Available online at www.sciencedirect.com
ScienceDirect
Information Fusion 10 (2009) 83–98
INFORMATION FUSION
www.elsevier.com/locate/inffus

Ontology-based situation awareness

Mieczyslaw M. Kokar [a,*], Christopher J. Matheus [b], Kenneth Baclawski [c]

[a] Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, United States
[b] VIStology, Inc., Framingham, MA 01701, United States
[c] Computer and Information Science, Northeastern University, Boston, MA 02115, United States

Received 1 August 2006; received in revised form 14 January 2007; accepted 26 January 2007
Available online 20 February 2007

Abstract

The notions of ''situation'' and ''situation awareness'' have been formulated by many authors in various contexts. In this paper, we present a formalization of situations that is compatible with the interpretation of situation awareness in terms of human awareness as well as the situation theory of Barwise and Devlin. The purpose of this paper is to capture the situation theory of Barwise in terms of an OWL ontology. This allows one to express situations in a commonly supported language with computer processable semantics. The paper provides a description of the classes and the properties in the ontology, and illustrates the formalization with some simple examples.

Keywords: Situation; Ontology; Situation awareness; Situation theory; Formalization

1. Introduction

Situation awareness was envisioned as the main part of Level 2 processing in the JDL model [1,2]. But only recently has it become the center of attention for information fusion research. As is typical with a new field of research, various studies on this subject have contributed results that are difficult to integrate into one coherent conceptual structure. In other words, the field of situation awareness needs a unifying framework that would play the role of a common theory integrating various research efforts.

Situation awareness research can be classified by the subject that performs this process – human or computer. For human situation awareness, the model proposed by Endsley [3] has been more or less accepted by the information fusion community. Moreover, this model has been used in various studies as a justification for structuring the computer-supported situation awareness process. While the human situation awareness model has been grounded in various studies of cognitive science, the computer situation awareness process still lacks a more systematic treatment. Moreover, the difference between human and computer processing is that the human situation awareness process needs to be measured and possibly supported, which is the main focus of Endsley's research, while the computer process needs to be defined and implemented.

Clearly it is necessary to develop unambiguous specifications, designs and implementations of situation awareness processes. One of the trends in this direction that became prevalent in recent years is that of using ontology-based computing as a paradigm on which to develop computer based situation awareness processes (cf. [4–14]). Although all of these efforts are based on ontologies as the main representational structure, they lack commonality in the repertoire of concepts used in the analysis and the synthesis of situation awareness processing.

Artificial intelligence (AI) has dealt with a notion of ''context'', which, according to [15], stands for the same concept as ''situation''. This line of AI research was started by McCarthy, cf. [16] and is still an active research field. The main idea of the AI approach is to introduce a

* Corresponding author.
E-mail address: mkokar@ece.neu.edu (M.M. Kokar).

1566-2535/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2007.01.004

predicate, $isp(c,p)$, that explicitly states the fact that the proposition p is true in the context c.

Sowa in his book [17] provides both a historical overview of the AI treatment of context and an approach to representing contexts (situations) in the formalism of *conceptual graphs* [18,19]. Conceptual graphs are patterned upon *existential graphs* developed by Charles S. Peirce. Similarly to McCarthy's approach, Sowa introduces a *description predicate*, $dscr(x,p)$, which captures the fact that the entity x is described by the proposition p. When the entity is a situation, then the proposition p describes that situation. This predicate is then used to state facts that hold in a given situation. Conceptual graphs are representable in a graphical form that is more human friendly than a computer-readable form called Conceptual Graph Interchange Form (CGIF).

The principal goal of this paper is to formalize the main concepts of situation awareness using a language that is both processable by computer and commonly supported. To achieve this goal, we first need to identify appropriate concepts that can be classified as part of the situation awareness domain. We have already mentioned a number of such concepts provided by Endsley [3]. Another source of information on situation awareness is the situation theory developed by Barwise and Perry [20–22], which was subsequently extended by Devlin [23]. Since the concepts of situation theory encompass most of the concepts discussed by Endsley, and since situation theory is described in a more formal language, in this paper we first provide a short overview of situation theory and then show how situation theory can be captured in a formal language with computer-processable semantics.

Computer support for logic is a popular theme in computer science, and there are many languages that have been developed for this purpose. Moreover, situation theory has already been expressed in terms of some existing logical languages. However, few of these languages have even been standardized, and fewer still are commonly supported by popular software tools and systems. Currently, the only languages that have such support are the languages of the Semantic Web [24]: the Resource Description Framework [25] and the Web Ontology Language [26], which is based on RDF. OWL improves on RDF by adding many new logical capabilities. One of the most important new capability is the ability to define classes in terms of other classes using a variety of class constructors such as unions, intersections and property values. Accordingly, we have chosen OWL as the language for formalizing situation theory, and in this paper we give examples to show how the reasoning techniques pioneered by Barwise and Devlin can be mapped to OWL class constructors.

