

Training Teams with Collaborative Agents

Michael S. Miller, Jianwen Yin, Richard A. Volz, Thomas R. Ioerger, John Yen

MS 3112
Department of Computer Science
Texas A&M University
College Station, Texas 77843-3112
{mmiller, jianweny, volz, ioerger, yen}@cs.tamu.edu

Abstract. Training teams is an activity that is expensive, time-consuming, hazardous in some cases, and can be limited by availability of equipment and personnel. In team training, the focus is on optimizing interactions, such as efficiency of communication, conflict resolution and prioritization, group situation awareness, resource distribution and load balancing, etc. This paper presents an agent-based approach to designing intelligent team training systems. We envision a computer-based training system in which teams are trained by putting them through scenarios, which allow them to practice their team skills. There are two important roles that intelligent agents can play; these are as virtual team members and as coach. To carry out these functions, these agents must be equipped with an understanding of the task domain, the team structure, the selected decision-making process and their belief about other team members' mental states.

1 Introduction

An integral element of large complex systems is that a team of humans is needed to manage them. Teams demand that the members be competent not only in their individual skills, but also in anticipating the needs of the team as if it were an entity by cooperating with other team members to act effectively. Teams can induce a large amount of stress on members that can lead to tragic consequences such as the shooting down of an Iranian airliner by the USS Vincennes. Stressors such as sensor overload, fatigue, time pressure, and ambiguity contributed to this accident [1]. In order to better manage these factors teams train together to be able to perform together effectively.

In heterogeneous teams, that is teams that require specialists in order to function, team members must not only be able to perform their own unique functions, but they must also be able to act as a cohesive part of the team. A team member may or may not be familiar with the functions of other team members but is competent within his own domain. In order for the team to become competent, the team members must practice together [2]. Team training is normally done using simulations of the system or the actual system with all of the human team members participating. When an intelligent tutoring system is added, it is typically in the role of training an individual to be able to understand his individual tasks before taking part in the team. In order to

train teams it would be useful to expand an ITS so that it can support team activities. In our approach, partial teams can be simulated using computed-based agents to represent team members and thus teach a trainee necessary team skills such as situational awareness, group decision-making, and communications efficiency without having to involve the entire human team for all training sessions.

By building computer-based simulation environments, trainees can be run through simulated scenarios, providing a type of hands-on experience. Intelligent agents serving as virtual team members can provide significant cost-savings through partial team training. However, significant challenges exist in developing such *intelligent team training systems* (ITTS). First, for agents to participate in the simulation (and provide believable interactions) as virtual team members, they must have an understanding of the team structure and the collaboration process, requiring multi-agent belief reasoning. Second, in order to diagnose problems with teams and provide dynamic feedback (e.g. coaching), things such as distributed plan recognition and interpreting individual's action in terms of their beliefs about their teammates must be done.

We envision a computer-based training system in which teams are trained by putting them through scenarios, which allow them to practice their team skills. Our proposed approach to training teams is to use an intelligent multi-agent based system that has a knowledge-based foundation. This ITTS allows the human trainee to build an understanding of his role within the team. The trainee is able to learn which other team members the trainee must monitor and when or where the trainee can provide support to the other team members without interrupting them in the performance of their duties. This also called a *shared mental model*, which is thought to be a key to effective teamwork [3]. Mistakes that the trainee makes can be caught by a coaching agent that can either use other virtual team agents to correct the trainee or directly interact with the trainee as a tutor within the system.

2 Teamwork

Our definition of a team is a group of entities (humans or agents) that are working together to achieve a goal that could not be accomplished as effectively (or at all) by any one of them alone [4]. Team members play unique roles, which may require unique skills and resources. Our focus is on teams that are hierarchical, with a clear chain-of-command and leadership or authority roles. Teams are also heterogeneous in that individual team members have different roles and responsibilities within the team. All teams have to deal in one way or another with sharing information and distributed decision-making (also called cooperation or collaboration) [5].

3 Team Training

In team training, the focus is not on each individual's skills (which are typically learned beforehand), but on optimizing interactions, such as situational awareness, communications efficiency, and the effectiveness of team decision-making [6].

Intelligent agents can help extend these methods to build Intelligent Team-Training Systems. There are two important functions that intelligent agents can play in such systems. First, we can have agents that can substitute for other team members. This allows for partial team training, which could provide huge cost savings, and allows for either individuals or sub-teams to train without the need for the rest of the team. A second major role is for a knowledge-based agent to play the role of coach [7]. This eases the burden of a human instructor from having to monitor both the trainee and the other virtual team members in the simulation. To carry out their roles, these agents must be equipped with an understanding of the task domain, the team structure, the selected decision-making processes and their belief about other team members' mental states.

4 Other Agent Based Teams

The agent-based teams that exist in the literature are focused on allowing rational agents to work together on a common goal. Such agents have a shared mental model of what each agent is able to contribute to the team. This shared mental model is a simplified model of the mental states of all the other members on the team. Agents must be able to query and establish team goals that the agents collaborate upon in order to achieve a shared goal that they would otherwise be unable to achieve.

A teamwork model must provide support for reasoning about team goals, plans, and states. It must also represent the roles and responsibilities of individual team members as this relates to other team members. Information needs of team members need to be fulfilled by the team by finding out who best can answer those needs. In the approaches listed below such information needs are not yet examined.

In the SharedPlans approach each agent maintains individual plans and shared plans [5]. Individual agents accomplish plans that require cooperation between such agents by building shared plans. Other team-based agents build on this foundation to construct general models of teamwork. COLLAGEN [8] uses a plan recognition algorithm in order to reduce communications during collaboration between a human and an agent. Using attention, partial plans, and clarification enables COLLAGEN-based agents to interact with humans in an intelligent fashion. COLLAGEN is an implementation of the SharedPlans theory.

The approach that STEAM uses is to find *joint intentions* between agents and create a hierarchy of joint intentions so that the agents can monitor other agents' accomplishments or failures in achieving these shared intentions [9]. Both systems provide a model of teamwork into which domain specific knowledge and agents can be added. STEAM is designed to be domain independent in its reasoning about teamwork. It is based on the *joint intentions* framework by Levesque [10]. STEAM also provides for capabilities to monitor and repair team plans.

PuppetMaster addresses the issue of reporting on student interactions within a team for use by an instructor [11]. A top-down approach is used to reduce unnecessary details and be able to recognize actions at the team level. The focus of PuppetMaster is not on the individual student's behavior but as an aid to an instructor to recognize team failures.

5 CAST – Collaborative Agents for Simulating Teamwork

We describe a computational system for implementing an ITTS called CAST, for *Collaborative Agent architecture for Simulating Teamwork*. We focus on humans as a part of the virtual team. We wish to model the individual's beliefs and actions within the context of the team. We also wish to automate the training process and allow individuals to practice alone without needing a large support staff to setup and monitor the exercise. We assume that a good description can be provided of the actions that a team and its members will be able to perform. Therefore, we assume that the team has a plan of what needs to be accomplished in the performance of the team mission, and we know who are the team members and what their roles will be.

We want to enable an individual new to the team to become a part of the team by increasing his situational awareness, showing him who or what to monitor, and how best to respond to the actions and requests of his fellow team members. The team exists in a domain in which each team member plays a specific role and responses need to be well rehearsed in order to overcome any difficulties that the team may encounter. We can best illustrate what such a team looks like with the following brief example.

6 An Example Team Domain

The NASA Mission Control Center consists of a team that is arranged in a hierarchical manner with clearly delineated roles for each team member. The Flight Director (FD) oversees 10 disciplines which each monitor functions on the Space Shuttle. These stations are manned continually during a Space Shuttle mission, which typically lasts less than 10 days. During scheduled events all relevant disciplines are fully staffed. During down times, such as when the astronauts are sleeping, only lighter staffing needs are required.

To examine the operation more closely, consider the PROP (Propulsion Systems Officer). The PROP is responsible for the operation of the Space Shuttle Orbital Maneuvering System (OMS) and Reaction Control System (RCS). These secondary engines are used for orbital corrections, docking operations, and the De-orbit burn. The PROP is assisted by the OMS/RCS Engineering Officer (OREO) and the Consumables Officer (CONS) as a sub-team [12]. The PROP knows the functions and duties of his sub-team members but instead typically focuses on interacting with the other disciplines. The PROP uses his sub-team to fulfill his requirements for information and allows them to manage their respective sub-systems.

The PROP officer is also in a vertical chain of command leading up to the Flight Director. The Flight Control Room (FCR) provides each FCR team member a headset with separate channels dedicated to different disciplines and needs. The sub-team members such as the OREO and CONS officers sit on consoles in a separate room from the FCR called the Multipurpose Support Room (MPSR).

As an example scenario during the launch stage, the FD asks the PROP officer for a status check to see if the discipline is ready for launch. The PROP officer checks with his sub-team. Each sub-team member checks his own console and reports to the

PROP officer. The PROP officer reports back to the FD that they are ready for launch. This is a simple example but shows the need for monitoring the needs of the team and having a situational awareness as to what functions each individual should be performing. Individuals are aware of what the team goals are and what their responsibilities and needs are in order to fulfill the team goals.

In order for the MCC team to be able to train as a team, the resources of the MCC at Johnson Space Center must be dedicated to running a training simulation. This can involve not only the MCC team members, but also astronauts in the Space Shuttle Trainer (which is located in a different building), building and computer support personnel, and the resources of the actual FCR. When such a training task is in progress no other work can be done with the facilities. Such team training is not done when a space shuttle is in flight. This will become a problem when resources must also be used for monitoring the International Space Station.

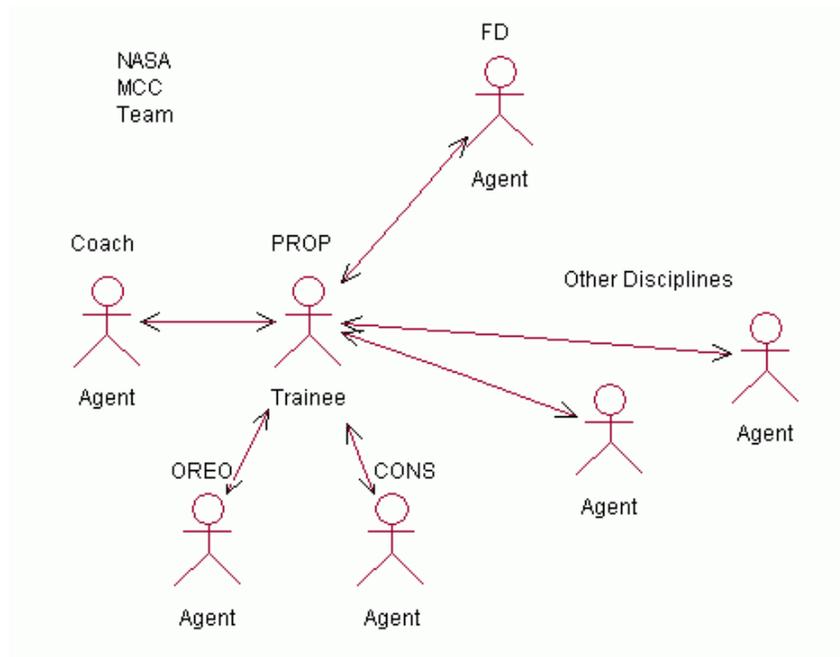


Fig. 1. A subset of the NASA MCC Team

7 The CAST Architecture

The approach we take in CAST is to model the team interactions of team members using Petri Nets. We propose to use a model of teamwork and reasoning agents with beliefs about the team and themselves in order to construct the ITTS. The virtual team agents must also be able to interact with human trainees and communicate with a

human counterpart. The explicit representation of goal hierarchies and intentions will be important for diagnosing problems with team behavior and providing useful feedback for coaching and evaluation.

7.1 Challenges in Developing an ITTS

An agent-based, team-centered ITS for training teams has certain challenges to overcome in order to be an effective training tool. First, the virtual team members have to generate reasonable interactions for human team members. Humans must be incorporated in the ITTS initially as one or more trainees, and eventually as other team members in order to allow sub-teams to practice among themselves. Second, the coaching agent should be “non-intrusive” by passively monitoring and interpreting all actions and interactions of the trainee instead of announcing itself as a coaching agent and asking the trainee about his intentions and reasons. And last, understanding the actions of an individual on a team is more complicated because their decision-making explicitly involves reasoning about the other members of the team (e.g. their beliefs, roles, etc.), and their actions may be implicitly in support of a team goal or another agent.

7.2 Components of CAST

The ITTS will have four major components. Intelligent agents are used to represent individual team members. The individual team members incorporate a model of teamwork in order to help identify points of communications and shared goals. The state of a simulated world within which the training will occur must be maintained. To be useful the simulation should also be able to interface into an existing simulation, or integrate into an actual system. This last approach is the one planned for use with the MCC NASA domain. A coaching agent also maintains a user model of the trainee and acts when appropriate to tutor the trainee on understanding his role as a team member.

7.3 Elements of a Knowledge-Based Model of Teamwork

An ITTS agent must reason not only about its goals and capabilities, but also about the goals of the team and other team members and about commitments or shared responsibilities. This requires what is known as *belief reasoning*, which we simulate in CAST.

First, we use the team-description language MALLETT [13] to provide a framework for modeling teamwork. Second, we encode this model of actions and interactions of a team into a representational structure using *Petri Nets*. Third, we use an *Inter-Agent Rule Generator* (IARG) to determine the interactions that will take place among the agents. Fourth, we incorporate a coaching agent to be able to detect when the trainee fails to act as a member of the team and provide feedback to the trainee to enable him to act appropriately.

