The Pennsylvania State University

The Graduate School

College of Information Sciences and Technology

INTELLIGENT COACHING AGENTS FOR ENHANCING HELPING BEHAVIOR IN HUMAN TEAMWORK

A Thesis in

Information Sciences and Technology

by

Cong Chen

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2006

The thesis of Cong Chen was reviewed and approved* by the following:

John Yen
University Professor of Information Sciences and Technology
Thesis Advisor
Chair of Committee

John M. Carroll
Edward M. Frymoyer Professor of Information Sciences and Technology

Brian K. Smith
Associate Professor of Information Sciences and Technology

Tracy Mullen
Assistant Professor of Information Sciences and Technology

Guohong Cao
Associate Professor of Computer Science and Engineering

Joseph M. Lambert
Associate Professor of Information Sciences and Technology
Chair of the Graduate Programs Advisory Committee in
College of Information Sciences and Technology

*Signatures are on file in the Graduate School

ABSTRACT

Teamwork is the joint work of individuals who act together productively. Recent technological advancement, global competition and world events made teamwork vitally important to the success of many organizations in both military and civilian sectors. Team training is arguably one of the most intensively studied topics for cognitive science researchers in the past decades, yet there are few effective software tools that automate team performance assessment and coaching.

The goal of this research is to develop an intelligent training framework where software agents are used to automate team performance assessment and coaching, with a focus on helping behavior. One of the design challenges in coaching for team training is that the performance of a team is affected by multiple factors, which include the quality of the team's plan and each individual's execution of the plan. To address this difficulty, our framework uses a two-phase training protocol that provides coaching for two phases: a mission planning phase and a mission execution phase. We adopt two user modeling approaches (overlay and error taxonomy) in Intelligent Tutoring System (ITS) for the two phases, according to different complexity involved in modeling expert behavior. In the planning phase, intelligent coaching feedback is generated based on an expert model for resource allocation. In the execution phase, coaching feedback is generated based on error taxonomy to assess team's execution performance of their planned activities. Due to the broad scope of team training, this research stresses one important dimension of teamwork—the helping behavior among team members, in case of an unbalanced workload and resource distribution. Coaching feedback is provided in a debriefing session at the end of each mission execution with a goal of improving trainee performance during the next mission.

We have implemented the framework within a team-based agent architecture and applied it to train helping behaviors for a simulated command and control (C2) task. To evaluate the effectiveness of the agent-based team training approach, we designed and conducted a human subject experiment that applied the agent-based two-phase training protocol and provided teams in the experiment group with feedback generated by the coaching agents. Results have suggested that the coaching agents have a positive impact on trainees' learning of how to effectively helping each other to achieve mission success in time-critical and complex task domains.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. John Yen and my committee members, Dr. Carroll, Dr. Smith, Dr. Mullen and Dr. Cao. I would also like to thank Dr. Shebilske, Dr. Volz, Michael Miller, Shruti Narakesari, and Ganesh Alakke as part of the MURI project. And last but not least, thanks for everyone working in the intelligent agent lab for their constant support.

This research was supported by DOD MURI grant F49620-00-1-0326, administered through AFOSR.

1 Introduction

1.1 Research Question and Motivation

Teams allow workloads to be distributed among individuals with different expertise, and effective teamwork prompts improved team performance. There is a growing awareness that teamwork is vital to the success of many organizations, especially for complex tasks that impose high mental or physical demands that go beyond individual capacity—and that only effective communication and the collaboration of multiple members can ensure mission success.

With rapid technological advancement and intensive global competition, team training that has been one of the most important topics in psychology, has drawn more attention from researchers in various other disciplines to study team training technology that can be applied in either the military or the civilian sector.

The nature of teams is fundamental to understanding team performance and training. A team is defined as a distinguishable set of two or more people who interact dynamically, interdependently, and adaptively toward a common goal, who have each been assigned specific roles or functions to perform, and who have a limited life-span of membership [1]. The notion of a common goal in team definition distinguishes a team from other working groups who also have a certain degree of interaction and exchange of information or resources. Another unique characteristic of a team is that some kind of organizational structure needs to be imposed on the team members, and there must be some form of task dependency in order for them to coordinate and accomplish their shared objectives.

Training that aims to enhance team performance is more complex than the training of individuals to master certain task skills. Traditional one-on-one tutoring focuses on the modeling of the students at the individual level and provides feedback on how students can go through a problem-solving process better. While intelligent tutoring has become an effective method for a student to acquire individual skills, such as the Intelligent Tutoring System (ITS) application in learning math, algebra or programming language, its application in team training is still challenging. It is often observed that a team of experts does not make an expert team in many real-world scenarios.

More than 30 years of team training research has provided a great deal of information concerning the characteristics of a team, the dimensions of teamwork, and the factors that could influence team performance. Many studies have been designed to investigate factors that influence team performance outcomes, yet these studies have not provided a common answer to what exactly are the collective skills in the team performance process that training is supposed to improve. In this study we focus on arguably one of the most important dimensions of teamwork — helping behavior— to guide our design and development of the intelligent training framework and its coaching assessment modules.

Team performance, defined by Nieva as goal-directed behaviors/function accomplished by a team in performing a task, has two major components: individual-level task behavior and team-level task function [2]. It is an outcome of dynamic processes reflected in the coordination and communication patterns that a team develops over time. Individual expertise, as a portion of the solution to a complex problem, is necessary but not sufficient to achieve good team performance. Beyond the individual skills needed to perform their assigned subtasks, team members need to make decisions about how to communicate efficiently and to coordinate their work flow. One typical example of coordinated work flow is a team member proactively offering help to other members who are in need of support [3].

Intelligent simulations provide interactive environments that are essential to automate team training. In such environments, trainees are able to interact with autonomous agents and other human team members and go through a set of realistic training scenarios where collective tasks need to be performed. Simulation-based training is less constraint by equipment cost or personnel availability that normally associated with human-to-human field training, and provides the potential for trainees to get training exercises almost anywhere anytime. To ensure trainees' effective learning and transferring of the target knowledge and skills, intelligent agents are designed in such training environment to play the role of human coaches, monitoring trainees' activities and providing feedback as necessary.

Intelligent tutoring system research provides a rich source of theories and practices that guide the development of automated tools in facilitating individual learning. However, it is not feasible to apply individual cognitive diagnosis, student modeling, and adaptive tutoring directly into team training due to the nature of team problem solving, such as the increased complexity, uncertainty, multi-layered knowledge spaces, and time stress. For example, initial attempts have been made to apply ITS concepts in developing intelligent team training technology, yet most of them focus on providing semi-automated instructions to reinforce human-based coaching, few succeeded in developing an intelligent training system where intelligent coach plays the role of a human coach/tutor [4].

Planning has been long recognized as an effective way to ensure success in any endeavor that needs preparation. Planning before undertaking a task can result in efficiency enhancement and cost deduction, such as individual's planning of a trip or budget or strategic planning in organization or tactical planning in the military. Reported in ITS literature, planning has been employed in guiding instructional design and discourse control [5, 6], yet the benefit of student's planning activities haven't been recognized in the course of solving a problem. The lack of attention in ITS development on individual planning in ITS development might root in the nature of individual-based problem domains: 1) little indeterminacy is involved 2) the individual task can be highly procedural 3) the learning outcome highly depends on individual's proficiency of a particular subject. Numerous such examples in ITS domain exist where planning doesn't have significant impact on the learning outcome, such as solving a mathematical problem or trouble shooting a device. On the other hand, team problem solving always involves anticipation of task situation and adaptation to the possible future outcome. Thus for most team tasks, the demands of team-level decision making require a great focus on planning, which guides team's proactive responding to potential interactions that might arise and plays a critical role in determining mission success.

Given the implications that planning has on training complex task skills, we proposed our "two-phase" training approaches that "divide and conquered" the team training problem at hand. In this approach, the training problem is decomposed into two sub-problems. Accordingly the whole training session is divided into a planning phase and an execution phase, each involves solving part of the training problem at different levels of complexity. During the planning phase, we allow trainees to practice on planning for their future mission and focus on their resource allocation skills in the context of helping overloaded team members. During the execution phase, trainees practice on execution of the detailed helping behavior, take advantage of the planned allocation strategies and try to adapt to changes evolved during the mission.

Besides the fact that the two-phase training has addressed the significance of planning activities as one important team training dimension, the distinction of the two sub-problems with different complexity also allow us to employ different ITS modeling techniques to suit the specific training needs in each phase. At planning phase, trainee's planning task concentrates on knowledge at the abstract level, which is the allocation of team resources where that they are most needed. Yet in this phase, little detailed timing information is involved about when to initiate the specific action points to realize the planned resource allocation. Thus we could employ the overlay approach and extract the problem solution as an expert model that captures the desired planning strategies and provide planning phase feedback by comparing trainees' actual planning strategies with the set of expert planning strategies. As more team dynamics are introduced in the execution phase, trainee actions might involve multiple layers of knowledge and skills. For example, the execution of their allocation plan entails team-level communication and coordination with the specific timing and outcome for each domain action. In the execution phase, however, it is not feasible to employ the overlay approach and build a comprehensive model of "correct" behaviors for an expert team. The characteristics of the execution-phase problem entail such difficulties: 1) Different than the traditional tutoring problem, there is no single right solution for the complex team task; 2) there are intensive interactions among team members and different combinations of team actions could lead to the same outcome. With the observation that the training needed at the execution level is not the provision of the entire solution set, we employ the error taxonomy approach to build trainee deficiency libraries at both individual level and the team level. To diagnose trainee deficiencies, we link multiple steps of event-based assessments to assess patterns of team' helping behavior, which allows us to provide trainees with feedbacks that address their helping-related deficiencies.

The complexity of human team training has been intensively studied in cognitive science community. To design and develop the intelligent training framework, we also adopted a set of theoretically well-founded principles as reported from literature [7]. These principles are generic yet vital to guide the design of human team training for various domains.

1.2 Identify the Scope of the Team Training Problem

Team training is hard problem to solve given the difficulties we described in the previous section (Section 1.2). Aiming to provide an automated training framework that addresses the team training problem, we need to carefully evaluate the scope of the research. Later in this section, we identify our target training goal by providing detailed characteristics of the training problem, including the nature of the team, the training environment, team task, target team dimension and the target skill level.

- The team being trained

We are looking at command and control (C2) teams that are responsible to accomplish complex team tasks by managing limit amount of resources within restricted time and space. Adequate amount of communication and collaboration are expected to facilitate team decision making and ensure success of the team mission. Examples of C2 teams include AWACS teams, air defense teams, first responder teams, incidence management teams, air traffic control teams, and NASA mission control center.

- The training environment

A realistic virtual environment is essential for simulation-based training and allows the modeling of individual operation, team collaboration and communication. Given a specific set of training goals, it is desirable that the simulation environment allows a range of adjustable parameters to configure the training scenarios. For example, in training of the helping behavior, it is necessary to have the situated collaboration context that requires team member's helping each other. The team assessment and diagnosis capacities within the training framework need to be transferable to multiple training domains with similar team settings.

- Characteristic of task demands

As determined by the nature of C2 teams, the stress induced on team personnel may have played a significant role on the overall mission success while performing tasks in dynamic, time-critical, and uncertain environment [8]. Among incidents in warfare that were reported to exemplify the impact of stress factors for C2 teams, Vincennes tragedy has a tremendous implication on the urgent demand of training under stress, which refers to the tragedy when civilian Iran air flight 655 was mistakenly shot down by the US navy in 1988 [9]. In this study, we introduced Dynamic Decision Making (DDD) where team members are overwhelmed with a large amount of incoming tasks during limited time period. Our training approach is based on the assumption that the introduction of stress in the tasks performed by trainees can reinforce the creation of valid simulation-based training environments that have positive impact on trainees' dealing with stress in actual task-performance situations.

- Legitimacy of helping behavior

One of the main focus of this research is to study helping behavior, "a team skill that is at the heart of teamwork" [10]. In designing of our training scenario, we need to ensure a social setting where team member's helping each other is legitimate and critical to achieve better team outcome. We focus on two aspects of a team that determine the need for helping— characteristics of the team's task and the characteristics of the team composition. Team task can be quantified as the amount of workload assigned to each team member and team composition determines the resource distribution within team. The legitimacy of helping need is created when the workload assigned to a particular team member exceeds the amount of resources allocated to accomplish the task. As a specific case of the above described helping legitimacy, when team members share the same amount of resources (even resource distribution), the uneven distribution of task demand creates a clear and distinct need for team members' helping each other [10]. Given that team's incoming task load can be anticipated, observed or learned, the imbalance between the workload demand and resource allocation can be corrected. Helping behavior is highly involved in situations when team member need to recognize which teammate has been over loaded and provide direct assistance to compensate his/her lacking of resources.

- Skills to be acquired—Taskwork skills vs. teamwork skills

Taskwork skills refer to the ability of an individual to perform domain tasks that involves declarative and procedural knowledge at individual-level, such as individual's awareness of a set of domain constraints specified by the task (declarative) or a sequence of actions to perform a domain specific operation (procedural). Teamwork skills refer to the collective capacity of the team in terms of information sharing and coordinated decision making. Compared to taskwork skills that might be acquired individually, teamwork skills represent higher level cognitive tasks such as team situation awareness, active communication and well-coordinated joint acts and have to be trained within an interactive team context.

1.3 Research Objectives

In this dissertation, we look at defining an agent-based team training framework and constructing coaching components to help trainee enhance performance in complex and time-stress domains where intense team interactions are required to ensure mission success. To design and implement the intelligent training system, we have two training phases each deal with a training sub-problem with different levels of complexity. In designing the assessment modules to diagnose trainee deficiencies in the second phase, we adopted the event-based training approach to link multiple steps of an event to assess patterns of helping behavior. Our design and implementation of the intelligent training tool allows further experimentation to validate training protocol and the effectiveness of coaching feedback.

The research addressed in this dissertation represents a rather limited yet in-depth coverage of many topics that concerns team performance and intelligent training. The two major team performance measures in training are problem diagnosis and skill development. Inspired by the psychological study about human team training, we diagnose the areas of difficulty in learning how to acquire team helping behavior. As an important dimension of teamwork, helping behaviors overlap with some other team dimensions, such as collaboration and coordination and involves a rich set of skill sets to be developed. To assist trainees with their skill development, we narrowed our research scope to be looking at C2 teams in a simulation environment. As from military teams in actual work situation, a collective skill set is identified to include command, control, communication, and coordination. Further, feedback is presented to a team at both the individual level and the teamwork level for optimal results.

We aim at building an intelligent coaching system that provides training feedback regarding a specific dimension of teamwork at both individual-level and team-level for a team of trainees. It helps them acquire appropriate knowledge and skills to perform highly interdependent tasks. We seek answers for two fundamental research questions about the design and evaluation of the intelligent team training system in this study: 1) Regarding an important dimension of teamwork (helping behavior), how can we build an intelligent training system that enables software agents to play the role of a human coach and provide appropriate coaching feedback for a team of trainees to improve their performance? 2) What kind of training protocol and experimentation we design to evaluate the effectiveness of such training system for our current training objectives?

As we described earlier, we adopted the *divided and conquer* approach to decompose the training problem into a planning sub-problem and a coaching sub-problem in recognition of the significance of team planning and to be able to apply different ITS approaches according to different levels of problem complexity. Accordingly, during the course of training, a two-phase training protocol was employed as having a planning phase and an execution phase. The modeling of trainee on two different aspects of a complex problem and the design of the corresponding training protocol allows trainees to plan ahead for their mission before they carry out the mission. We had the assumption that trainees could achieve better mission performance by taking advantage of the anticipation of mission information during the planning phase and proactively adapt to changes during mission. During the after action review session, coaching feedbacks are presented for both trainee’s planning of resource allocation and their online execution of the team mission.

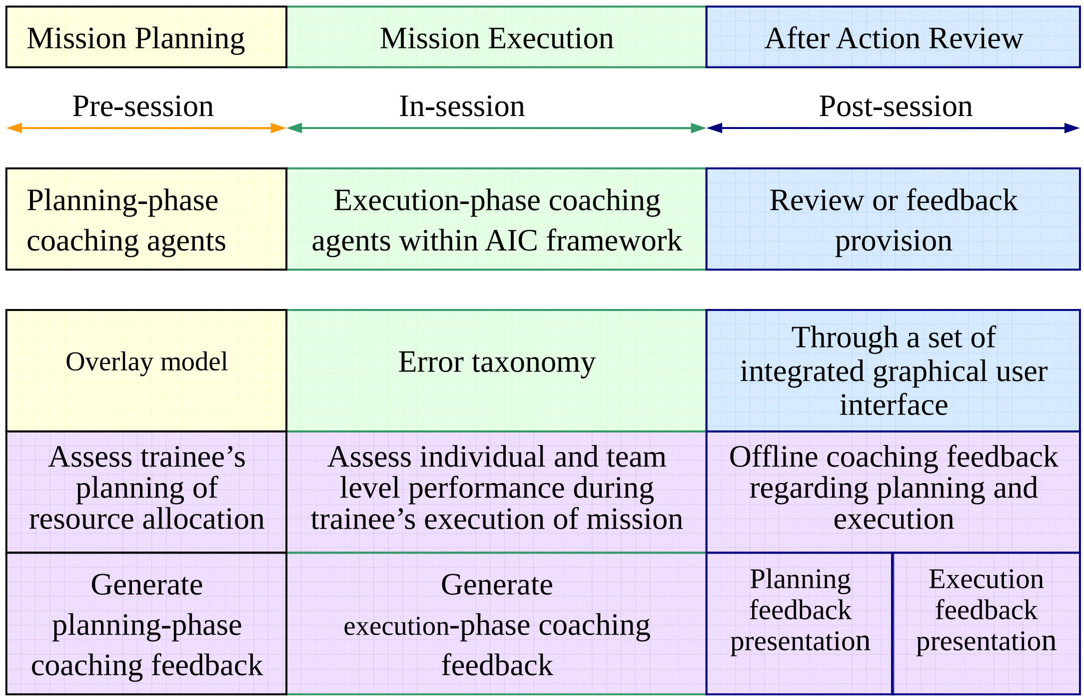


Figure 1 Two-phase Coaching within the Training Session Timeline

Figure 1 provides an overview of the training timeline of expected coaching and feedback sessions that occur in a complex tactical training environment. The whole training session consists of three sub-sessions, respectively pre-session, in-session and post-session. In the pre-session, trainees are allowed to plan about their mission-execution. Given a specific team plan, planning coach generates feedback regarding trainee's planning of team resource allocation. The in-session includes the generation of execution-phase coaching feedback as well as the actual training simulation execution. During the post-session, coaching feedback generated by both planning phase and execution phase agents is presented to trainee with the goal to help them achieve better performance during the next training session.

For the mission planning phase, we built coaching agents that can assess trainees' planning strategies and help them make better resource allocation plans for the next mission. Accordingly during the planning phase trainee, we provide trainees with information that helps them predict the upcoming mission, and allows them to make decisions about the allocation of resources during mission. The planning phase as part of the training problem is an abstraction of a trainee's execution knowledge at the resource allocation level, with only rough timing information involved. The diagnosis capacity of the planning phase coaching agent is built on top of an overlay model where trainees' planning strategies are viewed as a subset of the experts' planning strategies. Trainee deficiencies are identified by comparing the student planning model with the expert planning model and feedback will be generated accordingly aiming at helping trainee make better plans during the next mission. Offline coaching feedback will be provided to trainee at the end of the whole mission execution regarding their resource allocation strategies compared to a set of expert resource allocation strategies.

For the mission execution phase, we built coaching agents that can assess trainees' team performance during execution with a focus on their helping behavior and provide feedback regarding their performance deficiency both at the individual level and team level. The desired execution-level teamwork behavior involves intensive communication and interaction among team members and such a complex real-time problem does not imply a unique expert solution. To model trainee behavior during execution, it is not feasible to build a comprehensive overlay model to represent desired team behavior, thus a different type of assessment approach, error taxonomy model, is adopted. Within the error taxonomy model, a list of critical events regarding helping behavior is identified and a monitoring framework is developed for intelligent coaching agents to capture these critical events. We assess trainee’s helping pattern by linking multiple steps involved in each event category and reasoning about trainees’ deficiency regarding helping one another. During the after action review session, which is at the end of the whole training mission, trainees will also be given feedback regarding their helping behavior and other related individual performance deficiencies.

We also conducted human experiments to evaluate the effectiveness of our training protocol and the coaching feedback generated by the two-phase coaching agents within the intelligent training framework. The human experiment involved two groups of trainees that are both introduced to the two-phase training protocol, yet only the experiment group receives the coaching feedback generated by the two-phase coaching agents at the end of the training session. The underlying research hypothesis is that coaching agents within the agent-based intelligent training framework helped trainees achieve better team mission performance and specifically helped them learn how to proactively offer help to other members in a team context where such collaboration are desired and essential to accomplish the team tasks with high interdependency and complexity.

1.4 Accomplishment of this Research

In this study, we focused on command and control teams that perform highly interdependent tasks in time-critical domain. Each member of the team has clear responsibilities, but an important strategy for enhancing team performance and processes is for each individual member to dynamically adapt to changes in the task environment, contributing their resources and coordination capabilities as necessary for the team to accomplish their mission successfully.

In designing the intelligent coaching agents within our training system, we built a performance assessment model that includes not only assessment of individual member's task performance, but also a comprehensive team performance assessment that captures the collective variables that will significantly affect team performance during team's mission execution. For the highly interdependent team tasks, our training goal is to help individual trainees learn how to improve inner-team collaboration capabilities to achieve teamwork effectiveness that couldn't be achieved by team members individually.

We built our intelligent coaching agents on top of a multi-agent architecture. The generic coaching components can be applied to other teamwork-oriented domains where collaboration among members is key for overall mission success. Specifically we built a prototype training system that adds domain-specific coaching solutions for a command and control simulation. During the course of training, we applied the two-phase training protocols that allow trainees to first plan their mission execution and then carry out the plans they built as a team during mission. In planning phase, trainees will be given information that could be used to predicate characteristics of their incoming mission; in the execution phase, trainees carry out the plan they made as a team and make adjustments to adapt to the changes in the mission.

The intelligent training framework provides both planning phase coaching feedback and execution coaching feedback at end of each mission. The planning phase feedback addresses the issue of how to make a good plan that optimizes resource allocation of the overall team. Our training agents generate feedback about each individual team member's planning deficiency comparing the actual team plan to a set of expert planning strategies. During a trainee's execution of the mission, agents monitor human trainees' actions, analyze data collected for critical collaborative events and provide feedbacks to trainees about their online deficiencies concerning individual task performance or their lack of the collaborative skills involved in team processes that are vital for the overall mission success.

Human experiments were conducted to validate our research hypothesis that in a simulation-based team training environment, intelligent coaching agents that provide offline feedback about team's planning and execution of the mission have a positive influence on trainee's achieving better collaborative process and performance outcome for tasks that requires coordinated interaction among team members.

In summary, this research has made contributions in the following aspects:

1. Developed a generic team training framework that can be extended to other training domains for monitoring and giving feedback for a team of trainees to enhance both their team process and outcome.

2. Developed intelligent coaching agents within the team training framework that implements an empirical training protocol for command and control teams in a military simulation domain.

3. Conducted human subject experiments where teams of participants exercise with the above mentioned training protocol with or without the assist of intelligent coaching agents and validated our research hypothesis by comparing their performance process and outcome

2 Literature Review

2.1 Human Teamwork

A fundamental question that must be addressed in intelligent team training concerns the nature of human teamwork. Teamwork occurs when a group of people work together toward common goals. Compared to the loose interactions that might also be found in other types of group work, teamwork requires close interaction among its members to accomplish highly inter-dependent and cooperative tasks.

Teamwork has been intensively studied in human cognition science and many other disciplines, from business management to artificial intelligence, to information sciences and concurrent engineering. Consequently there is a diverse body of literature that defines team from different perspectives, which is core to the research of teamwork.

Katzenbach and Smith defines team as “a small number of people with complementary skills who are committed to a common purpose, set of performance goals and working approach for which they hold themselves mutually accountable” [11]; Salas defined team as “a distinguishable set of two or more people who interact dynamically, interdependently, and adaptively towards a common and valued goal/objective/mission, who each have been assigned specific roles or functions to perform, and who have a limited life-span of membership.” [1] One central point reflected in all of above team definitions is the notion of shared goals; it is about each member’ commitment to perform within-team interdependent tasks. To achieve teamwork effectiveness, team members need to dynamically change information, coordinate activities with team resources and constantly adjust to task demands.

To better understand human teamwork, activities involved in the course of performing a team mission can be decomposed into two categories—task-work and teamwork. The former one refers to the task-related independent activities that individuals carry out during the mission; normally these activities involve certain skills for performing some domain specific tasks. Task-skills are a prerequisite for individuals to work successfully together as a team. Beyond task-skills, there are team skills that explicitly oriented to the interaction among team members.

Teams are often more structured and organized compared to other groups. Such structure may be differentiated by task demands or its member capability. For example, a military team has a much more restricted hierarchical structure than a sports team that has a flat structure where team members mutually depends on each other.

2.1.1 Team Performance and Measurement

The goal of training is for the trainees to acquire skills. A set of performance measurement is essential to determine if trainees have well acquired the skills during their work performance. Traditionally, analysis on human performance focuses on individual’s behavior description, task requirement and situational constraints [12]. Due to the great needs to develop a set of team performance measurement from the team training perspective, it is important to investigate how to adapt individual performance measurement into a team context.

Two important questions concerning team's appropriate acquisition of the target knowledge and skill set are: 1) What are the specific knowledge and skills targeted in team training, and at which level? 2) What are the specific team processes and team outcome we measure to determine team's acquisition of such knowledge and skills? To answer these questions, we first compare team performance at individual level and that at team levels and then take a close look at the connection between team process and outcome.

Measurements are central to the evaluation of team performance. Normally, measuring of team performance can be approached by monitoring team process (and other internal team dynamics) and team outcome [13, 14]. Team process describes moment-to-moment behaviors of team members that work together and team process measures refer to measures of the strategies, steps or procedures used to accomplish a team task. Rather, team outcome measures refer to measures of the quality and quantity of the end result. Both measures are necessary for training evaluation whereas process measures provides diagnostic information that is critical for providing feedback; outcome measures are needed to identify whether part or the overall team mission is effective.

Much of the team performance research supports the general notion that effective team performance is an outcome of effective team process that represents synchronous interactions among team members to address the demands and constraints imposed by the task. Another important goal of team training is to link the team outcome with different team process measures and be able to rate how individual's teamwork skills to the overall mission performance. The types of measure could address how well the team works together as a group, the ability for the team to reach agreements, the effectiveness of team communication, and the team's supportive behaviors (e.g. helping behavior among team members).

Individual and team are two units of analysis in measuring performance. Historically cognitive modeling has strong roots in the individual performer and many task analyses evolved from individual actions performed. Yet to make an intelligent training system adaptive to individuals within a team context, it is essential for the training system to also monitor trainee actions beyond their knowledge and skills in performing individual tasks. Inferences need to be made about team-level factors that contribute to the quality of performance, such as team communication and interactions.

For the teams with minimal member interaction and lower task complexity, individual task performance may be a critical factor to determine team performance. At the other extreme, for the high interdependent teams with imposed organizational structure, member interactions may predominate. For many teams where performance is determined by both, team performance has to be measured by modeling both individual task process and team coordination process. The modeling of user at appropriate level is key to judge user's understanding of knowledge or mastery of skills. In the context of developing intelligent team training systems, a user model can be a powerful tool to make inference about user's task knowledge as well as team communication and coordination.

Table 1 illustrates the taxonomy of human–performance indicators at both team level and individual level of analysis [13].

	INDIVIDUAL	TEAM
PROCESS	• Cognitive Processes • Position-specific Taskwork Skills	• Information Exchange • Communication • Supporting Behavior • Team Leadership
OUTCOME	• Accuracy • Latency	• Mission Effectiveness • Aggregate Latency and Accuracy

Table 1 Taxonomy of Human Performance Measures

Without a theoretical framework, it is hard to describe what aspects of teamwork are to be modeled within an intelligent training system. Based on different training objectives, a set of essential team dimensions need to be identified through cognitive task analysis. Subject matter experts were asked to develop a list of teamwork behaviors that they consider important across various teams, to individually sort the identified behaviors into meaningful categories and to reach consensus.

The resulting seven teamwork dimensions and the corresponding teamwork behaviors for each dimension, shown in Table 2, are considered critical for effective team performance and outcome [15].

Communication	• Pass complete information to correct members • Respond to others’ request for information
Feedback	• Provide specific constructive suggestions to others
Monitoring	• Observe and keep track of performance of other team members
Coordination	• Pass relevant information to others in a timely/efficient manner • Facilitate performance of other team members
Team initiative /leadership	• Providing guidance or suggestions to team members • Stating team and individual priorities
Helping Behavior	• Provide others with assistance when needed
Situation Awareness	• Note deviation from steady state • Identify potential or anticipated problems • Recognize other members’ needs for some of one’s own information • Recognize need for action • Be proactive

Table 2 Teamwork Dimensions

2.1.2 Collaboration and Helping Behavior

Rather than cover a wide spectrum of teamwork behaviors, in this study, we focus on one essential dimension of teamwork— helping behavior. Helping behavior is identified as one of the essential dimensions of teamwork; it is not exclusive but closely related to other dimensions of teamwork, such as active communication, and shared mental models. In this section we draw connections between two overlapping teamwork dimensions – collaboration and helping behavior and define helping behavior within a team context.

Collaboration is central to effective team behavior. Pervading in daily life, it spans from the well-planned collaboration in sports, science and health care to the spontaneous collaboration among people who extemporarily find a problem better solved with joint efforts. Distinguished from interaction, which can be loosely acting on others, collaboration is to inherently work jointly with others [16]. Collaboration can be characterized as a group of people's intention to work together towards a goal when it is beyond the individual's capacity to achieve it.

From artificial intelligence perspective, typically collaboration can be modeled as a three-step process—the commitment to the joint activity, reaching consensus on the recipe about how to reach the joint goal and the commitment to constituent actions. From the team training perspective, collaboration normally refers to the process of working with other team members, which covers a much broader range of phenomenon within teamwork context. The main properties of collaborative activity can be described as we look at participants' relationship, knowledge, and capabilities during the course of collaboration:

1. Central control or master-servant relationship doesn't exist for collaborations
2. Most collaborative situations involve people who have different beliefs and capabilities.
3. To achieve the joint goal, collaboration entails both collaborative planning and acting.

As theoretical foundations established for computational representing and modeling collaboration process, intelligent collaborative systems were developed where individual agent can reason about other agents so as to work with them. A step further, it is more interesting and challenging for the multi-agent system community to understand how people achieve effective collaboration and develop computer systems that support human collaboration in a way that agents provide constructive feedback to enhance collaboration. Agents need to model human performance and deal with non-rational as well as the rational.