As mentioned above, situation theory has already been expressed in terms of some existing languages. While the argument in favor of OWL over these other languages is reasonably compelling, it is still worthwhile to consider some of the potential disadvantages of OWL relative to the alternatives. Two of the most commonly mentioned disadvantages of OWL are that it is wordy and unreadable. In fact, the wordiness of OWL is only a disadvantage from the point of view of people, not computers. To computers it becomes a significant advantage. There are various languages for representing OWL, but all of them share common features such as self-description, decoupling of facts from the containing document, and reduction to simple elementary statements. The first feature allows OWL to be parsed by commonly available generic parsers such as the ubiquitous XML parsers. The latter two features make it much easier to store and manage OWL facts in databases. These advantages easily outweigh the disadvantage of the wordiness of OWL. Concerning the unreadability of the XML representation of OWL, this is also only an issue for people, not computers. It is expected that people would usually neither read nor write OWL using the XML representation. However, it is still necessary sometimes, so it could be argued that some other language would be better. To deal with this problem, a number of alternative OWL syntaxes as well as GUIs have been developed that are much more readable and succinct and that map directly to the XML representation. The Abstract Syntax and N3 are two well known examples of syntaxes, and Protégé is a well known example of a popular GUI and IDE that supports OWL. Furthermore, these notations and GUIs are about as readable as possible given the requirement that the notation be self-describing.

The OWL language has three levels that have progressively richer semantics but are also progressively harder to process. Since situation theory requires that one model "classes as instances", it is necessary to use the highest OWL level, OWL Full. Furthermore, while OWL Full is sufficient for nearly all concepts required by situation theory, there are a few that even OWL Full cannot express. Those concepts can be formalized using a computer-processable rule language compatible with OWL such as RuleML. The concepts expressed in OWL and the ones expressed using rules together form a formal *ontology* for situation awareness. Since the intent of our ontology is to capture most of situation theory, we call it the *Situation Theory Ontology*, or STO, for short.

Such an ontology can play the role of a unifying theory of computer-based situation awareness. In this paper we describe all the concepts in this ontology. One of our claims is that STO is compatible with current thinking about situation awareness in the community. In particular, there are clear relations between the concepts in this ontology and Endsley's model of human situation awareness [3].

While the ontology discussed in this paper has useful characteristics, it is not complete, and experts in this field might have somewhat different opinions on which concepts should be included and how they should be represented. An ontology is valuable only if the majority of the community accepts its main concepts and structure. The most important aspect of our proposal is that the ontology is formally defined, i.e., it is expressed in a language with formal semantics. This fact makes it possible to ground the

discussion of the ontology in a precise and unambiguous language.

The secondary goal of this paper is to indicate how the STO can be used to develop situation awareness systems. The major point of this part is that a significant amount of flexibility can be achieved through the use of generic ontology-based tools. To achieve the secondary goal, we give examples of how the ontology based approach to situation awareness can be used. For this purpose we first describe a simple example (somewhat similar to the one used in Sowa's book [17]) and show how that situation can be represented. Then we show how automatic logical inference can be carried out using the formal description of the situation and of the ontology.

The paper is organized as follows. In Section 2, we discuss various meanings of "situation" and "situation awareness." In Section 3, we discuss Barwise's situation theory and how it is formalized by Devlin. In Section 4, we introduce the STO based on the situation theory of Barwise. Section 5 describes the example we use for explaining the use of the ontological approach to situation awareness. In Section 6, we show how situations related to the example are represented in the ontology. Section 7 presents examples of how such an ontology can be used for developing situation awareness systems. Finally, Section 8 discusses conclusions as well as some issues that require further research.

2. Situation and situation awareness

Although the notion of "situation awareness" is part of the data fusion lexicon, cf. [27], this term has been used with a number of different meanings. In this section, we identify and discuss some of the most common interpretations of this concept and relate them both to the JDL Data Fusion Model [28] and to the model of Endsley [3]. We use Fig. 1 to support this discussion.

Fig. 1 shows four planes, each referring to a different level of abstraction. The bottom layer shows the World, i.e., the physical (or abstract) world that is the subject of some inquiry. Although this figure suggests that the World is associated with a geographical region, it actually does not have to be so. It is just a symbolic depiction of the things that give rise to a situation. These can be either physical or conceptual things, or both.

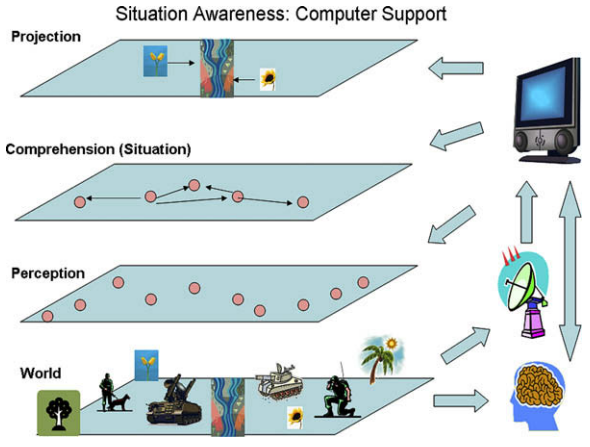

Fig. 1. Situations and perception.

