fulfillment of a decision that involves inter-agent and intra-agent activities, respectively. Another round of decision-making is triggered in the event that any unexpected condition develops from situational changes or action failures. Also, any behavioral (plan) adaptations that have been tested as workable will be used to adapt the relevant experiences.

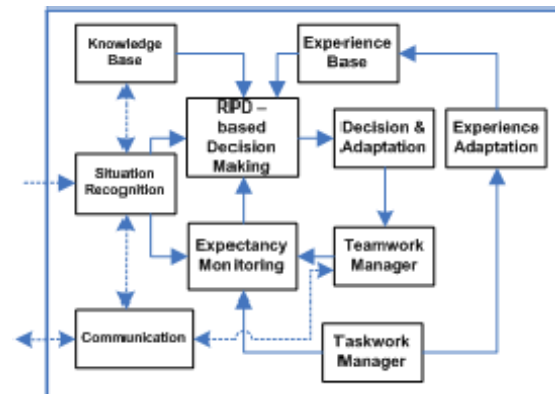


Figure 1. R-CAST Components (Fan, 2006)

In (Fan, 2006), R-CAST is described as a configurable architecture where many of the components can be replaced. Using R-CAST to develop a multi-agent system involves two major steps if the default R-CAST configuration is preserved:

1. Elicit domain knowledge and implement a domain adapter for the problem studied, and;

2. Elicit and represent domain expertise about decision making in terms of experiences.

Once the R-CAST architecture is understood, it is these two steps that will be employed to develop a collaborative agent system that aids in recommending the status of a project using color indicators (red, yellow, green) that are based on the progress and standing of the project-related constraints.

R-CAST Characteristics

The R-CAST architecture has three noteworthy features: it is cognitively inspired, it supports context awareness, and it provides a framework for agents to work as teams (Fan, 2007). Each of these features will be explored in some detail. Note that the majority of the presentation in this section comes from (Fan, 2007) and (Fan, 2005a) except where otherwise noted.

Cognitively Inspired Architecture: The Recognition-primed Decision (RPD) Model is a naturalistic decision-making model that offers a logical and rational process for the improved solving of ill-structured problems where there is no time for extensive reasoning. Dynamic information sharing problems and distributed team cognition problems can be further investigated by embedding team-wide collaboration opportunities into the RPD process. When making decisions in a dynamic environment, experts focus on recognizing the similarity between the current decision and previous decision experiences, a process that is captured in and is a primary focus of the RPD model.

RPD has two phases: recognition and evaluation. In the recognition phase, an agent is challenged with developing situation awareness and recognizing the course of actions that make sense for the situation. In the evaluation phase, an agent individually evaluates the course of action by imagining how it will evolve. In the case that a course of action does not work for the current situation, the agent can reject it and examine other alternatives to find a reasonable solution, or adjust the selected course of action.

The possibility exists that the decision maker could misinterpret the situation due to the dynamic and uncertain nature of the environment. Therefore, the decision maker monitors the status of expectancy so that the situation can be further diagnosed. From this perspective, the RPD model stresses on Simon's "satisficing" criterion, where the selected solution attempts to be adequate as opposed to optimal (Simon, 1955).

An R-CAST agent may start a new iteration of the recognition phase when the expectancy for a situation becomes invalid. *Expectancy* examines what will happen, serving as a gate condition for retaining the current recognition and is monitored until the selected course of action is completed.

The RPD model provided by R-CAST can also be described as a "collaborative RPD" process since an R-CAST agent has the ability to behave as a human's partner and/or team up with other R-CAST agents. Fundamentally, RPD is an individual process because it is within the decision maker's mental model. However, in a collaborative approach, if team members know RPD then they can establish a SMM about the dynamic RPD process and pursue opportunities to contribute to decision-making by seeking and sharing information. Collaboration becomes more necessary and typical as domain complexity increases and time pressures are involved.

Supporting Context Awareness:

Context awareness focuses on the manner in which the context is represented. Difficulties in context representation result from it not only having to support identification of relevant decision-making information, but also from the need to support dynamic context changes (emulating how humans adapt decisions in a fluid environment).

