By Paul Kogut, John Yen, Yui Leung, Shuang Sun, Rui Wang, Ted Mielczarek, and Ben Hellar

PROACTIVE INFORMATION GA HOMELAND SECURITY TEAMS

Imagine a software assistant agent (AA) that proactively provides relevant concise personalized information to an analyst or first responder and collaborates with other AAs to share knowledge and arrange person-to-person contact. The following descriptive scenario provides an example application of this concept. A report of an explosion at a chemical plant is received by a homeland security (HLS) analyst, local firefighters, and state police. The AA for the analyst "reads" the report and immediately retrieves information about what chemicals exist in the plant and a map of nearby dangerous and vulnerable facilities.

The firefighters arrive on the scene and identify the sector from which flames and smoke are emanating. An AA provides the firefighters with information on what chemicals are likely to be burning, how to extinguish the fire, and the potential health hazards of the smoke. The AA notifies the HLS analyst that it is a harmless gas. Plant personnel tell state police they saw a suspicious individual in a car in the parking lot. The police enter the license plate number of the suspicious car and the analyst AA immediately searches for aliases of the owner and links to terrorist organizations. A link is found and the HLS analyst's AA searches for an AA of an expert on that terrorist group. The terrorist expert AA notifies the HLS AA that an associate of the suspicious person is a chemical engineer that works in a nearby plant where another explosion has just been reported. The HLS AA discovers that if the two smoke plumes intersect they will create a deadly acid mist. The AA plots the smoke plumes on a map and notifies the HLS analyst that the combined plume will reach a crowded sports stadium in approximately 15 minutes. The AA immediately initiates a phone call between the HLS analyst and stadium security.

This scenario illustrates how a team of software agents could support a team of people to accomplish a time-critical task related to U.S. homeland security activities. One of the major technical issues in designing software agents with these capabilities is they need to be

THERING FOR

Supporting counterterrorism analysts with software agents that dynamically anticipate their information requirements.

able to anticipate and reason about information needs of teammates in a highly dynamic environment.

Realization of the Vision

Lockheed Martin and Pennsylvania State University's School of Information Sciences and Technology are collaborating on a project to realize this vision using CAST (Collaborative Agents for Simulating Teamwork) agent technology [4–6]. There are several similar agent research efforts: CoABs (see coabs.globalinfotek.com/) and STEAM [3] are two examples. One of the major features distinguishing CAST from these approaches is that CAST enables an agent to dynamically infer information needs of its teammates from a shared mental model about the structure and the process of the team, which is expressed in the knowledge representation language MALLET (Multi-Agent Logic-based Language for Encoding Teamwork).

Figure 1 shows the overall architecture of using CAST agents to assist homeland security teams. The user guides the creation of the shared mental model (that is, MALLET knowledge of CAST agents) by selecting components from an ontology that pertain to the user's role in the team. Incoming messages and documents are preprocessed by the M&DS Intelligent Information Factory (see mds.external.lmco.com/mds/products/gims/iif/index.html) to extract important entities, relations, and events. Based on the shared mental model captured in MALLET, the extraction results trigger appropriate information gathering actions by the CAST agent. The CAST agent then interacts with the reasoning

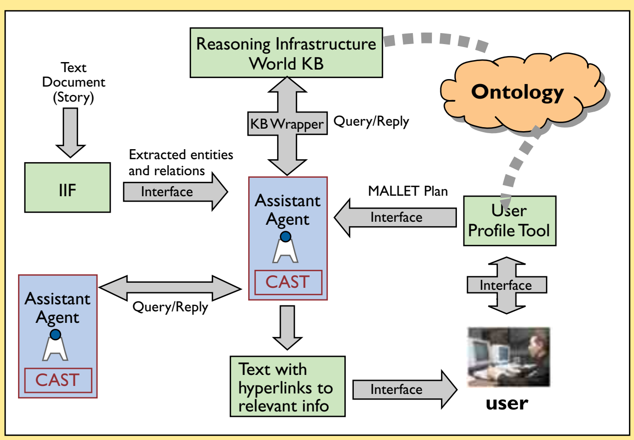

Figure 1. Architecture for proactive information gathering.

infrastructure/world knowledge base (RIWKB) or other CAST agents to retrieve and present relevant concise personalized information to the user. Heterogeneous knowledge from various sources is loaded into the RIWKB by techniques such as Web scraping, information extraction, and importing Semantic Web