To the right of the World plane, a human head depicts the fact that situation awareness actually takes place in the human's brain. The human observes some aspects of the World, and the human gets inputs from the computer, as shown in the figure.

The next layer is denoted as "Perception." The dots on this plane represent objects from the World that are observed through sensors and represented in computer memory. The arrow from the World plane toward the radar icon represents the sensory process, which then feeds the computer, which in turn generates the object representations. The label "Perception" represents the fact that this kind of representation is compatible with the output of the Perception process in Endsley's model [3]. In some discussions of situation awareness this kind of representation is considered to be the "situation," i.e., some people consider the situation to be the knowledge of all the objects in a specific area, and possibly their kinematic states.

This is not how this term is defined in dictionaries. For instance, Webster Dictionary [29] defines "situation" as

> 1 a: the way in which something is placed in relation to its surroundings.

Thus the emphasis in the dictionary definition is on relationships. The relations are viewed from the point of view of a thing, and they capture how other things in the surroundings of that thing are related to it; the thing is the focal object of the situation.

The JDL model [28] also recognizes the role of relations as the basic feature of situations:

> Situation Assessment: estimation and prediction of relations among entities, to include force structure and cross force relations, communications and perceptual influences, physical context, etc.

In this paper, we consider this kind of situation. In Fig. 1, this kind of notion of situation is represented by the plane labeled as "Comprehension." The lines that connect some of the points represent the relations. Again, this is just a view that symbolizes relations. Although the figure shows only lines connecting pairs of points, i.e., only binary relations, in fact relations can relate more than two objects. Moreover, the same set of objects can be related by many different relations. The label "Comprehension" indicates that this representation maps to the Comprehension part in the Endsley's model of situation awareness [3].

Note, however, that although the JDL model captures the role that relations play in the definition of "situation," it misses the essence of "awareness" in its formulation. For instance, the term "aware" provided in Webster's Dictionary [29] is explained as:

awareness implies vigilance in observing or alertness in drawing inferences from what one experiences.

In other words, a subject is aware if the subject is capable not only of observing some objects (experiences) but also of drawing conclusions (inferences) from these observations. The need for inference comes from the fact that not all information comes explicitly through experience. This is particularly true for relations. While it is typical that information about objects (or at least their properties) can be experienced, or observed directly, the relational information must be inferred. This aspect of awareness seems to be part (although not explicitly) of "comprehension" as defined by Endsley [3].

The top layer of Fig. 1 shows the plane labeled "Projection." This layer has a direct relationship with Endsley's model in which projection is defined as the capability of anticipating future events and their implications.

The importance of relations and inference for situation awareness can be easily observed in various scenarios in which humans can be said to be aware (or not). For instance, consider a scenario of watching a game, like American football or baseball, by someone who has never learned the rules and the strategies of these games. Although the person can clearly see where each player is and where the ball is, the person still has no idea of "what is going on" and thus cannot claim to be "aware" of the situation of the game being watched. The main part of being aware is to be able to answer the question of "what's going on?" As this example shows, in order to be able to do so, one needs to have data pertinent to the objects of interest, some background knowledge that allows one to interpret the collected object data and finally a capability for drawing inferences.

The essence of experienced vs. inferred information can also be expressed in formal terms. In mathematics, a relation is a subset of the Cartesian product of a number of sets. For instance, the Cartesian product of two sets A and B is the set of all ordered pairs $\langle a, b\rangle$, i.e.,

$$A \times B = \{\langle a, b\rangle | a \in A, b \in B\}$$

A relation R is then a subset of the Cartesian product, $R \subseteq A \times B$. A relation can be given (specified) either extensionally or intensionally. An extensional specification of a relation is given by explicitly listing all the tuples of the relation. An intensional specification of a relation, R, is given through a predicate, P. In that case R contains all those tuples r for which the predicate P is true. It is formally written as:

$$R = \{\langle a, b\rangle | P(a, b)\}$$

Now the question is how we know for a given pair $\langle a, b\rangle$ that the predicate P is true. This is where the power of inference comes to bear. Predicates are the main component of *sentences*, which in turn are part of *logical theories*. Relations, on the other hand, are part of *models*, i.e., interpretations of sentences. The process of inferencing, or reasoning, is carried out within a specific theory. A computer-based reasoning process is purely syntactic, i.e., an inference engine manipulates "facts" that are stored as strings. It matches "inference rules" to patterns in its current fact base and "derives" new facts according to the inference rule whose pattern has been matched. To make this possible, one must have:

1. A formal language in which all the facts used in the reasoning process are expressed, and
2. A formal specification of the reasoning process.