Another form of collaborative activity is helping one another in achieving a goal. Helping behavior comes in a variety of forms, as emergency aid, technical assistance, or peer learning. In different situations, helping can be deliberate or accidental, remote or local, altruistic or in exchange of favor. In a team context, helping, as a way to diffuse responsibilities, can be defined as assisting other team members in performing certain roles [17, 18]. A specially situated helping behavior deals with workload imbalance among a team of people; helping acts in such episodes have been characterized as both recognition and correction of the imbalance on workload distribution to achieve team mission success [10]; In an empirically-derived helping behavior taxonomy, Pearce and Amato classify helping behaviors as:

1. direct help (doing what one can) vs. indirect help (giving what one has)
2. planned help vs. spontaneous help
3. serious help vs. non-serious help

The first two dimensions describe helping from the helper’s perspective—whether the helper is doing what he/she can to directly help solving the problem or offer what he/she has as indirect resource assistance; or what kind of social settings around the helping acts. From help recipient's perspective, the third dimension captures the degree of the help needed.

Complementary to Pearce and Amato's taxonomy, helping can also be categorized as proactive helping vs. reactive help. It is reactive when initiated by an external request. A typical example would be the fire-fighting department's helping recipients putting out a fire after receiving their 911 call. Proactive help as an alternative, is not initiated by a helping request, rather, by having certain type of shared knowledge, the helper can anticipate other's needs and initiate helping actions even when there is no explicit request for her/him to do so.

In this study, we focus on helping behaviors that are situated in a team context where planned, direct, serious and proactive help is needed to balance team workload distribution.

2.2 Intelligent Tutoring System

2.2.1 Intelligent Tutoring System

Intelligent tutoring systems (ITS) are the earliest applications of artificial intelligence in education. ITS can be broadly defined as the computer systems that employ ideas and techniques from artificial intelligence in providing feedback to help student learn a task. Ideally, intelligent tutoring systems aim to provide each student with a learning experience similar to the one provided by a human tutor. The design and realization of ITS have gone a big step in its 30 years history, from where systems could only react to trainees in a specified way to the stage where the dynamical adaptability of the system is recognized beyond pre-design. For example, ITS is an outgrowth of the earlier computer-aided instruction or CAI model, the development of CAI systems only provided strictly didactic feedback in rigid instructional sequences.

The goal of various ITS is the use the knowledge of the domain, the student, and teaching strategies to support flexible individualized learning and tutoring [19]. There is an ongoing desire for tutoring systems to tailor their interactions to suit the individual learners. To effectively facilitate individual’s learning process, later themes in ITS development recognized the importance of reasoning about student’s underlying knowledge and skills, so as to provide one-on-one instruction that adapts to student’s evolving knowledge/skill set.

Traditionally intelligent tutoring system research focuses on modeling user at the individual level and on solving well structured problem where little collaboration is required. Applications regarding individual’s learning task has been built across many domains, including ITS as personalized information retrieval system [20], web-based argumentation system [21] and ITS that help medical students understand medical system [22].

Successful applications of intelligent tutoring system that assess the student’s knowledge and misconceptions from problem solving performance yield significant learning gains beyond traditional classroom environments. The cognitive apprenticeship system [23, 24] in which the computer acts as an adaptive coach to the student (the apprentice) who works through a series of problem-solving exercises. Examples of intelligent apprenticeship system span the domains of mathematics, science, and technology [25, 26], including the LISP tutor where probabilistic reasoning has been applied for knowledge tracing [27], the SHERLOCK maintenance tutor that participants in reflective dialog in assisting avionic technicians in troubleshooting electronic equipment [28], and the physics tutors that provides coached examples in teaching Newtonian physics to students at the US Naval Academy [29, 30].

Although there is no standard architecture for ITSs, four components emerge from literature as typical of an ITS. For conceptualization and design, it is often easier to view an ITS as consisting of several interdependent modules. A typical architecture of an Intelligent Tutoring System includes several basic components as shown in the following diagram [31].

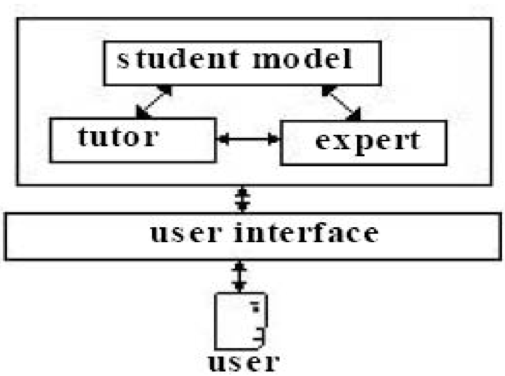


Figure 2 Standard Intelligent Tutoring System Architecture

The expert model is a computational model that represents domain knowledge needed for the generation of problems and/or solving the problem given to the student. It provides the desired behavior of an expert and provide basis for evaluating student’s behavior. The student model develops a model or road-map of the student's knowledge; it is a critical component that makes ITS adaptive to individual learner by providing a basis to reason about student actions and interpreting student behaviors. The pedagogical module is a model of the tutoring process. It manages the instructional interventions and contains the knowledge about how to advise students regarding their problem solving activities.

Based on the Advanced Computer Tutoring (ACT) theory [32], the ACT project group developed production system models that specify how student solves a problem in a given application area. They then defined a set of principles to guide construction of computer tutors around these cognitive models. The ACT theory held that cognitive skills consist in units of goal-related knowledge and that the traditional steps of building a cognitive tutor involves the following steps:

1. Selecting a problem-solving interface,
2. Constructing a curriculum under the guidance of a domain expert
3. Designing a cognitive model for solving problems in that environment
4. Building instructions around the productions in that model
5. Deploying the tutor in the classroom

However, early evaluations of these tutors usually but not always showed significant achievement gains.

2.2.2 User Modeling

User Modeling in general is a system view of user; specifically it refers to the ability of a user-adaptive system to construct knowledge about its user that facilitates the individualization of information and the interaction with user [33]. Numerous applications were developed that collect information regarding different aspects of user and exhibit different types of adaptation in the early work of user modeling [34]. Two areas of focus were addressed in early user modeling development—Distinction has been made between the specific user modeling and those system components that also perform other tasks; Generic frameworks were developed containing basic modeling functionalities that can be applied to multiple domains [35, 36].

In an intelligent tutoring system, student model is built as a specific case of user modeling to describe knowledge and belief about the student and guide pedagogical decision-making. In conventional education environment, teachers change strategies as their knowledge increase about how students learn. For the purpose of adapting to individual students, ITS researchers are most interested in how student learn and acquire knowledge. Yet it is difficult to reason about students' knowledge and learning as well as making performance assessment about their understanding of a target subject. A student model tracks student progress, monitors user-system interactions, and provides feedback adaptive to these learning interactions. Student modeling helps diagnose student's strength and weakness and make the corresponding suggestions. With the relevant information enclosed in a student model, ITS can infer about student's potential goals, misconceptions or performance deficiencies by reasoning about their learning styles and mental behaviors [37].

Advancement in artificial intelligence provided novel approaches for extensive development and use of student model. One of the major reasons to apply AI techniques is to infer the unobservable aspects of a student, such as student knowledge that is believed to underlie their learning behavior. Successful examples of student modeling have been identified in various application domains, such as multimedia learning environment and web-based instruction systems. A number of explicit models were built to elicit an accurate assessment of student knowledge at appropriate level of difficulty [38], such as constructing a bug library for students' misconception [37], or inducting a coherent student model that inherits techniques from the area of machine learning [39].

A student model can be categorized as either domain-specific or domain-independent. It is domain-specific if the representation of student's current state and level of knowledge relates to a particular concept [40]; it is domain independent if it focuses on a set of domain-transferable learning goals, cognitive aptitudes, mental states, and learning styles.

Due to different research focus in different problem domains, the contents of a user model varies—they can be built to recognize user's plans, to evaluate user performance, or to describe user constraints. It can be very costly to continuously monitor user behaviors throughout a task and thus it is essential to identify specific aspects of students to adapt to and modeling student information at appropriate level of granularity [41]. In our approach, the user modeling component focuses on evaluation of user performance regarding an important dimension of teamwork—helping behavior. We distinct trainees' knowledge at individual and team level, organize such knowledge into functionally divided training components and build explicit student models that can react to trainee behavior at team or individual level.

2.2.3 Difficulties in Team Training with Traditional ITS approach

ITSs apply advanced cognitive modeling and diagnosis to develop adaptive feedback to each student. The primary objective of this research is to extend approaches used in current ITSs to support team training in dynamic and complex contexts. Yet several fundamental differences between ITS and training system made it hard to directly employ traditional ITS techniques, which include the difference between individual task domains and team-based complex domains, scenario-driven and student-driven pacing of the problem-solving activities and the existence of multiple expert solutions coupled with a lack of explicit, objective evaluative criteria.

As mentioned previously (Section 2.2.2), a characteristic shared by many ITSs is that they infer a model of the student's understanding of the subject matter and use the individualized model to adapt instruction to the student's needs. There exist several traditional ITS approaches in construction of a user model, such as overlay models, differential models, perturbation models, and episodic learner model. Overlay models and differential models view user’s knowledge at any point as a subset of the expert’s knowledge and permit easy comparison between what the students did and what an expert would do [40, 42]. The difference between overlay models and differential models is that the latter identifies the knowledge that the learner is exposed to in addition to the learner’s existing knowledge. In domains where no subset of the expert knowledge could explain the incorrect procedure of the user, an error taxonomy, also called perturbation model is introduced to determine the significance of various user errors/error patterns and to choose appropriate remediation strategies [43]. This approach is theoretically based on impasse learning, repair theory and the notion of cognitive error in specific procedures [44]. Such error taxonomy or bug library is typical in a perturbation model. While the overlay model represents the learner only in terms of “correct” knowledge, a perturbation model combines knowledge of learner normally captured in standard overlay model with knowledge about learner misconceptions that go beyond the expert knowledge set [24].

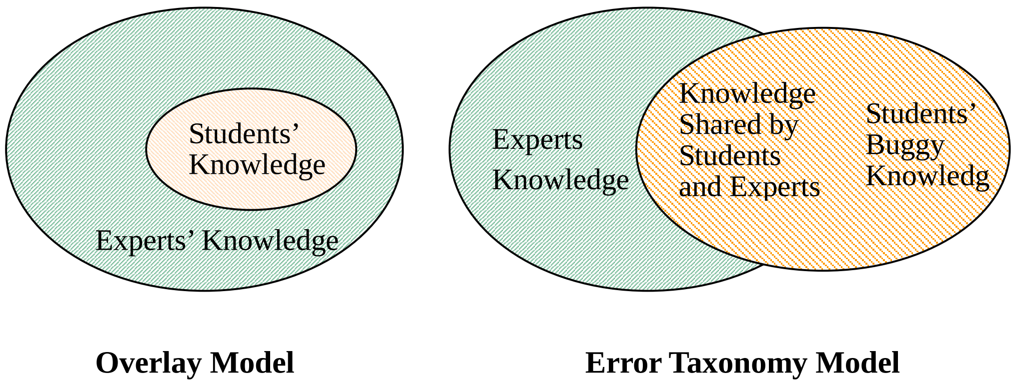


Figure 3 Overlay Model and Error Taxonomy Model

ITS research provides a good source of ideas about how to design student models and how to use it for tutoring, coaching, or training [45, 46]. However, there are underlying difficulties to apply the traditional ITS approaches to team training. As mentioned previously, the overlay model and the error taxonomy model are two commonly adopted modeling approaches in traditional one-on-one ITS paradigm (illustrated in Figure 3). The advantage of an overlay model is the low complexity when student knowledge can be easily represented as declarative knowledge of the domain and a set of production rules

[32]. Yet within an overlay model, feedback can only be provided regarding how student knowledge will exactly match that of the experts. This creates the difficulties in developing an overlay model for team training. Beyond the individual level knowledge and skills, student modeling for team training requires the representation of team dynamics including the sharing of information and the coordination of joint actions. As a typical interactive team model, C2 teams demonstrate intensive communication and collaboration when team members acting together.

Compared to traditional ITS domains, team environments introduce higher complexity, uncertainty, and time stress. Complexity can be viewed twofold: 1) the task demands required by the team task well exceeded the cognitive load of an individual student; 2) relevant team-level knowledge regarding communication and collaboration could not be easily represented by a few rules. Uncertainty refers to the situations when multiple knowledge paths can lead to the same individual action, or multiple solution paths can lead to the same team outcome. Time stress is characterized as the critical time constraints associated with fast-paced team problem solving. A typical example of a team problem could be the training for tactical team decision making, which demonstrates all the above listed features.

2.2.4 Agents in Intelligent Tutoring Systems

The use of intelligent agents as components within an ITS has drawn increasing interests in recent development of intelligent tutoring systems [47-50]. Within an ITS, intelligent agents have been used in modeling the students, building the expert model, and developing pedagogical strategies. Advantages of the tutoring/pedagogical agents may attribute to the flexibility, reactivity and learning capacity of agents in effectively achieving certain pedagogical goals. However most of the agent-based ITS development still falls into the one-on-one tutoring paradigm, with focuses on the individual learner and the issue of how agent technology can enhance the interactions between the individual learner and the learning environment [51, 52].

In this section, we briefly discuss STEVE agents and its extension in group learning environment as successful prototypes that incorporated intelligent agents as part of the individual-based tutoring systems.

STEVE (Soar Training Expert for Virtual Environment) was initially proposed and developed to facilitate student’s learning of individual task [53, 54]. Intelligent agent technology was used for tutoring to enhance the pedagogical process. Specifically, the reasoning and diagnosis capacities of STEVE were built on Soar [55], a generic cognitive architecture for developing intelligent systems.

STEVE pedagogical agents were incorporated into a virtual learning environment to enhance individual student’s learning experience. One of the distinct capabilities of STEVE is that the pedagogical agent is animated as a human within the 3D world. STEVE has a facial expression and a hand that can point to objects in the virtual environment and was developed to facilitate student’s individual task. It helps trainees learn shipboard operations in a navy training domain by monitoring the sequences of trainee actions and providing a set of pedagogical capacities. For example, it can point out student error or answer simple questions being asked, or show demos to guide students within the virtual world.

STEVE agent was then extended to play the role of virtual teammates as well as tutors of students in distributed virtual reality [56]. It stresses on building a new learning environment where students, tutors and virtual teammates cohabit, interact, and learn. STEVE now tracks the actions of multiple students, allows each individual interacts with the world through a set of separated software components and model communication as explicit speech among human and agents. The extension of STEVE creates opportunities for students to collaborate with each other, as well as agent tutors and virtual teammates. It considers the roles that individuals might play in the cohabited virtual reality. However, rather than focusing on how collaboration related team knowledge that is represented and accessed, it still focuses on the task knowledge of how students can locate and operate relevant equipment. The pedagogical intervention is also limited as that it provides instruction in a manner similar to one-on-one tutoring.

2.3 Simulation-based Training

Simulation often refers to the immersion of the user into a virtual environment where the users interact with each other or with computer generated images [57]. Compared to training in real world scenarios, training in virtual environments offers more flexibility and scalability to model human behavior during the course of training. Further, simulation allows training that entails great cost or high risk, such as performing dangerous tasks, incidence management responding to emergency, operating a complex piece of equipment or training of clinicians in health care domains.

Cognitively, training of individual in virtual environments is supported by providing trainee with computerized multi-sensory presentation and real-time interactions with the system [58]. For the purpose of team training, a rich set of computer-mediated communication channels (texts, audio or visual) allows trainees to share knowledge and collaborate in performing complex tasks. However, it is always hard to decide the fidelity of any simulation-based training where the transfer of knowledge and skills to the actual job needs to be ensured.

To design an efficient simulation environment that supports training, several issues must be taken into account, including the targeted training audience, the task requirements, and the training objectives. A number of architectures/models have been developed for modeling human behavior in simulated environments, including knowledge-base systems, mathematical models of human performance, psychologically inspired agent architectures and the basic ITS (Intelligent Tutoring System) concepts and methods. The core team competencies that are necessary to be acquired/learned can be classified as knowledge, skills, and attitudes [59].

2.4 Agent-based Teamwork

Early research on agent-based teamwork can be traced back to the 90s [60]. Among researchers in the Multi-Agent System (MAS) community, there has been growing interests in using intelligent agents to simulate and support human teamwork.

While multi-agent systems offer promising alternative for developing team training, research in this area has been limited, in this section, we will first briefly introduce theories that have been developed as the foundation of teamwork [60-63] and then survey major agent-based teamwork architectures, which allow communication and coordination among team members [64-67].

Agent-based teamwork systems emphasize agents' communication and collaboration as a team in pursuing certain common goals. Within the Belief-Desire-Intention (BDI) agency paradigm[68], several theoretical frameworks were proposed to capture the fundamental aspects of teamwork and system modeling of team behavior. Among those, Joint Intentions theory [60] and SharedPlans theory [63] are two widely accepted theories that provide foundation for modeling team behaviors in a computational environment. Based on them, several agent-based teamwork models have been proposed [65, 67, 69], which we will briefly survey later in this section.

2.4.1 Joint Intentions Theory

Joint Intentions theory specifies how agents can jointly act together as a team by sharing certain mental beliefs about their cooperative actions. Two types of agents' mental states are key to Joint Intentions theory— intentions and joint intentions. Levesque and Cohen define joint intention as a shared mental state among agents, which requires agents not only to have intention (internal commitment) to perform an individual action but also to have shared commitment to perform a joint action. The notion of joint commitment is core of the joint intention model, which allows agents to have their beliefs and goals about the world, yet commit to informing other team members whenever it detects the team's common goal has already been achieved or never will be achieved or becomes irrelevant.

In the joint intentions theory, for an agent to be part of the team, it must communicate its intention to achieve a joint persistent goal. A team is formed when all agents on the team has committed to a goal and believes that other agents have also committed to the common goal. Once a persistent goal has been adapted, the agent holds the belief that the goal hasn’t been achieved yet, and that all agents believe that it is achievable in the future.

The significance of the persistent goal and joint intention lies in twofold: 1) it motivates agents to communicate about their intentions and teamwork, and in this sense each individual cannot freely de-commit (without communication) from a joint persistent goal; 2) Team will try multiple attempts to achieve the joint persistent goal until it is achieved or certain conditions invalidate the goal.

2.4.2 SharedPlans Theory

SharedPlans is an agent teamwork model that provides a formalization of the collaborative activities among a team of agents. Grosz and Sidner argue that the shared mental states among collaborators are essential to ensure collaboration success. Thus SharedPlans are developed as a formal representation of the mental aspects of collaborative activities, including agents’ mental attitudes such as their mutual belief about the team goal, their desire, intention and commitment to the actions that are to be performed. Compared to the Joint Intentions theory, it focuses on teams’ collaborative planning and acting. The first order logic based formalism is consisted of primitive predicates, modal operations, meta-predicates and action functions.

One advantage of SharedPlans theory is the inclusion of the partial shared plans, which described the refinement process in collaborative planning. The collaborative planning can be decomposed into different levels where partial plans are modified over the course of planning by a team of agents involved in the collaboration. The definition of a partial plan elaborates on how plan-based reasoning can be accomplished when recipes for certain actions are incomplete (agents may have partial plan for certain actions in the recipe) or some sub-actions have not been assigned to any agent.

2.4.3 Agent-based Teamwork Architectures

In this section we will briefly review several well-known team-based multi-agent frameworks that aim to be reused across multiple domains, including human team training, multi-robotic mission and internet assistance.

Among the initial attempts in building agent-based teamwork, Tambe's group adopted joint intentions theory to have implemented STEAM model, an explicit, domain-independent teamwork model that enables agents to reason about commitments and responsibilities in teamwork, and flexible plan coordination and communication. In addition, the group has been interested in the organization of agents through team-oriented programming. STEAM has encoded the definition of team readiness and a set of task synchronization mechanism to establish joint commitment. For example, STEAM agents produce robust behaviors by reconciling members' beliefs about the team goals, they communicate with each other to preserve the coherent initiation and termination of team plans and monitoring each other to ensure the repair capacity in case of failures of a critical member agent. Teamcore, as an extension of STEAM, elaborated on the concept of team-readiness and addressed the challenges of constructing collaborative plans for large-scaled agent teams. The programming-oriented approach aimed at saving software development efforts in large-scaled organizational setting and has been tested in application domains such as simulated military evaluation and human collaboration assistance [70].

CAST (Collaborative Agents for Simulating Teamwork) is an agent-based teamwork architecture that has been developed to capture the key aspects of simulating and practicing basic collaborative teamwork skills in a simulation environment. A rich computational shared mental model is highlighted in CAST design, including adaptive team process modeling and a dynamic team structure representation of agents with different capabilities and roles. A high-level encoding language MALLET is developed to specify agents' behavior within the team process model, focusing on the dynamic team aspects. Powered by the team process transition net and the knowledge encoding language, CAST agents can observe and reason about other team members' behavior, communicate and maintain the shared-mental model. The design and implementation of CAST is inspired by all the above mentioned theory of agency, including the Belief-Desire-Intention model, Joint Intentions theory and SharedPlans theory. CAST agents focus on one important communication aspects, which is proactive information delivery. Specifically, it optimizes information exchanges by enabling agents who know some facts to communicate it to exactly those teammates who need the information to carry out their goals. Compared to the STEAM agent architecture, which uses a quantitative utility function regarding communication cost to make decisions about team communication, the proactive information delivery approach in CAST might involve more in-depth monitoring of other agents and some complex reasoning about their beliefs and goals.

Collagen [69] was originally designed to address the need of developing domain independent agent framework where agents can effectively work with people. To achieve this goal, the first step is to ensure a coherent flow of agent-user interaction within the system. Collagen solved the problem by applying human collaborative discourse principles to agents. By providing mechanisms for agents to mimic how people communicate and interact, agents become natural and easy to collaborate with from human user's perspective. What hasn't been addressed by Collagen is the robustness of the discourse processing and the extensibility to a distributed environment.

As a ubiquitous ground for agents' social interaction, RETSINA infrastructure aims at building an open agent community where any coordination is a result of agents' peer-to-peer interaction without centrally imposed constraints. It implemented distributed services that facilitate the interactions between different types of agents. As a heterogeneous team, agents fulfill a partial team plan by assigning themselves to team sub-goals according to authority or capacity constraints. RETSINA supports domain-independent usability and has been applied to many domains ranging from portfolio management to logistic planning. As a computational infrastructure that supports a mixture of agents and humans, RETSINA also aim to investigate a teamwork model that supports agent-based team aiding. Driven by this sub-goal, they identified the functional role-allocation among humans and agents, and applied human factor research on agent teamwork to determine team effectiveness and distinguish successful teams from unsuccessful teams. A case study was also conducted to test the architecture's applicability to agent-based software engineering. Three layers of sub-architecture were derived respectively at the individual, functional and infrastructure level. [71]

2.5 Computer Supported Collaborative Learning

CSCL (Computer Supported Collaborative Learning) is a research area that investigates how technology can support learning process underlying the joint efforts of students working together on specific tasks [72]. It is recognized by Koschmann as an emerging paradigm of educational technology [73]. Compared to the one-to-one ITS that interacts with one student and personalizes the tutoring to the needs of the student, CSCL represents a one-to-many collaborative learning environment where the system interacts with a group of students, imparting the subject knowledge using collaborative learning strategies.

It is another prosperous research community where the teamwork aspect of human users is concerned. CSCL systems are developed to help learners achieve better learning outcome by participating in team learning processes. Specifically, the goal is narrowed down to create learning environments where learners have the opportunities to acquire knowledge through intensive interaction with each other. CSCL focuses on supportive learning environments that help students learn collaboratively; whereas the intelligent team training system stresses more on the intelligent coaches' role to help trainees achieve better performance by monitoring, diagnosing and generating appropriate feedback on specific teamwork aspects. With a different research goal yet similar team context, CSCL systems also address a set of fundamental issues of great interests to the intelligent training community, such as how to help members achieve better teamwork interaction in terms of group situation awareness, informal communication, negotiation and helping.

3 An Agent-based Coaching Framework for Team Training

3.1 Overview

The training objective of the agent-based intelligent training system is to help trainees learn how to work together as a team and collaboratively achieve a mission. Specifically we design agents to monitor trainee's helping behavior and provide feedbacks regarding this important dimension of teamwork. The detailed data collected by coaching agents eventually has to be attached to a specific domain regarding trainee performance; yet the general team aspect we model—helping behavior with a team setting, is domain-independent. In designing and realizing the intelligent training system, we seek answers for a list of fundamental questions: 1) How to design an intelligent training framework that is applicable to many complex team domains? 2) How do we address different aspects of trainee performance that are highly interdependent yet have distinct sets of characteristics? For example, team performance regarding mission planning and team performance regarding mission execution. 3) How to design coaching modules to model trainee interactions, diagnose their performance deficiencies and generate coaching feedbacks?

To achieve the domain independency of this coaching framework, the first step is to make sure the overall training problem at hand has some degree of applicability to other similar domains. It leads to the choice of helping behavior as our training goal. The next step is the identification of the planning of team mission as a ubiquitous team process. Planning has been intensively addressed in multi-agent system (MAS) literature to facilitate team members' coordinated actions, yet it hasn't received adequate amount of attention in the ITS (Intelligent Tutoring System) literature. Planning and execution together makes a full solution path to a complex team problem, yet each is associated with different levels of complexities.

To better address different aspects of the training problem with different complexity, the training process is divided into a planning phase and an execution phase. The planning phase of training allows trainees to anticipate team mission and plan ahead. We built coaching agents that provide feedback regarding trainees' planning of team collaboration; the execution phase allows trainee to carry out the plan they made during the planning phase and make appropriate online adjustments to mission dynamics. For team performance in this phase, execution phase coaching agents were built for improving team's collaboration process.

Due to the complexity involved in team training (section 1.1 and 1.2) and the difficulties in directly applying the traditional ITS modeling approaches to team training (section 2.2.3), we divided the overall training problem into two sub-problems. Accordingly, during the course of training, we applied a two-phase training protocol that has a planning phase and an execution phase. For each of the training phases, we developed a specific type of coaching agents that can assess and diagnose trainee performance in particular aspects, as a way of solving the overall training problem.

During the planning phase, coaching agents look at a trainee's performance regarding planning of allocating their resources to balance workload for the whole team.

The complexity of the planning sub-problem is relatively low; trainees are required to identify the imbalanced team workload distribution and make decisions about where to allocate individual resources. Accordingly an overlay model can be developed in comparing trainee's resource allocation strategies with the set of expert planning strategies.

Compared to the planning phase coaching agents, the execution phase coaching agents assess trainee performance regarding much more complex team tasks. The execution sub-problem involves not only the detailed timing information for each domain action at the individual level, but also the team members' real-time interaction, such as team members' proactively helping each other and actively communicate with each other in facilitating collaborative behaviors. In this phase, the overlay model is no longer suitable for performance assessment and diagnosis. Instead the complex team problem with no single answer implies that we didn't aim at providing trainee a comprehensive set that contains every possible solution. Instead we adopted an error taxonomy approach and built a buggy library to diagnose training deficiencies. Within the error taxonomy/perturbation model, related trainee events are linked at both individual and team level to assess the pattern of helping behaviors.

The mission-planning-phase coaching focuses on evaluating the resource allocation plan that trainees develop at the beginning of each training session. System inputs include distribution of team workload and resource capacity of each decision maker. We implemented a scoring algorithm to decide whether trainee's placement plan is efficient by comparing trainee plan with a set of expert planning strategies. The offline coaching agent then generates feedbacks to trainees about how they could improve their planning of team resource allocation. The mission-planning phase coaching focuses on team's planning of mission based on roughly predicated mission characteristics (e.g. for our task domain, the incoming task pattern) without considering the detailed timing of mission execution for each trainee. During the mission-execution phase, trainees' team resource allocation plan can be used to determine whether trainees' deviation from their original plan is an effective adjustment due to execution dynamics or it is an undesired deviation from the expert execution model.

For the mission-execution-phase coaching, we developed CAST-ITT (CAST-Intelligent Team Training) System coaching agents to automate the event-based training approach. Basically an event-based training cycle begins with pre-mission information presentation, and then moves on to demonstration, practice, feedback and goal-setting for the next exercise [74]. In the following sections of this chapter, we will lay out the connections between each functional component of AITC (Agent-based Intelligent Team Coaching) framework and the corresponding step within the event-based training methodology.

The integration of CAST agent framework and the distributed simulation domain enabled us to collect both operational performance data and a set of collaboration related performance data during team's mission execution. The operational performance data includes individual's launching, moving and returning of their assets and their identification and attack of incoming tasks. To gather the collaborative performance matrices and ensure that collaboration is reinforced during the course of training, we designed the training scenario so that the team's workload is highly unbalanced during each mission execution— some individual might be overloaded, who has more workload than her/his own resource capacity could handle, some might be idle, who has less workload than her/his resource capacity could handle. If each individual only manages her/his asset to handle individual workload, the overloaded member's performance becomes the bottleneck of their team performance—it is impossible for this individual to finish the workload with the limited resources in a short period of time and thus the team mission cannot succeed. However, the overall team resource is adequate to cover the total amount of workload assigned for the team and team mission success can be achieved if members within the team actively work on developing strategies to assist the overloaded teammate(s) and help those as much as they can.

Based on the team's workload distribution and individual's resource capacity (how many assets each member has, how powerful these assets are, constraints on using these assets and etc), trainees need to identify which teammates is under critical situation where the individual's own resource capacity is far below her/his workload. To determine whether the trainee did a good job regarding helping, which is the central collaboration dimension in this study, our coaching agents quantitatively model trainees by detecting critical helping events, focusing on two domain specific tasks that contribute significantly to team's collective performance and team outcome.

Given the nature of the problem, we chose to apply both quantitative and qualitative approaches during the design, implementation and evaluation of the intelligent training system. During feedback generation, coaching agents take quantitative input (relevant trainee information) from our ITS components with different functionality, apply artificial intelligence (AI) reasoning, generate training feedback at both individual and team level and present them to trainees via a set of user-friendly graphical user interfaces (GUI). To evaluate the usability and efficiency of the system, we incorporated qualitative survey in which we collect user information and feedback from trainees, such as their game experience and their feedback about the agent-based planning and coaching tools.

3.2 Issues in Achieving Domain Independency

A theme underlying much of ITS research is domain independency, which refers to the ability of the intelligent training system to be extended and applied to multiple domains. To make an intelligent training system reusable for different domains, the system designers need to extract the essential characteristics across a set of the targeted pedagogical domains and achieve as much domain independency as possible for each ITS sub-component, such as knowledge encoding within the student model, or the deficiency diagnosis within the assessment model. We are aware of the importance of being able to apply our intelligent training system in to a set of team-based complex domains. Several issues are discussed below regarding how we achieve domain independency in designing the intelligent training framework.

3.2.1 Coaching to Enhance Helping Behavior

In this study we focus on arguably one of the most important dimension of teamwork — helping behavior, to guide our design and development of the intelligent training framework and its coaching assessment modules. Accordingly, our major training goal is identified as enhancing trainees’ planning of helping related resource allocation and performance regarding helping each other during mission. Helping behavior, defined as helping others perform their roles [17, 18, 75], represents a team generic dimension that occurs ubiquitous in many real-world team scenarios.

Helping behavior is a widely recognized as a generic team dimension whose legitimacy can be characterized as interactions between two domain-independent team inputs. In making helping legitimate in the team training context, we also looked at two generic aspects of a team in performing a common goal, including the team composition and the other is the team task characteristics. Specifically team composition in terms of resource implies the distribution or allocation among each member, team task characteristic implies how many workload each team member will be responsible or are expecting in achieving a mission. The interaction between the two types of team inputs has great impact in creating the needs of helping behavior. A typical example that makes helping behavior essential to achieve team mission is the imbalanced distribution of the amount of team resource and workload on each individual. If an individual hasn't been allocated enough resources to manage the workload assigned to her/him, helping behavior is expected from other team members (who may have extra resources) to ensure the overall performance efficiency of the team.