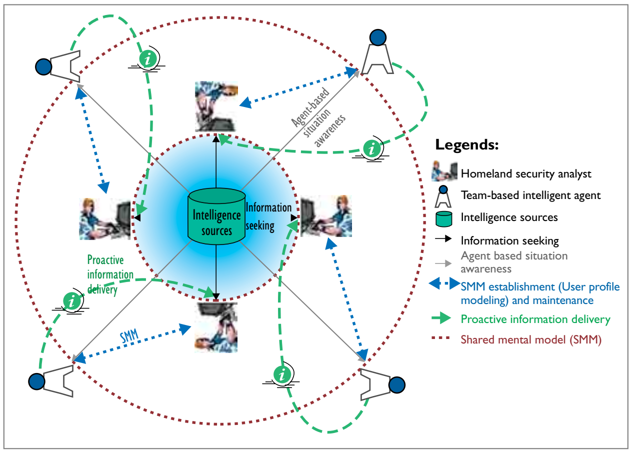

Figure 2. Shared mental models.

markup [2]. The output to the user is the original text with superimposed hyperlinks to fine-grained information (with drill-down links to original source documents).

Agents for Proactive Information Exchange

As mentioned earlier, CAST is an agent architecture that empowers agents in a team to have a shared mental model about the structure and the process of the team so they can anticipate potential information needs of teammates and proactively deliver it to them. Psychologists who studied teamwork have identified overlapping shared mental models as an important characteristic of high- performance teams [1].

To enable agents with such capabilities, CAST provides four key features. First, it uses a high-level language, MALLET, to capture knowledge about the team structure and the team process. Second, each agent establishes a computational shared mental model by transforming its teamwork knowledge in MALLET into a Prolog-like knowledge base and a predicate transition nets—a process representation that extends Petri Nets. Agents maintain their shared mental models about team states by dynamically updating the knowledge base and the predicate transition nets. Third, the CAST kernel provides domain-independent algorithms that enable agents to dynamically allocate responsibilities among members of the team, to infer information needs of teammates, and to proactively deliver relevant information to them. Finally, each agent uses a decision-theoretic communication strategy for determining how it should assist teammates regarding their information needs.

The CAST agents play the role of AAs by establishing and maintaining shared mental models with the human analysts as shown in Figure 2. The shared mental models contain knowledge about the responsibilities, information-seeking processes, and information-processing methods of human users. The notion of shared mental model extends the concept of user profiles in two ways. Fundamentally, it broadens the scope of user profiles to include dynamic process, tasks, roles, and responsibilities about human users. Additionally, it extends the profile of an individual user to a team.

Conclusion

CAST-enabled AAs will profoundly change the way HLS teams perform their missions. The architecture we have described here can be tailored to support applications in other domains such as teams of institutional and individual investors in the financial domain and enterprise knowledge management in large corporations with diverse technologies. The architecture could also be augmented with machine learning for automated adaptation of information needs. **C**

References

1. Cannon-Bowers, J.A., Salas, E., and Convers, S.A. Cognitive psychology and team training: Training shared mental models and complex systems. In *Human Factors Society Bulletin*, 1990.
2. Kogut, P. and Heflin, J. Semantic Web technologies for aerospace. In *Proceedings of IEEE Aerospace Conference*, 2003.
3. Tambe, M. Towards flexible teamwork. *Journal of Artificial Intelligence Research 7* (1997), 83–124.
4. Yen, J., Fan, X., and Volz, R.A. On proactive delivery of needed information to teammates. In *Proceedings of the AAMAS 2002 Workshop of Teamwork and Coalition Formation*, 2002.
5. Yen, J. et al. CAST: Collaborative Agents for Simulating Teamwork. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001, 135–142.
6. Yin, J. et al. A knowledge-based approach for designing intelligent team training systems. In *Proceedings of the Fourth International Conference on Autonomous Agents*, 2000, 427–434.

Paul Kogut (paul.a.kogut@lmco.com) is a research project lead at Lockheed Martin Integrated Systems and Solutions in King of Prussia, PA and an adjunct professor of software engineering at Pennsylvania State University.

John Yen (jyen@ist.psu.edu) is University Professor of Information Sciences and the professor in charge of the School of Information Sciences and Technology at Pennsylvania State University.

Yui Leung (yui.h.leung@lmco.com) is a software engineer at Lockheed Martin Integrated Systems and Solutions.

Shuang Sun (ssun@ist.psu.edu) is a Ph.D. student of information sciences and technology at Pennsylvania State University.

Rui Wang (rzw104@psu.edu) is a Ph.D. student of information sciences and technology at Pennsylvania State University.

Ted Mielczarek (ted.a.mielczarek@lmco.com) is a software engineer at Lockheed Martin Integrated Systems and Solutions.

Ben Hellar (bhellar@ist.psu.edu), formerly a software engineer at Lockheed Martin Integrated Systems and Solutions, is a student of information sciences and technology at Pennsylvania State University.