7.4 MALLET: A Multi-Agent Logic Language for Encoding Teamwork

The ontology underlying our framework is based on the BDI model [14] (Belief represents the knowledge of the agent, Desire represents the general goals of the agent, and Intention represents the selected plans of the agent). The purpose of using an ontology is to identify the general concepts and relationships that occur in teamwork across multiple domains, and give them formal definitions that can be used as the basis of a team-description language with predicates with well-specified meanings. MALLET is a language based on predicate logic that allows the encoding of teamwork. Being a logic-based language, MALLET provides a number of pre-defined terms that can be used to express how a team is supposed to work in each domain such as Role (x), Responsibility (x), Capability (x), Stage (x), etc.

7.5 Petri Net Representation of MALLET

The actions and interactions of a team can be encoded in *Petri Nets*, which are a natural representation for actions, synchronization, parallelism, etc. Petri Nets have previously been suggested as an appropriate implementation for both intelligent agents and teamwork [15]. Petri Nets are particularly good at representing actions in a symbolic/discrete framework. They can represent the dependence of actions on pre-conditions in a very natural way, i.e. via input places to a transition. The effects of the chosen action simply become output places in the Petri Net. We use an algorithm to transform descriptions of roles in MALLET into Petri Nets, including beliefs, operators, and goals, etc. We use a Petri Net for each role on the team, with beliefs specific to that agent.

7.6 IARG algorithm

The IARG (Inter-Agent Rule Generator) algorithm is used to detect information flow and generate team interactions. IARG uses both offline and online components. An agent analyzes the Petri Nets of all the other agents using the IARG algorithm in order to derive information flow and identify propositions that other agents need to know. We can define information flow as a 3-tuple: $\langle \textit{Proposition}, \textit{Providers}, \textit{Needers} \rangle$. *Proposition* is a truth-valued piece of information. *Providers* is the set of roles that can provide the information (i.e. perhaps has the responsibility of achieving and/or maintaining it). *Needers* is the set of roles that need this information. An agent is said to need a piece of information in the sense that the proposition maps onto an input place of a transition in the Petri Net corresponding to an action that that agent can execute to carry out one of its responsibilities. We believe that using a belief representation for handling communications can serve as the shared mental model that a team maintains. This can then reduce the *explicit* communications needed between team members by instead promoting *implicit coordination* among team members. The information flow computed by the IARG algorithm can be used to generate communications for information exchange.

8 Development of a Coaching Agent

An advantage of our approach in CAST is that a coaching agent can use the model of teamwork within CAST to facilitate user modeling and the detection of errors between team members. User models [16] exist in single-user training systems in order to detect [17] and correct errors [18] in the trainee's domain knowledge. In a traditional ITS, an overlay approach is often used, in which the user's actions are compared to those that would be generated by an expert, to identify discrepancies between the student's (user) model, and the expert model (typically involving trigger or production rules for deciding what to do). However, understanding the actions of an individual on a team is more complicated because their decision-making explicitly involves reasoning about the other members of the team (e.g. their beliefs, roles, etc.), and their actions may be implicitly in support of a team goal or another agent.

Our approach in CAST is to model team members as maintaining simplified models of the mental states of all the other members on the team. To avoid issues of computational complexity with belief reasoning (e.g. via modal logics), we use Petri Nets as an approximate representation of these mental states. Then when a team member needs to decide what to do, they can not only reason about what actions would achieve their own goals, but they can reason about the state and needs of others. In particular, we focus on two effects: by making teamwork efficient through anticipating the actions and expectations of others (e.g. by knowing others roles, commitments, and capabilities), and by information exchange (knowing who to ask for information, or providing proactively just when it is needed by someone else to accomplish their task).

The coaching agent focuses on observing an individual's activities within the context of the team goals. Actions that each virtual team member takes depend on beliefs those agents hold regarding the goals and state of the other agents. Actions that a trainee takes also depend on his beliefs as to what needs to be done at that time in order to achieve the team goals. But beyond these actions, we can attempt to detect and properly classify whether a trainee has failed to act because of either inaction on the trainee's part, or an assumption by the trainee that it was another's responsibility, or a failure to properly monitor another team member. We also use the individual's model of teamwork to support the user model. We can infer the state of the team mode for the trainee based on observed actions, and we can map incorrect actions to problems with the trainee's representation of the other team members in the trainee's model of the team that would explain them, and from there back to the team/domain knowledge. Finally, the coaching agent will provide corrective feedback based on an appropriate pedagogical model (e.g. dynamically through hints during the scenario, and/or through after-action reviews).