In summary, none of the characteristics of helping behavior has domain-dependent constraints in designing the intelligent training system and thus the choices of helping behavior as our main training focus set a solid foundation for the domain independency of the intelligent training approach.

3.2.2 Two-Phase Coaching

Planning is one of the most effective ways to increase individual or group productivity. There are growing interests in studying human behaviors involved in a variety of planning activities. For example, in business, the separation of planning from operations in middle management; in highly organized military teams, complex mission planning before any critical mission is carried out.

Our training approach identified the importance of team planning activities and divided the overall training problem as two sub-problems: one regarding team's planning of future mission activities, the other is the execution of the mission. Our hypothesis is that team can achieve better mission performance by taking advantages of the team decisions made at the planning phase, by better anticipating mission situation, and proactively adapting to online changes. The domain independent nature of planning reinforced the extensibility of our training approach to other team domains.

We distinguish the issues of how trainees plan out the collaboration with a focus on helping and how they carry out the plans by adapting to online team dynamics. We design the training protocol so that trainees are required to plan ahead about their collaborative behavior in the planning phase and then carry out their team plan while actively adjusting to the changes during mission execution. The distinction and separation of planning from execution is also key to address the problem of monitoring and assessing a team of trainees' helping behavior. The collaboration process is divided into the planning and execution of team collaboration, two types of feedbacks are generated each address a different aspect to enhance trainees collaborative performance. As two highly interdependent processes, mission-planning and mission-execution are both critical in achieving teamwork success.

Figure 4 illustrates the two-phase training approach and the interdependency of the two coaching phases.

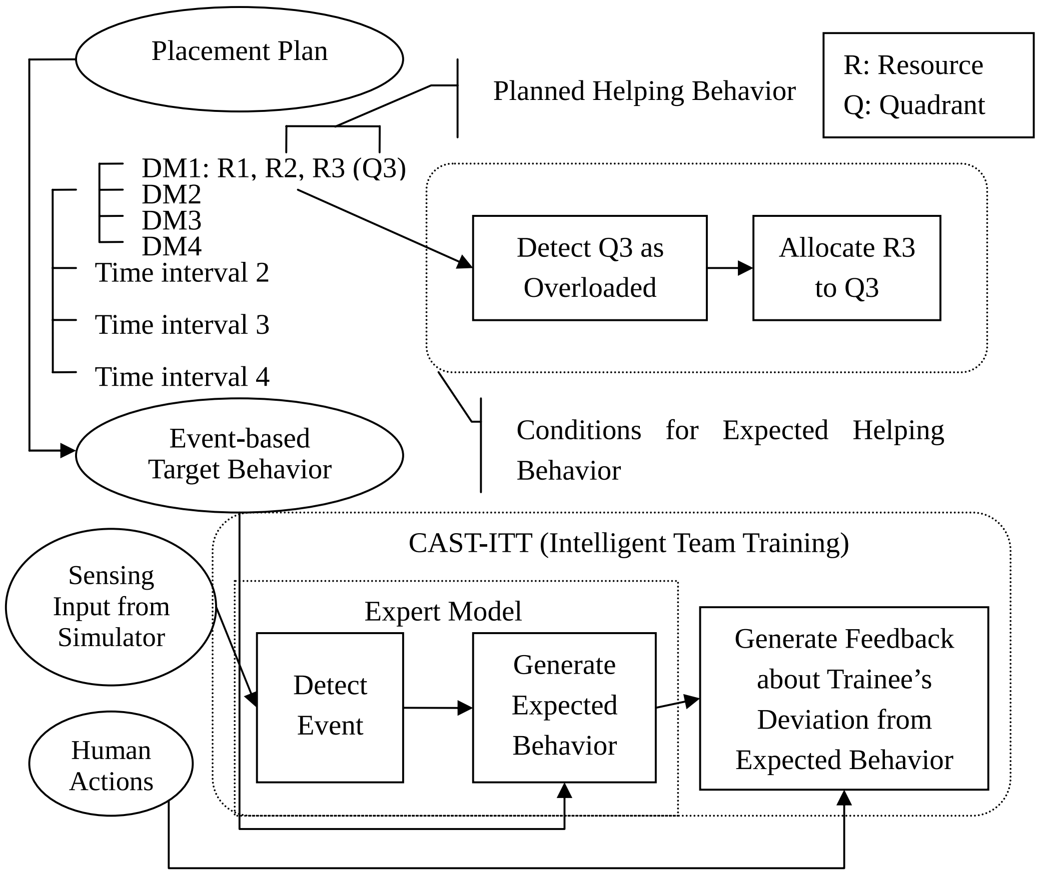


Figure 4 Two-Phase Coaching: Planning and Execution of Team Collaboration

3.2.3 Event-based Training Approach

During the mission-execution phase, coaching agents need to monitor trainees' online performance, diagnose their taskwork/teamwork deficiencies and provide feedback to trainees addressing their online execution deficiencies at both individual level and team level. The complexity of performance coaching requires that we identify a well-developed theory or methodology to guide the design of our execution phase coaching agents.

Event-based training provides a set of structured methods, strategies and tools that are essential for training and testing of specific knowledge, skills and attitudes [76]. In a simulated domain, it enables effective training through simulated scenarios where relevant events can be introduced and identified to test specific skills. The cornerstone of event-based training is a process-based performance measurement component that is linked closely to training objectives embedded in pre-specified scenario events [77]. An event-based training includes the following procedures:

a) Identify training objectives through cognitive task analysis

b) Translate training objectives into a set of scenario events

c) Identify performance criteria that represent achievement of targeted objective for each event

d) Translate performance criteria into event-based measures for team performance processes and outcome

e) Provide remediation with appropriate details regarding team performance

As we mentioned in the overview session, our training objective is to help trainees learn how to effectively plan and execute a complex mission as a team, with a focus on team helping behavior as one of the most important dimension of teamwork. We introduce the event-based training framework into our second-phase of training, and integrate it with an error taxonomy model to realize agent-based assessment and feedback regarding team performance during mission execution. Event-based training has been intensively studied by cognitive researchers in the field of human team training [76, 78, 79]. In the rest of this section, we briefly describe the event-based training cycle with respect to our specific training objective.

Event-based Training Cycle
1. Skill inventory and learning objectives
2. Scenario scripts and triggering events
3. Performance measures and criteria
4. Performance diagnosis
5. Feedback and remediation

Figure 5 Steps of an Event-based Training Cycle

The first step involves the identification of learning objectives and the creation of an inventory of the targeted skill set. Shown by the Cognitive Task Analysis (CTA) of a previous psychological study [80], helping behavior is a generic competency that is required for a team of trainees ranging over many tasks. Specifically when there is a mismatch between resources and capacity for individual members, yet at the team level there is no such resource and capacity mismatch, helping behavior among members on a team becomes a critical skill to ensure team's overall mission success.

As the learning objectives being identified, we need a set of trigger events to test the trainee's mastery level of operational task skills and collaborative skills such as helping one another within the team. For example, certain conditions can trigger a desired domain event. To test for possible deficiencies, our coaching agents monitor trainee actions around a particular event, compare trainee actions with the behavioral representation of the desired course of actions, and generate feedback to address the corresponding deficiencies identified.

Performance measurement criteria must be explicitly identified to assess trainee's performance that reflects a team's taskwork and teamwork competency. To automate the event-based performance measurement and assessment, we take the measurements identified through expert observations and semi-automation, and focus on the deficiency measurements to design the intelligent training system. We have identified the corresponding measures for a flexible and complex team domain and designed evaluation algorithms for both operational measurement and team process measurement.

As trainees go through the training scenarios, relevant process and outcome data is collected around a set of critical domain events. Our coaching agents then establish patterns of the individual and team behaviors of each team member, and determine whether these patterns meet the requirements of the desired behavioral representation that is referred to as the team performance model in the AITC framework. To build a team performance model, a set of deficiencies regarding mission execution objectives needs to be extracted from the DDD domain. Defined at the highest level, execution objectives can be identified as the goals of each DM during the game at both individual level and team level. Respectively at the individual level, the goal is to destroy all hostile tasks that enter their restricted quadrants and at the team level, it to help one another as much as needed to maximize team outcome. Additional individual level guidelines may include constraints in performing certain domain tasks or recommended domain actions to perform upon a certain event. To address our training of collaborative skills, it is crucial to develop a set of buggy heuristics (as opposed to expert strategies) that specifically address trainee's deficiency regarding helping performance when a set of necessary conditions for a helping event is recognized. We constantly improve our execution-level perturbation model as the coaching agents has been designed, implemented and used for training. Again, we may not have a complete set of every detailed trainee deficiencies regarding their taskwork or teamwork behavior, yet we believe that the current coaching feedback generated by the execution assessment module is of some degree of help to both individual's operational and collaborative performance.

A final stage for any training system development is the design of feedback generation and presentation. According to the diagnosis, constructive feedback can be provided immediately or at the completion of the task. In this study, we focus on tailoring our feedback to address collaboration related deficiency, especially those concerning desired helping behavior. There are quite a few design principles regarding feedback provision, we briefly discuss some of them in section 3.2.6 with a focus on our training objective.

3.2.4 An Agent-based Intelligent Coaching Framework

We discuss the design and implementation of the execution phase coaching agents in the context of an Agent-based Intelligent Team Coaching (AITC) framework. The AITC framework is initially proposed and designed as joint efforts among researchers working on a cross-discipline MURI project [66, 81-83]. As a generic team training approach, the AITC framework is also denoted as “CAST-ITT” framework s an extension of the CAST (Collaborative Agents for Simulating Teamwork) agent-based team architecture [84, 85].

We introduce the functional components designed to realize execution-phase coaching, and focus on how these intelligent components automate the event-based training approach, including the monitoring of trainee actions, diagnosing of trainees’ performance deficiencies, and the generation of feedback regarding the execution of a team plan.

As a high level training framework, AITC coaching framework is functionally divided into several intelligent training components. In the original CAST (Collaborative Agents for Simulating Teamwork) architecture, intelligent agents serve two major roles. One role is to be the virtual team members who perform similar tasks to reduce human trainee’s workload; the other role is to monitor trainee actions while trainees execute the mission. In designing the coaching components within the AITC framework, we utilize the monitoring capability of the interface agents from the CAST agent framework and add assessment capabilities to diagnose trainees’ performance deficiency with a focus on collaborative behavior and provide coaching feedback to help the team enhance its performance process and outcome.

In Figure 6, the functional components of execution phase coaching agent are shown within the agent-based intelligent team coaching (AITC) framework, which supports user modeling and coaching feedback to the team regarding members' teamwork performance; the underlying CAST agent components are shown on the right side of the diagram; the coaching agent components are shown on the left side of the diagram. In particular, the "needs anticipation" component of CAST is inherited by the coaching agent to identify events that require helping behaviors.

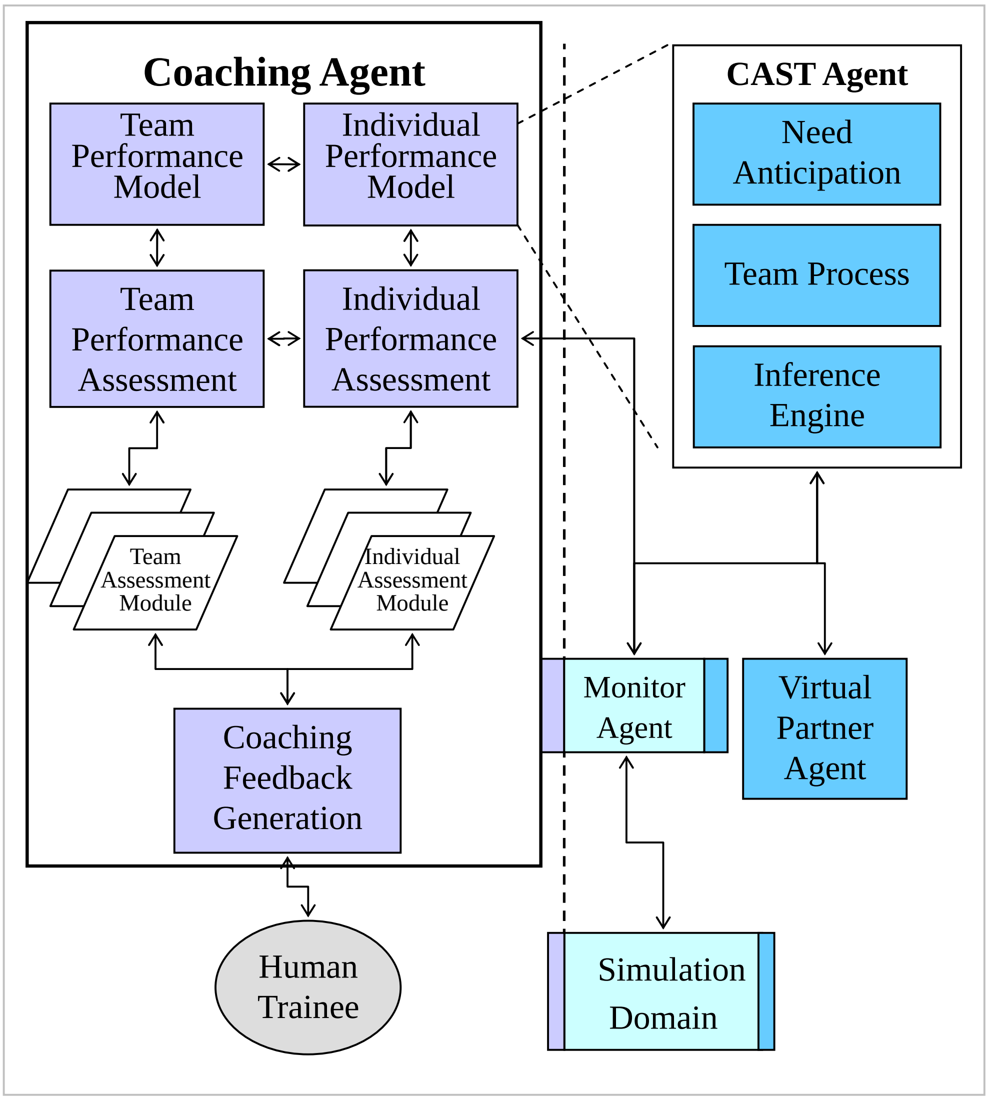


Figure 6 Agent-based Intelligent Team Coaching Framework for Performance Assessment

The coaching agent consists of four high level components: the individual performance assessment module, the expert model (for individual-level and team-level execution strategies), the team performance assessment module and the coaching feedback generation module. Outside the coaching agent box, a monitor agent is a bridging component between human trainees and other CAST virtual team members in the domain.

A monitor agent runs on the background for each trainee in the domain, who is responsible to dynamically collect trainee's online performance data and passes it on to the individual performance assessment module. In collecting performance criteria (Step 3 in an event-based training approach), we constructed performance models that contain error taxonomy of performance deficiencies, which indirectly help trainees achieve the desired online individual and team performance. It provides a set of performance measures to be compared against trainees' event-based actions captured by monitor agents.

In realizing performance diagnosis (step 4 in an event-based training approach), the individual performance assessment module analyzes data as it relates to critical domain events and identifies trainee's individual performance deficiencies by communicating with the performance module where a set of individual deficiency measurements is used to assess trainee's event-based individual performance. Each individual performance assessment module also serves as a student model that provides sufficient information about the trainee to the team performance assessment module.

With a collection of all the individual user models, the team performance module analyzes data as it relates to collaborative events. A set of trainee's team-level deficiencies are developed to serve as team performance measurements, e.g. team members' error in helping each other with particular domain operations to maximize team performance. Such trainee deficiencies are categorized and stored in team performance module, indirectly representing the desired team-level execution. The team performance model then provides the team-level performance assessment model with the deficiency measurements so that the team performance assessment module is able to detect trainees' team performance deficiencies.

As the last step of event-based training, the coaching feedback generation module takes the deficiencies diagnoses of both individual and team performance assessment modules and then generates appropriate feedback to help trainees enhance their performance. At the individual level, it contains remediation for trainee's operational deficiency; at the team level, it focuses on deficiencies in carrying out the helping related collaborative events. Feedbacks are provided to trainees at the end of the mission execution to help them improve next mission.

The individual and team assessment of performance realized the performance criteria and performance diagnosis in the event-based training framework. It established a team performance model of trainees that distinguishes our work from the traditional assessment modules where individual performance is the main focus. Coaching feedback generation module implements the last step in the event-based training approach and provides offline feedback to help trainees improve future mission performance. In a military training scenario with human coaches, an after action review (AAR) or debrief session is conducted at the end of each training session. With our intelligent coaching agents, offline feedback is provided as part of the AAR or debriefing. We believe that offline feedback can be less intervenient to trainees' time-stressful task, allowing them to make mistakes during mission execution and provide deficiency remediation for them to learn from these mistakes.

3.2.5 Link Event-based Approach with the Intelligent Coaching Framework

As we mentioned in section 3.2.3, event-based training provides a set of structured methods, strategies and tools that are useful for team based training and testing of specific knowledge, skills and attitudes [24]. In our designing of the execution phase coaching agents, we developed event-based error taxonomy by adopting the event-based training approach as part of our perturbation model to diagnose training deficiency. Table 3 lists several key steps in a generic event-based assessment approach and relates each step to the functionality of the intelligent team training components discussed in the agent-based intelligent coaching framework for team training (Figure 6).

Selected Steps in Event-based Assessment Approach	**Related Design in CAST-ITT framework**
Identify performance criteria that represent achievement of target objective for each event Identify measurement that can be used to assess trainee performance reflecting both taskwork and teamwork competency	*Individual Performance Model* – encode a set of task-specific misconceptions in agents' knowledge base, serving as the performance criteria to identify trainee's operational performance deficiency *Team Performance Model* -- encode a set of teamwork-oriented misconceptions in agents' knowledge base, serving as the performance criteria to identify trainee's collaborative performance deficiency
Translate performance criteria into event-based diagnosis for team performance processes and outcome Diagnose trainee deficiency regarding helping related performance	*Individual Performance Assessment Module* – collect relevant trainee performance data regarding critical domain events (e.g. a list of trainee's domain actions as part of a helping event) and assess trainees' individual performance by taking inputs from the *Individual Performance Model* *Team Performance Assessment Module* – a team model representing how team members interact during mission execution and assesses trainees' collective performance. Take inputs from the *Individual Performance Assessment Module*, and team-level *Performance Model*, it and diagnoses a set of trainee deficiencies regarding team collaboration
Provide feedback at appropriate granularity level – Operational taskwork and collaborative teamwork	*Feedback Generation Module* – organize and present feedback to the team based on the diagnosis results generated by both individual and team performance assessment modules

Table 3 Event-based Assessment Approach and Related Components in AITC (Agent-based Intelligent Team Coaching) Framework

The execution-phase coaching agents need to diagnose trainees' performance deficiencies and provide feedback to improve trainee's teamwork performance and outcome. Yet due to the complexity of team interactions within the open problem space, it is not feasible to build a comprehensive model of expert behaviors that lead to a unique correct solution. On the other hand, it is also not efficient to build a comprehensive model of trainee behavior regarding each domain operation conducted. To address the difficulty of building the expert model, we employed a widely adopted ITS (Intelligent Tutoring System) approach—the perturbation or error taxonomy model where we can focus on trainee deficiencies instead of providing them the comprehensive solution. To address the difficulty of building a comprehensive student model, we took the event-based perspective, which emphasizes the analysis of desired trainee actions by looking at crucial steps to achieve an execution outcome and the trainee deficiencies that significantly imply trainees' helping-related misconception.

Each intelligent training component in Figure 6 is designed to realize a particular step in the event-based training paradigm. First an event is identified by the monitor agent. The performance assessment modules then evaluate trainee actions regarding the desired performance criteria. The performance assessment modules consist two components at different assessment level: Individual Performance Assessment Module that assesses individual's task performance and Team Performance Assessment Module that assesses the collaborative performance of the team based on the assessment results collected from each individual performance assessment modules. The desired performance criteria are encoded in two expert models— Individual Expert Model and

Team Expert Model, correspondingly they can be accessed by the individual performance module and the team performance module for deficiency diagnosis purpose.

3.2.6 Design of Feedback Provision

One of the novelties of this research is that we designed different types of coaching agents to address different aspects of the overall training problem. The abstract level of coaching feedback is twofold: 1) individual-level feedback vs. team-level feedback, addressing taskwork and teamwork performance respectively. 2) planning-phase feedback vs. execution-phase feedback, addressing team's planning of mission execution or their online execution of mission. As part of the training protocols, the interactions between planning and execution imply the underlying time constraints: trainees first go through planning phases and then carry out their plan during the execution phase. Feedback regarding each phase can be generated at both individual and team-level.

The above listed characters of coaching feedback within our ITS framework rooted in a domain independent model of feedback provision, which basically contains three levels of analysis: a) provide feedback at the appropriate abstract level; b) provide feedback at the right time; c) provide feedback in the right format. In the rest of this section we elaborate on each level of analysis and demonstrate that the design choices we made for feedback provision are also guided by the domain-independency principles.

When a team task fails, it could be attributed to trainees' failure to comprehend the specific domain knowledge or rules, or to their failure to collaboratively act together as a team. It is desired that coaching agents identify the cause of certain team performance deficiency, convert the diagnosis results into constructive feedback and communicate the remediation to trainees to help them focus the improvement at a specific abstract level, which could be either their lacking of essential task skills or their lacking of team-oriented active interaction.

Research shows that context specificity is very important to learning [86]. Thus the question of when to provide coaching feedback has been discussed intensively in the literature. Instructions could be given before each problem, in the middle of trainee's solving the problem or after the problem has been solved (or failed to be solved). The answer to this question is highly dependent on the training goal and the specific domain tasks that are involved in the training process. (e.g., for information-intensive tasks, trainees may find it interfering with their own problem solving if feedbacks are presented at the precise point whenever a mistake is committed [27]). To avoid disturbing students from their task focus during training and to help them learn from their own mistakes, our coaching agents present feedbacks at the end of each training session where the coach agent has collected adequate amount of information from trainees during mission execution. In analogy to the military training debriefing scenario, the offline feedback is provided in the after action review (AAR) session where team members are presented with the remediation generated by agents and allowed to discuss accomplishments and failures in a previous mission right after the completion of that session.

Feedbacks can be presented in a variety of forms. They could be remediation of trainees' deficiencies organized into different categories, a demonstration of a success scenario with detailed explanation, or a question-and-answer session if further help is requested by trainees. Later in this chapter, we will lay out possible feedbacks that can be generated as a team performs differently with regard to our training objective and its performance leads to different deviation levels from the original placement plan. However the design of the generic intelligent training system would not constrain the forms of feedbacks that can be provided.

3.3 The Task Environment—DDD Simulation Domain

Our research explicitly focuses on the measure of effective teamwork performance and helping behavior among team members. The Airborne Warning and Control System (AWACS) team has been intensively studied in terms of member roles, responsibilities, and interdependencies and is among our choices as one of the most effective domain to study team helping behavior. AWACS team serves as a vital Command and Control (C2) node, providing airborne surveillance functions for tactical defense forces. They detect, identify, track, and intercept airborne threats. To perform such complex and dynamic tasks under resource and workload imbalance, team mission success can only be achieved if each team member helping each other to balance the team resource distribution with their individual workload.

Among the simulation domains available, DDD (Distributed Dynamic Decision-making) replicates a fast-paced, stressful, and complex warfare environment [87]. Given the domain independent nature of team helping behavior and the research assumptions we have, we chose DDD simulation as our experiment domain, and we believe that it provides an appropriate context for us to design and exercise our training protocol and to conduct experiments that can evaluates the effectiveness of the tool-oriented coaching system.

The generic DDD simulation is developed by the Department of Defense for research and training purposes. It is a real time command-and-control simulator that has wide flexibility to play training scenarios with various complexities. Each participant is a member on a decision making team and plays the role of monitoring incoming tasks though radar and works interdependently to protect a restricted airspace from hostile tasks. Each decision maker is in charge of a particular quadrant and is responsible to detect, identify, attack and disable enemy tasks using available resources, including a number of assets with different capacity. There are two major domain tasks – identification (ID) of incoming tasks and attack of incoming hostile tasks. Different assets have different capacities in terms of these two domain operation, specifically they have different radar ranges in performing these actions. For the attack operation, one of the important domain rules is to use adequate powered asset to attack incoming hostile task. Different assets have different power level and these power levels need to be considered when choosing the appropriate attack asset for an identified incoming hostile task.

We set up our experiment in a way to make teamwork essential. Individuals are required to help each other to ensure team’s mission success. Specifically, for each scenario, the highest workload distribution is shifted among different time intervals when all the teammates share the same level of resources; thus the uneven levels of task demands create clear and direct needs for helping behaviors.

The game screen is a grid geographically partitioned into four quadrants of equal area (NW, NE, SW, and SE). Each Decision Maker (DM) is located in one of the quadrants and is assigned to protect that area. The main goal is to prevent hostile tasks from entering team's restricted quadrants, including an outside green square and an inside red square. The red square is located in the center of the screen where team defense score will decrease rapidly when enemy tasks stay in this critical area; similarly, inside the green square team defense score will also decrease at a steady and relatively slower speed.

For each DDD game, there is a scenario file that controls the pattern of the incoming tasks and the resource capacity of each decision maker. Enemy pattern includes the number and power of the incoming tasks, when and where they will appear on the screen, their movement specification before they disappear. Resource capacity of a DM refers to the number and the capacity of the asset, which includes asset's observation, identification, attacking range, attacking power, and moving speed. Each specific scenario used in this research is divided evenly into four time intervals. During each time interval, the decision making team will encounter unequally distributed tasks among its members. An individual DM is considered overloaded when it is expected that a large amount of incoming tasks will come to her/his quadrant during a specific time interval. In this case, she/he needs help from the other three decision makers with comparably lower load. Among the three team members who are not overloaded, one has a very light load, the other two have medium amount of loads.

At individual level, each team member is responsible for protecting the restricted quadrant in her/his own quadrant from incoming tasks; at team level, all team members as a whole are collectively responsible for protecting all the restricted quadrants in four quadrants. In our study, we set up the experiment so that in each time interval, one quadrant that is particularly overloaded. This creates the necessary condition for collaborative and supporting behavior. In other words, the team overall capability and resources can successfully tackle the problem at hand; yet each member needs to identify the mismatch between individual resource and task-load and help the overloaded teammate by identifying or attacking tasks. Team mission can only be achieved when individual resources are allocated in a way that maximizes team resource usage. The computer supported DDD team can be characterized as a tactical decision making team where: 1) Team members make decisions under critical time pressure 2) Team members must coordinate and help each during the course of team mission execution 3) The achievement of effective team mission is beyond the capability of the team if each member chooses to carry out tasks independently.

Despite the fact that our laboratory context is a computerized simulation, the various features of DDD allow us to objectively measure the trainee’s helping behavior in mission execution phase.

3.4 Realization of Two-phase Coaching

In this section, we discuss the detailed implementation of the two forms of coaching agents. Each corresponds to one of the training phases of the team task—coaching agents in mission-planning phase and coaching agents in mission-execution phase for trainees’ execution performance. The mission-planning phase coaching agents provide trainees with feedback about their planning of team resource allocation and the execution phase coaching agents provide trainees with feedback regarding their execution of the resource allocation plan, their team performance and outcome. To allow trainees improve their planning and execution strategy for the next mission, both types of coaching feedbacks are provided during the After Action Review (AAR) session at the end of each training session. Coaching agents for both phases have been implemented and tested in human subject experiments at this time; the details about the human experiment will be discussed in chapter 4.

During planning phase, we provide trainees with a planning tool and additional information about incoming task pattern to help them come up with a team resource allocation plan. The planning-phase agents examine their plan and diagnose the plan deficiencies that might exist in terms of how the team allocates their limited resources. The diagnosis is generated for any team resource allocation plan trainees made, and the feedback is presented to the team at the end of their mission execution.

The mission-planning phase addresses a critical element among various team performance measures—team assignment based on each member’s workload and capability. Inappropriate assignment of responsibilities would result in unacceptable performance for both the individual team member and the team as a whole. On one hand, coaching agents evaluate the feasibility of trainees' resource allocation plan, and help the team identify early mistakes during planning session; On the other hand, they gather a set of resource allocation strategies that can be used as part of the expert model to measure trainees’ collaborative performance during mission execution session.

The mission-execution phase is more complex in terms of modeling the trainees, as team members dynamically go through the scenario and carry out their resource allocation plan. In this phase, coaching agents monitor trainee actions during their online execution and generate feedback for trainees to review after each mission is complete. We take individual performance measures and assessment as part of our after-action-review feedback as it contributes to team performance and outcome. Feedback on team's collaborative behavior is generated by organizing assessments into several team event categories. Such collaborative events include assist identification (ID), assist attack, active communication and etc.

3.5 Coaching Agent for Planning the Team Collaboration

To facilitate trainees' learning in the planning phase, we provide the team with additional information in forms of an intelligence report and a software planning tool to facilitate their generation of a team resource allocation plan. The intelligence report gives the trainees an overview of the task load in each quadrant, and detailed information about when and where (in which quadrant) the incoming tasks will arrive [88, 89].

Via the planning tool interface, information contained in an intelligence report is graphically displayed on four quadrants each corresponds to each DM's responsible area on the game screen. Each individual can have an idea of the distribution of team workload, as a team, they can then communicate, negotiate and make decisions about team resource allocation.

Each team mission is equally divided into four time intervals; for each time interval, there will be a wave of tasks coming into team's restricted zones. Individual team members are encouraged to identify overloaded quadrant for a particular time interval and decide if they need to allocate assets to help DM with critical domain tasks (e.g. Identification or ID). Overloaded quadrant in a specific time interval is the one with the maximum number of incoming tasks. With a careful analysis of the intelligence report, team members can predicate the incoming task pattern, identify who will be overloaded in a particular time interval during the mission and plan about which assets should be offered for helping the overloaded quadrant if help for a particular domain task is needed. They use a planning action panel to select targeting quadrant for each one of the assets; these decisions are then encoded as a text-based file where team planning decisions about resource allocation are recorded.

Thus at the end of the mission planning phase, the team has produced a resource allocation plan that records the planned team asset allocation in terms of which quadrant to place a team asset during each time interval. The plan captures team decisions about who is going to help, who is going to get the help and what types of assets to be offered for help. The mission-planning coaching agents will then evaluate the team resource allocation plan and provide feedback to help trainees improve their resource allocation planning. Later in this section, we will explain the details about how the planning-phase coaching agent evaluates a team resource allocation plan.

The mission-planning phase coaching addresses the critical issue of resource allocation by evaluating the feasibility of a team's resource allocation plan for each training session. Feedback is presented at the end of each mission-execution to improve their planning of team asset allocation. To evaluate each resource allocation plan, the number of needed assets within a proper range is first acquired and significant deficiencies are identified if the number of planned assets is not within that range. To provide more detailed feedback, a scoring algorithm has been developed to decide whether team's resource allocation plan can balance team resources and work load—after a redistribution of resources among members, team would be able to achieve mission success of destroying all the incoming tasks. The algorithm evaluates each trainee resource allocation plan based on a list of prioritized expert strategies regarding how each member allocates her/his resource to balance the overall workload and capacity as a team. Further description will be given for the algorithm when we discuss the detailed implementation of planning-phase coaching.