A formal language is given by a grammar and a notion of interpretation. A *grammar* is given by a number of rules for constructing compound sentences out of elementary sentences. An *interpretation* is a function that maps all of the elements of the formal language to a relational structure, called a *model*. In particular, an interpretation maps each predicate to a relation. A sentence is said to be true for an interpretation if its corresponding relation holds in the model. For a set of sentences A, a model of A is any such relational structure for which every sentence in A holds.

The specification of the reasoning process is given by the notion of *entailment*. A set of sentences A *entails* a sentence s if and only if for every interpretation of A, whenever all sentences of A are true, the sentence s is also true. In the context of situation awareness we will use the term "entailment" to indicate the process in which a sentence is determined to be entailed by some set of sentences.

In addition to reasoning about relations, situation awareness involves the use of the concept of situation in real life. While a situation can be defined as a set of relations with other objects, both the objects and the relations change with both time and location. For instance, one is in different situations when one is driving home and when one is hiking in the mountains. To make use of situation awareness, especially for decision making, one must be able to recognize situations, assess their impact on one's goals, memorize situations, associate various properties with particular situations, and communicate descriptions of situations to others. This leads to two additional requirements with respect to the representations of situations:

1. Situations can be classified by Situation Types, and
2. Situations can be treated as objects, like physical objects or conceptual objects.

These requirements support the idea of modeling situations as typed objects within the object-oriented paradigm.

A number of philosophers and logicians introduced concepts similar to that of a situation, including von Mises [30] in 1949 and Bunge [31] in the 1970s. However, the earliest formal notion of situation (although not situation awareness) was introduced by Barwise and Perry as a means of giving a more realistic formal semantics for speech acts than what was then available [20–22]. In contrast with a

"world" which determines the value of every proposition, a situation corresponds to the limited parts of reality that we perceive, reason about, and live in. As Barwise explains [22]:

> One of the starting points for situation semantics was the promotion of real situations from second class citizens to first class citizens. By a situation, then, we mean a part of reality that can be comprehended as a whole in its own right – one that interacts with other things. By interacting with other things we mean that they have properties or relate to other things.

While Barwise's situation semantics is only one of the many alternative semantic frameworks currently available, its basic themes have been incorporated into most of the others.

3. Barwise's situation semantics

We now present a formalization of Barwise's situation semantics in terms of an ontology, with some parts using mathematics and rules. We call the resulting ontology the Situation Theory Ontology (STO). Most of our interpretation of the meaning of situation semantics is based upon Devlin's book [23] and an unpublished paper [32]. Devlin formalizes a number of concepts developed by Barwise and Perry, subsequently extended by Devlin; we will refer to these concepts as *situation theory*, or the *situation-theoretic framework*.

While we have attempted to be fully faithful to situation theory, it is not possible to give a completely rigorous formalization. As Devlin explained, "Although described as a 'theory,' situation theory is more profitably approached as a set of mathematically-based tools ..." Accordingly, our mapping can only be a correspondence, not a formal equivalence.

Barwise and Perry began with the assumption that "people use language in limited parts of the world to talk about (i.e., exchange information about) other limited parts of the world. They call those limited parts of the world *situations*. Events and episodes are situations in time, scenes are visually perceived situations, changes are sequences of situations, and facts are situations enriched (or polluted) by language." Devlin stresses that "the appearance of the word *parts* in the above quotation is significant. Situations are *parts* of the world and the information an agent has about a given situation at any moment will be just a *part* of all the information that is theoretically available. The emphasis on partiality contrasts situation semantics from what was regarded by many as its principal competitor as a semantic theory, possible worlds semantics.

3.1. Basic notions and relationships

In situation theory, information about a situation is expressed in terms of *infons*. Infons are written as

$$\ll R, a_1, \ldots, a_n, 0/1 \gg$$

where R is an n-place relation and $a_1, \ldots, a_n$ are *objects* appropriate for R. Since situation theory is multi-sorted, the word "appropriate" means that the objects are of the types appropriate for a given relation. The last item in an infon is the *polarity* of the infon. Its value is either 1 (if the objects stand in the relation R) or 0 (if the objects don't stand in the relation R). Devlin states that "infons are not things that in themselves are true or false. Rather a particular item of information may be true or false about a situation." Infons may be recursively combined to form *compound infons* by using conjunction, disjunction and situation-bounded quantification. Devlin does not have a term for infons that are not compound. We say that such infons are *elementary*.

To capture the semantics of situations, situation theory provides a relation between situations and infons. This relationship is called the *supports* relationship which relates a situation with the infons that "are made factual" by the situation. Given an infon σ and situation s the *proposition* "s supports σ" is written as

$$s \models \sigma.$$

The relation between a situation (in the world) and a representation of the situation (in a formal framework) is relative to a specific agent. In situation theory, it is the agent who establishes such a link. This link is defined by *connections* that link entities in the world to formal constructs of the situation-theoretic framework. These connections are not part of the formal theory. Thus they cannot be part of any formal theory such as a situation awareness ontology. One refers to situations within a formal theory by using *abstract situations*, although the qualifier "abstract" is usually dropped in most discussions of situation theory, and we will generally do so as well in the following discussion.

3.2. Supports vs. derives

Before we proceed further with our formalization of situation theory, we need to provide some clarification regarding our formalization vs. Devlin's. Since our formalization makes use of OWL, whose semantics is specified in the classical model-theoretic way, our formalization can be claimed to *resemble* situation theory rather than faithfully implement it. To clarify the relationship between the two forms of semantics, we need to discuss the relation between the notion of "supports" in situation theory and "models" in model theory. A possible confusion about these two terms comes from the fact that both approaches use the same symbol, $\models$, to describe two different concepts.

In the classical model-theoretic semantics the symbol, $\models$, stands for "satisfies." In other words, the meaning of this symbol is that a given relational structure, called the *model*, satisfies a set of sentences. A set of sentences A is said to *entail* a set of sentences B if any model of A contains the

model of B. A relation parallel to the entailment relation is called *derives*, or *logically implies*. While entailment is defined on both sentences and models, logical implication is defined only on sentences (i.e., within *theories*). Derivation is defined in terms of *inference rules*. A set of sentences A *derives* a set of sentences B, written as $A \vdash B$, if each sentence in B can result in "true" by the application of the inference rules and other sentences from A. The *models* and the *derives* concepts are related, i.e., a derivation system needs to be *sound*, meaning that only entailed sentences can be derived. A desirable feature for a logical system is *completeness*. For a complete system, all entailed sentences can be derived. If a set of inference rules is both sound and complete, then entailment and derivation coincide.

In situation theory, the symbol, $\models$, stands for *supports*. This is a softer requirement than "satisfies;" it admits some incompleteness of the relational structure. Situations in the world play the role of models (relational structures). Infons, on the other hand, are sentences, i.e., they are part of logical theories. Relations among infons can thus be modeled using the "derives" relation. The model-theoretic meaning for "derives" is provided by the "satisfies" relation in the usual model-theoretic sense.

Our interpretation of situation theory can be discussed using Fig. 2. The figure shows an oval representing the World and a rectangle representing an Agent. The Agent is connected with the World; this fact is captured by the arrows annotated with the label *supports*. The Agent represents information about the World in terms of infons. The figure indicates that at a given time t_1 and location l_1 a situation s takes place in the World. This is captured by the infon

$$\ll \texttt{type}, \texttt{s}, \texttt{Situation}, \texttt{l}_1, \texttt{t}_1 \gg$$

We used here the term "type" that in OWL has the meaning of "instance of" and the term "Situation" to indicate that this is a class of situations.

Since our Agent is a logical agent, it uses formal logic to derive facts about situations. In this particular case, the Agent *derives* that the relation r holds using logical inference. This fact is expressed by the *entails* relationship between two infons – from the fact that an infon about situation s holds, another infon can be entailed, i.e., one that captures the fact that in this situation, a relation r among objects o_1 and o_2 holds. The formalization of entailment is given by the standard model-theoretic semantics.

Therefore, in our framework, we capture the basic concepts of situation theory, i.e., the "supports" relation, the partiality of the knowledge of real situations (it is a feature or imperfection of the Agent), the inference of facts from situations, and the fact that particular facts hold in a situation.

We next discuss the basic components of situation theory. This will be followed by a discussion of the situation theory ontology.

3.3. Objects and types

The basic elements of situation theory are *objects* (also called *uniformities*) and *types* [23]. We start the presentation of our formalization of STO by showing the top-level structure of STO. In particular, we first explain how *objects* and *types* are interrelated within STO. In Fig. 3, we show the pattern that is used throughout the construction of STO. The main idea is that the ontology has two *meta-levels*. The class `TYP` is the top-level class representing *types*. It has a number of subclasses (subtypes) as described in the next section. In Fig. 3, we show only one subtype, `IND`. Instances of this class are classes. In this case, `Individual` is the class that is an instance of `IND`. In the figure, the relation of "instance of" is represented by an arrow and a label, `io`. The class `RelevantIndividual` is a subclass of `Individual`. The "subclass" relation is represented by an arrow with a label, `isa`. In STO, each kind of object has two associated classes – the type of object class (analogous to `IND`) and a class that collects instances of a given type (analogous to `Individual`).

For instance, consider the class called `Dog`. Instances of this class are representations of particular dogs. The class `Dog` is an instance of `IND`. We can also have subclasses of `Dog` such as `YellowLab` which contains descriptions of various dogs of this breed. The class `YellowLab` is an instance of `IND`.

By virtue of the fact that the STO is expressed in OWL, one can construct classes using the OWL class constructors. These constructors are sufficient for the most common forms of type construction in situation theory. When the

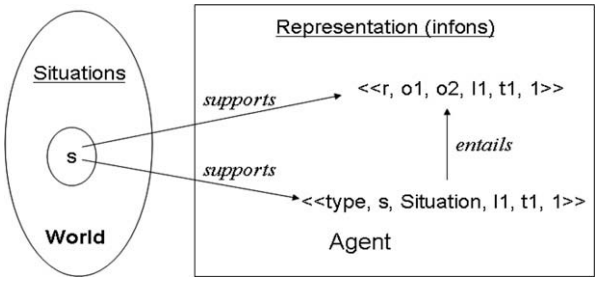

Fig. 2. Agent's connection to the World.

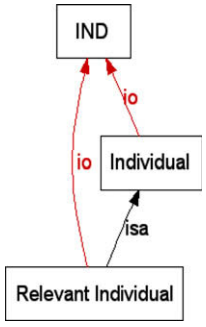

Fig. 3. A fragment of STO showing two (meta)levels of representation: `IND` (type); `Individual` and `RelevantIndividual` (objects).

OWL class constructors are insufficient for constructing a situation-theoretic type, one can use rules. Both the OWL constructs and the rules have fully specified and computer-interpretable semantics.

3.4. Basic types

Now we discuss all the types of objects of situation semantics.

TYP – the class of types, as described in Section 3.3.

IND – the type of individuals. The corresponding class, `Individual`, is an instance of **IND**. These are entities perceived by an agent using its *connections* (as explained earlier).

In situation theory individuals are usually denoted as *a, b, c,...* In our ontology we provide a class called `Individual`. So particular individuals *a, b, c,...* are instances of this class. Examples of individuals are Rex (an instance of the class `Dog`), and Fluffy (an instance of the class `Cat`). In STO, these facts would be expressed in OWL. In mathematical notation, the facts about Rex and Fluffy would be presented by the unary predicate expressions `Dog (Rex)` and `Cat (Fluffy)`

REL^n – the type of n-place relations. STO represents these using the `Relation` class. In situation theory, the first position of an infon specifies the relation of the infon. STO has a property `relation` from `ElementaryInfon` to `Relation` that serves the role of the first position of an infon. Elementary infons must satisfy the constraint of compatibility of arguments (see the discussion of parameters below).

Relations, denoted as $P, Q, R, \ldots$ are described by their names and the types of the objects allowed in particular attribute places. An example of a relation is `chases`. An example of a tuple of this relation is <Rex, Fluffy>. In STO, the tuple would be represented using a binary predicate, `chases (Rex, Fluffy)`. The fact that this tuple is a member of the `chases` relation would be represented as `relation (chases (Rex, Fluffy)) = chases`.

ATTR – the type of attributes. Situation theory provides types specifically devoted to capturing locations (**LOC**) and time instants (**TIM**). In STO, these two types are subtypes of **ATTR**. To capture instances of locations and time instants, STO provides classes `Location` and `Time`, respectively. For other attributes STO provides the class `Attribute`, which is a superclass of both `Location` and `Time`. OWL has a rich collection of time notions as a result of its support for the XML Schema datatypes.

In situation theory, infons may include information about locations and time of occurrence of a particular situation. For instance, the situation in which the dog, Rex, is chasing the cat, Fluffy, can be expressed by the proposition

$$s \models\!\ll \texttt{chases}, \texttt{Rex}, \texttt{Fluffy}, \texttt{l}_1, \texttt{t}_1, 1 \gg$$

The location of the situation is l_1 and the time is t_1.

VAL – the type of values. Situation theory does not have such a type. In situation awareness, however, one needs to speak of such values as 5 m/s or 30 km. In order to be able to model this kind of thing, STO provides the **VAL** type and the `Value` class. We model the type **POL** – polarity – of situation semantics as a subtype of **VAL**. Polarity represents the truth value of an infon; it can be either 0 or 1. In STO, polarity is represented using the `Boolean` data type of XML Schema.

DIM – the type of dimensions. This type captures dimensions (information about the systems of units) in terms of which particular values are expressed. Although situation theory does not provide such a type, it is a very important type for modeling various physical and other phenomena. Instances of this type are elements of the class `Dimensionality`. Examples of such instances are [m/s] and [km].

SIT – the type of situations. This type corresponds to the `Situation` class that has already been discussed. The semantics of the `Situation` class is the same as in Barwise, i.e., an abstract situation is the set of those infons that are supported by the same concrete situation *s*. According to situation theory, the word "the same" in this sentence means "whatever the agent perceives as the same." In other words, the extent of this class is deferred to the capabilities of the agent. In our approach, the constraints placed on the `Situation` class capture our understanding of the agent's perceptual capabilities, i.e., they capture what the agent perceives as "the same situation." We provide an example of a situation later in the paper, after we introduce the notions of "utterance situation," "focal situation" and "resource situation."

Situations in situation theory involve objects and relations among the objects. In order to model them, STO has the `relevantObject` and `relevantRelation` properties. We require that a situation must have at least one relevant object and at least one relevant relation. These kinds of constraint are expressed in STO as *existential* restrictions on the `Situation` class.

PAR – the type of parameters. Situation theory uses parameters as a mechanism for constructing types. Parameters are used in infons; they serve the same role as variables in rule-based systems and languages. Because STO is expressed in OWL, it has a rich set of mechanisms for class construction that does not rely on variables. As discussed above, these mechanisms are sufficient for most class constructions. When these mechanisms are not adequate, one can specify a class using rules. Rules use variables in much the same way that situation theory uses parameters.

Situation theory provides the notion of *restricted parameters* to allow one to restrict the ranges of parameters used in an infon. STO has properties `parlType`, `par2Type`, ..., whose domain is `Relation` and the range is a (restricted) type, that allow one to restrict the slots of a relation to specified classes.

Parameters of situation theory are variables which can be set to specific object values. The setting of one or more parameters to specific objects is done by means of an

anchor which maps the variables to the desired values. We will say that an infon is *anchored* when all of its parameters have been anchored to objects. STO uses the properties `anchor1`, `anchor2`,..., whose domain is `ElementaryInfon` and whose range is `Object`, to specify the object values in the slots of an instance of `ElementaryInfon`. The instances of `ElementaryInfon` that can be part of a given relation will be restricted exactly as in situation theory, i.e., the type of the value to which a triple is anchored to must be exactly the same as the type of the slot of the relation that the tuple is part of. We require that both `par<n>Type` and `anchor<n>` properties be functional.

In situation theory the parameters representing individuals, situations, location or time are denoted by $\dot{a}, \dot{s}, \dot{l}, \dot{t}$, respectively. They can be of any type of object that is part of situation theory. An example of a parameter could be `FourLeggedAnimal`. In that case the relation `chases` could be restricted to consider only four-legged animals that chase each other. In STO, a class `FourLeggedAnimal` would be constructed as a subclass of `Individual` and restricted to those instances of `Individual` for which the property `numberOfLegs` has the value 4.

INF – the type of infons. In situation theory, this includes both elementary and compound infons. We introduce a class `ElementaryInfon` for elementary infons, and we use OWL class constructors and rules to deal with compound infons. For example, consider the general form of an elementary infon:

$$\ll R, a_1, \ldots, a_n, 0/1 \gg$$

We can gain an understanding of what is possible to represent in STO by considering all possible fillers for particular slots in the above representation of an infon.

The first slot, R, can be filled with a representation of a relation. In STO, this is an instance of the class `Relation`. Since STO is expressed in OWL, any OWL property can also fill this slot. Such a property is always binary as an OWL property, but in STO, it can have additional slots, such as the time when the property holds for two individuals. We have already seen an OWL property, `type`, in this role.

The slots $a_1, \ldots, a_n$ can be filled with: individuals, relations, location (spatial and temporal), situations, and types of all of the above. Compound infons can be expressed using OWL expressions as well as rules. Examples of these will be discussed later.

3.5. Reasoning within situation semantics theory

In situation theory, an agent reasons by applying knowledge that is expressed with *constraints*. Constraints link situation types. A constraint that situation type S_1 is linked to another situation type S_0 is written as

$$S_0 \Rightarrow S_1,$$

and one says that S_0 *involves* S_1. The meaning of such a constraint is that whenever a situation s_0 is an instance of S_0, there is a situation s_1 that is an instance of S_1. When s_1 is the same situation as s_0, i.e., $s_1 = s_0$, Devlin refers to the involvement as being *reflexive*. Constraints play the role of laws of the world, e.g., physical laws. In the STO, the reflexive constraints correspond to *subclass* relationships among situation types. In other words, to express that S_0 reflexively involves S_1 one specifies that S_0 is a *subclass* of S_1. A subclass relationship is a special kind of rule which specifies that if a situation belongs to one situation type, then it also belongs to another one. Reasoning using such rules is called *subsumption*, and it is the basis for description logic [33], which is the underlying logic for OWL. While subclass relationships are a "built-in" feature of OWL, there is no relationship in OWL corresponding to non-reflexive involvement, although either kind of involvement can be expressed using rules.

4. The situation theory ontology

Now we show how situation theory is formalized as an ontology; we call it the *Situation Theory Ontology* (STO). A graphical representation of STO is shown in Figs. 4 and 5. Since STO is a relatively complex ontology, in these figures we show only some of the classes in partial views.

The ontology is visually represented using a Protégé plugin called OntoVIZ. The boxes in this notation represent classes. A class is interpreted as a set of instances that satisfy all the constraints and restrictions associated with the class. The rectangles show class names. Arrows represent properties. Names of properties appear as labels on the arrows. In OWL, properties are binary relations. The class at the tail of an arrow is the domain of the relation and the class at the head of the class is the range of the relation. The complete ontology is available at `http://vistology.com/ont/2006/STO/STO.owl`.

`Situation` is the central class. Instances of this class are specific situations. This class is a direct counterpart of the *abstract situation* concept in situation theory. The second class is the `Individual` class, which is a counterpart of the individuals in situation theory. Similarly, `Relation` captures the *n*-ary relations. In order to provide a means for inferring relations we introduce the class `Rule`. Instances of this class capture axioms of the domain that can be used for inferring whether a given relation holds in a situation or not. `Attribute` is a generalization of locations and time instants in situation theory. Instances of this class are attributes of individuals and situations. An attribute may have a dimension associated with it (e.g. [m/s] or [m^2]). For this purpose, we introduce the class `Dimensionality`. We also introduce the class `Polarity`. This class has only two instances that correspond to the two possible values associated with a tuple, either that a given tuple holds or that it does not hold. In situation theory these polarity values are denoted as '1' and '0'. The fact that polarity is a special case of Value is specified

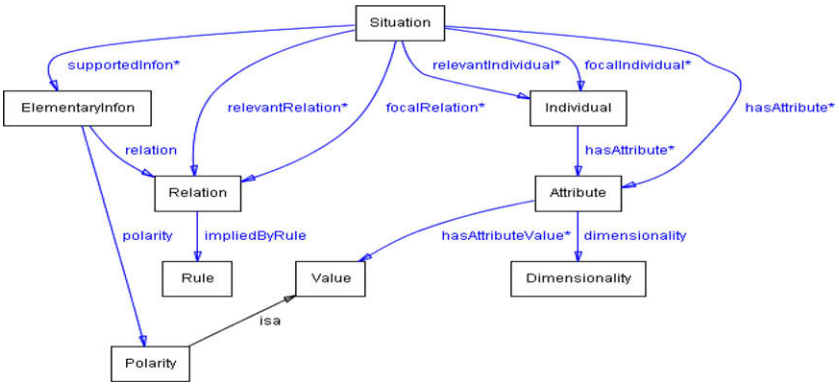

Fig. 4. Main classes and properties of STO.

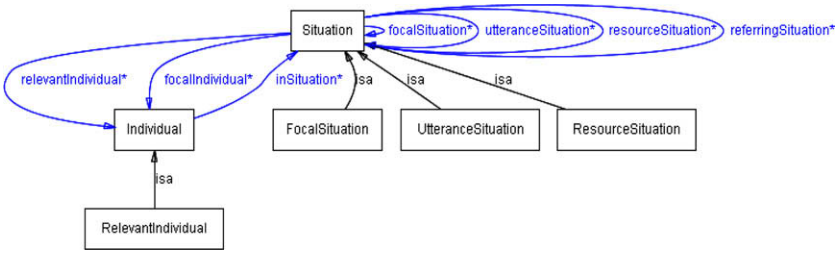

Fig. 5. Situation types (partial view).

in OWL using the `subClassOf` property. In the OntoVIZ notation this is shown by the `isa` label.

Classes of STO are related through a number of OWL properties. Situations are linked with four kinds of entities. First, the property `relevantIndividual` captures the individuals that participate in a situation. The property `relevantRelation` is used to assert that a given kind of relation is relevant to a given situation. Since situations are objects, they can have attributes of their own. Attributes of situations are captured by the `hasAttribute` property. The domain of this property also includes `Individual`. Attributes have `dimensionality` as well as `hasAttributeValue` properties.

4.1. Subclasses of situation

Situation semantics is a study concerned with utterances. In situation semantics, three kinds of situations are distinguished: *utterance situation*, *resource situation* and *focal situation*. Consequently, in our ontology, we provide three subclasses of the class `Situation`: `UtteranceSituation`, `ResourceSituation` and `FocalSituation` (see Fig. 5). These are provided primarily to show that the STO is rich enough to express Barwise's situation theory. If ones primary concern is not with utterances, then one can introduce other subclasses of `Situation`.

In situation theory, the meaning is acquired through the speaker's connections (links) to a situation. To be compatible with Devlin, these links are captured in the STO by means of the `focalIndividual` and `focalRelation` properties.

The most important type of situation to Devlin is the *utterance situation*. "This is the context in which the utterance is made and received." In situation theory, an utterance is represented as an expression, Φ, in some language, e.g., in natural language. In order to acquire the meaning, this expression needs to be linked to a real situation, s, and represented as a (typically compound) infon. The links between Φ, s and the infon are in the so called *speaker's connections*.

In our formalization of situation theory we interpret utterances as queries which come from the user of a formal situation awareness system. Queries thus partially provide the "speaker's connection". In our SAW Core Ontology [4] we used to call this class `Goal`. It is like a perspective that gives focus to what should be considered as *relevant* for a specific situation.

In some cases, an utterance refers to another situation, i.e., another situation is used in a support role. The kind of situation that an utterance situation is referring to is called a *resource situation*. It is a situation that is used as a kind of background for reasoning with the current situation. And finally, a *focal situation* (or *described situation*) is that part of the world that is relevant to a given utterance. Because our emphasis is on understanding a situation based on sensory data, the focal situation is the most important type of situation. As Devlin explains, "Also known as the described situation, the focal situation is that

part of the world the utterance is about. Features of the utterance situation serve to identify the focal situation.''

We introduce two pairs of properties that are inverses of each other (inverses are denoted by '↔') whose domain and range is `Situation`:

- The properties `focalSituation` ