Situation Assessment: Procedural vs. Logical

Mieczyslaw M. Kokar
Department of Electrical & Computer Engineering
Northeastern University
360 Huntington Avenue
Boston, MA 02115
mkokar@ece.neu.edu

Abstract – *This paper discusses issues and challenges in automatic situation assessment. First, a number of problems have been identified. Then each of the problems is briefly discussed, including main approaches to their solution. Two kinds of solutions are compared - procedural and logical.*

1 Introduction

Various terms have been used to refer to Level 2 fusion processing: *situation refinement*, *situation awareness*, *situation development*, *relation estimation* and other. All of these terms have a common part in their definition, i.e., all of them require that the definition should include the knowledge of all the relevant objects and their kinematic states. This is essentially a Level 1 function, so it will not be discussed here. Some of the definitions, but not all, include the requirement of knowing relationships among the objects. This brings three problems: 1. the *relevance problem*: there are so many possible relations - which ones are relevant? 2. the *the resource problem*: where can we get the necessary information resources, both data and processing, that can be used to assess the current situation? and 3. the *derivation problem*: how do we derive whether a particular relation holds or not? And even fewer definitions capture the aspect of *awareness* as defined in the Webster dictionary [1], where awareness is explained as "AWARE implies vigilance in observing or alertness in drawing inferences from what one experiences." In other words, a *subject* is *aware* if the subject not only observes (experiences) the *objects* but also is capable of drawing conclusions from these observations. We call this the *inference problem*: how can we infer the implications of a specific situation on the tasks that we are pursuing? In the following, we briefly discuss all of these problems and indicate directions for the search of solutions.

2 The Relevance Problem

As a simple example consider a situation with 20 objects involved. How many possible (binary) relations can there be? A relation is a subset of the Cartesian product. The size of the Cartesian product in this case is equal to $20 \cdot 20 = 400$. The number of subsets in this Cartesian product is equal to $2^{400} = 2.5822 \cdot 10^{120}$. Clearly the number is prohibitively large for considering all possible subsets. Typically, only a relatively small subset of these relations would be relevant. Which ones should be checked by the situation assessment system?

Two approaches to this problem are: 1. the system developer makes a selection of the relations from this subset and encodes procedures for the assessment of these relations, and 2. the system has access to the knowledge of the domain and then analyzes dependencies among the relations and picks the ones that can be traced to the relations that are indicated as relevant by the human user. In our work we use the latter approach (cf. [2, 3]).

3 The Resource Problem

A typical interpretation of the Level 2 processing is that a Level 2 system must be able to process information that comes from Level 1, which in turn is responsible for the processing of the information provided by the sensors directly attached to the system. This seems like a rather narrow formulation of the situation assessment problem. This is definitely not the case in human processing where we use all the knowledge that we can access in order to properly estimate the situation. In particular we access all kinds of data and knowledge - data bases, books, libraries as well as our sensory inputs and our own memory. In addition to this, we often rely on the information that we can obtain from other humans, i.e., other peoples' processing capabilities.

Similarly, a situation assessment system could access all kinds of data. Instead of just processing whatever is presented to the system, it could also actively seek additional data that are relevant to a specific situation. Additionally, it might share processing responsibilities with other processes (some call them *agents*), including humans. Consequently, a situation assessment system could have have connections to humans, other processors and data sources. Data sources may include sensors, data bases and knowledge bases, but also unstructured data sources, like text. Sensors may be connected directly to the system, but they might also be connected to it indirectly through other data/knowledge sources. In order to assess a specific situation, a situation assessment system might need to access any of the existing (somewhere in the universe) data/knowledge source and any process.

Where do humans get information from? In the past, the only sources of information were other humans. Later in history, written information became the primary source of knowledge, followed by media, like radio and TV. This situation has changed in recent years due to the development of the Internet and the World Wide Web. Now, whenever we are faced with a problem to solve, our first step is to search the Web for various answers. However, while the information on the World Wide Web is appropriate for human processing, due to the fact that it is mainly in the form of text and images, it is not particularly suited for computer interpretation. The realization of this problem is behind the concept of the Semantic Web [4]. According to the vision of the Semantic Web, the Web is one large memory that keeps a huge number of information *resources* that include both data and processes. Each resource has a Unique Resource Identifier (URI), which can be used by a computer based agent to access the resource. It seems then natural to expect that situation assessment systems should be able to take advantage of all this wealth of web based information resources.

To make the Web accessible to computer interpretation, the W3C has developed a new XML-based language called the Web Ontology Language (OWL) [5]. OWL is an emerging standard for knowledge representation. It has a formally defined semantics, which makes it possible for computers to interpret OWL expressions. The expressions are built out of relations (called *properties*), *classes* and instances of those. OWL was designed with the focus on representing *ontologies*. Ontologies capture potential objects and potential relations among the objects. This means that ontologies do not describe what is in the world but rather what can be in the world. Ontologies, however, can be used to *annotate* or *markup* descriptions of instances of the world in what are called instance *annotations* or *markups*. In summary, the Semantic Web paradigm allows situation assessment systems access all the information resources available on the Web.

4 The Derivation Problem

Since in logical terms ontologies and markups define *logical theories*, formal logical inference can be carried out within such ontologies. Note that the reasoning capability cannot be accomplished with purely syntactic languages such as XML Schema since an XML Schema specification does not have any domain specific semantics. Reasoning can be accomplished with OWL, since it has formally defined semantics.

In the logical approach, a query about a specific relation can be posed to an *inference engine* (or a *theorem prover*). The inference engine then returns an answer, possibly with some variable bindings. A number of inference engines for OWL have been developed and/or are under development. Examples include JTP [6], Fact [7], Racer [8] and others.

In typical data fusion applications the derivation problem is solved in a different way than through logical reasoning. It is usually solved in a procedural way, i.e., in order to determine whether a particular relation holds or not, a procedure is invoked, which returns either a "yes" or a "no" answer, possibly also including some return parameters. While this approach may turn out to be more efficient in terms of time complexity, it lacks the genericity that the logical approach has. The limitation comes from the fact that only those queries for which procedures have been explicitly coded by the system developer can be answered. The logical approach is termed *declarative programming*, while the procedural approach is called *procedural programming*.

5 The Inference Problem

In the logical approach, the inference problem is closely related to the relation derivation problem. A logical query regarding any feature of a situation is posed to an inference engine. The query language for OWL is called OWL-QL. The number of types of queries is only limited by the complexity of the ontology that captures the domain knowledge. The queries are built our of the class expressions and property expressions using logical connectives that are part of the ontology language. This practically provides for an unlimited number of types of queries that can be formulated and then answered by an inference engine. Again, the advantage of the logical approach is that the query engine is not designed to answer a specific set of queries, but it is rather generic, capable of answering any query that is expressible in the query language. This is not the case in the procedural approach, where only those queries that have been formulated at the design time can be resolved by the system.

6 SAWA: A Situation Awareness Assistant

The observations presented in this paper have been collected during the two year period of working on the Situation Awareness Assistant (SAWA). In most general terms, SAWA is an ontology based situation monitor. Its main goal is to monitor a "standing relation", i.e., a query formulated in terms of an underlying ontology. SAWA collects information (events) and invokes its inference engine that derives whether the relation holds or not. The reasoning mechanism of SAWA combines logical inference with Bayesian belief propagation. A number of findings from this project have been published in papers [9, 2, 3].

7 Challenges

Although the logical approach advocated in this paper gives a promising approach to solving a general situation assessment problem, still, a number of issues need to be resolved in order to make the logical approach scalable up to the real world problems. Some of these issues are discussed in the following paragraphs.

A number of challenges were discussed in another paper (see [10] and for this reason they have been omitted from this presentation. They include such issues as objectification of situations (treating situations as first-class citizens) and treatment of uncertainty of derived relations and complexity of derivation algorithms.

Consistency and Ontology Mapping. For a logical approach to work, the underlying theory must be *consistent*. In short, it means that it must not be possible to derive the conclusion 'false'. Although the inconsistency problem is undecidable in the worst case, in the context of a stand-alone program, this is not a very serious problem since consistency checking tools (cf. ConsVISor [11]) can detect inconsistencies and even help the ontologist fix such problems. The problem is complicated by the fact that in the Semantic Web paradigm the whole Web is viewed as one large ontology. It is unrealistic to expect that the whole Web will be consistent. The best one can hope for is to be able to identify subsets of the Web such that are consistent. Unfortunately, even this seemingly simple approach cannot be applied within the means provided by the OWL language. The reason for this is that OWL does not have a mechanism for selective inclusion of portions of a number ontologies into one ontology that could then be checked for consistency. The only mechanism that OWL provides is the *import* construct, which is equivalent to including a whole ontology into another ontology. Another problem that is related to this operation is the how can two ontologies be integrated so that the resulting ontology preserves the semantics of both components? This problem is also referred to as the *ontology mapping problem.*

The Identity Crisis . Level 2 information may be reported to the situation assessment system using various kinds of reference. For instance, the object referred to as "the leader of the organization" might be the same object as the one pointed to as "the number one". The question then is how to resolve this kind of ambiguity? In Level 1 this kind of a problem is known as *data association*. Unfortunately, the Level 1 data association methods are not directly applicable to this kind of problem. The main reason for that is the lack of quantitative characteristics associated with the objects that are normally required for the use of the Level 1 data association methods. Consequently, other methods need to be applied. Within the logical approach, one can use the characteristic of consistency to resolve the issue of whether two references point to the same object. In short, one adds the hypothesis that two of the references refer to the same object and then checks the consistency of the resulting theory. If the result is "inconsistent", the belief in the hypothesis is low. On the other hand, one also can hypothesize that two references refer to two different objects. If the resulting theory is inconsistent, again, the belief in this hypothesis is low. Some results on this kind of approach is presented in [12].

Representational expressiveness vs. computational complexity . While the use of a generic reasoning mechanism provides a generic mechanism for checking relations and inferring consequences of a given situation, it also poses very high demand on the reasoning engine. This is because in general logical reasoning is undecidable. In simple terms, this means that an inference engine might never terminate its process of answering a specific query.

The problem of undecidability was a great concern to the designers of OWL. OWL comes in three flavors: OWL Full, OWL-DL and OWL Light. The three flavors differ in both the expressive power and computational complexity. OWL Lite is the simplest of the three, with the least expressive power. But still OWL Lite is in the EXPTIME complexity class, meaning that in the worst case a deduction will require time equal to $O(2^{p(n)})$, where $p(n)$ is a polynomial in the size n of the ontology. OWL-DL is more expressive than OWL Lite, but it is in the NEXPTIME complexity class (non-deterministic EXPTIME). which means that one can check a deduction (i.e., check the correctness of a proof) in time $O(2^{p(n)})$. Finding the proof takes OWL Full is the most expressive of the three OWL languages, but it is undecidable. Undecidability means that there is no bound on the amount of time or space that may be needed to perform a deduction, in the worst case. In practice, even OWL Full is not sufficiently expressive, and it is necessary to use rules to express some of the axioms of the ontology. Unfortunately, if rules are added to any of the three languages, including OWL Lite, then the computational complexity of deduction is undecidable.

In order to make systems based upon such generic reasoners scalable, the issue of complexity of reasoning must be resolved so that reasoning conclusions are derived within the constraints imposed by the situation assessment time requirements. There are various approaches to such a problem known in the literature. In general, such approaches are based upon a tradeoff between the quality of the solution and the computation time. More research in this area is needed to make the logical approach to situation assessment scalable to real world applications.

The Problem of Trust . When a situation assessment system accesses various information services on a web (even if the web is limited to a community of interest instead of the WWW) it must have means for determining whether it should rely on the information provided by such a service. It should be noted that the issue of trust goes beyond the problems of computer security. The problem of trust is present even if we know that the service provider is not a rogue agent. In some cases the system can simply trust a given (known) process with some degree. Or it can ask the service for explanation on how it derived a given conclusion. One of the methods is to provide a trace of the inference chain. The issue of trust is one of the research topics being investigated within the Semantic Web community.

8 Conclusion

This paper identified a number of problems with automatic situation assessment in the context of multi-source information fusion. Two approaches to the solution of these problems were discussed - procedural programming and declarative programming. Advantages and disadvantages of each of these approaches were discussed. While the declarative logic based approach was advocated, a number of challenges have been identified that need to be addressed in order to make this approach scalable to real-world applications.

References

[1] Merriam-Webster Online, 2004. Available at: `http://www.m-w.com/`.

[2] C. J. Matheus, M. M. Kokar, and K. Baclawski. A Core Ontology for Situation Awareness. In *Proceedings of the Sixth International Conference on Information Fusion*, pages 545–552, 2003.

[3] C. J. Matheus, K. P. Baclawski, and M. M. Kokar. Derivation of ontological relations using formal methods in a situation awareness scenario. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2003*, pages 298–309. SPIE, 2003.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[5] OWL-ref. OWL Web Ontology Language Reference, 2003. `http://www.w3.org/TR/owl-ref`.

[6] R. Fikes, J. Jessica, and F. Gleb. Jtp: A system architecture and component library for hybrid reasoning. In *Proceedings of the Seventh World Multiconference on Systemics, Cybernetics, and Informatics*, 2003.

[7] I. Horrocks. The FaCT System, 2005. Available at: `http://www.cs.man.ac.uk/ horrocks/FaCT/`.

[8] V. Haarslev, R. Mller, and M. Wessel. Racer, 2005. Available at: `http://www.sts.tu-harburg.de/ r.f.moeller/racer/`.

[9] M. M. Kokar, C. J. Matheus, K. Baclawski, J. A. Letkowski, M. Hinman, and J. Salerno. Use Cases for Ontologies in Information Fusion. In *Proceedings of the Seventh International Conference on Information Fusion*, pages 415–421, 2004.

[10] M. M. Kokar. Situation awareness: issues and challenges. In *Proceedings of the Seventh International Conference on Information Fusion*, pages 533–534, 2004.

[11] K. Baclawski, C. J. Matheus, M. M. Kokar, J. Letkowski, and P. Kogut. Towards a symptom ontology for semantic web applications. pages 650 – 667. Lecture Notes in Computer Science, LNCS 3298, Springer, 2004.

[12] M. M. Kokar, C. J. Matheus, J. A. Letkowski, K. Baclawski, and P. Kogut. Association in Level 2 fusion. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, pages 228–237. SPIE, 2004.