We focus on two major domain tasks to evaluation of the feasibility of trainee's resource allocation plan and provide helping behavior related feedback. The domain tasks are identification and attack respectively. To ensure execution success of either task, trainees need to allocate the appropriate asset that is suitable for the particular task to a quadrant that needs the help. Such resource allocation behavior needs to be reflected in the team plan that they developed during the mission-planning phase.

3.5.1 Diagnose Significant Deficiencies:

At a high level of diagnosis, evaluation of trainees' planned resource allocation can be qualified by detecting whether the number of assets for a particular type of domain task is within an appropriate range. The expected ranges are derived from domain knowledge of subject matter experts, specifically their resource allocation strategies.

Domain Task Categories	**Identification**	**Attacking**
Minimum number of assets	1 identify asset	2 attack assets
Maximum number of assets	3 identify assets	6 attack assets

Table 4 Quantify General Placement Plan for Domain Task Categories

Assets with certain operational capability can be used to fulfill a specific domain task (e.g. identification or attack) in either trainee's own quadrant or one of her/his teammates' quadrant. As an active member in the team, a trainee needs to ensure a resource allocation plan that can protect her/his own quadrant, at the mean time she/he needs to identify other team members who do not have the appropriate asset to perform certain domain task. If such situation is detected, a trainee is expected to plan of sending the corresponding asset to help; if such resource allocation activity is not reflected in her/his plan, the coaching agent will generate corresponding feedback to help trainee realize such resource allocation deficiency.

To coach about mission-planning, the first step is to detect significant errors by comparing the number of assets planned in each quadrant with the minimum and maximum number of appropriate assets that can be used to assist team member's corresponding domain tasks, as identified above. Take the overloaded quadrant as an example; feedbacks are listed below about potential trainee planning deficiency in this regard:

<table>
<tr><th>Reminder</th><th>M = Number of attack assets
M <2</th><th>M = Number of attack assets
2< M<6</th><th>M=Number of attack assets
M>6</th></tr>
<tr><td rowspan="2">N = Number of identification assets
N<1</td><td>Overloaded quadrant needs asset for assist attack</td><td>Appropriate number of assets for attack in overloaded quadrant</td><td>Non-overloaded quadrants needs asset for attack</td></tr>
<tr><td>Overloaded quadrant needs asset for assist ID</td><td>Overloaded quadrant needs asset for assist ID</td><td>Overloaded quadrant needs asset for assist ID</td></tr>
<tr><td rowspan="2">N = Number of identification assets
1<=N<=3</td><td>Overloaded quadrant needs assist attack</td><td rowspan="2">Appropriate number of assets for both ID and Attack in overloaded quadrant</td><td>Non-overloaded quadrants needs asset for attack</td></tr>
<tr><td>Appropriate number of assets for ID in overloaded quadrant</td><td>Appropriate number of assets for ID in overloaded quadrant</td></tr>
<tr><td rowspan="2">N = Number of identification assets
N>3</td><td>Overloaded quadrant needs assist attack</td><td>Appropriate number of assets for attack in overloaded quadrant</td><td>Non-overloaded quadrants needs asset for attack</td></tr>
<tr><td>Non-overloaded quadrants needs asset for ID</td><td>Non-overloaded quadrants needs asset for ID</td><td>Non-overloaded quadrants needs asset for ID</td></tr>
</table>

Table 5 Inferences about trainee deficiency

The texts in grey indicate significant trainee deficiency and the corresponding reminder. In the following two cases, feedback is generated to address one of trainee’s significant asset allocation deficiencies listed in the above table.

For those cases listed in the first column, the number of team assets for attack is not enough in overloaded quadrant. Coaching agents need to show trainees how their plan would fail without helping the overloaded team member with attacks in her/his quadrant, resulting in team’s low offense and defense scores.

For the cases listed in the first row, the number of team assets for identification is not enough in the overloaded quadrant. Coaching agents should remind trainees that they are committing an error of not providing enough assist ID to help the overloaded quadrant and that it is necessary to allocate at least one (and no more than 3) idled asset for ID purpose from one of the non-overloaded quadrants, e.g. recommending that the lowest loaded (or the second lowest) quadrant's allocating one of its ID assets to the overloaded quadrant

Expert allocation strategies also point out the kind of allocation deficiency due to the capacity constraints of certain assets. For example, team resource allocation plan is evaluated based on whether the team has allocated speed-limited asset to help overloaded quadrant. Allocation strategies against the above expert recommendation are considered to a possible deficiency that leads to mission degradation. Coaching agents need to remind trainees that even though an asset has enough power to help with attack, the asset's speed limitation can result in loss of team defense scores before it ever reaches the overloaded quadrant and attacks the targeting task.

3.5.2 Scoring Algorithm for Resource Allocation Planning Evaluation

Besides the first-step quantitative evaluation and feedback described above, the second step of team resource allocation evaluation focuses on the assets planned to be used for the most critical domain task (attack in DDD domain) in a particular quadrant. A set of expert strategies is gathered via cognitive task analysis regarding team asset allocation, ordered by resource allocation priority. Given an asset allocation plan, we are able to evaluate whether the planned asset allocation by each decision maker has taken the workload and capability mismatch into consideration and whether the team assets as a whole can possibly perform the expected domain task, which is to destroy all the incoming enemy tasks for a particular scenario in DDD.

A scoring algorithm has been implemented to reflect these expert strategies. The algorithm takes team asset allocation plans as input and detects if there is a potential attack of an incoming task by a suitable asset— there exists an asset with the adequate power and it is available considering the timing of attacking an incoming hostile task. The algorithm generates a score for each team's planned asset allocation. For each potential attack (a match between a team asset and an incoming task), it scores one point for the resource allocation plan and marks the vehicle as allocated so as to reflect the change of asset availability in the same time interval for later analysis. A set of expert allocation strategies are collected by cognitive task analysis and the order of these strategies reflect the priority when experts make decision on asset allocation and potential engagement in the execution session. The prioritized expert strategies are used by the scoring algorithm to identify a potential match between a hostile task and a team asset. Each score increase predicates a potential coverage of an incoming hostile task by a particular team asset during the execution phase. Higher score indicates a better chance of attacking more hostile tasks during execution if the resource allocation plan is followed. Thus team resource allocation plan with a higher score is considered better planning phase decision making about where to allocate team assets. The design of the training scenarios requires team efforts to balance individual level workload and capacity mismatch by helping each other among the team, thus a higher score also shows a better helping pattern in trainee's planning of resource allocation. The set of expert strategies that has been adopted by the scoring algorithm is shown below (ordered by their priorities):

1. Match an incoming hostile task with an equal-powered unallocated asset
2. Match a hostile task with an equal-powered asset that has been arranged to attack a task that appears early. Arrange another attack with the same asset if there is a reasonable time interval (acquired from cognitive task analysis) in between the two adjacent attacks
3. Match a task with a higher-powered unallocated asset
4. Match a task with a higher-powered asset that has been arranged to attack a task that appears early. Arrange another attack using the same asset if there is a reasonable time interval (acquired from CTA) in between the two adjacent attacks

According to which quadrant a particular asset belongs to, we refined the above expert strategies by adding another dimension to distinguish non-local asset (assets that are planned to help other quadrants) and local asset (assets that are planned to perform domain tasks in a decision maker's own quadrant). Higher priorities are set for planning local assets first if multiple engagements are possible. The reason is that given a sufficient time interval in between the appearance of two incoming hostile tasks, it is more time-efficient to launch and move local assets to perform attacking task in a DM's own quadrant. Our training scenario is designed so that such multiple usages of local asset is allowed, yet there are still a number of cases when the time interval between two incoming tasks are insufficient and it is when helping behaviors are expected from other team members to assist overloaded DM with certain domain task (identification or attack).

3.5.3 Deficiency Diagnosis for Resource Allocation Planning

3.5.3.1 Deficiency Diagnosis for Attack Task

To further explain the feedback generation for trainee's planning of resource allocation, we list the general conditions that hold regarding team asset allocation and the abbreviation of each term used is explained in Table 4. These conditions are identified before a comprehensive set of deficiency categories can be developed regarding team resource allocation.

Nav	Number of allocated team attack vehicles/assets
Nmv	Number of matched team vehicles/assets
Niv	Number of invalid team vehicles/assets
Net	Number of total incoming enemy tasks
Nmt	Number of matched enemy tasks
Pth	Highest power of incoming hostile tasks
Ptl	Lowest power of incoming hostile tasks
minV	*Minimum number of vehicles(assets) required* = ROUND_UP (the number of power 1 tasks/2) + ROUND_UP (the number of power 3 tasks/2) + ROUND_UP (the number of power 5 tasks/2) ;

Table 6 General Conditions and Abbreviations of the Classification Tree

We classify trainee's resource allocation deficiencies into categories and provide the corresponding feedback for each category. As we mentioned, the evaluation focuses on two major domain tasks—identification and attack of the incoming tasks. Figure 7 illustrates the classification of resource allocation deficiencies regarding trainee's attacking task; the classification conditions for each node can be determined by running the above mentioned scoring algorithm that predicates potentially attacks during mission execution.

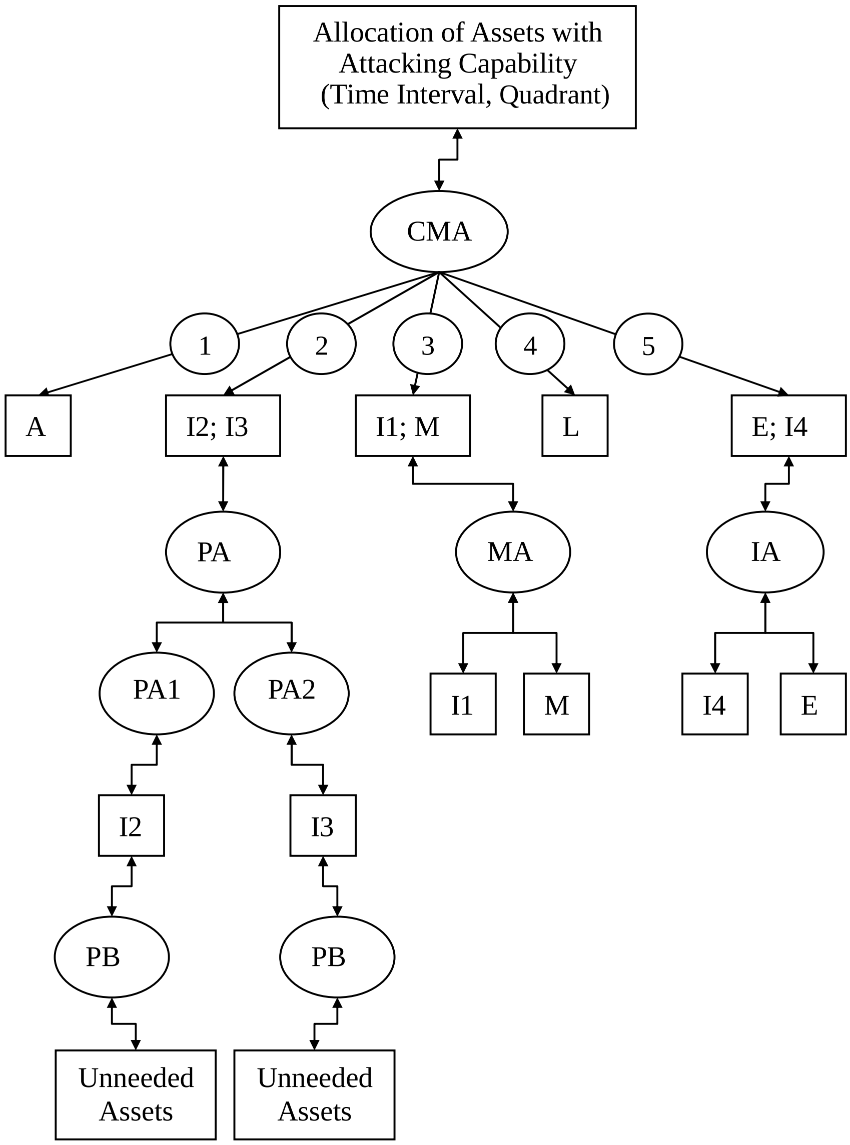


Figure 7 Deficiency Diagnose Tree for Planned Resource Allocation

Figure 7 represents the diagnose tree of resource allocation deficiencies, in which square nodes represent different types of trainee's resource allocation deficiencies and circle nodes represent a set of conditions to classify these resource allocation deficiencies. It illustrates how the diagnosis of trainees' resource allocation deficiencies is comprehensively classified into eight categories given the general conditions listed in Table 4 and the asset-task matching conditions listed below:

1. Whether all incoming hostile tasks can be potentially attacked by allocating one of the team assets
2. Whether all the team assets are allocated for potentially attacking these tasks
3. Whether the number of allocated assets is greater than or equal to the minimum number of assets required
4. Whether team allocates any assets with the attacking capacity
5. Whether the unallocated assets have the adequate power

The following table (Table 5) represents the detailed classification conditions for each node on the deficiency diagnose tree shown in Figure 7

Node	**Conditions**
1	A: Adequate number of attack assets allocated 1. Nmt==Net (all the enemy tasks were matched) 2. Nav==Nmv (All the team assets has been used to match, none of them left unmatched or invalid)
2	L: Less than enough attack assets allocated 1. Nmt<Net (not all the enemy tasks can be matched) 2. Nmv==Nav (all the assets allocated to this zone has been matched to the incoming enemy tasks, but still cannot cover all of them)
3	1. Nmt<Net (not all the enemy tasks can be matched) 2. Nmv< Nav (or Niv>0, given Nav=Nmv+Niv) (Team has allocated invalid assets in a particular zone that cannot be matched to any enemy tasks)
4	1. Nmt<Net (not all the enemy tasks can be matched) 2. Nmt==Nmv==0 (<Net) (when no enemy tasks can be matched to any assets)
5	1. Nmt==Net (all the enemy tasks were matched) 2. Nav>Nmv There are some attack asset(s) that haven't been matched because they're not needed.
PA	Invalid asset allocated Allocated assets with *power > Pth or* Allocated assets with *power <Ptl*
PA1	I2: invalid attack asset allocated and the total number of assets allocated is less than the minimum requirement *Nav<Minimum number of assets required*
PA2	I3: Invalid attack assets allocated and the total number of assets allocated is equal or above the minimum requirement *Nav>=Minimum number of assets required*
MA	I1: No enemy tasks can be matched yet some attack assets allocated Nav > 0 (team allocated some attack assets)
	M: No enemy tasks can be matched with no team assets allocated Nav=0 (team hasn't allocated any attack assets)
IA	E: Too many team attack assets allocated, with adequate power unmatched assets all with the power (*Ptl < =power <=Pth)*
	I4:Too many team attack assets allocated, with the inadequate power unmatched assets all with the power (*power< Ptl or power>Pth)*
PB	Algorithm to identify the unneeded assets for I2 and I3

Table 5 Conditions to Diagnose Resource Allocation Deficiencies

For example, node 1 represents the case when all the team assets have been allocated and all the incoming hostile tasks can be potentially attacked given the planned resource allocation (as a result of the scoring algorithm).

We provide trainees with the type of feedback marked as “A” if conditions represented by node 1 meet, which concludes that trainees have allocated an adequate number of assets with adequate attacking capacity in a particular quadrant.

3.5.3.2 Deficiency Diagnosis for ID (Identification) Task

Besides the attack of incoming hostile tasks, the other important domain task that we focus on is the identification of all the incoming tasks. The error categorization table for the ID task is shown in Table 6 for the mission planning phase. An A indicates that the number of assets allocated in a particular quadrant is adequate for ID. An L indicates that the number of assets is not enough for the ID task. An E indicates that there are more than enough assets for ID task.

Tracks/AWACS	0	1	2	3	4
2	A	E	E	E	E
3	A	E	E	E	E
4	L	A	E	E	E
5	L	A	E	E	E
8	L	L	A	A	E
9	L	L	A	A	E

Table 6 Categorization of Planned ID Resource Allocation Assessment

3.5.3.3 Feedback Generation for Resource Allocation Planning

The corresponding feedback is generated for diagnosed deficiency categories described in the previous sections regarding trainees' asset allocation planning for the two major domain tasks. Table 7 and Table 8 list the feedback in each of the categories. Table 7 shows the feedback on trainee deficiencies regarding attack resource allocation planning, and Table 8 shows the feedback on trainee deficiencies regarding ID resource allocation planning. These feedbacks will be presented to trainees via a set of graphical user interfaces, which we will describe in the next section.

Case	Expert Evaluation	Expert Guidance (based on the analysis of the number, power and timing of the tasks provided in the intelligence report)
A		Good Job!
E	TOO MANY ATTACK ASSETS!	***Diagnose***: This quadrant has too many attack assets. ***Remediation***: Team should consider reusing assets more within quadrant with too many attack assets and assisting quadrant (I1, I2, I3) with more attack assets.
L		***Diagnose***: This quadrant has too few attack assets. ***Remediation***: Team should consider assisting this quadrant with more attack assets and planning less reuse of attack assets within this quadrant.
I1	NO ATTACK POSSIBLE!	***Diagnose***: None of the assets placed in this quadrant has enough power to attack any of the incoming hostile tasks. ***Remediation***: Team should allocate assets with power equal to or greater than the incoming hostile tasks in this quadrant.
I2	NUMBER AND POWER OF ASSETS INSUFFICIENT!	***Diagnose***: This quadrant has neither a sufficient number of attack assets nor the right mixture of assets to attack all the incoming hostile tasks. ***Remediation***: Team should allocate assets with power equal to or greater than the incoming hostile tasks in this quadrant.
I3	ASSET EXCHANGE NEEDED!	***Diagnose***: The number of attack assets in this quadrant is enough, but the mixture of assets is not sufficient to attack all tasks in this quadrant ***Remediation***: Team should assist this quadrant with assets that better match the tasks and allocate unneeded assets to assist other quadrant.
I4	UNECESSARY ASSETS PLACED!	***Diagnose***: The number of attack assets is more than needed. Some of the allocated attack assets do not have enough power to attack tasks in this quadrant. ***Remediation***: Team should consider allocating the asset that does not have enough power to attack any tasks in the quadrant (or assets with adequate power but will not be needed) to assist other quadrants
R	SPEED-RESTRICTED ASSET OUT OF ITS OWN QUADRANT	***Diagnose***: This quadrant has speed-restricted asset(s) from other quadrant ***Remediation***: Team should make a plan that allocates speed-restricted asset(s) within their own quadrant—these assets are too slow to help effectively in other quadrants.
N	NO ATTACK ASSETS!	***Diagnose***: This quadrant has no attack assets. ***Remediation***: Team should consider assisting this quadrant with attack assets.

Table 7 Feedback Table for Planned Attack Resource Allocation

Case	Enough for ID?	Expert Evaluation	Expert Guidance
A1	Adequate as the number of ID-only assets>0		**Recommendation:** In addition to using <number of ID-only asset> ID-only assets allocated in this quadrant, DM should consider using home base or other attack assets' ID ring
A2	Adequate when the number of ID-only assets =0		You have no <name of ID-only asset> in this quadrant; DM should use home base ID ring or other attack assets to ID
L1	Too few when the number of ID-only assets =0		This DM should place ID-only assets in its own quadrant; Team should assist this quadrant with <number of recommended ID-only assets> more ID-only assets. <Quadrant with too many ID-only assets> should allocate <name of ID-only asset> to help this quadrant
L2	Too few when the number of ID-only asset >0		Team should assist this quadrant with <number of recommended ID-only assets> more <name of ID-only assets>. <Quadrant with too many AWACS> should allocate <name of ID-only asset> to help this quadrant
E	Too many	TOO MANY AWACS!	This quadrant has too many AWACS; The team should assist <quadrant with too few ID-only assets> with ID by allocating <name of ID-only assets> that are not needed in this quadrant.

Table 8 Feedback Table for Planned ID Resource Allocation

3.5.3.4 Feedback Presentation of Resource Allocation Planning

In this session we first summarize the general approach of feedback presentation for mission-planning-phase coaching agent. We then address the issues of trainees' learning of team collaboration, and discuss some detailed design choices we made to present the above generated assessment results in an efficient way.

During the training session, trainees perform knowledge intensive tasks in a complex domain DDD and tend to be information-overloaded most of the time. For example, they need to respond to various information updates from the simulator, and other information updates from the performance support tools provided, such as the TAP (Task Assignment Panel). Thus it is an important design decision to make the coaching interface as simple and direct as possible, and avoid overloading trainees with additional diagnosis and feedback information during their performing of the complex tasks.

To avoid imposing additional information load on trainees during their mission, we provide the resource allocation feedback in the after-action-review (AAR) session at the end of each training mission. To ensure trainees' quick digest of the feedback, we convert the text-based feedback into graphics and create a user interface similar to the generic DDD simulation interface. We present the feedback on four different quadrants similar to the layout trainees perceived on the simulation screen and present each deficiency categories with a unique icon, e.g. a smiling face indicates that the team did a good job in terms of ID or Attack. We also provide trainees with detailed explanation in texts, which give them an overview of some critical resource allocation related statistics and help them better understand how to improve their planned resource allocation as recommended.

As we distinguish the resource allocation needs for different domain tasks, feedback regarding the two major domain tasks (identification and attack) is presented to trainees one at a time. On the main feedback presentation window, trainees can click radio buttons to switch between the resource allocation assessment for attack tasks and the resource allocation assessment for ID tasks. Within the feedback window regarding either domain task, feedback is organized for each time interval and presented on different tabbed-panes, each one of time-interval tab corresponds to feedback generated for that particular time interval. On the feedback screen, trainees can click different tabs for feedback on their resource allocation in different time intervals (from time interval one to time interval four). Figure 8 and Figure 9 are the screen shots for the GUI of coaching feedback for trainee's mission-planning. The task-switching buttons are aligned on the red bar in the lower screen and the panel-switching tabs for different time interval are located on the upper-left corner of the screen;

Figure 8 GUI for ID Feedback Presentation

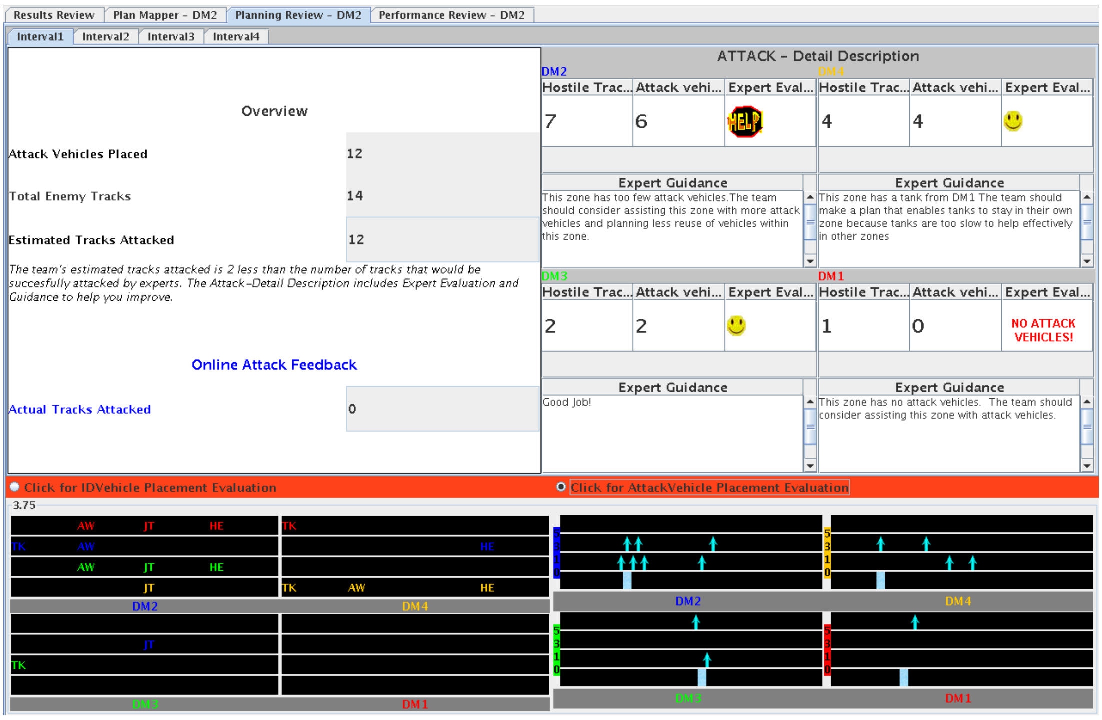


Figure 9 GUI for Attack Feedback Presentation

Individual and Team presentation

On the bottom of the screen (Figure 8 and Figure 9) the trainees' placement plan and the corresponding Intelligence Report are presented as a reference for the team. It reminds trainees how the team has planned for their resource allocation. The upper screen (above the red bar) is divided into two parts. General feedback is presented on the upper left as an overview summarizing important statistics about the planned resource allocation at the team level. For ID resource allocation assessment, the statistics presented to the trainees are the number of total incoming tasks, and the number of main ID assets placed in a particular time interval; for attack resource allocation assessment, the statistics include the number of hostile incoming tasks and the number of team attack assets allocated, the number of hostile tasks that the team can possibly attack (as the result of running the scoring algorithm discussed in section 3.5.2), compared with the number of hostile incoming tasks that an expert team can possibly attack.

For each individual trainee, the feedback provided by mission-planning-phase coaching agent is presented on the upper-right area on the screen. It is divided into four quadrants similar to the layout of the DDD simulation task screen. The detailed feedback includes three major parts—individual statistics, expert evaluation and expert guidance. Individual statistics include the number of hostile/incoming tasks and the number of assets allocated for a particular domain task in the individual DM's quadrant. Expert evaluation is summarized as an icon that indicates whether the planned resource allocation is appropriate for a particular quadrant. The evaluation is a result of the team plan diagnosis described in section 3.4.3.3. Expert guidance explains the recommendation in details about how trainees can improve their planning of resource allocation for either one of the major domain tasks.

3.6 Coaching Agent for Team Collaboration Process

3.6.1 Beyond the Planning-phase Expert Model

The team resource allocation plan trainees developed in mission-planning phase explicitly addresses the problem of allocating team assets to certain quadrant and makes implicit assumptions about team's better anticipation of mission with the planned team-level helping behavior during the mission execution. However, the interdependency between mission planning and mission execution leads to a variety of combinations regarding team's planning outcome and execution outcome. For example, trainees can succeed in making a good resource allocation plan but fail to achieve a good mission performance; there might also be situations when trainees fail to make a good resource allocation plan yet during their execution of the team mission, they succeed by adjusting to a good online resource allocation that is different (and better) than what they have planned. To achieve our training goal of helping trainees improve their team performance, it becomes an inescapable step to develop coaching mechanism to monitor and assess relevant team processes during trainees' mission execution.

As we have discussed in Chapter 2, each team member's domain specific operation skills are necessary yet not sufficient for team's overall mission success. Thus our training protocol involves training of each individual to acquire domain specific individual-level skills as a pre-requisite before they can exercise higher-level collaborative strategies. To help trainees develop adequate individual skills to perform basic domain tasks, we provide trainees with a list of operational level strategies that they could apply during their mission execution.

The following table lists a set of general principles that the DDD domain experts act upon during mission execution to ensure that they perform effectively for relatively independent tasks.

1. Each DM needs to try launching all assets at the beginning of each time interval (during which team will encounter a wave of incoming tasks).
2. Assets can be launched only one by one— the second launch command can be issued only after the finishing of the first launch command and the issuing of the move command for the asset to go to the target quadrant.
3. If an asset is to be used for a subsequent attack, check to see if that asset is returned to base immediately after the previous attack.
4. All the assets need to be returned to the base at the end of each wave when all tasks are leaving.
5. Constantly engage assets with identifying and attacking incoming tasks; the speed and number of incoming hostile tasks require each DM to perform multiple domain operations in parallel.
6. ID assets need to be sent to the place where the incoming tasks are most likely to come. Check the status of asset that doesn't have attacking power, refuel and rearm if necessary.

Table 9 Individual-level Execution Strategies

Given each team member's workload, the more detailed mission-execution strategies are listed as: a) The least-loaded quadrant allocates an identification asset to help with identification. b) The least-loaded quadrant allocates an attack asset to the overloaded quadrant with the observation that this asset can be used in the overloaded quadrant to attack multiple incoming tasks (need to return to base and re-launch for each successive attack). c) The heavier medium loaded quadrant allocates an attack asset to the overloaded quadrant to help with attack; asset may be needed for attack in the medium loaded quadrant. Based on the timing of the two attacks, an asset needs to do the local attack first or return to base and locally attack another task. d) The other medium loaded quadrant (with slightly heavier load than the least-loaded quadrant) allocates one of its assets with both identification and attacking capacity to the overloaded quadrant to help with identification and attack of a task.

Here is an example of event-based deficiency detection for tasks in the DDD simulation domain:

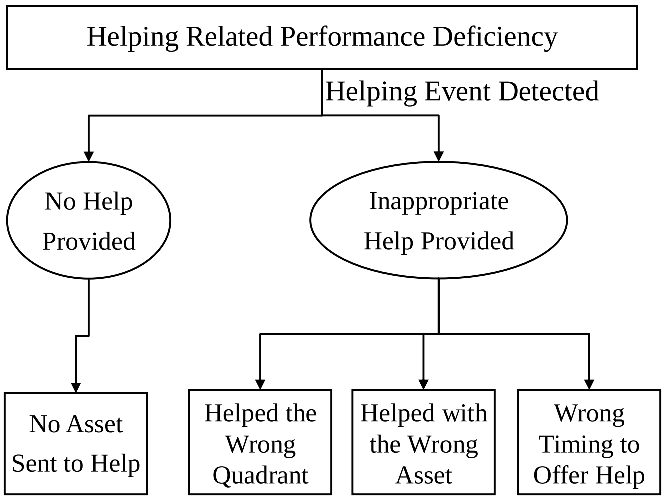


Figure 10 Deficiency Categories for Helping Behaviors (DDD domain)

Figure 10 illustrates a partial deficiency taxonomy that will organize the possible trainee errors regarding helping behaviors among team members. Upon detecting a set of conditions associated with a helping event, the expected helping behaviors can be identified by going through the teamwork knowledge encoded in the team performance model. The individual assessment module or the team assessment module can then collect the actual trainee performance data related to this event from the monitor agents. Thus the coaching agent can compare the desired helping behavior with the actual trainee actions and provide feedback that addresses trainee deficiency if needed. For example, if help with attack is desired yet no asset is provided to help with the domain task in the overloaded quadrant, the coaching agent needs to identify who can potentially be one of the helpers, inform the potential helper who did not provide help (regarding the missing actions) about who is overloaded, what types of assets need to be sent out, and that the trainee can potentially be one of the helper(s). If inappropriate helping actions are detected, the coaching agent needs to figure out the specific deficiency (e.g. whether it's the inappropriately powered asset or the asset was sent to a quadrant that does not need help, or the asset was not sent on time, and etc), feedbacks including the above information will be provided about trainee's specific failure to help.

