Editor: **Daniel Zeng,** University of Arizona, zeng@email.arizona.edu

Situation Awareness and Cognitive Modeling

Mieczyslaw M. Kokar, *Northeastern University*
Mica R. Endsley, *SA Technologies*

Situation awareness has been recognized as one of the important, yet unsolved, issues in many different domains, including human-controlled and -monitored mobile communication networks, social networks, physical and cyber-security systems, disaster monitoring and recovery, epidemic monitoring and control, intelligent transportation systems, financial and investment services, and tactical and operational battlefield command and control. These were just some of the situation awareness scenarios discussed at the 2nd IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (see http://www.cogsima2012.org/pag.html).

The common feature of all such systems is that they need to react to a dynamic environment that changes its state independently of whether the human or computer agents act on it. The agents want to act on the environment so that its evolution, at least in the area of interest to the agents, leads to the satisfaction of their goals. Towards this end, the agents need to collect information about the environment (usually from many different sources), make decisions based on the collected information and their knowledge, act according to their decisions, collect feedback from the environment in response to the actions, and update their knowledge (learn) to make better decisions in the future.

Since the amounts of information and the decision-making processes are complex, both humans and computers may be overwhelmed by the challenges of information processing. Thus, computer agents and humans need to collaborate—that is, share the responsibilities on information processing tasks (see Figure 1).

Collaboration means working together towards the achievement of common goals. It requires exchanges of information among the collaborating parties, which in turn requires the agents be interoperable. This means they need to follow some established protocols so that messages are delivered intact and can be interpreted by the receiving nodes. For example, in communications, the interoperability of two communicating nodes at a specific protocol layer means that both nodes execute the protocols associated with that layer in the open systems interconnection protocol stack—for example, in the physical layer or the data link layer. Since the application layer typically hosts the interfaces for both computer agents and human GUIs, interoperability requires a common protocol for this layer as well.

However, it isn't sufficient to just follow a common protocol. To achieve common goals, the computer and human agents need to have both a shared understanding of the goals and a shared understanding of what is relevant for a particular goal, so they can perform tasks that lead to achieving the goals. The understanding of goals and tasks is referred to as a *mental model*,[1] as Figure 1 indicates. For computers, we call these models *computer models*; both can be viewed as types of *cognitive models*.

In this article, we are trying to explain the problem of human-computer collaboration towards improving situation awareness (SA) which would in the end support better understanding of situations the humans are faced with and to support their decision-making tasks. We begin by briefly discussing the interpretation of some of the terms needed for the presentation. Our objective is to analyze the role and form of cognitive models in the process of collaborative development of shared situation awareness.

Basic Concepts

According to Merriam-Webster, the word *aware* "implies vigilance in observing or alertness in drawing inferences from what one experiences" (www.merriam-webster.com/dictionary/aware). This definition captures the essence of the meaning

1541-1672/12/$31.00 © 2012 IEEE
Published by the IEEE Computer Society

of this term and suggests that an aware subject (agent) observing an environment should possess the following features:

- It must be vigilant: it should be actively looking for information, presumably relevant to its goals. However, not all inputs come from intentional observations; some are imposed by the environment, so that the agent can experience various inputs without seeking them.
- It must be prompt in drawing conclusions from its observations. This implies that the agent must possess some capability of *inference* (deriving new propositions from others that are considered to be true).
- They must not only capture relevant data about the environment (Level 1 SA), but also understand the meaning or significance of that information (Level 2 SA) and be able to project near-term changes to the system (Level 3 SA) that are important for proactive decision-making.[1]

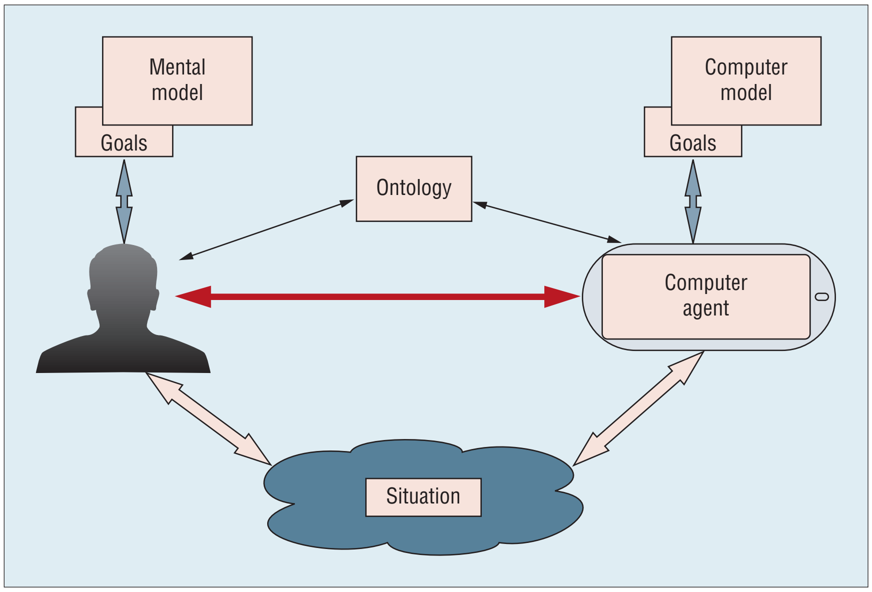

Figure 1. Human-computer situation sharing. Human and computer agents develop models of situations. Goals determine boundaries of situations. Situations then can be described in terms of the shared ontology and communicated to other agents.

Based on how humans develop good situation awareness in complex and dynamic environments,[1] a computer model of situations that attempts to achieve this goal needs several specific characteristics.[2] First, it needs to include a model of the system and environment that defines what is relevant about them, provides for dynamic information prioritization, and provides a mechanism for integration of low-level data to create meaning (for example, an understanding of the significance or importance of low-level data and projections of possible and likely future situation states). Such a model needs a process of active learning to maintain and refine it as new things about the system are encountered. In addition, where recognized classes of situations exist, they need to be linked to the model for rapid processing of well-defined situations (likely requiring a hybrid model). The more extensive system model can be used in circumstances where there isn't a good fit with known cases.

Second, to be successful, these models need to capture an understanding of relevant goals. Without goals, sensed data has no independent meaning, making strictly bottom-up information integration and fusion nearly impossible. Goals define information's relevance (separating signal from noise) and let meaning be established regarding that information. Most human roles have multiple goals that dictate the types of decisions they need to make, and thus what information they need to attend to and how they process it. So, the goals to which low-level data is being applied largely dictate the higher levels of situation awareness (comprehension and projection).

Finally, as there can be multiple and sometimes competing goals, the computer model will need to include a mechanism for goal prioritization, along with knowledge of which data states are pertinent for indicating which goals are the most critical at a given time. Creating a robust computer model of situations isn't easy, but many of these capabilities do exist in existing computer science approaches and can be combined into a successful model.

Case Studies

A few examples will help illustrate how these concerns interact in practice.

Development of a Situation Model using Fuzzy Cognitive Mapping

One approach to developing a situation model is to apply a fuzzy cognitive mapping (FCM) to the agent's internal representation of the world,[3] creating a so-called "SA-FCM" model. In one example, SA-FCM models were created for infantry operations. The models were comprised of concepts and weights that were categorized into three types of layers:

- The *input layer* contained the concepts that were directly connected to the external world.

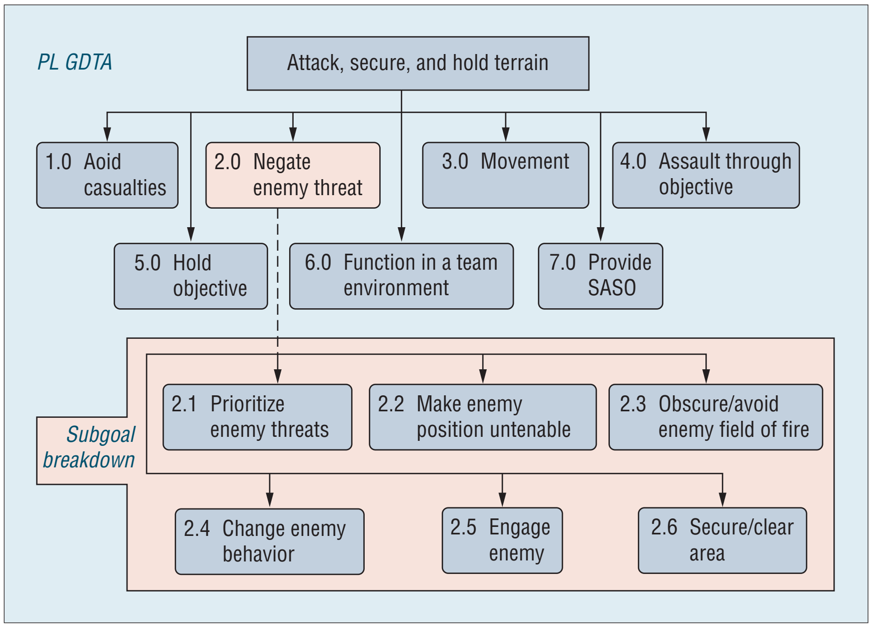

Figure 2. Platoon leader goal submap structure. The goals and subgoals for a platoon leader are mapped out using a goal-directed task analysis.

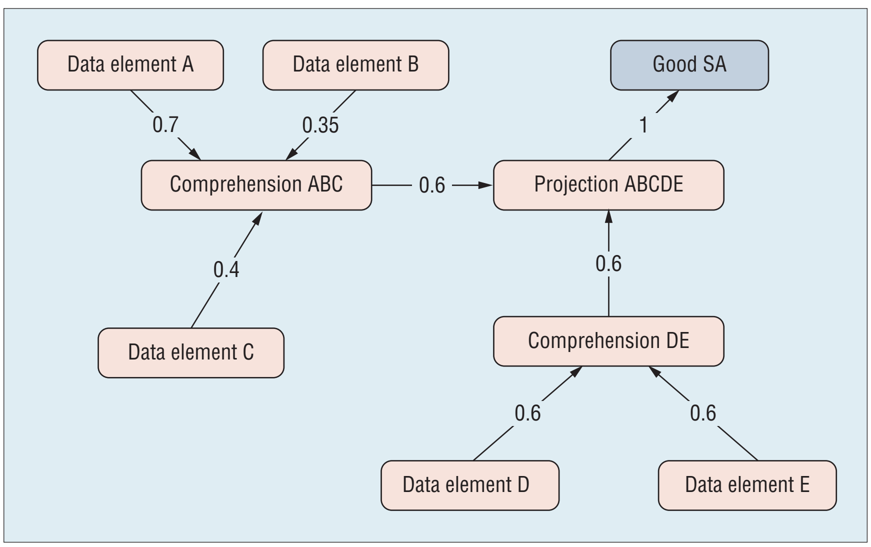

Figure 3. Fuzzy cognitive mapping structure captured from the goal directed task analysis, showing the key situation awareness requirements at all three levels (perception, comprehension, and projection) tied to each decision and subgoal.

- The *middle layer* was a processing layer that integrated concepts from the input layers and directed them to the output layer.
- The *final layer* was the output layer; its values were directed into the external world or back into the input layer.

This approach created situation awareness agents that functioned as autonomous software components designed for runtime context acquisition, situation analysis, and triggering the reactive behavior of the system. The SA agents included not only low-level data, but also mechanisms for assigning meaning and significance to that data and for making projections (in other words, to have the higher levels of situation awareness corresponding to the higher levels of the JDL fusion model).[4]

The FCM included a goal submap that defined the main goals, the subgoals, and how each goal influenced the others. A graphical network of SA requirements for the role, as determined from a goal-directed task analysis conducted to determine what constitutes good SA,[5] was linked to the goal submap. The SA requirements network included not just which data was important, but also how that data combined to form significance (level 2 SA) and future projections needed for each of the key decisions linked to the subgoal map. Specific weights between nodes reflected the input of subject matter experts.

The SA-FCM created in this example was used in a simulated mission to create course-of-action plans for an operation by a military platoon. A Turing test evaluated whether an independent subject matter expert—an experienced military leader—could tell whether the operational plans came from another expert or from the SA-FCM. In all cases, he was unable to distinguish which plans came from the computer. This indicates that the SA-FCM is a viable approach for modeling goals, decisions, and SA requirements across the three SA levels and then translating that information into a complete actionable model.

In practice, such a system would be highly useful for creating and evaluating different options in various types of military scenarios. While this example didn't include all the desired characteristics of a situation model (such as a learning mechanism and a connection to a case-based model), it did embody many of the necessary characteristics, primarily a mechanism that included goals and mapped them to the higher levels of SA needed to translate low-level data into higher-level meaning. Such an approach may be integrated with other modeling approaches, and extended with other needed features, to create a robust computer capability for situation modeling.

Representing and Reasoning about Situations Using Ontologies

This case is based on a scenario discussed by Jon Barwise and Jerry Seligman.[6] We have extended the scenario a bit to make it applicable to discussion of collaborative human-computer situation analysis.

In this scenario, Judith injures her leg by tumbling a hundred feet while hiking in the mountains. She clearly understands her situation: she's injured, it's already 4 p.m., she needs to get to a hospital, but she's far from the hospital and from anyplace she just could ask someone for help. She can't move on her own, and she doesn't have a phone connection. At the same time, she's aware of her own capabilities and of things within her reach that might be useful to achieve her goal of getting to a hospital. In particular, she has a flashlight that she could use to send an SOS, using Morse code, into the twilight.

We extend this scenario by assuming the existence of an Emergency Monitoring System (EMS) that is continuously scanning the environment for some indications of possible emergencies, such as forest fires, explosions, tornadoes, or other events. In particular, this EMS has the capability of recognizing Morse code letters (sequences of dots and dashes), words, and even the meanings of some words. For instance, the EMS can recognize the S-O-S sequence as a distress signal. Moreover, we assume that the EMS can act appropriately to also recognize the situation. The EMS then uses its localization sensors in order to determine the source of the SOS signal and accesses a geolocation database to identify the features associated with the location. In this case, it can determine that the specific location is in the mountains, far from any towns or even single houses, and that there are no roads that could be used to get to this location by car. The EMS also uses some plausible assumptions about the situation, for example that the source is a human and that the person is not doing this as a prank.

In this case, a computer-based EMS and a human are interacting in order to convey the information about a specific situation. The human sends a message, and the computer intercepts and interprets it and develops its own representation of the situation. Then the EMS sends the description of the situation to the mountain rescue team's computer, which displays it to a human in charge of rescue operations.

In order to understand how this whole process can be implemented, we need to show how the situation is represented in the computer and described by the human, how the additional information about the situation is inferred by the computer, and how the situation description is conveyed from the human to the computer, then to another computer and finally to another human.

Situation representation. While there are many different ways to represent knowledge, our method is a logic-based scheme as used in the Semantic Web. First we develop an ontology, and then we represent particular pieces of information about the scenarios in terms of this ontology. Since the goal is to capture situations, we use the Situation Theory Ontology (STO)[7] with certain extensions.[8] Figure 4 presents the main concepts of this ontology.

The representation consists of boxes and arrows with labels. The boxes represent *classes*, or concepts, and the arrows represent *properties*, or possible relationships between instances of the classes. The tails of the arrows indicate the classes that are the domains of the relations, while the heads point to the ranges.

The central class of this ontology is *Situation*. Instances of this class have properties of *relevantIndividual*, *focalIndividual*, *relevantRelation*, and so on. The other classes include *Individual* (individuals involved in a particular situation), *Attribute* (attributes of individuals or situations), *PropertyRelation* (relations that are relevant to the situation), and *Rule* (conditions that need to be true for a particular relational tuple in a given situation, and the inferences that are drawn when the conditions hold). Attributes can have *Dimensionality* and *Value* (for example, "meter" and "25," respectively). *ElementaryInfon* represents queries, or goals—statements that give focus to a particular situation. An example of such a statement could be a query, "Is Judith safe?" *Polarity* is a special kind of value; it can be either 1 (meaning the infon is satisfied) or 0 (not satisfied), corresponding to the whether statement represented by the ElementaryInfon is true or false, respectively. The notions of ElementaryInfon and Polarity come from Jon Barwise's situation theory.[9,10]

Representation of the example scenario. STO is a general-purpose ontology for representing situations. To represent emergency situations like the example, STO needs to be extended by some additional classes and relations. Classes should be subclasses of the STO classes, while relations can be either subproperties of the STO properties or new relations.

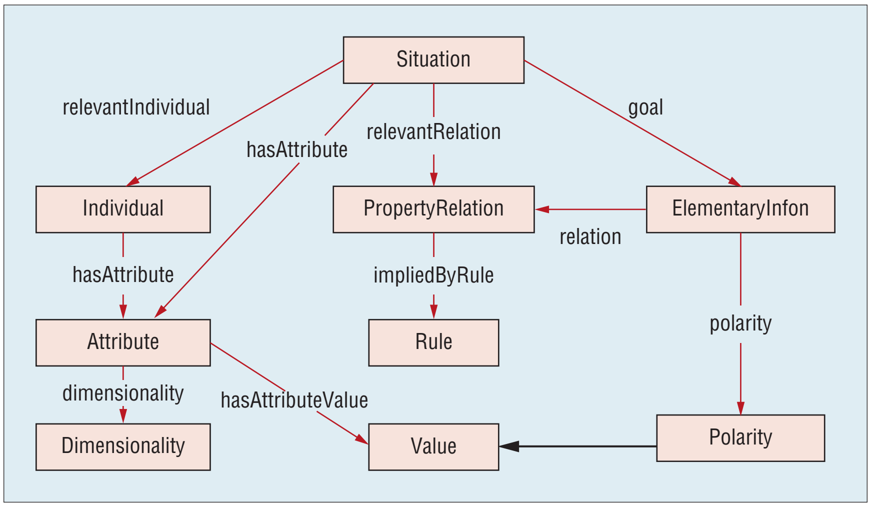

Figure 4. Top level of the Situation Theory Ontology. The boxes represent classes, or concepts, and the arrows represent properties, or possible relationships between the classes.

We might also need to supplement the definitions of classes and relations with rules. In this case, we extended STO by adding classes: *Emergency-Situation* (a subclass of Situation), *DistressSignal* (a subclass of Individual), and its subclass *SOS*. We also added a restriction stating that an instance of such a signal associated with a situation is a sufficient condition for inferring that the associated situation is an instance of Emergency-Situation. We call the extended ontology STO-X

The first piece of information we need to represent in STO-X is the distress signal. This information is created by the EMS after recognizing an SOS and adding *J-SOS* as an instance of DistressSignal. Moreover, we assume the existence of a rule by which the reasoner will create an instance of EmergencySituation in case a signal of this kind is added to the ontology—a "where there's smoke, there's fire" rule. In this case we assume this rule resulted in the creation of *J-Sit*, an instance of EmergencySituation.

In the next step the reasoner will invoke the axioms of the ontology shown in Figure 4 in order to infer the various aspects associated with the Situation class. In particular, the axioms will result in inferring the location of the SOS signal and the time it was sent. Through invoking other (background) knowledge, also represented in the form interpretable by the reasoner—in this case, in the Web Ontology Language (OWL)—it will become explicit that the location is in the mountains with no access by car and so on, as outlined previously. For instance, using rules similar to the smoke-fire rule mentioned above, the reasoner will infer that there is a *RelevantIndividual*, not known by name as yet, who sent the SOS signal and who is in this emergency situation now; that the intent of the person is to be in a safe place; that one of the relevant relations is *distance-ToRoad* (ternary relation between Judith, closest road, and distance); and that some of the relevant attributes of the situation are Location and Time, with the appropriate values and dimensionalities.

After inferring all the relevant information, the computer agent now needs to convey the description of this situation to the Mountain Rescue Team's computer, which in turn needs to present this description to a dispatcher. The message in this case is expressed in the XML serialization of the OWL expressions.

In this scenario, then, the following steps occurred:

1. The information about Judith's situation was conveyed from Judith to the EMS via just a simple SOS signal.
2. The description of the situation was inferred by the EMS computer system locally.
3. The inferred situation was sent over to the Mountain Rescue Team computer (all in OWL).
4. The situation was displayed to the dispatcher on duty.

All these transmitted messages were partial representations of the mental or computer model, as we discussed before. Not all the information in the model had to be sent over the communications channels: because both the computer and the human agents are assumed to share some ontological concepts and possess the inference capability, the implicit information was inferred by the agents locally.

The scenario ends with the dispatcher making an informed decision to send a helicopter to transport Judith to a safe place.

Currently available means of situation representation and communication can partially solve the scenarios and the approaches to dealing with situation management we have described. However, some of the steps in these scenarios still require further research. First, the computer agent that inferred the description of Judith's situation had to rely on the STO and the extensions specific to the scenario. Work is needed to develop more comprehensive ontology extensions so that many situations, at least in a specific domain such as emergency response, can be represented without new extensions.

Second, reasoning about various situations requires special-purpose rules. As with ontologies, rules should be developed that would cover a wide range of domain specific scenarios. Third, templates for typical situations in particular domains would be beneficial for the efficiency and reliability of the situation- assessment process.

Finally, there's the issue of confidence. Means for constructing and representing confidence in a situation assessment from the reliability of particular sources of information and credibility of particular pieces of information need to be developed.

References

1. M.R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors J.*, vol. 37, no. 1, 1995, pp. 32–64.
2. M.R. Endsley, "Bringing Cognitive Engineering to the Information Fusion Problem: Creating Systems that Understand Situations," *Plenary Presentation to 14th Int'l Conf. Information Fusion* (Fusion 11), IEEE, 2011; www.fusion2011.org.
3. R.E.T. Jones et al., "Modeling Situation Awareness for Army Infantry Platoon Leaders Using Fuzzy Cognitive Mapping Techniques," *Proc. Conf. on Behavior Representation in Modeling and Simulation* (BRIMS 10), BRIMS Society, 2010, pp. 216–223.
4. A.N. Steinberg and C.L. Bowman, "Revision to the JDL Data Fusion Model, *Handbook of Multisensor Data Fusion, 2nd ed., Second Edition*, M.E. Liggins, D.L. Hall and J. Llinas, eds., CRC Press, 2009, pp. 45–68.
5. M.R. Endsley and D.G. Jones, *Designing for Situation Awareness: An Approach to Human-Centered Design*, Taylor & Francis, 2012.
6. J. Barwise and J. Seligman, *Information Flow: The Logic of Distributed Systems*, Cambridge Univ. Press, 1997.
7. M.M. Kokar, C.J. Matheus, and K. Baclawski, "Ontology-Based Situation Awareness," *Information Fusion*, vol. 10, no. 1, 2009, pp. 83–98.
8. B. Ulicny et al., "Augmenting the Analyst via Situation-Dependent Reasoning with Trust-Annotated Facts," *Proc. 2011 IEEE Int'l Multi-Disciplinary Con. Cognitive Methods in Situation Awareness and Decision Support* (CogSIMA 11), IEEE, 2011, pp. 17–24.
9. J. Barwise and J. Perry, *Situations and Attitudes,* MIT Press, 1983.
10. K. Devlin, *Logic and Information,* Cambridge Univ. Press, 1991.

Mieczyslaw M. Kokar is a professor of electrical and computer engineering at Northeastern University. Contact him at m.kokar@neu.edu.

Mica R. Endsley is president of SA Technologies in Marietta, Georgia. Contact her at mica@satechnologies.com.