RPD provides the *decision-making process context* within R-CAST. To assist in the reuse of domain knowledge used in decision-making, context related to this knowledge is organized into two separate but related representations: *hierarchical experience context* and *inference context*. The decision process, hierarchical experience, and inference contexts combined enables R-CAST to incorporate various contexts for

identifying information relevant to decision making, for adapting decisions to a dynamic environment, and for facilitating reuse of context-related domain knowledge.

The decision process context provides the human peer a view of the progress of the current decision-making activity, and also governs the selection of the appropriate inference context and experience context.

The hierarchical experience context captures domain-specific experience used by the RPD model. Experiences are organized in a tree-like structure to support multi-level decision making and to facilitate reuse.

The inference context, which describes the context/situation for drawing various kinds of inference, relates high-level information needs to low-level information while identifying missing relevant information. It is also used to aggregate lower-level information to higher-level features.

Structure for teamwork: Teams of agents are initially equipped with a profile of team processes. The issue of a partial teamwork process is addressed in R-CAST by allowing a team to dynamically negotiate on "who works on what" when the team proceeds to a point where the agent-role (task) mapping is still incomplete.

Managing inter-agent communication is also a vital part of the R-CAST architecture because an agent is required to inform other teammates of its task progress at critical points. The agents' capability to exchange information regarding their teamwork progress allows them to attain a global picture of the dynamic progress of the committed team activity, thus supporting shared situation awareness.

Integration of Team Intelligence

The capability for team intelligence in R-CAST is provided by integrating the three principles (architectural flexibility, teamwork adaptability, and context reasoning) with the three characteristics just described (cognitively inspired, context aware, and teamwork ready) (Fan, 2007). This integration is achieved by incorporating these six components in the R-CAST architecture implementation.

As identified in (Fan, 2007), there are at least two benefits to integrating team intelligence within R-CAST. First, R-CAST provides a flexible solution to developing decision aids for supporting human decision-making teams under time stress (Fan, 2007). Second, R-CAST can play the role of a virtual teammate to human team members in information and knowledge intensive domains (Fan, 2007). The next section highlights these benefits.

Applications of R-CAST

This section provides brief explanations of examples of the application of R-CAST and how the architecture provides the benefits associated with: 1) integrating team intelligence, 2) the ability to provide a flexible solution to develop decision aids for supporting human decision making under time pressure, and 3) the ability to act as a virtual teammate to human members in rigorous information and knowledge exhaustive domains.

Command and Control (C2) Teams and R-CAST: The capability of R-CAST as a decision aid was evaluated in a battlefield environment simulation (Fan, 2007, 2005b). The components of this study were comprised of a DDD (Distributed Dynamic Decision-making) environment as the testbed, and scenarios that were designed to involve a blue (friendly) force consisting of three battle functional areas (BFAs): the intelligence cell (S2), the operations cell (S3), and the logistics cell (S4).

An experiment was performed to find out whether R-CAST could improve the performance of Command and Control (C2) teams when a time constraint is present (Fan, 2007). In the experiment two teams were formed. One type of team, called H, consisting of only human subjects playing the roles of S2, S3, and S4; and another team called HA, that was made up of 3 R-CAST agents playing the roles of S2, S3, and S4, where the S3 agent was paired with a human subject. The HA team had the capability of human-agent and agent-agent collaborations. (Fan, 2007)

From the experiment, it was learned that when the time pressure was minimal, both

teams were able to identify and destroy all enemies. However, when the time pressure became more intense the performance of the HA team increased while the performance of the H team decreased. The result confirmed that R-CAST agents, when used as decision aids, can be used to enhance the performance of C2 teams working under varying time pressure in handling threats (Fan, 2007, 2005b).

3-Block Challenge and R-CAST:

The 3-block challenge study was conducted to assess whether the context-awareness feature of R-CAST is a suitable aid to C2 teams that make real-time decisions that include multiple dynamic contexts (Fan, 2007). The study also used roles of the S2, S3, and S4 components described in the previous example. Using these components, C2 teams were assumed to function in complex urban terrain, reacting to possible threats related with crowds, IEDs (Improvised Explosive Device), and key insurgents (Fan, 2007). The study was hailed as the '3-block challenge' because within the boundary of a three block area in a city, officers in command must react to a steady flow of intelligence reports and make well-timed decisions for three kinds of tasks including combat, peace keeping, and humanitarian missions (Fan, 2007, 2005c).

