Sun, S., Councill, I., Fan, X., Ritter, F. E., Yen, J. (2004). Comparing teamwork modeling in an empirical approach. In Proceedings of the Sixth International Conference on Cognitive Modeling. 388-389. Mahwah, NJ: Lawrence Erlbaum.

Comparing Teamwork Modeling in an Empirical Approach

Shuang Sun (ssun@ist.psu.edu) **Isaac Councill (icg2@psu.edu)**
Xiaocong Fan (zfan@ist.psu.edu) **Frank E. Ritter (frank.ritter@psu.edu)**
John Yen (jyen@ist.psu.edu)
School of Information Sciences and Technology, the Pennsylvania State University
Information Sciences and Technology Building, University Park, PA 16802 - 6823

Introduction

Modeling teams is becoming an increasingly active topic for the research communities of multi-agent systems, cognitive modeling, and decision making. Many research efforts have tried to use software-agents to model teamwork. However, we found little work has been done comparing different methods with common experiments. The Agent-based Modeling and Behavior Representation (AMBR) project has compared different cognitive process modeling in a military simulation environment (Gluck & Pew, 2001), but it has only examined single agent behavior so far. In teamwork modeling, different architectures or models are often evaluated with different domains, which are not comparable. For example, STEAM (a Shell for Teamwork) used a helicopter combat scenario as a simulator (Tambe, 1997), while CAST (Collaborative Agents for Simulating Teamwork) used a Wumpus grid-world as its test-bed (Yen, Yin, Ioerger, Miller, Xu, & Volz, 2001). Consequently, it is difficult to evaluate teamwork performance between two architectures.

As a first attempt to address the issue, we conducted an empirical study to compare team performance. First, we built agent teams with two different architectures: CAST and Soar. Then, we tested them within the dTank simulation (Councill, Morgan, & Ritter, 2004). dTank is a tank game simulator for agents in a distributed environment. In the experiments, we compared the agents' knowledge and behavior.

Teamwork Modeling

We take a Unified Theory of Cognition (UTC) view of an agent as being composed of two parts: architecture and knowledge. We use the term architecture to refer to mechanisms and structures that process content (knowledge) and generate behaviors.

CAST. CAST has been designed to study teamwork related issues emerging from teams with well-defined structure and process, distributed expertise, and limited communication in time-stress domains. It implements a teamwork model that enables agents to anticipate potential information needs among teammates and to exchange information proactively. For example, the precondition of the "attack-enemy" plan includes the location of enemy. By sharing the condition knowledge, an agent will proactively deliver the location of detected enemy to other agents. Messages from CAST agents are realized through JAVA RMI with a Knowledge Query Manipulation Language or KQML format (Finin, Fritzson, McKay, & McEntire, 1994) that defines both the content of the messages and performatives of the communication.

Soar. Soar is a general cognitive modeling architecture that is designed to model human cognition (Laird, Newell, & Rosenbloom, 1987). Soar does not include any particular functions for teamwork. However, there have been several models of teamwork created in Soar. For instance, Tambe (1997) and his research group developed STEAM—a teamwork model that is based on Soar and joint intention theory (Cohen & Levesque, 1991). Implementing teamwork models using Soar, such as STEAM and Team-Soar, require writing Soar rules to implement collaboration and communication. For example, STEAM includes about 300 domain-independent Soar rules. For the purpose of this study, we developed a Soar team, which contains 22 productions encoded as communication knowledge.

dTank Teams. For comparison, agents in the CAST and Soar teams have similar procedural and declarative domain knowledge. The procedural knowledge contains a two-phase plan: search and attack. A team of agents wander around and search for enemy tanks. Once an enemy is found, the team-members communicate to inform each other of the enemy location and to coordinate their attack. Next, team members attack the target together and destroy it. This process iterates until all the enemies are destroyed. Declarative domain knowledge includes moving directions and obstacle locations.

Procedure. Both the CAST and Soar teams compete against a group of simple agents. We tested each team with the same scenario for 20 times. Each scenario is created with a randomly generated map and a set number of enemies. Agents' actions are recorded for further analysis. The simple agents are also implemented in Soar, except that they have no teamwork but a simpler knowledge representation.

Results

In our study, team agents' behavior is compared with their sequence of actions. In Figure 1, example agents are compared side by side. We find similar patterns for team operations across both teams. For example, initially both teams coordinate with a "hello-team" message, followed by a search plan that starts with a "turn" operator. This similarity results from equivalent collaborative knowledge included in both teams. We also noticed some consistency among attack behavior. For example, the circle in Figure 1 indicates that agents from both teams have a similar aim-and-shoot behavior pattern.