3.6.2 Diagnosis and Feedback

As discussed in section 3.1, the successful performance of some mission-specific task is essential (but not sufficient) to ensure effective team performance. Thus besides the event diagnosis at the collaboration level, the execution-phase coaching agent also needs to diagnose trainee deficiencies regarding some crucial events at operational level and provide feedback about team performance regarding those domain-specific sub-goals that play a crucial role on team's mission success. Only after trainees have mastered the essential operational level knowledge and skills, coaching feedback addressing trainees helping behavior performance at a higher level can then show a great impact on team's mission performance.

This mission-level event analysis focuses on a set of explicit domain-specific objectives or sub-goals to improve team outcome in performing the mission. Five general mission objectives (listed in Table 10) are identified by our previous conducted human experiments. It is shown that by learning these mission objectives trainees can significantly improve their training outcome.

Five Mission Objectives in DDD domain	1	Destroy hostile tasks immediately after they enter the green restricted zone
	2	Destroy hostile tasks as quickly as possible in red restricted zone
	3	Do not destroy hostile tasks outside the green restricted zone
	4	Do not destroy neutral tasks anywhere
	5	Check the asset status and avoid own asset's running out of fuel

Table 10 Mission Objectives in DDD Domain

In Figure 11, the mission objectives listed in Table 11 are further broken down into a set of trainee actions recorded by the monitoring agent. Ovals represent mission-level deficiencies or feedback. The shaded ovals represent the trainee deficiencies before decomposition; the dotted ovals show the potential diagnosis and feedback. Rectangles represent the deficiency diagnosis that decomposes the mission-level deficiencies to a set of detailed trainee actions that might serve as causes for the higher level mission deficiencies. Causes for trainee deficiencies have been collected by domain experts' observing a large set of trainee behaviors during their mission performance.

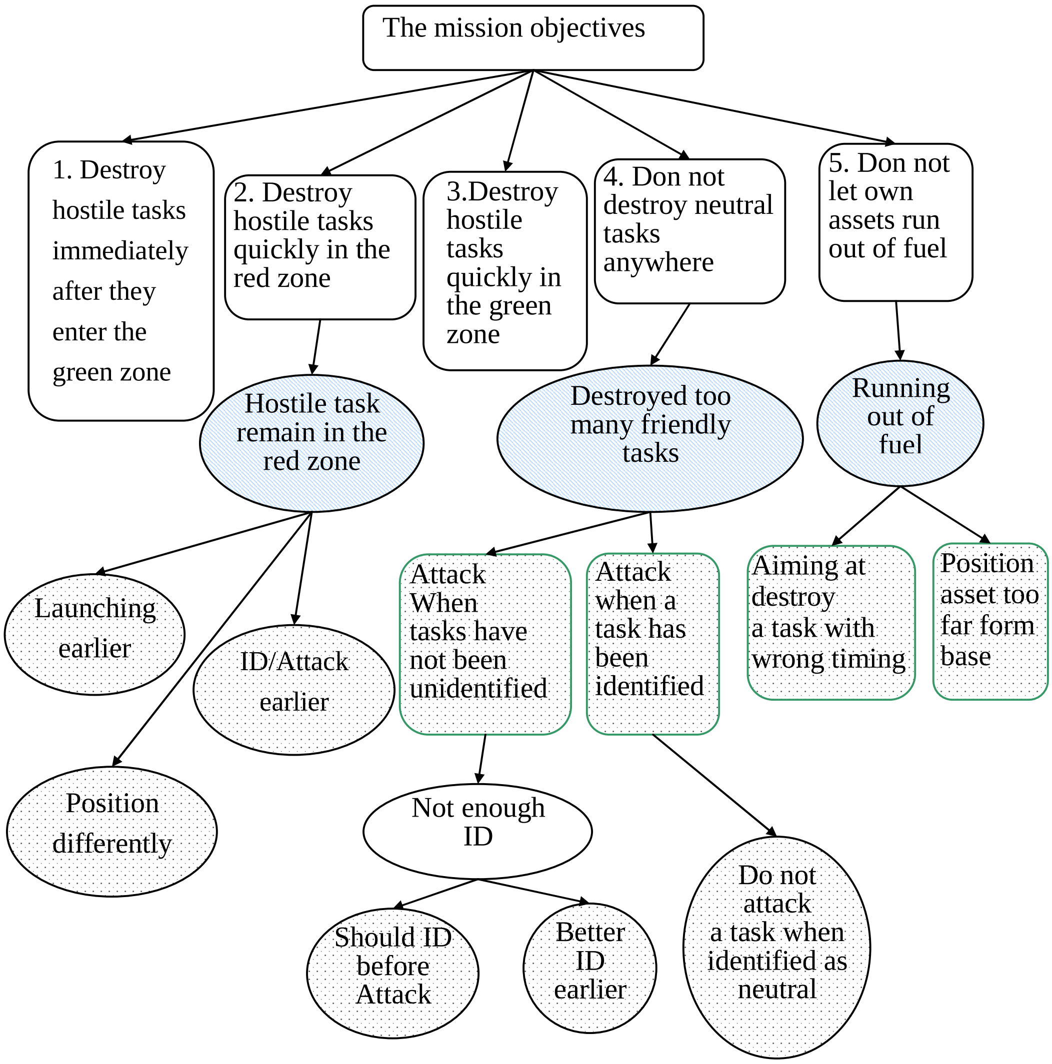


Figure 11 Mission Objectives and Event-based Coaching Feedbacks

Event-based assessment identifies deficiencies for each critical event captured by the monitor agent. At the task-specific event level, trainee deficiencies are associated with one of the above listed mission objectives (e.g., minimize the time of hostile tasks’ staying in the red zone; avoid destroying a neutral task and etc). If trainee performance deficiency is detected for one of these objectives, the coaching agent will analyze potential causes for the deficiency and diagnose the cause by linking them with other event-based assessment. For instance, if the team destroys neutral tasks very frequently, this may be caused by attacking un-identified tasks frequently when the team does not have enough assets for identification. Different causes can lead to this type of deficiency. Either the team planned to allocate less-than-enough assets for the identification purpose in the particular quadrant, or the team failed to send the ID asset during execution even though such resource allocation was planned. The causes can be narrowed down by collecting assessments from multiple domain events.

This approach allows the execution-phase coaching agent to record a set of domain-specific actions, assess and diagnose the potential trainee deficiencies. The mission-specific feedback explains why trainees' performance deficiencies can lead team to lower sub-scores representing the outcome of each of the mission objectives. It ensures that trainees get feedback to master important mission-specific skills, which sets the ground for team collaborative level behavior and feedbacks (such as team helping behavior deficiencies).

As we mentioned, each team's mission success is twofold: to ensure the individual level performance regarding specific mission objectives and to ensure the team-level collaborative performance such as helping each other balancing workload and resource capacity. One of the criteria to distinguish trainee's helping behavior from other task-related behaviors in the domain is whether these mission objectives are achieved in their own quadrant. Feedback can also be categorized as whether trainees performed well in achieving these mission objectives to protect their own quadrant, or to help other quadrants (especially the overloaded quadrant) regarding the above mission objectives when such help is needed.

This section has explained the basic concepts of event-based error detection and deficiency diagnosis exemplified within the DDD domain. We focused on the design diagram rather than a detailed description of the event-based assessment /feedback implementation. The implementation details will be discussed in the following section.

3.6.3 Intelligent Modules for Performance Assessments

In the Agent-based Intelligent Training/Coaching framework, each trainee has a monitor agent and a set of assessment modules that analyze trainee performance focusing on different aspects. The performance assessment modules include task specific deficiency analysis at the individual level, the resource allocation analysis and the helping behavior assessment at the team level.

3.6.3.1 Task Deficiency Analysis

Monitoring trainees' performance is fundamental for coaching agents' generation of effective team feedback. At the low end of team's performance matrices, task performance refers to domain specific activities that trainees perform to accomplish individual goals; it serves as the basis for fulfilling trainee's responsibility within the team and achieving good team performance. To build our coaching agent, we need an intelligent module that can identify a list of domain events that are critical to achieve the mission-specific sub-goals. The task level performance monitoring also involves the recording of domain operations that play significant roles in achieving team's collaborative performance (e.g. helping one another in case of workload and resource capacity mismatch).

Task specific diagnosis includes trainee actions regarding the mission specific sub-goals (see Table 10 Mission Objectives in DDD Domain). At the operational level, trainee deficiencies will be recorded. If trainees attack tasks outside the green restricted zone, or attack tasks without identification or allow an asset running out of fuel. These task specific errors may lead to operation failure and further affect the teamwork performance if certain conditions apply. Frequent occurrence of the above mentioned deficiency might indicate trainees' unawareness of some specific domain rules or their failure to perform the expected operation under stress. For example, if a trainee frequently attack tracks with insufficient power, it is possible that she/he is not aware of the attack after identification rule or that she/he is not aware of the attack with equal or higher power rule even after the task has been identified, or in another case, the trainee is overwhelmed with task situation and eager to attack regardless of any domain rules.

3.6.3.2 Resource Allocation Analysis

During the planning phase of training, trainees planned about their mission and make decisions about where to allocate their assets during each time interval of mission. As we mentioned in Section 3.4, the planning phase coaching agent will analyze the resource allocation plan that trainees made and comment on whether it is an effective planning compared against a set of expert planning strategies. During mission execution session where trainees carry out their plans, execution phase coaching agents monitor trainees' online execution of resource allocation, and provide them feedback about how well their team performance is regarding the allocation of team assets.

The first layer of analysis is whether trainees have followed recommended asset allocation specified by their resource allocation plan. Two types of inputs are needed—the planned allocation that is recommended by planning phase coaching agents and the actual assets allocation that trainees carried out during mission. The recommended plan elements are passed to execution phase coaching agents (specifically the resource allocation analysis module) from the planning phase coaching agent. To decide where trainees allocate their resource during mission, execution-phase coaching agents monitor trainee's allocation events, each of which is consisted a list of domain actions including the initial launch of a particular asset and subsequent move commands that lead the asset to a different quadrant. For each team asset, the resource allocation analysis module keeps an allocation event history with specific timing for each operation. Feedback is generated for trainee's resource allocation deficiencies listed below.

An asset is considered to be idling if the allocation event history of an asset is empty. Coaching agents will check for the recommended resource allocation for the idle asset if it is in the trainee's resource allocation plan. In case the trainees did not make a good resource allocation plan to include such allocation, coaching agents will refer to the online task pattern and generate the recommended allocation events with estimated timing for each action. Finally feedback will be provided including the recommended allocation events and inform trainees that the resource-constraint mission requires that team members maximize their asset usage in a timely fashion so as to balance team workload and resource capacity.

Trainees will more likely commit errors where they allocate an asset yet to quadrants where this asset will not be needed. Execution-phase coaching agents will diagnose such allocation deficiency by analyzing the task pattern and the availability of trainee assets during mission. If trainees followed their resource allocation plan and performed non-optimal allocation as part of the planned activities, feedback will be generated for trainees to refer to their planning feedback regarding resource allocation. Otherwise, coaching agents will propose an alternative resource allocation based on its analysis of the incoming task pattern (including task type, speed, appearing time, moving directions and etc) and the resource allocation of other related team assets.

3.6.3.3 Helping Behavior Assessment

Resource allocation feedback captures team helping behavior at a high abstraction level. It only addresses the issue of which quadrant to allocate an asset within a particular time interval. No specific timing included regarding when a helping event occurs or is expected to occur. Another important dimension of teamwork performance feedback has not been addressed concerning more important details about team members helping each other. Later in this section we discuss the remaining helping behavior assessment and feedback, including help with identification, help with attack or help communicating the identified task.

The scenario file for each training session is designed so that with a careful allocation and multiple engagements for all assets, the team can destroy all the incoming tasks during mission. At the end of the training session, some tasks might have been missed during mission, trainees either do not have the adequate resource to attack the task with the right asset or they failed on performing the domain specific tasks within an adequate amount of time. Thus every missed task is an indicator of potential operational or collaboration deficiency. The helping behavior assessment module collects asset allocation and engagement information for all the team assets during mission and starts the analysis from the missed task set at the end of the mission execution session.

The goal of the analysis is to diagnose whether team activities carried out during mission are adequate in terms of resource allocation and task engagement. Further more, this module aims at providing a better allocation and engagement that could yield a full coverage of all incoming tasks (effective identifying and destroying all hostile tasks).

Specifically, for each missed task, the helping behavior assessment module tries to identify an adequate team asset that can be used for an effective attack (of that missed task). Based on the appearing and remaining time of a missed task, the helping behavior assessment module first identifies an idle asset during that period of time that can be used for such an attack. The related asset engagement information is passed from the task analysis module and the helping behavior assessment module, for example, if an asset has been engaged in an attacking task in between two move operations that send this asset to a different quadrant. An asset is considered idle for a domain operation when it is not engaged to any domain operations, and will not be engaged in future activities until the potential domain operation can be completed. The identified asset can then be marked as assigned (no longer idle for that period of time). If an assignment can be made for the missed task, the algorithm then tries to assign potential attacking asset for another missed task until it depletes the missed task set. If a global assignment can be made for all the missed tasks with idled assets during each time interval, trainees' allocation and engagement activities carried out during the mission are then considered adequate. Coaching agents will then provide team level feedback about how the team can cover all the missed tasks by following the recommendation to maximize each asset usage (use the idle time of each asset to perform more effective attacks).

The execution activities of team during mission are not considered optimal if the algorithm could not identify an asset that can be engaged for potential attacks during the time when the missed task remains in team's restricted zones. The time required for launching, moving and attacking engagement limits each team asset's usage during each time interval, so that the algorithm can eventually go through the complete set of idled time slots for all the team assets. In this case, instead of trying to come up with a different mission engagement map with reallocation and a re-usage with the assigned and non-idle team assets, the allocation algorithm will prompt a hint to team that there are non-optimal elements involved in team's execution engagement, and the team should consider allocating asset differently according to the planning phase recommendation provided by the mission-planning coach.

3.6.3.4 Feedback Generation

In the previous sections (3.6.3.1, 3.6.3.2 and 3.6.3.3), we briefly introduced the different layers of monitoring and diagnosis mechanism for each intelligent assessment module as part of the execution-phase coaching agent within the AITC framework. In this section, we discuss the detailed design decisions that we made about the content, organization and presentation of the execution phase coaching feedback.

We design interfaces that present coaching feedback to trainees about their asset allocation and engagement regarding two major domain tasks— the identification of task power, and the attack of a task. They can switch between two windows each addressing their planning and execution of identification-related and attack-related activities.

Within each feedback category, we organize feedback into the four evenly divided time intervals during which the team will encounter a wave of incoming tasks. At the task level, coaching agents provide a summary about trainees' attack or identification of the incoming tasks. A list of each incoming task will be provided, including its power information (whether it is hostile and what type of asset might be used to attack it), which quadrant is responsible for it (either identify or attack), and whether it has been successful identified or attacked during that time interval. At the teamwork level, the content of feedback includes the online resource allocation feedback and the helping behavior assessment feedback.

Online resource allocation feedback first provides a summary about the individual's asset allocation by relating the actual asset placement to trainee's planned resource allocation. Feedback is organized into a table where it recaptures trainee's resource allocation plan and provides training information such as whether a certain type of team resource (attack or identification asset) has been allocated during mission. Compared to either team's good planning strategies or the requirements from the dynamic task pattern, coaching agents provide the recommended quadrant where this type of asset can be allocated.

The helping behavior feedback includes team process monitoring feedback and recommendation for effective helping. Both aspects critically reflect the quality of team's helping performance. In terms of team helping process monitoring, for attacking tasks, coaching agents provide an overview of the number of assist attacks and its success rate; for identification tasks, the overview includes the number of assist identifications and the number of transferred task identifications. A domain operation is considered to be a type of assist behavior when the decision maker is performing such operations in other teammate's quadrant and when she/he is not individually responsible for the task involved in the operations. Transfer of identified task information is considered an effective communicative behavior to avoid team members' duplicated operations (specifically identification of incoming tasks) and save time for the team to perform other time-critical operations (such as attack of tasks). Thus an overview focusing on the frequency of the above mentioned helping related events is a strong indicator of whether the team has a good collective performance in carrying out the mission. It informs trainees about their actual helping performance, and indicates potential improvement in this dimension by providing additional information, such as which quadrant is overloaded in a particular time interval.

The most constructive helping behavior feedback is highlighted for each time interval. We organized the content into four quadrants that correspond to the same screen layout in the simulation game. One of the quadrants is a trainee's responsible zone, where the trainee see the operational level performance recommendation as discussed earlier. In the rest of the quadrants, helping related feedback is provided in two cases. Suggested helping behaviors are promoted in case a trainee may serve as potential helper to assist a foreign quadrant with either identification or attacking task. From another perspective, a trainee may need help from others when its own quadrant is overloaded and in this case coaching agents will provide hint to the trainee about where the potential help may come from. The two types of helping behavior feedbacks aim at providing trainees with a high level mapping of the team resource allocation and their role in fulfilling the team mission by balancing workload and resource at the team level.

Figure 12 shows the graphical user interface (GUI) for the execution-phase coaching feedback presentation. There are two types of critical domain task categories, including identification and attack. To switch between attack feedback and identification feedback, trainees can click the radio buttons at the bottom of the screen. Task analysis feedback is located in the lower left panel and an event history is recorded for each team asset containing a list of critical domain operations regarding the specific task category. The total number of attempts and the successful rate for a particular domain task category (attack or ID) are listed together with the number of assist attempts and success rate for the corresponding helping related attack or identification events. Trainee deficiencies are prompted in red and a successful operation within an event is prompted in green. Team resource allocation related events are organized into the resource allocation analysis table, which locates on the upper right. The resource allocation table contains information such as to which quadrant trainees allocated their assets during mission, whether the actual allocation is consistent with the planned allocation, or whether the allocated asset was used for the major domain tasks (attack or ID). The detailed helping behavior analysis is located on the bottom right panel. As described earlier, the four quadrants are organized according to the game screen layouts. Within each quadrant, helping feedback is given either in form of the desired helping actions that need to be carried out, or in form of the potential help that the decision maker should expect from other team members. For example, in case the decision maker (DM) is overloaded, the helping behavior window will inform the DM that he does not need to allocate his resources to his teammates' zone, instead one or more of other zones could send assets to help. More specifically, it will show the DM a potential better usage of particular assets if they were idle during mission.

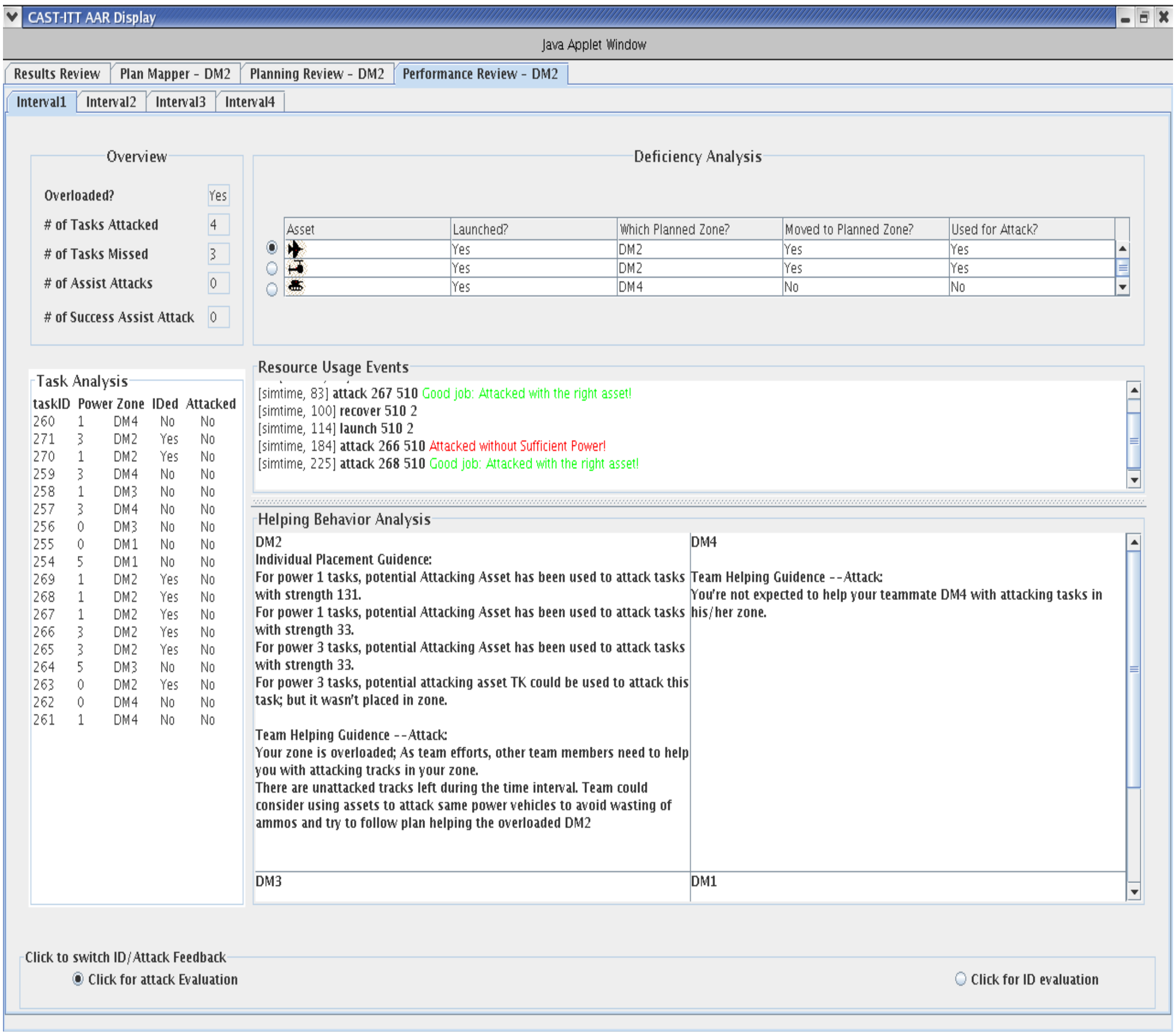


Figure 12 Graphical User Interface (GUI) for Execution-phase Coaching Feedback

From a different perspective, we associate feedbacks generated by coaching agents in the two training phases. For teams that make good resource allocation plans, they can be subdivided broadly into two categories according to their performance during mission execution:

1. Teams that deviate from the placement plan
2. Teams that follow the placement plan

Team deviate from plan

We estimate the percentage of the mission that a team follows the resource allocation plan and put them into the four categories according to different degrees of deviation. For teams that deviate from the placement plan (deviation>10%), we provide team members with the corresponding feedbacks:

Deviation level	Reminder
Slight deviation (10%-25%)	Identify the critical event during mission where they deviated from the plan and if they made attempts to get back to the plan
Moderate deviation (25%-80%)	Identify deficiencies during mission that hinders them from following the plan
Total deviation (80%-100%)	Coach the importance of planning and how it affects performance in mission execution, each resource allocation plan results from analyzing patterns of the incoming tasks, which is critical for team to succeed during mission execution

Table 11 Deviation of trainee in mission performance and the corresponding feedback

Team that follows the plan

1. Team that follows the plan and achieves good mission execution outcome

Provide trainees feedback that highlights trainees' good performance strategies that reflect the important connection between mission planning and mission execution. E.g. one of the non-overloaded quadrants needs to allocate an attacking asset to the overloaded quadrant and determine when to launch and allocate this asset by analyzing incoming task pattern from the intelligence report.

2. Team that follows the plan but fails to achieve good mission execution outcome

Identify timing deficiencies in trainee's online realization of the planned resource allocations (good timing management results from analyzing incoming tasks pattern) and provide coaching feedbacks to individual trainees according to the aforementioned execution-phase expert strategies.

Below is feedback in terms of the execution-phase expert strategies (also listed in Table 9):

1. It is recommended that each quadrant launches as many assets as needed at the beginning of each time interval. If coaching agents fail to detect such events during each time interval, feedback is generated to remind trainee that launch is a relatively time-consuming operation and in order for the team to be prepared for multiple incoming tasks appearing in a short period of time, each member needs to have all assets ready before hostile tasks appear, which allows them to immediately select the appropriate one for either identification or attack as soon as they observe incoming tasks.

2. Always allow enough time for a particular asset to reach the planned quadrant by analyzing incoming enemy pattern with a focus on the approximate

appearing time of the target task. After launching and moving of the asset to the target quadrant, launch the next asset if needed. Coaching agents need to detect if the corresponding events happen in the trainee's quadrant. If not, feedback is generated to remind trainees that intelligence report is a useful tool for them to predicate incoming task pattern and based on the analysis, they need to manage launching and moving of the suitable asset to the target quadrant before the estimated appearing time of the task.

3. An asset is returned to base immediately after engaged in attack if it will be used for subsequent attacks. Coaching agents first identify if the asset has been planned to do a successive attack, if so, detect if a return command is immediately issued after each attacking engagement, and generate feedback to remind the trainee such expert execution strategies if she/he fails to do so.

4. All the assets need to be returned at the end of each wave. If coaching agents fail to detect such events, feedback is generate to remind trainees that the analysis of incoming tasks patterns is critical for mission success. They may observe very few tasks at the end of each time interval, and it is a time for them to return assets back to base for refuel and re-arm and get ready for the next task wave.

5. Remind trainee that the task domain is a time-critical and stressful battle simulation. Every decision maker is expected to multi-task different operations at the same time to ensure that the team can manage to destroy all the hostile incoming tasks during mission as soon as they enter the critical zones.

4 Experiment and Results

As a final step to achieve our research objectives, we provide answer to the last research question listed in chapter 1, which concerns the validity of the intelligent training system. In this chapter, we discuss details about the human subject experiment conducted in evaluating the intelligent training approach and test our research hypotheses regarding the effectiveness of the two-phase coaching agents within the training framework. Data is collected regarding both individual and team level training interventions in a highly flexible and extensible real-time team task domain.

4.1 Scenario Design and Experiment Conditions

After design and implementation of the coaching agents within the intelligent team training framework as we discussed in Chapter 3, we have conducted human subject experiments to evaluate the effectiveness of the agent-based training, deploying the two-phase training protocol in the DDD simulation domain.

Our coaching agents are designed to provide training feedback to individual trainees in a team regarding their collaborative behaviors in a simulated environment. During the experiment, each trainee interacts with the simulation game and our intelligent agent software via a computer interface. We have tested our training protocol with ten four-member teams (10x4 = 40 trainees in total). For each team, the experimental session lasted about 150 minutes. The total data collection time for the experiments was 40 x 150

= 6000 minutes. Each trainee participates four training sessions as a member of a team with a total number of four decision makers.

At the beginning of each experiment, trainees go through a pre-training session, where we provide trainees with information about the experiment, and get them familiar with the important domain operations to achieve mission success—we first provide instructions about how to use mouse or keyboard to perform a set of basic operations while playing the simulation game and give trainees 20 minutes to practice the basic domain operations.

The rest of the experiment contains four training sessions and each training session consists of three parts—planning, mission-execution and after action review.

During the planning session, we developed a software tool that provides trainees Intelligence Report with detailed mission information that may help them predicate their incoming task pattern during the mission. We encourage trainees to analyze the intelligence report and make decisions about how to allocate their team resources so that as a team they balance overall workload with team resource capacity. The planning tool records where the team planned to place a particular asset (in case of helping, an asset is allocated to other team member's quadrant, who is in need of help with either attack or ID). Later during the mission, team's planned resource allocation decisions will be displayed via a GUI as a reminder to each member.

During the training session, trainees go through a set of randomized training scenarios with the goal to maximize their team outcome by helping one another to balance the overall workload and team members' resource capacity. The fifteen minutes game time is evenly divided into four time intervals (each lasts about 3.75 minutes). In each time interval, trainees are expecting one of the incoming task waves, where enemies concentrate attacking one or two of the team's quadrants. The scenario files are designed that each team member has a restricted number of resources (see Table 21 in Appendix A Training Material for Human Subject Experiment). For the overloaded quadrant, the decision maker can not effectively protected his own quadrant because of the unbalanced workload and resource distribution— regardless of this DM's domain skills, s/he doesn't have enough assets to attack all incoming tasks appear in her/his own quadrant. Thus it is a key to achieve mission success by helping one another and try to balance the workload and resource distribution at the team level; the scenario file is designed so that collectively as a team, trainees have enough assets to protect the four team quadrants.

For each mission, the coaching agent software will monitor trainee actions, record their performance matrices about certain events, including a set of mission-specific events and a set of teamwork-oriented events that provide data about how teams help each other, especially the decision maker who is overloaded. Upon each helping related domain event, coaching agents update the number of assist attacks, assist ID and their transfer of identified task information. At the same time, DDD simulation will update trainee's overall performance outcome in terms of total team defense and offense score (see Appendix A Training Material for Human Subject Experiment and a set of individual team sub scores that address important mission objectives and represent degrees of mission success).

At the end of each training session, we have a debriefing (or after action review) session where trainees will review their mission performance especially some of the mission deficiencies and discuss about how to improve team performance and outcome for next mission. For the control group, they will lead a discussion about the pervious performance deficiencies and the strategies to improve next mission without intelligent agents' assistance. For the experiment group, trainees will receive feedbacks generated by the two forms of coaching agents, including feedbacks that address their planning-phase resource allocation deficiencies and feedbacks that diagnose trainees' online execution deficiency focusing on helping them improve their collaborative behavior within the team.

We expect that trainees would improve their team performance each time as they go through a number of training sessions, more importantly we hypothesize that team learns better by receiving feedbacks generated by the two-phase coaching agent at the end of each training session. In subsequent sections, we will describe in details the procedure and results of a one-week human subject experiment for our hypothesis.

4.2 Procedure

We recruited students as participants in our experiment several weeks before its start. As a compensation for the experiment, we offer participants $8 per hour with the requiring about 2.5 hours of each person's time (8*2.5=20 per person per experiment). We aimed at recruiting students with adequate amount of computer experiences, we heavily advertised for the experiment on campus— recruiting emails were sent out to the list-server of several departments including information sciences and technology, computer science and engineering, electronic engineering, mechanical engineering, and several other department at Pennsylvania State University. The advertising campaign successfully netted us 44 volunteers for the experiment (and we recruited 40 as participants in the experiment).

Human subjects were randomly assigned to teams and half of the teams are randomly selected as control groups and the remaining half as experiment groups. Each team will go through a total number of four trials, after the baseline mission, trainees will go through three more training sessions— sequentially as Baseline (B), Training (1), Training (2), and Training (3). A Latin Square is composed where each line in the square shows a unique sequence of the scenario files. For each trial, a team will encounter a different scenario and the order of running these scenarios is randomized as each team follows a line in the Latin Square. Such randomization ensures that for each trial, both control group and the experiment group will have enough teams that collectively encountered a complete set of scenarios. As an average score is calculated for each trial, we minimized the bias that might be introduced by the order of the scenario files given any differences that exists among scenarios— some scenario might be easier than others and if by chance we ordered scenarios differently for teams in the control group and experiment group, the result might not reflect the team’s skill difference rather than which sequence of the scenario might be helpful.

Each trial consists of a 5-min planning session, a 15.5-min mission execution (30 second grace period at the beginning of each mission for trainees to launch assets) and a 10-min after action review. In between the two training sessions, the after action review for the previous training session intends to help trainees improve their team performance for the next mission. During the Planning session, team used an interactive software planning tool to plan the allocation of their assets based on their analysis of an intelligence report. During the mission, we display a reminder of team's resource allocation plan about how team planned to allocate their assets as well as the intelligence report regarding incoming task pattern while teams defended their restricted zones. During the Debriefing session, the teams followed a checklist in discussing lessons learned from the mission. The sequence of events was: instruction and pre-training (30-40 minutes) Baseline, Training 1, Training 2, and Training 3 (110-120 minutes). Including breaks (participants can ask for a break in between training sessions), it makes the total time of the experiment about 150 minutes.

The independent variables were the type of training protocol, the training trials with different scenario files and the dependent variables were the measures of teamwork. The training protocols were identical for control and experiment groups for pre-training session, the planning and mission sub-session during training, the only difference between control and experiment groups lies in the after action review sub-session during training. The control groups debrief without receiving feedback generated by intelligent coaching agent, the experiment groups debrief with a thorough review of the feedbacks generated by the agent regarding their planning of resource allocation and their performance focusing on team helping behaviors during mission-execution. To evaluate teamwork performance for each team, we include helping behavior measurements such as the number of assist ID/Attack, Communication of ID attempts and the corresponding success rates for each one of the above helping behavior types. For the experiment group, we'll provide the above performance measures to each member during the after action review session, where the performance measurement is further divided into each time interval. For the control group, the helping behavior measurement won't be displayed.

For all teams participated in the experiment, we'll internally record their performance measures into a set of log files for later analysis.

We also include measures of team mission performance in terms of individual/team defense and offense scores, which include a set of sub-scores that explicitly address team mission objectives. The total defense and offense scores both start at zero and get updated each time as any sub-score changes, thus they reflect the overall team performance and outcome. The defense sub-scores include a penalty of 25 points (-25 points from the defensive scores) for each attack of the enemy outside the restricted zones, a penalty of 25 points for each asset's running out of fuel, a penalty of 25 points for each attack of neutral tasks, a penalty of 1 points per second for each hostile task's staying in the green restricted zone, a penalty of 2 points per second for each hostile task's staying in the red restricted zone. The individual offense score will increase 50 points for each successful attack of enemy tasks within the green restricted zone and the team offense score is a sum of the individual offense scores of all members.

At the end of the experiment after all training sessions have been completed, trainees were asked to fill out a post-experimental questionnaire regarding their game experiences, their self evaluation as a team player, and their feedback to the mission-planning and mission-execution phase coaching agents. The qualitative results we collected from human participants via the questionnaire will be discussed later in this section.

During the whole experiment session, trainees were allowed to talk with each other during either planning, execution and after action review session, especially when they collaboratively work on a team resource allocation plan and review about their performance deficiencies in the after action review session. During mission execution, the observation was that trainees talked less when they were under the time-pressure of the team task and the most effective communication among team members was through the DDD interface, e.g. communicate the power information of a task they identified by performing the operation of transferring an ID. After the first two trials, trainees were given a short period of time to take a break.

To depict the above human subject experiment in terms of hardware equipment and software configuration, we provide a brief summary of the team training laboratory setting. We have four computers running Redhat Linux 9.0; each runs a DDD client for a particular quadrant and the corresponding coaching agent for DM in that quadrant. We run the main experiment control program on one of the machines as the DDD server. All the computers are connected with Ethernet and share data within an NFS file system.

4.3 Data analysis and Results

Two major categories of trainee data were analyzed – one category concerns team outcomes. Team defense score and offense scores are recorded and decomposed at individual level during each trial. The other category concerns the team process; coaching agents keep track of trainees' collaborative events such as assist attack, assist ID, transfer ID and etc, and record the number and successful rate for each event type. As we mentioned earlier, teams are randomly assigned to experiment group and control group (randomly select half of the teams to be in the experiment group). The results show improvements of the experiment group over the control group for each training session (Baseline, training1, training2, training3) where each trial of the training session contains a set of scenario files randomized via Latin Square Randomization.

4.3.1 Team Outcome Improvement

We measure team outcome by looking at the individual offense scores and the individual defense scores during each training session. Each trial contains four time intervals within a particular scenario. During each trial, individual team member will encounter different workloads for each time interval, yet the overall workload averaging over all the time intervals remains equal. Team performance improvement can be broken down into individual level outcome shown in Figure 13 and Figure 14, which shows the average individual offense score and defense score for the team members within the control group and those within the experiment group. The score is averaged over four different trials, biases introduced by which is counterbalanced by Latin-square Randomization. Our data contains a set of individual performance scores; each score represents trainee's performance for a particular trial (individual offense scores are shown in Figure 13 and individual defense score are shown in Figure 14).

The red bar represents the mean and 95% confidence interval (estimated using 1.96*standard error) of the individual performance scores for trainees in the experiment group. The blue bar represents the mean and 95% confidence interval of the individual performance scores for trainees in the control group. The bar chart gives us an overview of the advantages of individual scores within the experiment group compared to those within the control group. At the beginning of the baseline mission, neither control team nor the experiment team received any coaching feedbacks. Thus we expect similar baseline performance for the control group and the experiment group due to the randomized assignment of individual team members to both groups. As shown from Figure 13, the average scores for both groups in baseline session are about to be the same, and the confidence interval for the means have much overlap. For the three training sessions that follow the baseline, the bar charts (Figure 13 and Figure 14) show that both control and experiment group improve their team outcome after each mission by going through a set of trials—respectively Baseline, Training 1, Training 2 and Training 3.

To statistically compare the two group means, a two-way ANOVA test was conducted on our sets of data with treatment (control group and experiment group) and trial (Baseline, Training 1, Training 2, and Training 3) as the two factors. For the main factor treatment it yields a p-value of 0.000, for the main factor trial, it yields a p-value of 0.000 and for the interaction, a p-value of 0.010 (see Table 12 for the corresponding confidence intervals, Figure 25 and Figure 27 in Appendix D Two-way ANOVA Test Results – Confident Intervals). The p-value of the treatment factor shows significant differences between the two groups of trainees, and supported our hypothesis that trainees who receive the coaching feedback from agents achieve better individual offensive outcome within a team collaboration context. The p-value of the trial factor shows significant differences between the different trials, and supported our hypothesis that trainees' individual performance score increased from trial to trial. The p-value of the interaction shows that the different treatment has an effect on the score increase from trial to trial, particularly Figure 13 shows us that the score increases for the experiment group from trial to trial is better than the score increase for the control group.

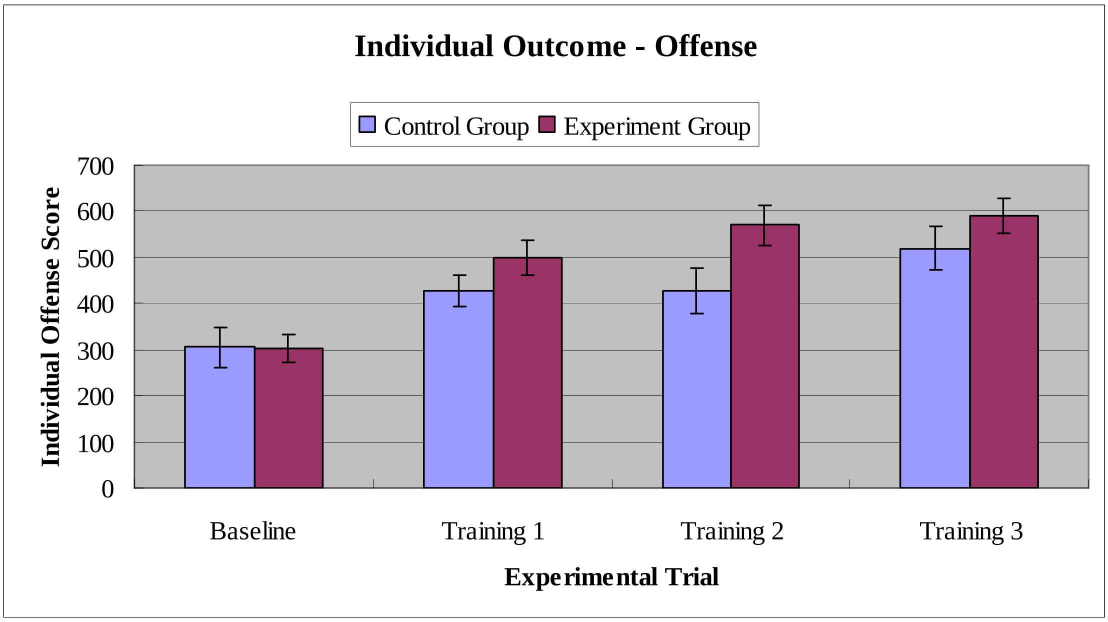


Figure 13 Individual Outcome in Team-context (Individual Offense Scores)

Source of Variation	**DF**	**Sum of Squares**	**Mean Square**	**F**	**P-value**
Treat	1	199516	199516	22.08	0.000
Trial	3	1394672	464891	51.44	0.000
Interaction	3	105172	35057	3.88	0.010
Error	152	9037	1373625		
Total	159	3072984			

Table 12 Two-way ANOVA Test Results (Individual Offense Scores)

As we mentioned earlier, team defense score consists of a set of mission specific objectives addressing different performance aspects to better achieve team mission (Figure 19). Table 8 listed the overall team defense scores that summarize how team achieved those mission objectives. For the overall team defense mission, teams in both the experiment and the control group start about the same. Figure 14 shows the mean and

95% confidence interval (estimated using 1.96 * standard error) of individual defense scores for trainees in both groups. For the baseline trial, we can see that the mean of individual scores within the two groups are about to be the same, and the confidence intervals have much overlap, thus it doesn't show significant difference among the two groups for baseline trial. At the end of training session 1, member in both teams has a score increase (see Figure 14). For the following two trails (Training2 and Training 3), improvement of defense outcome for trainees in both groups had reduced to a lower rate as seen from each individual member's performance score increase.

The two-way ANOVA test yields p-values of 0.001, 0.000 and 0.311 respectively for the treatment factor, the trial factor and the interaction between these two factors (see Table 13). The p-value of the treatment factor (0.001) shows a significant score differences between the two groups, which corresponds to better individual performance for trainees in the experiment group in term of defending team's restricted zones. The p-value of the trial factor (0.000) shows significant differences between scores from trial to trial, which corresponds to significant improvement of team's overall defensive performance from trial to trial. The p-value of the interaction (0.311) shows no support for significant interactions between the treatment factor and the trial factor. To see the evidence of the different defensive score improvements between the two groups, another two-way ANOVA test is conducted in which we take the score differences of later trials from initial trial as the response (instead of taking the absolute scores for each trial as the response measurement).

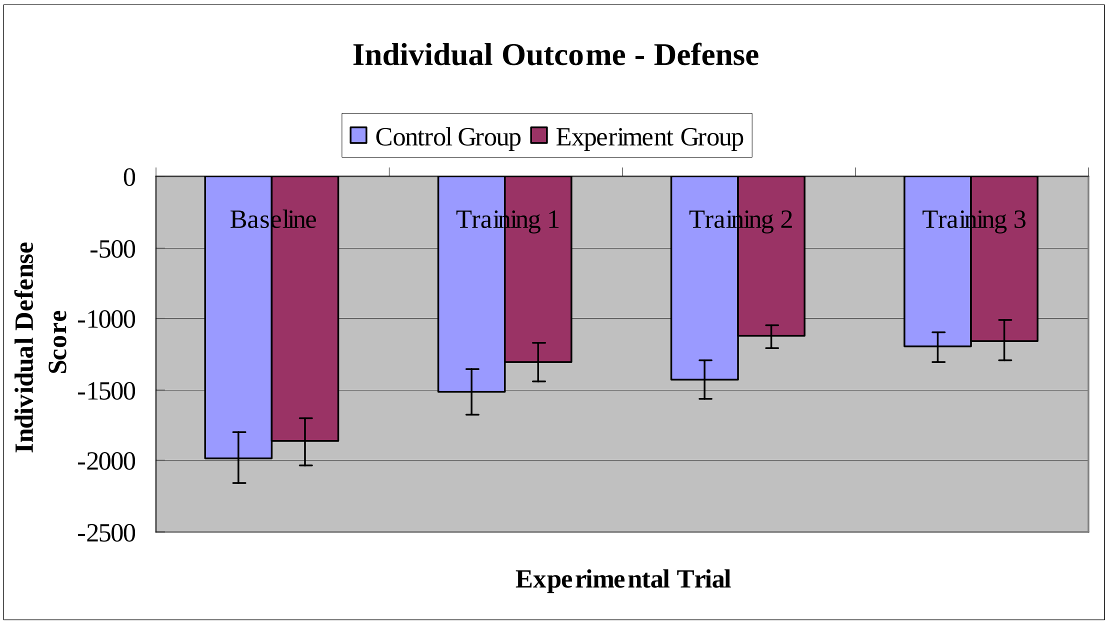


Figure 14 Individual Outcome in Team-context (Individual Defense Scores)

Source of Variation	**DF**	**Sum of Squares**	**Mean Square**	**F**	**P-value**
Treat	1	1148363	1148363	10.86	0.001
Trial	3	13213429	4404476	41.65	0.000
Interaction	3	381596	127199	1.20	0.311
Error	152	16073757	105748		
Total	159	30817145			

Table 13 Two-way ANOVA Test Results (Individual Defense Scores)

Figure 15 and Figure 16 listed the average team offense score increase and the average team defense score increase. We observe that trainees in both groups improved on their team scores from trial to trail. There’s a leading advantage of performance improvement for trainees who receives coaching feedback during the after-action-review session for all the trials except the baseline. Another interesting observation is that the rate at which trainees improve team performance decreased for later trials, especially after teams are passing average performance and reaching to the successful performance range. The performance benchmarks (e.g. offense and defense scores that indicate successful performance) have been gathered by similar human subject experiment in DDD simulation domain, a range of performance success is listed in Appendix B Debriefing Check Lists.

For individual offense score improvement, the two-way ANOVA test yields p-values of 0.000, 0.003 and 0.290 respectively for the treatment factor, the trial factor and the interaction between these two factors (see Table 14). The p-value of the treatment factor (0.000) shows significant differences of the individual offense score improvement of the two treatment groups, which corresponds to better individual performances for trainees in the experiment group given the means of individual offense score improvements shown in Figure 15. The p-value of the trial factor (0.003) shows significant differences between individual offense score improvements from trial to trial, which corresponds to significant improvement of team's overall offensive performance from trial to trial. The p-value of the interaction (0.290) shows no support for significant interactions between the treatment factor and the trial factor, taken the individual offense score improvement as the response.

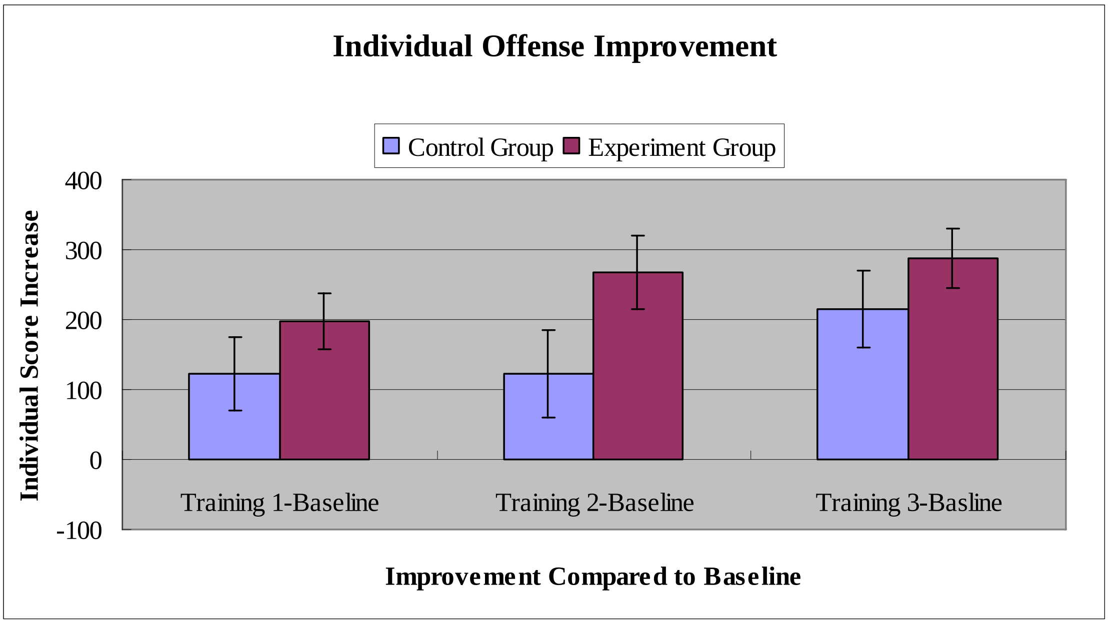


Figure 15 Individual Offense Score Improvement

(Offense Score Increase per Training Trial Compared to Baseline)

Source of Variation	**DF**	**Sum of Squares**	**Mean Square**	**F**	**P-value**
Treat	1	285187	285187	21.07	0.000
Trial	2	169542	84771	6.26	0.003
Interaction	2	33875	16937	1.25	0.290
Error	114	1543375	13538		
Total	119	2031979			

Table 14 Two-way ANOVA Test Results (Individual Offense Score Improvement)

For individual defense score improvement, the two-way ANOVA test yields p-values of 0.003, 0.006 and 0.701 respectively for the treatment factor, the trial factor and the interaction between these two factors (see Table 15). The p-value of the treatment factor (0.003) shows significant differences of the individual defense score improvements of the two treatment groups, which corresponds to better individual performances for trainees in the experiment group given the means of defense score improvement shown in Figure 16. The p-value of the trial factor (0.006) shows significant differences between individual defense score improvements from trial to trial, which corresponds to significant improvement of team's overall defense performance from trial to trial. The p-value of the interaction (0.701) shows no support for significant interactions between the treatment factor and the trial factor, taken the individual defense score improvement as the response.

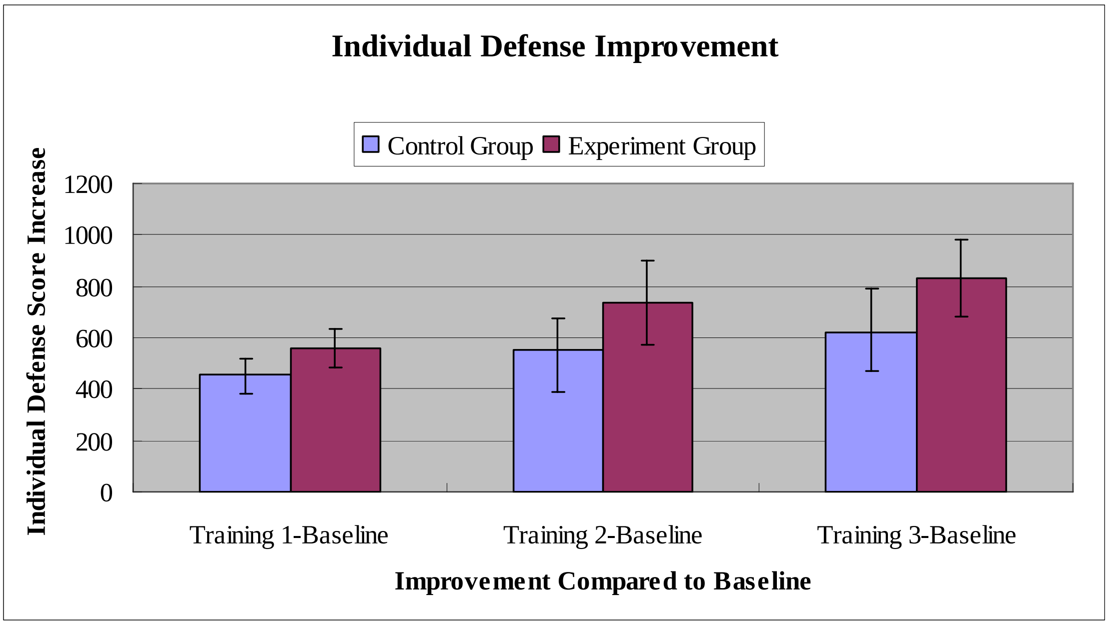


Figure 16 Individual Defense Score Improvement

(Defense Score Increase per Training Trial Compared to Baseline)

Source of Variation	DF	Sum of Squares	Mean Square	F	P-value
Treat	1	838508	838508	9.50	0.003
Trial	2	951277	475638	5.39	0.006
Interaction	2	62875	31437	0.36	0.701
Error	114	10063074	88273		
Total	119	11915734			

Table 15 Two-way ANOVA Test Results (Individual Defense Score Improvement)

4.3.2 Team Process Improvement

In this section of data analysis, we focus on the process measurement about team member's helping behavior during mission. As we mentioned in the experiment design (see section 4.2). The experiment is designed so that to achieve better mission performance, members on a team have to help each others and balance the team's overall workload and resource capacity. During trainee's mission execution, coaching agents collected data about the following helping events:

1. Assist Attack (Team member helped others with attacking incoming tasks outside his/her own quadrant)
2. Assist Identification (Team member helped others with identifying incoming tasks outside his/her own quadrant)
3. Communication of Identification (Transfer identification/ID information to team mates when a decision maker cannot attack the incoming task right away, e.g. he or she doesn't have the right powered asset)

As one of the most important domain operations, attack has a significant impact on team's defensive and offensive scores— teams destroy incoming hostile tasks so as to avoid their staying in the restricted zones and for each successful attack, individual decision maker (and team) receive 50 points offense score increase. Figure 17 and Figure 18 show an improvement of experiment team's performing more frequent attacks that help others with tasks outside DM's own quadrant. Specifically, Figure 17 shows the mean of total number of assist attacking attempts for each trial (Baseline, Training 1, Training 2, and Training 3) and Figure 18 shows the mean of the total number of assist attacks that succeed (attack hostile tasks with the right powered asset and with the right timing).

During the feedback presentation session, coaching agents provide an overview of the attack related performance to each individual team member, including the number of assist attacks that the decision maker (DM) performed during each time interval, the number of assist attack attempts and the successful rate of these assist attacks. The attack feedback also includes whether this DM is overloaded—in case a DM has more assets than needed, coaching agent provide suggestions about using his/her own idled assets to help with attack in others' quadrant and thus improve the assist attack and the successful rate. As we mentioned in section 3.6, coaching agent will also provide detailed helping guidance to trainees regarding attack based on trainee's specific resource usage and online performance.

A two-way ANOVA test was conducted on the number of assist attacks for trainees in both groups from trial to trial. It yields p-values of 0.000, 0.000 and 0.614 respectively for the treatment factor, the trial factor and the interaction between these two factors (see Table 16). The p-value of the treatment factor (0.000) shows significant differences of the number of assist attack attempts of the two treatment groups. It corresponds to better individual helping performances (assist attack) for trainees in the experiment group if we look at the means of assist attack number shown in Figure 17. The p-value of the trial factor (0.000) shows significant differences between the numbers of assist attacks from trial to trial, which corresponds to significant improvement of team's helping performance regarding assist attack from trial to trial. The p-value of the interaction (0.614) shows no support for significant interactions between the treatment factor and the trial factor regarding the number of assist attacks.

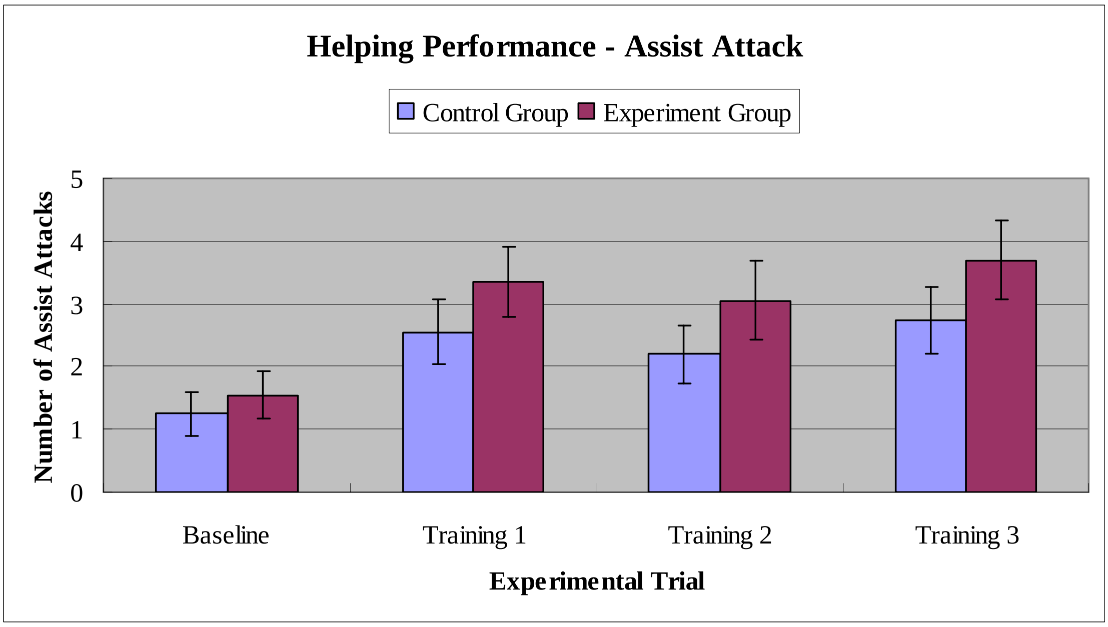


Figure 17 Helping Related Performance (Assist Attack)

Source of Variation	**DF**	**Sum of Squares**	**Mean Square**	**F**	**P-value**
Treat	1	21.02	21.02	15.05	0.000
Trial	3	77.75	25.92	18.56	0.000
Interaction	3	2.53	0.84	0.60	0.614
Error	152	212.30	1.40		
Total	159	313.60			

Table 16 Two-way ANOVA Test Results (Assist Attack)

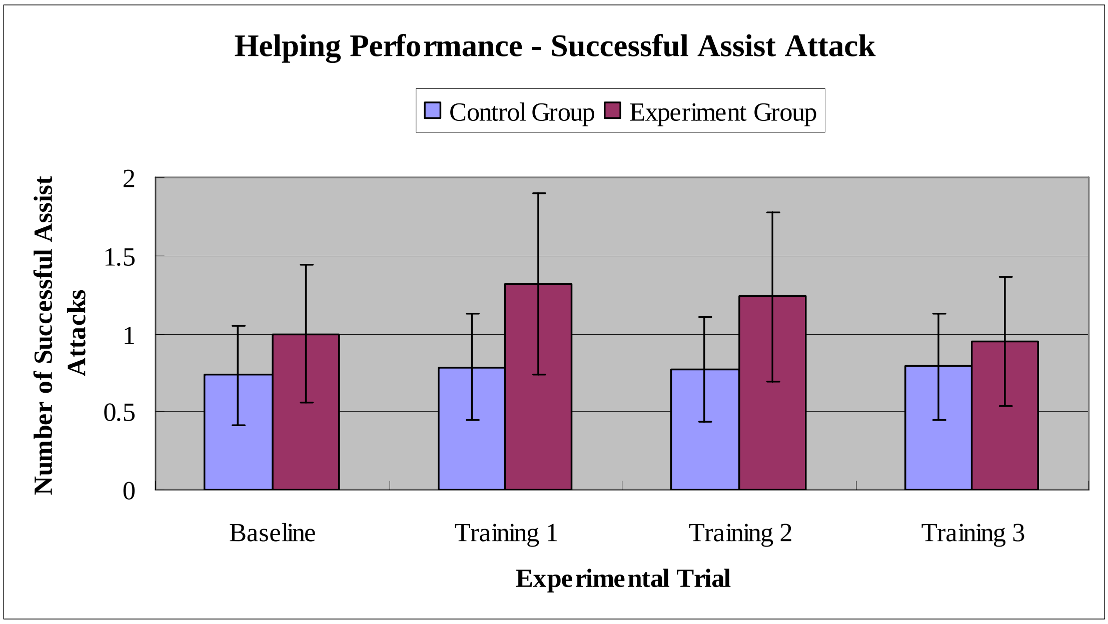


Figure 18 Helping Related Performance (Successful Assist Attacks)

Source of Variation	**DF**	**Sum of Squares**	**Mean Square**	**F**	**P-value**
Treat	1	12.100	12.100	12.87	0.000
Trial	3	18.525	6.175	6.57	0.000
Interaction	3	2.250	0.750	0.80	0.497
Error	152	142.900	0.940		
Total	159	175.775			

Table 17 Two-way ANOVA Test Results (Successful Assist Attack)

Similarly we did a two-way ANOVA analysis on successful assist attack. The significant differences of the related helping performance between two groups and the differences from trial to trial can be seen from the p-values for the treatment and trial factors (yield p values of 0.000 and 0.000 in Table 17). The p-value of the interaction (0.497 in Table 17) shows no significant interaction between these two main factors.

Analysis was also made on the assist ID and the communication of identified task information (successful rate of ID isn't included since the domain rule ensures that every attempt of identification leads to a successful ID operation). The ID operation is essential for successful attacking of any incoming tasks— direct attack without ID may lead to attacking of neutral tasks or attacking higher-powered tasks, both lead to attack mission failure. Figure 19 shows that the experiment group has significant improvement on assisting other quadrant with ID, which leads to a lower identification/ID burden on the already overloaded DM(s) and a higher rate of successful attacks in the overloaded quadrant.

The ANOVA test yields p-values of 0.000, 0.000 and 0.077 for the treatment factor, the trial factor and the interaction between the two factors. It shows that the trainees in the two groups performs differently regarding assist ID for all the trials, and the bar chart in Figure 19 shows that for each trial the experiment group has a much better performance. Regarding the trial factor for trainees in both groups, their assist ID performances differ from trial to trial, specifically the bar chart in Figure 19 shows that trainees improved their assist ID performance from trial to trial. The p-value of 0.077 shows no significant interaction between the main factors of treatment and trial.

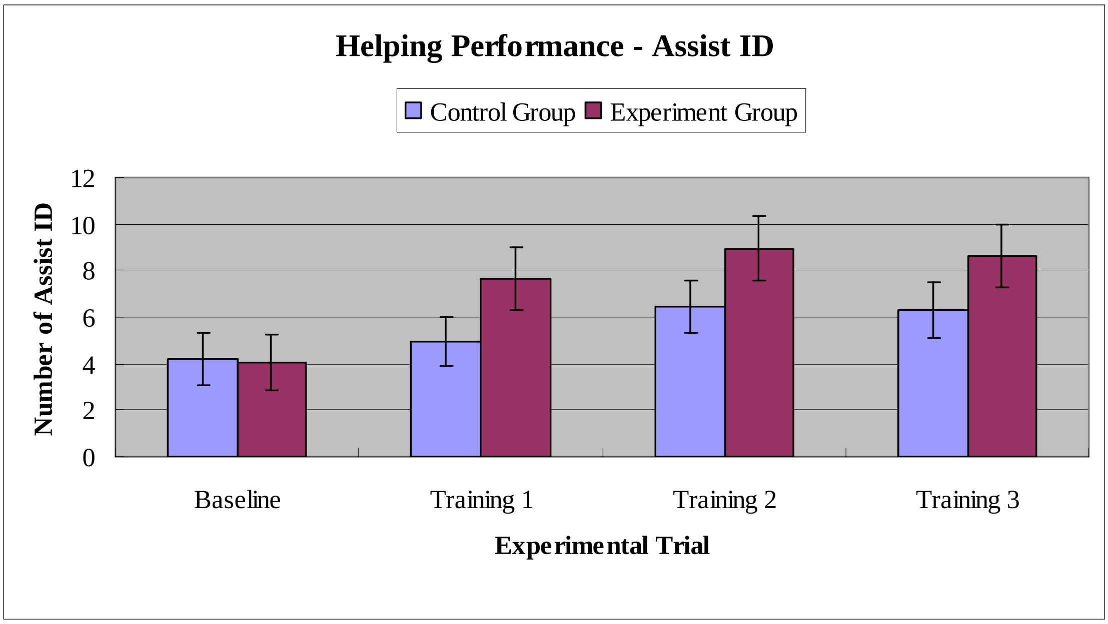


Figure 19 Helping Related Performance (Assist ID)

Source of Variation	DF	Sum of Squares	Mean Square	F	P-value
Treat	1	136.90	136.90	17.72	0.000
Trial	3	321.25	107.08	13.86	0.000
Interaction	3	53.95	17.98	2.33	0.077
Error	152	1174.30	7.73		
Total	159	1686.40			

Table 18 Two-way ANOVA Test Results (Assist ID)

Assist ID can lead to an assist attack performed by the helping DM or an attack performed by another DM other than the one who did ID. In the second case, the transfer of ID is an inescapable step to let other members know the power information of identified task. In this case, each successful attack consists of the identification of the task by one team member, the communication of such information among the team, and the actual attack of that task by another member of the team. Figure 20 shows an advantage of the experiment group over the control group regarding the above mentioned collaboration process in attacking hostile tasks.

From the two-way ANOVA test, we see the p-values of the two main factors are both quite small (0.000 and 0.000 in Table 19), which supports our hypothesis that trainees' helping performances regarding transferred ID differ significantly between two groups for all the trials, and that for trainees in both groups, their performance regarding transferred ID differ significantly from trial to trial. The p-value of 0.001 (Table 19) indicates there is an interaction between the two factors. Figure 20 shows the constant improvement of transfer ID for trainees in the experiment group, and a relatively randomized transfer ID pattern for trainees in the control group.

Another interesting observation is that better team processes are associated with better team outcome. It supports our hypothesis that selected team process measurements contribute to the overall team outcome and that these helping events have a positive impact on team outcome by balancing team workload and resources distribution and by saving significant mission time. Taken the transfer of identification information for example, teams who actively engaged in these helping related activities have much better team offense and defense scores.

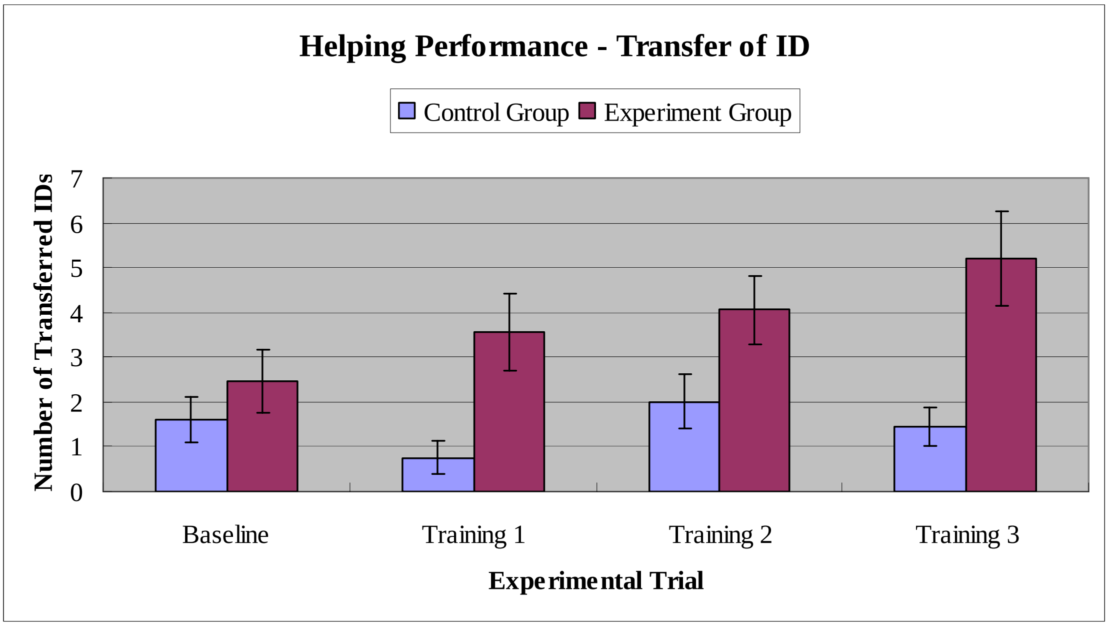


Figure 20 Helping Related Performance (Communicated ID)

Source of Variation	**DF**	**Sum of Squares**	**Mean Square**	**F**	**P-value**
Treat	1	223.26	223.26	88.94	0.000
Trial	3	49.42	16.47	6.56	0.000
Interaction	3	45.02	15.01	5.98	0.001
Error	152	381.55	2.51		
Total	159	699.24			

Table 19 Two-way ANOVA Test Results (Communicated ID)

4.3.3 Survey Results:

As we mentioned in section 4.2, we conducted a survey at the end of the experiment where participants were asked to fill out a short questionnaire. Information we collected includes participants' self-evaluation of their game experiences and team play skills, their feedbacks about the debriefing session planning review and execution review (for participants in the control group), their feedbacks about planning-phase coaching agents and the execution-phase coaching agents (for participants in the experiment group). Figure 21, Figure 22, Figure 24 and Figure 24 show the results of the post-experiment survey. Most participants consider themselves as good team players who had adequate amount of game experiences for participants in both control and experiment groups. For participants in the experiment group, more than 85% of the participants view the planning-phase and the execution-phase coaching agents as helpful. (See Appendix C Post Experiment Questionnaire for details)

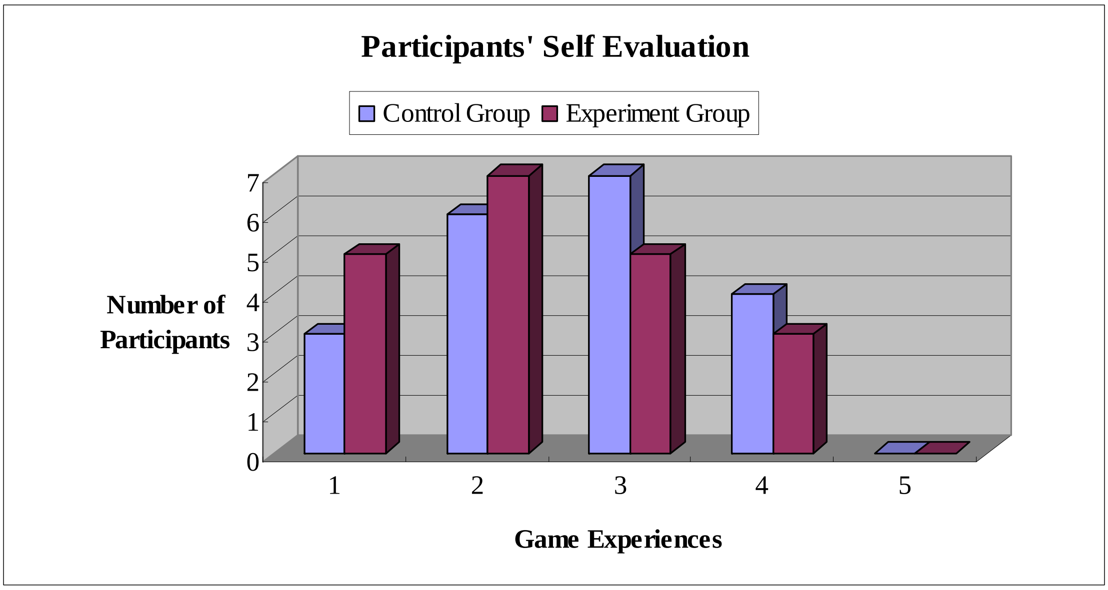


Figure 21 Participants’ Self-evaluation of Game Experiences

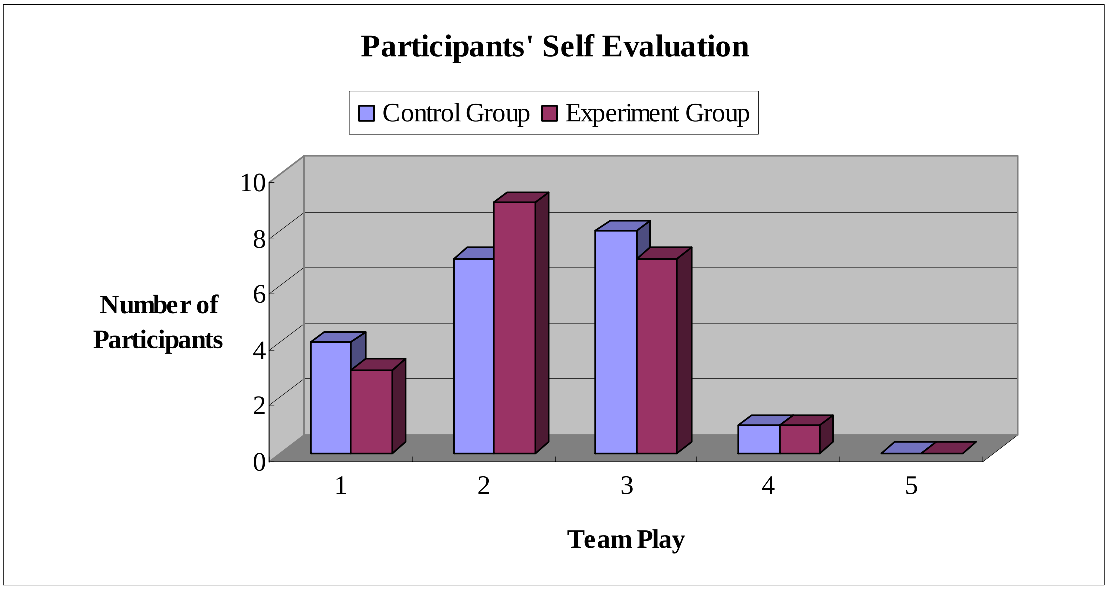


Figure 22 Participants' Self-evaluation of Team Play Skills

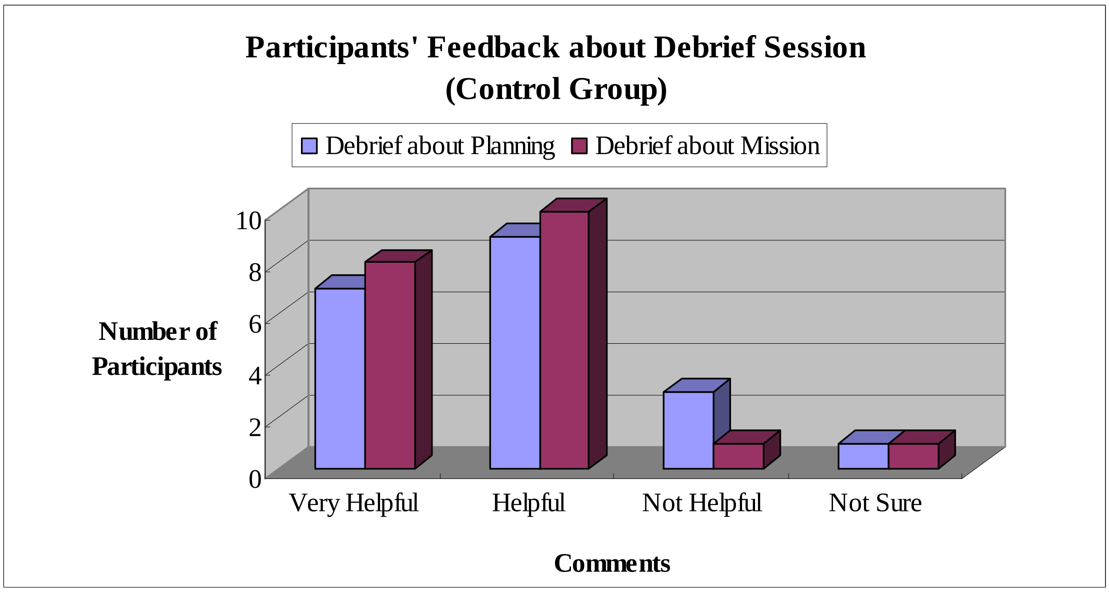


Figure 23 Participants' Feedback about the Debrief Session (Control Group)

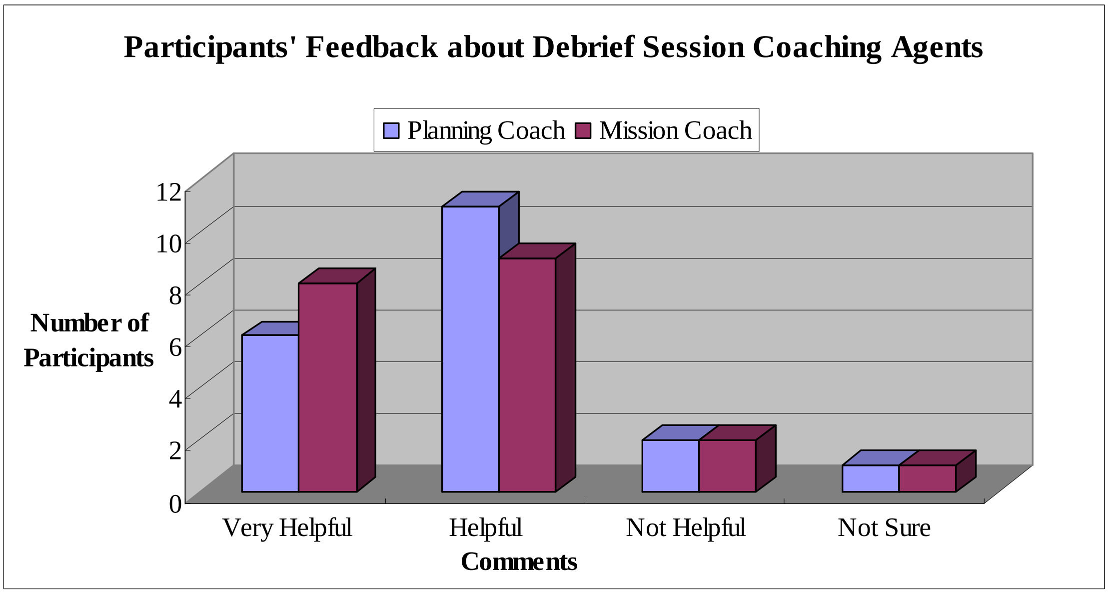


Figure 24 Participants' Feedback about the Two-phase Coaching agents

4.4 Discussion

The advantage of the experiment group over the control group is twofold:

1. The overall team outcome improved significantly—trainees who had coaching feedbacks during after-action-review sessions performed better for the defensive and offensive missions than trainees in the control condition on all three assessment missions (Training 1, Training 2 and Training 3).

2. The helping related team processes improved significantly—trainees who had coaching feedback during after-action-review session show significant online improvement regarding their helping behaviors during the next mission-execution session.

As we discussed in the result section, the overall team defense and offense scores represent the degree of mission success and indicate trainees' collective performance regarding five mission objectives (see Table 10). For these mission specific sub-goals, execution-phase coaching agents generated individual level feedback that addresses trainee deficiency regarding their mission-specific task performance. The improvement of average team offense and defense scores compared to the ones of the control group thus supported our hypothesis that coaching feedback generated by the intelligent training system helped trainees improve their mission performance regarding explicit sub-goals.

As a result of team process monitoring, coaching agents recorded critical events about team members' helping behavior. Given a specific scenario with unbalanced workload assigned on each member, coaching agents generate feedback about how such helping events might contribute to the overall mission success. In this study, coaching agents focus on four measurements to assess team performance regarding helping events—assist attack (attempt to help other member with attacking tasks), successful assist attack (the assist attack with the right asset and timing), assist identification (help other member with identifying tasks), transfer identification (communicate with other team members about the identified task information). The results show that the experiment group has a significant advantage over the control group regarding the average of the four measurements. Together with results showing team outcome improvements, the advantages of experiment group in terms of these four team process measurements show a positive correlation between team's online performance improvement and the two-phase training protocol. More specifically, the helping related event analysis shows strong support to our hypothesis that the after-action-feedback generated by the coaching agents can help trainees improve not only their team outcome but also the team process regarding critical collaborative events.

In summary, the significant advantage of the trainees who received coaching feedback over those trainees under the control conditions supports the effectiveness of the two-phase coaching agents and supported our design choices regarding feedback generation and presentation (e.g. content, organization timing and etc)

5 Conclusion and Future Work

In this research, we designed, implemented and tested coaching agents within an agent-based intelligent team training framework that provides offline planning and performance feedback regarding trainees' helping behavior as one of the important dimension of team collaborative behaviors.

The agent-based intelligent team training approach proposed in this dissertation provides automation support to human teamwork training. The selection of a two-phase training protocol and the implementation of the coaching agents provide a solid foundation that allows us to exercise the full circle of the event-based training approach. Through cognitive task analysis, we identified our explicit team-oriented training goal as facilitating trainee's improvement on within-team helping behaviors. The individual and team monitoring capability of the coaching agents enabled us to collect data about trainee performance around critical domain events that address either their operational performance or their collective performance at the team level. The intelligent assessment modules within our intelligent coaching framework compare trainee actions against a set of performance criteria and diagnose trainees' performance deficiencies addressing different aspect of team performance. The team-oriented feedbacks are organized to address both taskwork and teamwork performance with a focus on teamwork oriented events. The feedbacks are represented via a set of user friendly interfaces.

We tested our implementation of the two-phase coaching agent within a command and control (C2) task environment and conducted human subject experiment to validate the effectiveness of such training approach. The data collection and analysis of the experiment is twofold: we evaluate the improvement of team collaborative processes as well as team outcome—team defense and offense scores reflect the overall mission success and team process statistics show strong indication about whether a team is actively engaged in helping each other during mission. Results showed advantage of the experiment group relative to the control group regarding the overall team outcome and the helping related team processes improvements. The distinct team outcome and process improvement of the experiment group has a strong support to our hypothesis that our training protocol and the coaching agents we used to facilitate training have a positive impact on team's mission performance and outcome— trainees who had coaching agent feedback during after-action-review sessions achieve better team outcome for the next mission and have significant improvement on their helping each other within the team, as one of the important collaborative team processes.

5.1 Significance of Research

This research demonstrates a full circle of event-based training approach via the design and construction of intelligent coaching agents within a team-oriented training framework. We use intelligent software agents to substitute the roles of human coaches and validated the effectiveness of our training approach and the feedback generated by coaching agents. Our design, implementation of intelligent coaching agents (and the agent-based training approach in general) is a successful prototype in automating human team training facilitated by intelligent agent coaches.

Knowledge modeling is one of the most important research issues that we targeted to address in this research. In building the agent-based coaching framework and the corresponding coaching agent components, we modeled the team related knowledge with adequate applicability to other team-oriented domains. We established our work by making the following major design decisions in building the agent-based intelligent training system.

Due to the broad scope of teamwork research, we narrowed our scope to study the helping behaviors of teamwork. As one of the domain independent teamwork dimension, helping behavior is general enough to contain many other overlapping teamwork dimensions that are of our interests, such as communication and collaboration, yet it is specific in a degree that we can observe trainees' actions and behaviors in a context of workload and resource imbalance when team members are required to help each other to achieve the overall team mission. During the helping process, we allow trainees to communicate in coordinating their helping plans, monitor their helping related performance during mission and provide them with feedbacks that address their helping deficiencies.

The choice of a time-stressed command and control domain allows us to model trainee performance in two interdependent phases. In many other common team domains without intensive time-constraints, people can acquire knowledge through work practice. In such cases, e.g., for scholars to collaborate on writing a paper together, it is arguably true that training might not be necessary or even helpful. For command and control teams, given the time stress involved in performing the mission and the great impact on team performance of the critical decisions that to be made, training is essential and team performance can be greatly improved if team members are given extra time during the planning phase to collaborate on a team plan with specific goals. Specifically in this study, the performance goal concerns the planning and execution of team's helping behaviors. On the other hand, our design of the two-phase training protocols also allowed us to study planning, as an explicit activity that occurs before the performing of tasks. Such attempts haven't been made in either the intelligent tutoring literature or the team training literature.

Successful teamwork is only achievable with team members' adequate individual tasks skills and more importantly the ability to effectively work together as a team. Included in the designing our intelligent coaching agents, we have not only the essential mechanism to facilitate the monitoring and assessment of task specific work for each individual, but also a generic teamwork model that captures the performance monitoring and assessment regarding one of the most important dimensions of teamwork— the helping behavior among team members. The interactions of various intelligent assessment and diagnosis modules are domain-independent which implies that the explicit team model in our design can be applied to other team-oriented task domains.

In this research, we distinguish taskwork and teamwork in every step before providing well-tailored feedbacks to trainees at both individual level and team level. We gradually construct intelligent coaching agents' knowledge about trainees by monitoring individual's task specific actions and the collective event oriented actions at team level. Within the intelligent assessment module, we also have distinct sets of performance criteria to diagnose trainee's task or collaborative deficiency. The effectiveness of our intelligent agent-based training approach thus relies on coaching agents' ability to generate well-organized feedback regarding both trainee's taskwork skills and teamwork skills.

5.2 Future work

In studying the helping related behaviors, we focused on trainees' planning of helping behavior in terms of their resource allocation plans and trainee's execution of their helping behavior regarding several helping related assist behavior. Trainee knowledge was modeled at individual and team level yet no explicit high level cognitive support was addressed. Beyond the monitoring of trainee actions and the inference about their potential causes of deficiencies, cognitive decision making framework can be incorporated to better understand trainees shared mental model towards helping behavior. For example, a naturalistic decision making framework such as the Recognition-Primed Decision (RPD) Model can be considered and adopted to help improve future research.

The training protocol we proposed distinguishes the planning phase and execution phase of a mission. The results of our human experiment showed that a combination of planning phase and execution phase coaching agents improved trainees' performance process and outcome as a team. It would yield interesting implications to the design of such training protocol if we make an extension of our work to evaluate the effectiveness of coaching agents on each phase separately. For example, human subject experiment can be conducted separately for trainees who only have the planning phase coaching support and for trainees who only have the execution phase coaching support.

Bibliography

1. Salas, E., Dickinson, T. L., Converse, S. A., and Tannenbaum, S. I., *Toward an understanding of team performance and training*, in *Teams: Their training and performance*, R.W. Swezey, and Salas, E., Editor. 1992, Ablex Publication Corp: Norwood, NJ. p. 3-29.
2. Nieva, V.F., Fleishman, E. A., & Rieck, A., *Team Dimensions: Their identity, their measurement, and their relationships*. 1978, Advanced Research Resources Organization: Washington, DC.
3. Sycara, K.a.L., M. *Modeling teams of specialists*. in *Proceedings of the Tewnty-Seventh Hawaii International Conference on System Sciences*. 1994.
4. Zachary, W., Cannon-Bowers, J., Bilazarian, P., Krecker, D., Lardieri, P., and Burns, J., *The Advanced Embedded Training System (AETS): An intelligent embedded tutoring system for tactical team training*. International Journal of Artificial Intelligence in Education, 1999(10): p. 257-277.
5. Cho, B., Michael, J. A., Rovick. A. A, Evens, M. W. *An Analysis of Multiple Tutoring Protocols*. in *Intelligent Tutoring Systems: 5th International Conference*. 2000. Montreal, CA: Springer.
6. Woo, C.W., *Instructional Planning in an Intelligent Tutoring System: combining global lesson plans with local discourse control*, in *Computer Science Department, Illinois Institute of Technology*. 1991.

7. Smith-Jentsch, K.A., Zeisig, R. L., Acton, B., and McPherson, J. A., *Team dimensional training*, in *Making Decisions under Stress: Implications for Individual and Team Training*, J.A.C.-B.a.E. Salas, Editor. 1998, APA Press: Washington, DC.

8. Johnston, J.H., Poirier, J., and Smith-Jentsch, K. A., *Decision Making Under Stress: Creating a Research Methodology*, in *Making Decisions under Stress: Implications for Individual and Team Training*, J.A. Cannon-Bowers, and Salas, E., Editor. 2000, American Psychological Association: Washington, DC.

9. Collyer, S.C.a.M., G. S., *Tactical Decision Making Under Stress: History and Overview*, in *Making Decisions Under Stress: Implications for Individual and Team Training*, J.A. Cannon-Bowers, and Salas, E., Editor. 2000.

10. Porter, C., Hollenbeck, J.R., Ilgen, D.R., Ellis, A.P.J., West, B.J., and Moon, H.K, *Towards understanding backing up behaviors in work teams: the role of personality and the legitimacy of need for backing up others.* Journal of Applied Psychology, 2003. **88**: p. 391-403.

11. Katzenbach, J.R., and Smith, D. K., *The Wisdom of Teams: Creating the High-Performance Organization*. 1993, Boston, Massachusetts: Harvard Business School Press.

12. Fleishman, E.A., and Quaintance, M. K., *Taxonomies of Human Performance: The Description of Human Tasks*. 1984, Boston, MA: Academic Press.

13. Cannon-Bowers, J.A., and Bell, H. H., *Training decision makers for complex environments: Implications of the naturalistic decision making perspective*, in

Naturalistic decision making, G.A.K. Caroline E. Zsambok, Editor. 1997, Lawrence Erlbaum Associates, Inc: Hillsdale, NJ. p. 99-110.

14. Smith-Jentsch, K.A., Johnston, J. H., and Payne, S. C., *Measuring Team-Related Expertise in Complex Environments*, in *Making Decisions under Stress: Implications for Individual and Team Training*, J.A. Cannon-Bowers, and Salas, E., Editor. 2000, American Psychological Association: Washington, DC.

15. Johnston, J.A., Smith-Jentsch, K. A., and Cannon-Bowers, J. A., *Performance measurement tools for enhancing team decision making*, in *Team performance assessment and measurement: Theory, methods, and applications*, E.S. M. T. Brannick, & C. Prince, Editor. 1997, LEA: Mahwah, NJ. p. 311-327.

16. Grosz, B., *Collaborative Systems: 1994 AAAI Presidential Address.* 1996. **2**(17): p. 67 - 85.

17. Dickinson, T.L.M., R. M., *A conceptual framework for teamwork measurement*, in *Team performance assessment and measurement: Theory, methods, and applications*, E.S. M. T. Brannick, & E. Prince, Editor. 1997, Lawrence Erlbaum Associates, Inc: Mahwah, NJ. p. 19 - 43.

18. McIntyre, R.M., & Salas, E., *Measuring and managing for team performance: Lessons from complex environments*, in *Team effectiveness and decision-making in organizations*, R.A.S. Guzzo, E., Editor. 1995, Jossey-Bass: San Francisco. p. 9-45.

19. Bruslovsky, P., *Adaptive and Intelligent Technologies for Web-based Education.* Kunstiche Intelligenz, Special issue on Intelligent Systems and Teleteaching, 1999. **4**: p. 19-25.

20. Bueno, D., David, A. A. *METIORE: A Personalized Information Retrieval System*. in *Proceedings of the 8th International Conference on User Modeling*. 2001.

21. Zukerman, I. *An Integrated Approach for Generating Arguments and Rebuttals and Understanding Rejoinders*. in *the 8th International Conference on User Modeling*. 2001. Sonthofen, Germany.

22. Zhou, Y.a.E., M. *A practical student model in an intelligent tutoring system*. in *11th IEEE International Conference on Tools with Artificial Intelligence*. 1999. Chicago, IL.

23. Collins, A., Brown, J. S., & Holum, A., *Cognitive apprenticeship: Making thinking visible.* American Educator, 1991. **6-11**(38-46).

24. Collins, J.A., Greer, J.E., and Huang, S.X. *Adaptive assessment using granularity hierarchies and bayenesian nets*. in *Proceedings of Intelligent Tutoring Systems (ITS'96)*. 1996. Montreal Canada.

25. Graesser, A.C.a.V., .K, Rose, C. P., Jordan, P. W. , and Harter D., *Intelligent Tutoring Systems with Conversational Dialogue.* AI Magazine, 2001. **22**(4): p. 39-52.

26. Corbette, A.T. *Cognitive Computer Tutors: Solving the Two-Sigma Problem*. in *Proceedings of the 8th International Conference on User Modeling 2001*. 2001: Lecture Notes In Computer Science.

27. Anderson, J.R., Corbett, A. T., Koedinger, K. R., and Pelletier, R., *Cognitive Tutors: Lessons Learned.* The Journal of the Learning Sciences, 1995. **4**(2): p. 167-207.

28. Lesgold, A.M., Lajoie, S. P., Bunzo, M., and Eggan, *SHERLOCK: A coached practice environment for an electronics troubleshooting job*, in *Computer assisted instruction and intelligent tutoring systems*, J. Larkin, and Chabay, R., Editor. 1993, Lawrence Erlbaum Associates, Inc: Hillsdale, NJ.

29. VanLehn, K. *Conceptual and meta learning during coached problem solving*. in *Proceedings of the 3rd International Conference on Intelligent Tutoring Systems*. 1996: Springer.

30. VanLehn, K., *Cognitive skill acquisition.* Annual Review of Psychology, 1996(47): p. 513-539.

31. Burns, H.L.a.C., C. G., *Foundations of intelligent tutoring systems: an introduction*, in *Foundations of intelligent tutoring systems*, M.C.a.R. Poison, J. J., Editor. 1988, Lawrence Eribaum Associates: Hillsdale, NJ. p. 1-19.

32. Anderson, J.R., *Rules of the mind*. 1993, Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

33. Wenger, E., *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. 1987, Los Altos, CA: Morgan Kaufman Ed.

34. Cohen, P.R.a.P., C. R., *Elements of a Plan-based theory of speech acts.* Cognitive Science, 1979. **3**: p. 177-212.

35. Kass, R. *Student modeling in intelligent tutoring systems - implications for user modeling*. in *User Models in Dialog Systems*. 1989. Berlin, Heidelberg: Springer-Verlag.

36. Finin, T.W. *GUMS: A general user modeling shell*. in *User Models in Dialog Systems*. 1989. Berlin, Heidelberg: Springer-Verlag.

37. Baffes, P.a.M., R., *Refinement-Based Student Modeling and Automated Bug Library Construction.* Journal of Artificial Intelligence in Education, 1996. **7**(1): p. 75-116.

38. Joseph, B., Stern, M., and Woolf, B.P. *Using the Student Model to Control Problem Difficulty*. in *User Modeling: Proceedings of the Sixth International Conference*. 1997. Vienna, New York: Springer Wien.

39. Kono, Y., Ikeda, M. and Mizognchi, R., *THEMIS: a nonmonotonic inductive student modeling system.* Journal of trtificial Intelligence in Education, 1994. **5**(3): p. 371-413.

40. Brusilovsky, P., *The construction and application of student models in intelligent tutoring systems.* Computer and System Sciences International, 1994. **32**(1): p. 70-89.

41. Ohlsson, S., *Some principles of intelligent tutoring.* Instructional Science, 1986. **14**(3-4): p. 293-326.

42. Carr, B., and Goldstein, Ira P. *Overlays: a Theory of Modelling for Computer Aided Instruction*. in *5th International Joint Conference on AI*. 1977.

43. Stevens, A., Collins, A. and Goldin, S.E., *Misconceptions in students' understanding*, in *Intelligent Tutoring Systems*, D.H.a.B. Sleeman, J.S.(eds.), Editor. 1982, Academic Press: London. p. 13-49.

44. VanLehn, K., *Mind Bugs: The origins of procedural misconceptions*. 1990, Cambridge, MA: MIT Press.

45. VanLehn, K., *Student models*, in *Foundations of Intelligent Tutoring Systems*, M.C.a.R. Polson, J. J., Editor. 1988, Lawrence Erlbaum Associates, Inc: Hillsdale, New Jersey.

46. Self, J., *Student Models: What Use are they?*, in *Artificial Intelligence tools in education*, P.a.L. Ercoli, R., Editor. 1987: Amsterdam, North-Holland.

47. Hospers, M., Kroezen, E., Nijholt, A., op den Akker, R., and Heylen., D., *An agent-based intelligent tutoring system for nurse education*, in *Whitestein Series in Software Agent Technologies*. 2003, Birkhauser Publishing Ltd: Basel, Switzerland. p. 143-159.

48. Capuano, N., Marsella, M., and Salerno, S. *ABITS: an agent-based intelligent tutoring system for distance learning*. in *Proceedings of the International Workshop on adaptive and Intelligent Web-based Educational Systems, International Conference on Intelligent Tutoring Systems (ITS 2000)*. 2000. Montreal, Canada.

49. Jafari, A. *Conceptualizing Intelligent Agents for Teaching and Learning*. in *Proceedings of the International Conference on Intelligent Agents*. 2001. Las Vegas, USA.

50. Capuano, N., De Santo, M., Marsella, M.; Molinara, M. and Salerno, S. *A Multi-Agent Architecture for Intelligent Tutoring*. in *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*. 2000.

51. Lester, J.C., Towns, S. G., Fitzgerald, P. J., *Achieving Affective Impact: Visual Emotive Communication in Lifelike Pedagogical Agents.* International Journal of Artificial Intelligence in Education, 1999. **10**: p. 278-291.

52. Chou, C., Chan, T., and Lin, J., *Redefining the learning companion: the past, present, and future of educational agents.* Computers and Education, 2003. **40**(3): p. 225-269.

53. Rickel, J.a.J., W. L., *Animated agents for procedural training in virtual reality: Perception, cognition, and motor control.* Applied Articial Intelligence, 1999: p. 343-382.

54. Rickel, J.a.W.L.J. *Intelligent tutoring in virtual reality: A preliminary report*. in *Proceedings of the Eighth World Conference on Articial Intelligence in Education*. 1997: IOS Press.

55. Laird, J.E., and Newell, A. *A universal weak method: Summary of results*. in *International Joint Conference on Artificial Intelligence*. 1983.

56. Rickel, J.a.J., W. L., *Extending Virtual Humans to Support Team Training in Virtual Reality*, in *Exploring Artificial Intelligence in the New Millennium*, G.L.a.B. Nebel, Editor. 2002, Morgan Kaufmann Publishers: San Francisco. p. 217-238.

57. Shea, C., Shebilske, W.L. and Worchel, S., *Motor Learning and Motor Control*. 1993, Englewood Cliffs, New Jersey: Prentice Hall.

58. Kalawsky, R.S., *The Science of Virtual Reality and Virtual Environments*. 1993, Wokingham: Addison-Wesley,.

59. Brannick, M.T., *Team Performance Assessment and Measurement : Theory, Methods, and Applications*. Series in Applied Psychology. 1997: Mahwah, N.J. Lawrence Erlbaum Associates, Inc.

60. Cohen, P.R., and Levesque, H. J., *Teamwork*, in *Handbook of Multi-agent Systems*. 1991. p. 487-512.

61. Georgeff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge, M. *The Belief-Desire-Intention Model of Agency*. in *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*. 1999: Springer-Verlag.

62. Wooldridge, M., and Jennings, N. R. *Towards a theory of cooperative problem solving*. in *Proceeding of Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-94)*. 1994. Odense, Denmark.

63. Grosz, B.J.a.K., S., *Collaborative plans for complex group action.* Artificial Intelligence, 1996. **86**(2): p. 269-357.

64. Tambe, M. *Teamwork in Real-world, Dynamic Environments*. in *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS)*. 1996.

65. Yen, J., Yin, J., Ioerger, T.R., Miller, M., Xu, D., and Volz, R.A. *CAST: Collaborative Agents for Simulating Teamwork*. in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*. 2001.

66. Yen, J., Volz, R.A., Ioerger, T.R., Miller, M., Fan, X., Xu, D.,Sun S., and Rozich, R., *CAST: Empowering Agents with Shared Mental Models for Proactive Teamwork Behavior*. Journal of Artificial Intelligence Research, 2003.

67. Tambe, M., *Towards Flexible Teamwork.* Journal of Artificial Intelligence Research, 1997. **7**: p. 83-124.

68. Rao, A.S.a.G., M. P. *Modeling rational agents within a BDI-architecture*. in *Proceedings of Knowledge Representation and Reasoning*. 1991. San Mateo, CA: Morgan Kaufmann Publishers.

69. Rich, C.a.S., C. *COLLAGEN: When agents collaborate with people*. in *Proceedings of the International Conference on Autonomous Agents (Agents'97).* 1997.

70. Tambe, M., Pynadath, D.V., and Chauvat, N., *Adaptive Agent Architectures for Heterogeneous Team Members.* In Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000), 2000: p. 301-308.

71. Sycara, K., Paolucci, M., Velsen, V., and Giampapa, J. *The RETSINA MAS Infrastructure*. in *Autonomous Agents and Multi-Agent Systems*. 2003.

72. Resta, P. *Project CIRCLE: Student Mentors as a Strategy for Training and Supporting Teachers in the Use of Computer-Based Tools for Collaborative Learning*. in *Proceedings of the Computer Supported Collaborative Learning*. 1995. Indiana University, Bloomington, IN.

73. Koschmann, T., *CSCL: Theory and practice of an emerging paradigm*, in *Paradigm shifts and instructional technology: An introduction*, T. Koschmann, Editor. 1996, Lawrence Erlbaum Associates: Mahwah, NJ. p. 1-23.

74. Salas, E.S., R.W., *Teams: Their Training and Performance*. 1992, Norwood, NJ: Ablex Publishing Company.

75. Morgan, B.J., Glickman, A., Woodard, E., Blaiwes, A., and Salas, E. *Measurement of team behaviors in a navy environment*. 1986. Orlando, FL: Naval Training Systems Center.

76. Oser, R.L., Gualtieri, J. W., Cannon-Bowers, J. A. and Salas, E., *Training team problem-solving skills: An event-based approach.* Computers in Human Behavior, 1999. **15**: p. 441-462.

77. Fowlkes, J.E., Lane, N. E., Salas, E., Oser, R. L., & Prince, C. *Targets for aircrew coordination training*. in *Proceedings of the 14th Interservice Industry Training Systems Conference*. 1992. Washington, DC: National Security Industrial Association.

78. Oser, R.L., Cannon-Bowers, J. A., Dwyer, D. J., and Salas, E. *Establishing a learning environment for JSIMS: Challenges and considerations*. in *Proceedings of the 19th Annual Interservice/Industry Training, Simulation and Education Conference*. 1997. Orlando, FL.

79. Oser, R.L., Dwyer, D. J., Cannon-Bowers, J. A., & Salas, E. *Enhancing multi-crew information warfare performance: An event-based approach for training*. in *RTO HFM Symposium on Collaborative Crew Performance in Complex Operational Systems*. 1998. Edinburgh, United Kingdom.

80. Pearce, P.a.A., P., *A taxonomy of helping: A multidimensional scaling analysis.* Social Psychology Quarterly, 1980. **43**(4): p. 363-371.

81. Fan, X., Yen, J., Miller, M. S., Ioerger, T. R. and Volz, R. A., *MALLET - A Multi-Agent Logic Language for Encoding Teamwork.* IEEE Transaction on Knowledge and Data Engineering, 2006. **18**(1): p. 123-138.

82. Yen, J., Fan, X., Sun, S., Wang, R., Chen, C., Kamali, K., Miller, M., and Volz, R. A., *On Modeling and Simulating Agent Teamwork in CAST.* in Proceedings of the Second International Conference on Active Media Technology, 2003a: p. 18-33.

83. Yen, J., Fan, X., Sun, S., Hanratty, T., and Dumer, J., *Agents with shared mental models for better tactical decision-makings.* Journal of Decision Support System, 2004.

84. Miller, M.S., and Volz, R. A. *Team-oriented Agents for Training Command and Control Teams*. in *International Conference on Computing, Communications and Control Technologies*. 2004. Austin, Texas.

85. Miller, M.S., Yin, J., Ioerger, T. R., Yen, J., and Volz, R. A. *Training Teams with Collaborative Agents*. in *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems*. 2000. Montreal, Canada.

86. Lave, J., and Wenger, E., *Situated Learning: Legitimate Peripheral Participation*. 1990, Palo Alto, CA: Institution for Research in Learning.

87. Miller, D., Young, P., Kleinman, D., and Serfaty, D., *Distributed Dynamic Decision-Making (DDD) Simulation.* 1998.

88. Xu, D., Volz, R. A., Miller, M. S., and Plymale, J. *Human-Agent Teamwork for Distributed Team Training*. in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*. 2003.

89. Xu, D., Miller, M., Volz, R.A., and Ioerger, T.R., *Collaborative Agents for C2 Teamwork Simulation.* International Conference on Artificial Intelligence (ICAI03), 2003: p. 723-729.

Appendix A

Training Material for Human Subject Experiment

Score Information					
Score Components	Individual/ Group/Team				
Defense Scores (Starts at 0)	**-1 pt/sec** when enemy task in the green no-fly zone	**-2 pt/sec** when enemy task in the red/critical no-fly zone	**-25 pt** when attacked friendly tasks	**-25 pt** when attacked outside the green no-fly zone	**-25 pt** when one of your assets run out of fuel
Offensive Scores (Starts at 0)			**+50 pt** for a successful attack		

Table 20 Team Defense and Offense Scores

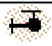
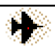
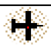
Assets Information					
Types of Assets	Icon	Power	Speed	Function	Amount
TK (Tank)		**5**	Very slow	Attack *Only*	1
HE (Helicopter)		**3**	medium	Attack & ID	1
JT (Jet plane)		**1**	fast	Attack & ID	1
AW (AWCS radar plane)		**0**	Very fast	ID *Only*	1

Table 21 Resources for DDD Domain Task (Assets Amount and Functionality)

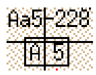
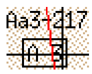
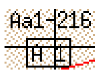
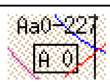
Task Information						
Types of tasks	Icon		Power	Speed	Function	Amount
	IDed	Not IDed				
A5			**5**	Direct ratio to the length of the task's tail	Enemy	unknown
A3			**3**	Direct ratio to the length of the task's tail	Enemy	unknown
A1			**1**	Direct ratio to the length of the task's tail	Enemy	unknown
A0			**0**	Direct ratio to the length of the task's tail	Friend	unknown

Table 22 Tasks in DDD domain

Asset Operation	**Step 1**	**Step 2**	**Step 3**	**Step 4**
Launch	Move your mouse on top of your base and **Middle click**.	An asset window pops up and you will see **a list of abbreviations** for different type of assets that you can launch	To select an asset: **Left click** the **“>”** button to the right of the asset that you wantto launch To de-select an asset **Left click** the **“<”** button to the right of the asset will de-select it	**Left click “OK”** on the bottom left of the window
Move	**Put your curser** on top of your asset and **right click**	A Menu appears, and **select the Move(fast) option**	Your cursor now appears as a small black “+”, **Move your cursor** close to the task	**Left click** on the spot you want to go and the asset starts to move
Return	**Put your cursor** on top of your asset and **right click**	A Menu appears, and **select the Return**	A pop-up window will appear	**Left click “OK”** on the bottom left of the window

Table 23 Basic Assets Operations

<table>
<tr><th>Task Operations</th><th>Step 1</th><th>Step2</th><th>Step3</th><th>Step4</th></tr>
<tr><td rowspan="2">Identify</td><td>Put your cursor directly <u>on top of the task</u> and right click</td><td>A Menu appears, and select the <u>Identify</u></td><td>Left Click <u>Fused</u> (square button); this will fuse the task power information on the screen</td><td>Left click “OK”</td></tr>
<tr><td colspan="4">Make sure the task is inside the blue/ID ring</td></tr>
<tr><td colspan="5">After the task has been IDed and the power information has been fused onto the screen, now we will learn how to transfer ID to transfer the task’s power information to your teammates</td></tr>
<tr><td rowspan="2">Transfer ID</td><td>Put your cursor directly <u>on top of the task</u> and right click</td><td>A Menu appears, and <u>select the Transfer Infor</u></td><td>A popup window appears, <u>select All Linked DMs</u></td><td>Left click “OK”</td></tr>
<tr><td colspan="4">Make sure the blue/ID ring still surrounds the task; wait a few seconds for the information to be transferred</td></tr>
<tr><td colspan="5">Only after a task has been IDed and/or the information has been transferred, you can attack the tasks</td></tr>
<tr><td rowspan="2">Attack</td><td>Put your cursor directly <u>on top of the asset</u> and right click</td><td>A Menu appears, and select the <u>Attack</u></td><td>Your cursor now appears as a big black “X”, Move your cursor <u>close to the task</u></td><td>Left click the task and wait a little while for the result</td></tr>
<tr><td colspan="4">Make sure the task is inside the red/Attack ring If you used the adequate powered asset, you will see the task exploding on the screen before it disappears, and your offense score will increase 50 for each hostile task you destroyed ; if not, you will see an error icon and your offense score won’t increase</td></tr>
</table>

Table 24 Basic Task Operations

Appendix B

Debriefing Check Lists

DEBRIEFING CHECKLIST –Experiment Group

1. Was your team's mission
   a. Very Successful(Team Offensive **2800** Team Defensive **-2300**)
   b. Successful (Team Offensive **2500** Team Defensive **-3500**)
   c. Average (Team Offensive **1900** Team Defensive **-5000**)
   d. Unsuccessful(Team Offensive **1200** Team Defensive **-7000**)
   e. Very unsuccessful(Team Offensive **less than 1000** Team Defensive **–less than -8000(for example -8500 is less than -8000)**)

2. **Discuss following strategies that could lead to efficient attacks in the green or red restricted zones**
   a. Launch assets early-*** keep in mind the fuel time for each asset
   b. Attack enemy promptly after they enter restricted zones
   c. Move asset to planned quadrant with good timing
   d. ID incoming tasks and transfer ID to teammate if not able to attack it right away
   e. Intercept a task at the shortest distance

3. Enemies in the restricted zones for a long time: **the defense score gives the number of points lost as a result of enemies staying in the green zone or the red zone.**
   a. Discuss why the enemies in your most recent mission remained in the restricted zones as long as they did. Select more than one if needed
      i. They were not identified
      ii. They were identified but not destroyed
      iii. They were too far from our assets and/or bases
      iv. Nobody had time for them
      v. Our attack asset was of the wrong power
      vi. Our attack asset was too slow
      vii. Other reasons
   b. What changes might prevent these long invasions? **Discuss placing your AWACS to ID quickly, allocating sufficient assets to cover all tasks in all four quadrant. Discuss planning efficient execution schedules for**

launching each DM's assets in the best order for a given intelligent report

i. Identifying tasks before they enter the restricted zones
ii. Making plans that take into account the load in each DM's zone
iii. Move the shortest distance to get the tasks within the attack ring
iv. Attacking with right power

4. How many times did you destroy a friendly or an enemy outside the restricted zone? **The penalty for this type of attack is a decrease of 25 defensive points in each case**
   a. Why did these mistakes occur
      i. Tasks not identified before attacking
      ii. Attacked the wrong task
      iii. Did not notice the border of the restricted zone
      iv. Desperate to attack some task
      v. Don't care just want to attack
   b. What changes might prevent these mistakes? **Discuss using the zoom option to clearly view and identify the task before attacking and do not attack tasks outside the green zone.**

The next section is evaluation of your plan and execution by the computer coach.

1. **Click "Planning Review" Tab to see agent coach's feedback to your planning session.**
2. **Click "Performance Review" Tab to see agent coach's feedback to your online performance.**
3. **Click "Interval#" (# from 1 to 4) Tab underneath to view feedback/results for each time interval**
4. **Select either the "ID" or "Attack" evaluation located inside the "Planning Review" or "Performance Review" page.**

DEBRIEFING CHECKLIST— Control Group

1. Was your team's mission
   a. Very Successful(Team Offensive **2800** Team Defensive **-2300**)
   b. Successful (Team Offensive **2500** Team Defensive **-3500**)
   c. Average (Team Offensive **1900** Team Defensive **-5000**)
   d. Unsuccessful(Team Offensive **1200** Team Defensive **-7000**)

e. Very unsuccessful(Team Offensive **less than 1000** Team Defensive **– less than -8000(for example -8500 is less than -8000)**)

2. **Discuss following strategies that could lead to efficient attacks in the green or red restricted zones**
    a. Launch assets early-*** keep in mind the fuel time for each asset
    b. Attack enemy promptly after they enter restricted zones
    c. Move asset to planned quadrant with good timing
    d. ID incoming tasks and transfer ID to teammate if not able to attack it right away
    e. Intercept a task at the shortest distance

3. Enemies in the restricted zones for a long time: **the defense score gives the number of points lost as a result of enemies staying in the green zone or the red zone.**
    a. Discuss why the enemies in your most recent mission remained in the restricted zones as long as they did. Select more than one if needed
        i. They were not identified
        ii. They were identified but not destroyed
        iii. They were too far from our assets and/or bases
        iv. Nobody had time for them
        v. Our attack asset was of the wrong power
        vi. Our attack asset was too slow
        vii. Other reasons
    b. What changes might prevent these long invasions? **Discuss placing your AWACS to ID quickly, allocating sufficient assets to cover all tasks in all four quadrant. Discuss planning efficient execution schedules for launching each DM's assets in the best order for a given intelligent report**
        i. Identifying tasks before they enter the restricted zones
        ii. Making plans that take into account the load in each DM's zone
        iii. Move the shortest distance to get the tasks within the attack ring
        iv. Attacking with right power

4. How many times did you destroy a friendly or an enemy outside the restricted zone? **The penalty for this type of attack is lose 25 defensive points in each case**
    a. Why did these mistakes occur
        i. Tasks not identified before attacking
        ii. Attacked the wrong task
        iii. Did not notice the border of the restricted zone
        iv. Desperate to attack some task
        v. Don't care just want to attack

b. What changes might prevent these mistakes? **Discuss using the zoom option to clearly view and identify the task before attacking and do not attack a task outside the green zone.**

5. Can you improve your next plan and mission with lessons learned from this mission about the following?
   a. Allocating your assets better **(Discuss about potential placement for each asset during the planning phase of the mission, both in your own quadrant and other's quadrant)**
      i. TK
      ii. HE
      iii. JT
      iv. AW
   b. Rearming and refueling (**While planning make sure you take into account refueling and rearming time while building the execution schedule of the assets for the mission)**
      i. TK
      ii. HE
      iii. JT
      iv. AW
   c. Requesting and offering help (**DMs can help one another for identifying or attacking tasks or both)**
      i. During planning (**During planning discuss your role in another quadrant and how you would coordinate it with your responsibilities in own quadrant)**
      ii. During Mission
         1. **If you're helping to ID tasks, remember to transfer the information to all DMS.**
         2. **If you're helping to attack tasks in other quadrant, if the task has been IDed by other DMs, attack the task with right-powered asset; Otherwise Id the task first, transfer info to all other DMs, and attack if the task's power is equal or lower than that of your asset**
   d. Coordinating your plan and mission
      i. Following your resource allocation plan (**Remember to plan and follow an execution schedule of launching your assets in the best order for the mission**)
      ii. Adapting to unexpected events**(Discuss backup plans that may be needed if any DM fails to execute his/her plan and incorporate any positive adaptations made during the previous mission)**

Appendix C

Post Experiment Questionnaire

Post-Experiment Questionnaire –Experiment Group

1. What is the experience that you have with video games? Please make estimation about how many hours per week that you play video games.

   A. Very good (More than 3 hours)

   B. Good (More than 1 hour and less than 3 hours)

   C. Average (Less than 1 hour)

   D. Not good (0 hr– if you don't have any experiences playing video games)

   E. Not sure

2. Are you a good team-player? What will you evaluate yourself in terms of collaborating/working with others?

   A. Very good

   B. Good

   C. Average

   D. Not good

   E. Not sure

3. Do you think *Planning Coach* is helpful for you to perform better during the mission?

   A. Very helpful (I made much better resource allocation plans)

   B. Helpful (It helped me to improve next mission planning)

C. Not helpful (It doesn’t help me to improve next mission planning)

D. Not sure (I don’t understand how it works)

4. Do you think the *Execution Coach* is helpful for you to perform better during the mission?

   A. Very helpful (It helped me to review different types of mistakes that I made during the previous mission and provided me useful tips about how to assist my teammates. My next mission significantly improved by making appropriate adjustments)

   B. Helpful (It helped me to review different types of mistakes that I made during the previous mission and provided me useful tips about how to assist my teammates.)

   C. Not helpful (I would like to review what I did in the previous mission, but I don’t think it helped me to improve)

   D. Not sure (I don’t understand how the *Execution Coach* works)

Post-Experiment Questionnaire –Control Group

1. What is the experience that you have with video games? Please make estimation about how many hours per week that you play video games.

   A. Very good (More than 3 hours)

   B. Good (More than 1 hour and less than 3 hours)

   C. Average (Less than 1 hour)

   D. Not good (0 hr– if you don't have any experiences playing video games)

   E. Not sure

2. Are you a good team-player? What will you evaluate yourself in terms of collaborating/working with others?

   A. Very good

   B. Good

   C. Average

   D. Not good

   E. Not sure

3. Do you think *Debrief Session* about your *Mission-Planning* is helpful for you to perform better during the mission?

   A. Very helpful (I made much better resource allocation plans)

   B. Helpful (It helped me to improve next mission planning)

   C. Not helpful (It doesn't help me to improve next mission planning)

   D. Not sure (I don't understand how it works)

4. Do you think the *Debrief Session* about your *Mission-Execution* is helpful for you to perform better during the mission?

   A. Very helpful (It helped me to review different types of mistakes that I made during the previous mission and provided me useful tips about how to assist my teammates. My next mission significantly improved by making appropriate adjustments)

   B. Helpful (It helped me to review different types of mistakes that I made during the previous mission and provided me useful tips about how to assist my teammates.)

   C. Not helpful (I would like to review what I did in the previous mission, but I don't think it helped me to improve)

   D. Not sure (I don't understand how the *Execution Coach* works)

Appendix D

Two-way ANOVA Test Results – Confident Intervals

Two-way ANOVA: Individual Offense Score versus Treat, Trial

```
                Individual 95% CI
Treat           Mean   -------+---------+---------+---------+----
C               420  (------*------)
E               491                    (------*------)
                     -------+---------+---------+---------+----
                          420       450       480       510

                     Individual 95% CI
Trial           Mean   ------+---------+---------+---------+-----
0               304  (---*---)
1               464                 (---*---)
2               499                     (--*---)
3               555                           (--*---)
                     ------+---------+---------+---------+-----
                         320       400       480       560
```

C=Control Group;   E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 25 Two-way ANOVA Test for Individual Offense Score (Confident Intervals)

Two-way ANOVA: Individual Defense Score versus Treat, Trial

```
                         Individual 95% CI
Treat          Mean   -+---------+---------+---------+---------+
C             -1530    (--------*--------)
E             -1360                     (--------*--------)
                     -+---------+---------+---------+---------+
                  -1600     -1520     -1440     -1360     -1280


                        Individual 95% CI
Trial          Mean   -+---------+---------+---------+---------+
0             -1922   (---*---)
1             -1407                    (---*---)
2             -1276                        (---*---)
3             -1174                           (---*---)
                     -+---------+---------+---------+---------+
                  -2000     -1750     -1500     -1250     -1000
```

C=Control Group;   E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 26 Two-way ANOVA Test for Individual Defense Score (Confident Intervals)

Two-way ANOVA: Individual Offense Score Improvement versus Treat, Trial

```
                        Individual 95% CI
Treat          Mean  ----------+---------+---------+---------+-
C              153   (------*-------)
E              251                       (-------*------)
                     ----------+---------+---------+---------+-
                            160       200       240       280

                        Individual 95% CI
Trial          Mean  ----------+---------+---------+---------+-
1              160   (--------*--------)
2              195          (--------*--------)
3              251                      (--------*--------)
                     ----------+---------+---------+---------+-
                            160       200       240       280
```

C=Control Group;   E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 27 Two-way ANOVA Test for Individual Offense Score Improvement

(Confident Intervals)

Two-way ANOVA: Individual Defense Score Improvement versus Treat, Trial

```
                     Individual 95% CI
Treat         Mean   --+---------+---------+---------+---------
C             543   (---------*--------)
E             710                     (---------*--------)
                     --+---------+---------+---------+---------
                     480       560       640       720

                     Individual 95% CI
Trial         Mean   ---------+---------+---------+---------+--
1             509    (--------*--------)
2             645               (---------*--------)
3             725                      (--------*---------)
                     ---------+---------+---------+---------+--
                           500       600       700       800
```

C=Control Group;   E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 28 Two-way ANOVA Test for Individual Defense Score Improvement

(Confident Intervals)

Two-way ANOVA: Assist Attack versus Treat, Trial

```
                   Individual 95% CI

Treat         Mean  ------+---------+---------+---------+-----
C             2.19  (--------*--------)
E             2.91                      (--------*--------)
                    ------+---------+---------+---------+-----
                       2.10      2.40      2.70      3.00


                    Individual 95% CI
Trial         Mean  ------+---------+---------+---------+-----
0             1.40   (----*----)
1             2.95                     (----*----)
2             2.63                 (----*-----)
3             3.22                        (----*----)
                    ------+---------+---------+---------+-----
                       1.40      2.10      2.80      3.50
```

C=Control Group; E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 29 Two-way ANOVA Test for Assist Attack (Confident Intervals)

Two-way ANOVA: Successful Assist Attack versus Treat, Trial

```
                    Individual 95% CI
Treat        Mean   --+---------+---------+---------+---------
C            1.69   (--------*-------)
E            2.24                      (-------*--------)
                    --+---------+---------+---------+---------
                    1.50      1.75      2.00      2.25
```

```
                    Individual 95% CI
Trial        Mean   ----------+---------+---------+---------+-
0            1.37   (-------*--------)
1            2.13                     (--------*-------)
2            2.15                      (-------*--------)
3            2.20                       (--------*--------)
                    ----------+---------+---------+---------+-
                           1.40      1.75      2.10      2.45
```

C=Control Group;   E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 30 Two-way ANOVA Test for Successful Assist Attack (Confident Intervals)

Two-way ANOVA: Assist ID versus Treat, Trial

```
                    Individual 95% CI
Treat          Mean  ----------+---------+---------+---------+-
C              5.48  (------*-------)
E              7.33                     (-------*------)
                    ----------+---------+---------+---------+-
                         5.60     6.40     7.20     8.00

                    Individual 95% CI
Trial          Mean  ---------+---------+---------+---------+--
0              4.13  (-----*----)
1              6.30              (-----*-----)
2              7.70                      (----*-----)
3              7.48                    (-----*-----)
                    ---------+---------+---------+---------+--
                        4.50     6.00     7.50     9.00
```

C=Control Group; E=Experiment Group
Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 31 Two-way ANOVA Test for Assist ID (Confident Intervals)

Two-way ANOVA: Transfer ID versus Treat, Trial

```
                  Individual 95% CI
Treat         Mean   -------+---------+---------+---------+----
C             1.45   (---*---)
E             3.81                            (----*---)
                    -------+---------+---------+---------+----
                       1.60      2.40      3.20      4.00

                   Individual 95% CI
Trial         Mean   -----+---------+---------+---------+------
0             2.02   (-------*-------)
1             2.15     (-------*-------)
2             3.02                 (-------*--------)
3             3.32                     (-------*--------)
                    -----+---------+---------+---------+------
                     1.80      2.40      3.00      3.60
```

C=Control Group; E=Experiment Group

Trial 0=Baseline; Trial 1=Training 1; Trial 2=Training 2; Trial 3=Training 3

Figure 32 Two-way ANOVA Test for Communicated ID (Confident Intervals)

CONG CHEN

cchen@ist.psu.edu

324 East, IST Building
School of Information Sciences and Technology
University Park, PA 16802
(814)865-6178

Home Address
425 Waupelani Dr Apt 126
State College, PA 16801
(814)883-7864
http://www.personal.psu.edu/cuc124

EDUCATION

Ph.D. Information Sciences and Technology December 2006 **GPA 3.77/4.0**
The Pennsylvania State University, University Park, PA
Dissertation: "Intelligent Training Agents for Enhancing Helping Behavior in Human Teamwork"

B.S. Computer Science and Technology Sep 1997 - Jun 2001 **GPA 3.58/4.0**
Peking University, Beijing, China
Thesis: "A Mobile Agent-based Infrastructure for Information Retrieval on the Internet"

TECHNICAL SKILLS

Programming languages: Java, C/C++, SQL, MATLAB, XML, VB, Prolog, Pascal
Platforms: Windows, LINUX (Fedora, Redhat, Debian), UNIX, DOS
Software packages: MS Office, MS Project, MINITAB

RELEVANT EXPERIENCE

Research Assistant, Laboratory for Intelligent Agents, Penn State University, PA, Aug 2001- present

Intelligent Agents for Team Training

- Designed an intelligent training framework to monitor sequences of trainee actions and to analyze data for performance assessment in a distributed simulation
- Incorporated event-based training approach and user modeling techniques into CAST (Collaborative Agents for Simulating Teamwork) and DDD (Distributed Dynamic Decision-making) simulation environment where intelligent agents serve the roles of interactive partners and offline coaches
- Implemented intelligent coaching agents that assess trainees' collaborative planning of resource allocation and team members' online helping behaviors and provide the corresponding feedbacks to help improve individual performance within a team
- Conducted human subject experiment to validate the intelligent training system

Teaching Assistant, Penn State University, PA, Aug 2005 - Dec 2005

IST 210 Assistant Instructor

- Taught undergraduate students set theory, Boolean algebra, basic programming concepts and HTML
- Designed course material, gave lectures, developed and maintained course website
- Supervised lab exercise, graded quizzes/assignments/exam and advised students during office hours

The 4th Annual Summer School on "Building Intelligent Tutoring Systems"
Carnegie Mellon University, Pittsburgh, PA, Jun 2004 - Jul 2004Designed and implemented a prototype cognitive tutor based on ACT-R theory

Additional Research Projects

- Designed and implemented agents with different trading strategies in a double auction market
- Designed an adaptive mind/machine interface for driving performance