For this experiment, using the H team (Human) and HA team (Human-Agent) defined in the previous example as a foundation, two types of C2 teams were formed from 30 subjects recruited from a US Army ROTC (Reserve Officer Training Corps) Organization at random. One type of team consisted of S2H-S3H where each team has a subject playing S2 and a subject playing S3. The other type of team consisted of S2A-S3HA groups where each has an R-CAST agent playing S2, and a human together with an R-CAST agent playing S3. A 2x3x3 factorial design involving 3 factors (team type, context switching frequency (CSF), and task complexity) was used as a design scheme. The CSF and task complexity factors each have three levels that varied the frequency of keeping a C2 team up to date with field information and the number of thriving hazards that posed threats in the field, respectively. (Fan, 2007)

The researchers observed that the performance dropped for both HH and HA teams as the context switching frequency increased. The performance decline continued to intensify for both teams under more time-stressed situations. However, the performance of HA teams at each CSF level was considerably better than the HH teams. When it came to performing secondary tasks, it was observed that HA teams recognized significantly more threats than HH teams. (Fan, 2007)

The result of this experiment illustrates that the R-CAST agents, as both teammates (S2) and decision aids (S3), can play an essential role in easing the impact of a human's cognitive aptitude on the performance of decision-making tasks that engage multiple contexts (Fan, 2007, 2006).

3. R-CAST Project Management Example

In this section, a project management-related simulation of the R-CAST architecture is presented. Following the mode of the examples in the last section, an example of the R-CAST system architecture populated with project management knowledge is presented. A tutorial for R-CAST (Prindible, 2006) that was developed by the researchers at The Pennsylvania State University was used as a guide for developing the simulation.

Example Scenario

For this simulation, the following scenario was created to set parameters of the knowledge needed to populate the knowledge base and experience knowledge base with specific project management domain knowledge:

While performing a project, the status of the project can be determined at a point based on the condition of its factors. The conditions determine the project status. The status of the project can be Green, Yellow, or Red. Using the factors of schedule and cost to determine the status, a project with a 'Green' status has a high performing schedule and an actual budget that is performing

under the planned budget. A project with a 'Yellow' status has a schedule that is performing to mediocre standards but is still under budget. A project that has a 'Red' status has a schedule that is low performing and is over budget.

R-CAST Syntax

As explained in Prindible (2006), it is necessary to become familiar with some of the syntax associated with R-CAST. The fundamental R-CAST syntax is defined as follows:

- Variables: denoted by a '?' (?position, ?condition)
- Symbols: name of each variable, quotes or underscores can be used to join multiple words ("Grand Canyon", grand_canyon)
- Comments: denoted on each line by a ';' or '#'
- Functions: standard arithmetic and string comparison operators (+, -, *, /, rand, dis, =, eq, <, <=, >, >=)

The following describes the active knowledge base of R-CAST and how knowledge is transferred into it for this simulation.

Active Knowledge Base (KB)

The objective of the active knowledge base is to clearly characterize knowledge through its information dependency relations on two levels. The first level, called the *FactType*, is a schema that represents knowledge at an abstract level to provide a comprehensive description of the agent's environment. The second level of representation, which is called a *Fact*, can be defined as an instance of a *FactType* when using the analogy of an object oriented programming language. Declarative knowledge can be populated into the agent's knowledge base by combining the two aforementioned levels. Rules, which are considered optional, can also be declared within the knowledge base to apply constraints or define meanings within the agent's environment. An example of a *FactType*, *Rule*, and *Fact* used for this example scenario is provided. (Prindible, 2006)

FactType: The *FactTypes* of an agent's environment must be defined before facts can be asserted into the knowledge base (Prindible, 2006). The following is an example of a *FactType* defined in the simulation example:

(FactType schedule (?status ?sched))

For the *FactType* schedule, the variables *status* and *sched* are used to define the project status where *status* denotes the color indicator (Red, Yellow, or Green) of the schedule with respect to the schedule's performance denoted by the variable *sched* with the values of low, med, high.