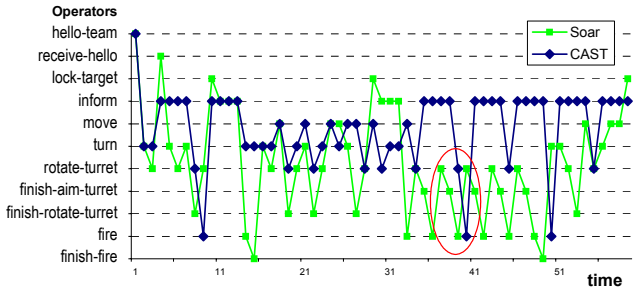

Figure 1: Behavior comparison by process model.

By comparing the frequency of each operator (Figure 2), we found that the most frequent domain non-communicative operator of both teams is rotate-turret. Other operators have comparable counts except the operators that are specially designed for Soar agents such as finish-fire. The most important and surprising difference is the communication operator, inform: a CAST agent communicates much more frequently than a Soar agent does. The CAST architecture is designed to communicate efficiently and a CAST agent should communicate less frequently. Why in this experiment do we see a contradictory result?

Actions	F2_Soar	F2_CAST
finish-aim-turret	22	0
finish-fire	5	0
finish-rottate-turret	12	0
fire	33	14
hello-team	1	1
inform	21	104
lock-target	5	0
move	11	14
receive-hello	1	0
rotate-turret	40	26
trace-target	6	0
turn	18	16

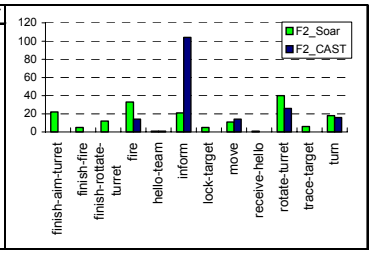

Figure 2: Frequency comparison of operators.

By examining the traces, we found the reason. A Soar agent needs to define the communication explicitly in productions. In this case, we defined the productions for communication after an agent locked onto a target. In contrast, the communication policies in CAST are embedded: whenever an agent observes a new piece of information, it sends the information to others who may need this information. Therefore, although implementing team communication for a CAST agent is easy, it sends more messages and it can be less efficient than ad-hoc designs that are less verbose. More importantly, the difference suggests that some team behavioral differences are difficult to resolve either with knowledge or with architecture alone.

Conclusions

As individual models become aggregated into teams of models, comparing architectures for teamwork modeling will become more important for cognitive engineers making decisions on choosing tools and implementing models. Through a set of experiments, we compared two architectures and their behaviors. Compared with Soar, CAST has more features that are designed specifically for modeling teams, and performed well where increased communication was useful.

Furthermore, we grouped the knowledge into domain-dependent knowledge and domain-independent knowledge. The domain independent knowledge is needed to compliment certain features from the architecture. In the Soar team, for example, knowledge about how to aim a gun at an enemy is domain-dependent; knowledge on how to compose messages is domain-independent. In CAST, communication is a part of the architecture. Therefore, the boundary between domain-independent knowledge and architecture can be blurred. Different architectures may be able to capture or use different domain-dependent knowledge. When we compare the knowledge coded for CAST and Soar, we find that Soar can incorporate more productions for making choice decisions.

Although the above findings are not conclusive, we have learned lessons on the relations between architecture and knowledge. (a) By including equivalent knowledge, a Soar team can perform collaborative behavior that is partially similar to a CAST team. (b) Some team behavioral differences are difficult to resolve with knowledge alone. (c) The experiments suggest that capturing teamwork behaviors as a part of the architecture or as a part of the agent's knowledge is important and perhaps equivalent decision choice for team modeling. (d) Human teams may vary on the teamwork behavior. Is it also affected by differences in knowledge? The question will motivate us to collect human data and compare with the models.

Acknowledgments

This research was supported by ONR (contract N000140210021), an AFOSR MURI grant (No F49620-00-1-0326), and a grant from Lockheed Martin Inc.

References

Cohen, P. R., & Levesque, H. J. (1991). Teamwork. *Nous, 25*(4), 487-512.

Councill, I. G., Morgan, G. P., & Ritter, F. E. (2004). *dTank: A Competitive Enviorment for Distributed Agents* (Technical Report ACS2004-1). University Park: The Pennsylvania State University. acs.ist.psu.edu/dTank.

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). KQML as an agent communication language. *In Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*. 456-463. Gaithersburg, MD. ACM Press

Gluck, K. A., & Pew, R. W. (2001). Overview of the Agent-based Modeling and Behavior Representation (AMBR) Model Comparision Project. *In Proceedings of the 10th Computer Generated Forces and Behavioral Representation Conference*. 3-6. Orlando, FL.

Laird, J., Newell, A., & Rosenbloom, P. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence, 33*(1), 1-64.

Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research, 7*, 83-124.

Yen, J., Yin, J., Ioerger, T. R., Miller, M. S., Xu, D., & Volz, R. A. (2001). CAST: Collaborative Agents for Simulating Teamwork. *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*. 1135-1142. Seattle, WA.