9 Conclusions

The CAST system is currently being implemented as a distributed system in JAVA and RMI. We are using the domain of the NASA MCC to demonstrate this approach. We believe that this system can be a useful complement to traditional approaches in

training teams. The agent-based teamwork model can not only be used to implement virtual team members in an intelligent team training system, it can also serve as the “expert teamwork model” for a coaching agent to assess the actions and the performance of a team being trained.

An ITTS cannot replace an actual human team. But it can reduce the time and overall cost of training individuals in a team staff for domains such as control centers and other team-centered applications. An eventual goal is to run the ITTS system in parallel with real-time operations in order to allow agent-based virtual team members to follow, monitor, and advise the actual human team members as they perform their duties.

10 Acknowledgements

This research was partially supported by GANN fellowship grant P200A80305 and seed funds from the Texas Engineering Experiment Station for the Training System Sciences and Technology Initiative.

References

1. Cannon-Bowers, J. A., Salas, E.: Making Decisions Under Stress: Implications for Individual and Team Training. American Psychological Association, Washington, DC (1998)
2. Van Berlo, M. P. W.: Systematic Development of Team Training: A Review of the Literature. Tech. Rep. TM-96-B010, TNO Human Factors Research Institute, Soesterberg, The Netherlands (1996)
3. Blickensderfer, E., Cannon-Bowers, J. A., Salas, E.: Theoretical Bases for Team Self-correction: Fostering Shared Mental Models. In: Beyerlein, M., Johnson, D., Beyerlein, S., (eds.): Advances in Interdisciplinary Studies of Work Teams. JAI Press, Greenwich, CT (1997) 249-279
4. Cohen, P. R., Levesque, H. J.: Teamwork. *Nous*, vol. 25, no. 4 (1991) 487-512
5. Grosz, B., Kraus, S.: Collaborative Plans for Complex Group Action. *Artificial Intelligence*, vol. 86, no. 2 (1996) 269-357
6. Salas, E., Driskell, J. E., Huges, S.: Introduction: The Study of Stress and Human Performance. In: Driskell, J. E., Salas, E., (eds.): Stress and Human Performance. Lawrence Erlbaum Associates, Inc., Mahwah, NJ (1996) 1-46
7. Mengelle, T., DeLean, C., Frasson, C.: Teaching and Learning with Intelligent Agents: Actors. In *Intelligent Tutoring Systems '98*, San Antonio, Texas (1998) 284-293
8. Lesh, N., Rich, C., Sidner, C. L.: Using Plan Recognition in Human-Computer Collaboration. In *Seventh Int. Conf. on User Modeling*, Banff, Canada (1999) 23-32
9. Tambe, M.: Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, vol. 7, no. 1 (1997) 83-124
10. Levesque, H., Cohen, P., Nunes, J.: On Acting Together. In *American Association for Artificial Intelligence (AAAI '90)*, Boston, MA (1990) 94-99
11. Marsella, S. C., Johnson, W. L.: An Instructor's Assistant for Team-Training in Dynamic Multi-Agent Virtual Worlds. In *Intelligent Tutoring Systems '98*, San Antonio, Texas (1998) 465-473

12. Schmitt, L. J.: Prop Position. In: (eds.): Shuttle Prop, JSC-17238. NASA, Houston, Texas (1998) I.1.1-1 - I.1.1-13
13. Yin, J., Miller, M. S., Ioerger, T. R., Yen, J., Volz, R. A.: A Knowledge-Based Approach for Designing Intelligent Team Training Systems. In Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona, Spain (2000)
14. Rao, A. S., Georgeff, M. P.: Modeling rational agents within a BDI Architecture. In 2nd International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, MA (1991) 473-484
15. Covert, M. D., McNelis, K.: Team Decision Making and Performance: A Review and Proposed Modeling Approach Employing Petri Nets. In: W. Swezey, R., Salas, E., (eds.): Teams: Their Training and Performance. Ablex Pub Corp, (1992)
16. Wenger, E.: Artificial Intelligence and Tutoring Systems. Morgan Kaufmann Publishers, Inc., Los Altos, California (1987)
17. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, K.: The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In 14th Annual Conference on Uncertainty in Artificial Intelligence, Madison, WI (1998) 256-265
18. Baffes, P. T., Mooney, R. J.: Using Theory Revision to Model Students and Acquire Stereotypical Errors. In 14th Annual Conference of the Cognitive Science Society, Bloomington, IN (1992) 617-622