Rules: According to Prindible (2006), depending on the situation, rules within an agent's KB are not always required. In this example, rules are required to define the instances when the schedule is performing at a high, medium, or low level. A rule consists of a rule name, which must be unique; an antecedent, the conditional statement; an "antecedent-consequence separator (->), which denotes "therefore"; and a consequent state (Prindible, 2006). The following is an example of a rule used in the simulation to determine a high performing schedule:

```
(Rule "determineHighPerfSched" # Rule def
  (daysVar ?daysVar) # Antecedent
  (< ?daysVar 3)
  -> # Antecedent-
  # consequence
  # separator
  (schedule green high) # Consequent
  # state
)
```

In this rule named *determineHighPerfSched*, the logic and conditions necessary are present for the agent to make decisions about whether a project schedule is performing at a high, medium, or low level. A variable *daysVar* is defined to indicate the number of days a project is behind schedule. This stored variable is compared to determine if the schedule is less than 3 days off schedule. If it is true, the variables *?status* and *?sched* in an instance of schedule will be set to 'green' and 'high', respectively. The agent can generate a series of implied facts by combining rules and facts (Prindible, 2006).

Facts: Facts are defined as instances of predefined FactTypes. Defining a fact permits each variable or parameter in the FactType to be assigned an explicit value (Prindible, 2006). The following provides an example of the definition of the daysVar variable, which indicates that a project is 2 days behind schedule:

(Fact daysVar (2))

Now that the knowledge base is defined, it can be matched with the experience knowledge base. The RPD experience base and how knowledge is transferred into it for this simulation is described in the ensuing section.

RPD Experience Knowledge Base (EKB)

According to Prindible (2006), the purpose of the RPD EKB module is to have a reasonable abstraction level where this specific type of decision-making can be modeled, and to recognize the information requirements during decision-making. The agent can cue a course of action based upon existing conditions and previous experiences through the RPD component. The organization of the cues are similar to the configuration of the KB, except that the EKB contains a hierarchical experience space of experiences and their cues (episodic memory), expectancies, anomalies, goals, actions, and results; instead of knowledge and explicit facts (declarative knowledge) contained in the KB. The experience is organized on the two perspectives of experience space, where the agent uses the *experience space* to organize experiences that are at a higher level to calculate the degree of a match (recognition), and *experiences* that are found in the definition of the experiences in the experience space.

ExperienceSpaces: The *ExperienceSpace* is comprised of the unique ExperienceSpace definition; a cue, the specific condition in the agent's knowledge base that extracts a past experience; an expectancy, conditions the agents should anticipate in the present situation; anomalies, certain conditions that could result in a divergence from expectancies; and experiences, used to connect specific experiences to other ExperienceSpaces (Prindible, 2006). An example of an

ExperienceSpace used for the simulation to get the project status is defined below.

```
(ExperienceSpace getProjStatus
  (Cue (getProjStatus yes))
  (Expectancy (getProjStatus yes))
  (Anomaly (getProjStatus no))
  (Experience (statusGreen) (statusYellow)
    (statusRed))
)
```

The getProjStatus ExperienceSpace becomes active when the variable getProjStatus is defined as 'yes'. Once the match is made, the agent traverses a flat experience base structure where it has the choices of experiences statusGreen, statusYellow, and statusRed. The experiences are defined in the next section.

Experiences: At the lowest level of each ExperienceSpace structure are experiences that use cues, expectancies, anomalies, and use fuzzy logic to calculate a match. The agent is notified of the goal and course of action to accomplish that goal once a particular experience is recognized. The agent uses a mental simulation to authenticate the course of action and implements it if the result is acceptable. (Prindible, 2006)

According to Prindible (2006), an *experience* has the following structure: experience definition, that must be aligned with an experience in the ExperienceSpace; cues, the conditions that the agents employs to calculate a match link between the present environment and this experience; expectancies, what the agent should look forward to after the COA; anomalies, what could take place that would elicit the agent to pass up selecting the COA; goals, the aim of the COA; actions, a COA connected to a Plan in the Process Manager; and result, the outcome in the past experience. The following is an example of an experience used in the simulation:

```
(Experience statusGreen
  (Cue (schedule green high)
    (cost green under))
  (Expectancy (project continue))
  (Anomaly (project halt))
  (Goal (project continue))
  (Action (projStatus Green))
  (Result success)
)
```


The experience statusGreen is recognized as a match when its *Cues*, defined as "(Cue (schedule green high)(cost green under))", evaluates to be true where, FactType schedule is defined as 'green' because it is performing at a high level; and the FactType cost is green and under, meaning that the cost of the project is under the budgeted cost thus it is green. In the statusGreen experience structure, the *expectancy* defined as "(project continue)" means that the agent should expect the project to continue when the COA is carried out. The *anomaly* component of the experience, defined as "project halt", means that if the variable project is defined as halt, "(project halt)", then the agent will avoid the COA. The *goal* component, defined as "(project continue)" means that the goal of the statusGreen experience is for the project to continue. The *Action* component of the experience defined as "(projStatus Green)", defines the name of the COA plan that is defined in the Process Manager. Finally, in the *Result* component of the experience, defined as "success", conveys the outcome of the statusGreen experience.

Example Summary

In this example, the knowledge base (KB) and experience knowledge base (EKB) of the R-CAST architecture was populated with knowledge related to the project management domain. When populated with this knowledge, R-CAST was able to reason over the knowledge and choose courses of action based on defined experiences.

4. Knowledge Acquisition Process

This section describes the methods used to elicit knowledge from the project management subject matter experts that will later be used to populate the R-CAST system. The R-CAST architecture allows information in the form of cues, goals, actions, and expectancies to be programmed into the configuration of its components. In this research, this information will be derived from the acquired knowledge of subject matter experts through interviews.

Knowledge Acquisition Scenarios

For this research, 3 scenarios with their own project schedule and Gantt chart were

developed in the Microsoft (MS) Project tool, along with a set of general survey questions to aid in the knowledge acquisition process. (For this paper, the information on the survey questions has been omitted in an effort to focus on the identification of cases that will help shape the red, yellow, and green project statuses.) The goal of developing the scenarios and questions are to enable the subject matter experts to make decisions on the scenarios by referencing prior experiences during an interview. Another goal of the scenarios is for the subject matter experts to identify key components that they feel are necessary to be present when deciding on the status of an IT project. The acquired information will allow the development of a scenario that takes into account some of the components that the experts feel are pivotal for a project, and is tailored to fit into the configuration of the R-CAST architecture.

Knowledge Acquisition – Interview Process

For this research, 10 professionals with experience in project management were selected to serve as subject matter experts (SME). The total time planned for each interview was 30 minutes. However, in most cases, many of the interviews went as long as 45 minutes to 1 hour.

Subject Matter Expert – Experimental Group: In order to establish an experimental group of subject matter experts, the participants had to meet the requirement of being Project Management Professional (PMP) certified. The PMP is a certification administered by Project Management Institute (PMI) to certify professionals as project managers (PMI, 2008). The amount of experience that the SME has accumulated was not a determining factor in the selection of the expert. However, this information is recorded to use as a deciding factor in what information to use in determining a resolution to conflicting thoughts when developing rules for the knowledge system. For this research, 10 subject matter experts were interviewed.

Interview Approach: Before the start of the interview, the SME was not provided with a copy of the research questions, scenario document, or Microsoft (MS) Project documents. This approach was

taken with the intent of eliciting the tacit knowledge associated with the SME's project management approach. Meaning, by providing the materials ahead of time, the risk of having experts provide answers that are explicitly stated in text is present. Also, the idea is that having experts answer questions they are not expecting, allows them to simulate the role of a project manager that is determining the status of an IT project based solely on the information they are provided. Therefore, the experiences that the expert recalls to reach the decision are intuitive and uncultivated.

Before conducting the interview, the SME was given a research agreement that ensures the protection of their identity. Next, the purpose of the research was introduced to the expert and then the experts answered the first set of general survey questions. After answering those questions, the SME was given the scenario document and MS Project documents simultaneously. The reason for this approach was to get an visual indication of the preference of the expert through observation on whether they preferred to read the scenario document first or immediately begin looking at the MS Project documents. This information will be used to determine how critical the Microsoft Project tool or any project management tool could be to understanding the scenario and developing a conclusion of the scenario's project status.

Knowledge Acquisition – Scenario Framework and Results

As mentioned in the previous section, the goals of the scenarios and survey questions were to facilitate the knowledge acquisition interviews. This section will describe the framework of the scenarios through the description, expected decision of status, and intent of the interview questions scenarios. The results acquired from the interviews will also be discussed.

Design of Scenario Framework:

For the expected decision of status, the expert has the choices of "Red", the project is at risk for not being completed on-time or on schedule; "Yellow", where the project is at risk but possible to bring it back on schedule; and "Green", where the project is on schedule and there are no immediate

high impact risks that can degrade the project's performance.

Also, each scenario has some ambiguous information and pieces of information missing from the scenarios and MS Project documents to elicit and aid the analysis process. By identifying missing information, the SME may be able to reference prior experiences and identify more factors that they feel are necessary to determine the status of a project. In this section, we will discuss the expected outcome, intent, and results of the scenarios for each of the 3 scenarios. The following is an example of a scenario used in the knowledge acquisition process.

Scenario #1:

Project Start Date: February 8, 2007

Completion Date: August 2007

Current Date: June 26, 2007

Resources: 1 project manager; 2 technical developers – 1 expert, 1 moderate expertise

Budget: \$50,000

Budget Spent: \$20,000

ABC Inc. is a company that loans money to students and gives them advice on when to pay their bill, the amount to pay in installments, and when to consolidate their loan. Project ABC is a project in which the goal is to develop and application that can process the daily operating activities of ABC, Inc.

The project team has encountered a problem with the software that they were planning to use to implement the consolidation portion of the application. Due to incompatibility of versions of other COTS products used to develop the system, the project team will need 3 extra weeks to wait for the updated version of "Consolid8", the consolidation software, to be delivered and tested for functionality.

What is the status of the project?

The expected status for this project is "Yellow." The intent of this scenario is to incite the expert to think about project management activities such as budgeting, and resource availability.

In the scenario, the completion date is defined as August 2007. The purpose of this designation is to get the SME to wonder if it

is at the beginning or end of the month, as this would affect the determination of the status. It is also defined as such to see if the SME refers to the MS Project documents for the defined date. Additionally, Scenario #1 is the only scenario to show the '% complete' of each task on the MS Project Gantt chart. This is also the only scenario that identifies project team resources.

Scenario #2: In scenario #2, the SME is presented with a scenario where the schedule is impacted negatively due to trouble with getting approvals to do work. The SME is challenged with providing the status for a project where the schedule has the potential to have up to two months added to its completion date.

The expected status for this project is "Red." The intent of this scenario is to incite the expert to make points about scheduling issues. The scenario conveys to the expert that an "extra month" is being added to the project in addition to 2 more months to conduct an audit. That information is included to find out if it affects the SME's evaluation of the project schedule.

Scenario #3: In scenario #3, the SME is presented with a scenario where a vendor and a customer are negotiating a proposal to upgrade a database solution. The SME is challenged with providing a status for a project where there is no indication of any foreseeable obstacles.

The expected status for this project is "Green." The intent of this scenario is to present the expert with a project scenario that appears to be on schedule.

Scenario Results: As shown in Figure 2, for each of the scenarios, the expected outcome of the scenario was not a unanimous choice. In the cases of Scenario #2 (where the expected outcome is Red) and Scenario #3 (where the expected outcome is Green), 70% of the SMEs concluded that the project status was 'Red' and 'Green', respectively. (Note: The 'Other' category in Figure 2 for Scenario #3 reflects that one SME did not provide a status classification.) Interestingly, in the case of Scenario #1, where the expected outcome is Yellow, only 40% of the experts concluded that the project was 'Yellow' and 50% of the

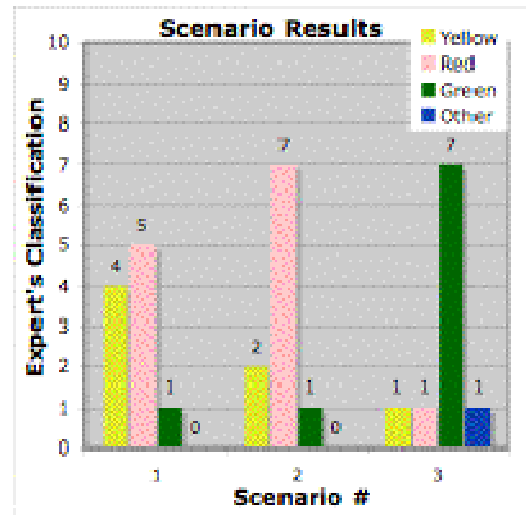


Figure 2. SME Scenario Results

experts concluded that the project was 'Red'. Of the experts that concluded that the Scenario #1 status was 'Red', 80% of them have the most years of experience in the field of project management out of the group of interviewed experts.

From the results, we can infer that the SMEs were consistent in diagnosing the project status when the expected outcome was Red or Green. However, in the case of a Yellow project status, there was more inconsistency present. When visualizing the range of Green-Yellow-Red, there is less doubt in the instances where the project can be categorized near the ends of the range (i.e. Green or Red). On the other hand, there is more ambiguity in determining the status when the outcome falls on the borderline of two status indicators. This decision uncertainty is exactly the situation we are trying to alleviate by populating the R-CAST system with project management domain knowledge. For instance, we want the system to help in determining the project status where it is not clear if a project manager should naturally or unnaturally terminate a project or make a plan to get it back on track (Cleland, 1999).

5. Conclusion and Future Work

R-CAST supplies a flexible solution to developing cognitive aids for supporting

human-centered teamwork in information and knowledge exhaustive domains. It provides this capability through its incorporation of various forms of team intelligence including shared teamwork process and progress, dynamic context management and information dependency reasoning, and a recognition-primed collaborative decision mechanism (Fan, 2007). The work presented here is a first step toward using R-CAST to develop a collaborative agent system that will recommend the status of a project. Through our example, we have shown that R-CAST has the capability to reason over knowledge that is specific to the project management domain.

The knowledge acquisition activities demonstrated that ambiguity is present in the scenario where the expert had to decide that the status of a project is Yellow. From these findings, research needs to be conducted to determine how the intensity of the scope, schedule, and cost project constraints influence a project manager's decision on the status of a project when the outcome could fall on the borderline of two status indicators: Green-Yellow and Yellow-Red. The ability to account for this uncertainty will be addressed in future research.

Future plans for this research include the following:

- Integration of Fuzzy Methods: a strategy will be devised to introduce fuzzy methods into the research. At this moment, one of the primary visions for using fuzzy methods will be to use fuzzy logic to determine the range of the Red, Yellow, and Green status indicators by inferring how the state of a combination of environment variables (i.e. scope, schedule, cost, performance, etc.) will influence the project status. The use of fuzzy logic can help in reducing cognitive dissonance by assigning a *degree of truth* to a set of *crisp* propositions acquired from a SME (Standford, 2008; Cox, 1995). In the case of determining what constitutes a "Yellow" project, the crisp values acquired from the SME can be defined as a fuzzy set, as a mechanism to reduce vagueness, that specifies the range of values for the concept of a "Yellow" project as well as the degree of

membership for any suspect characteristic value (Cox, 1995; Martin, 1988).

- Ultimately, develop a scenario that refines the scenario presented in the paper to allow further investigation into the capability of the R-CAST architecture. The scenario will be used for an interview with a subset of the SMEs already used in the knowledge acquisition process to get their decision on the scenario's project status.
- Empirical results will be obtained by translating the scenario and acquired knowledge into the R-CAST architecture and comparing the results obtained to evaluate the efficacy of the R-CAST system. In the end, we want to validate the use of R-CAST as a tool that will aid project managers in their decision-making process.

References

- Burgess, T., Byrne, K., Kidd, C. (2003) "Making Project Status Visible in Complex Aerospace Projects." *International Journal of Project Management*, Vol 21, Issue 4, pp. 251-259.
- Carlsson, Christer and Efraim Turban (2002) "Introduction: DSS: Directions For The Next Decade" *Decision Support Systems* vol. 33, issue 2, pp. 105-110.
- Cleland, David (1999) *Project Management Third Edition: Strategic Design and Implementation*. The McGraw-Hill Companies, USA
- Cox, Earl (1995) *Fuzzy Logic for Business and Industry*. Charles River Media, Inc., Mass.
- Fan, Xiaocong and John Yen (2007) "R-CAST: Integrating Team Intelligence for Human-Centered Teamwork" *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)*, 2007.
- Fan, Xiaocong and Bingjun Sun, Shuang Sun, Michael McNeese, John Yen, Rashaad Jones, Timothy Hanratty, and

- Laurel Allender (2006) "RPD-Enabled Agents Teaming with Humans for Multi-Context Decision Making." In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06), Japan, May 8-12, pp. 34-41.
- Fan, Xiacong, and Shuang Sun, John Yen (2005a) "On Shared Situation Awareness for Supporting Human Decision-Making Teams" In Proceedings of 2005 AAAI Spring Symposium on AI Technologies for Homeland Security, Stanford, CA, March, pp. 17-24.
- Fan, Xiacong, Shuang Sun, Michael McNeese, and John Yen (2005b) "Extending Recognition-Primed Decision Model For Human-Agent Collaboration" In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems, The Netherlands, July 25-29, pp. 945-952.
- Fan, Xiacong, Shuang Sun, Bingjun Sun, Guruprasad Airy, Michael McNeese, and John Yen (2005c) "Collaborative RPD-Enabled Agents Assisting the Three-Block Challenge in Command and Control in Complex and Urban Terrain." Proceedings of 2005 Conference on Behavior Representation in Modeling and Simulation (BRIMS), Universal City, CA, May 16-19, pp. 113-123.
- Felsen, Jerry (1976) Decision Making Under Uncertainty: An Artificial Intelligence Approach. CDS Publishing Company, New York.
- Hayes, J.R. (1962) Human Data Processing Limits in Decision Making, Report No. ESD-TDR-62-48. Air Force Systems Command, Electronics System Division, Bedford, Mass.
- Kerzner, Harold (2006) Project Management: A Systems Approach to Planning, Scheduling, and Controlling. John Wiley & Sons, Inc., New Jersey.
- Lewis, Bob (2008) "The 70-percent failure." <http://www.infoworld.com/articles/op/xml/01/10/29/011029opsurvival.html>. Retrieved August 2008.
- Martin, James and Steve Oxman (1988) Building Expert Systems: A Tutorial. Prentice Hall, NJ.
- Prindible, Matt (2006) "An Introduction to R-CAST: Developing Simple Agents using Decision-making modeling and the R-CAST Architecture" The Laboratory for Intelligent Agents, College of Information Science and Technology, The Pennsylvania State University http://agentlab.psu.edu/rcast/rcast_tutorial.pdf
- Project Management Institute (PMI) (2004) Project Management Body of Knowledge. Newtown Square, Pennsylvania.
- Project Management Institute (PMI) (2008) "The Project Management Professional (PMP) Credential." <http://www.pmi.org/CareerDevelopment/Pages/Obtaining-Credential.aspx>. Retrieved August 2008.
- Shim, J.P., Merrill Warkentin, James F. Courtney, et al., (2002) "Past, Present and Future of Decision Support Technology." Decision Support Systems, vol. 33, issue 2, pp. 111-126.
- Simon, Herbert (1955) "A behavioral model of rational choice" Quarterly Journal of Economics 69:99-118.
- Snedaker, Susan and Nels Hoenig (2005) How to Cheat at IT Project Management. Syngress Publishng, Inc., Rockland, Mass.
- Stanford Encyclopedia of Philosophy (2008) Search: Fuzzy Logic. <http://plato.stanford.edu/entries/logic-fuzzy/>. Metaphysics Research Lab, CSLI, Stanford University. Retrieved August 22, 2008.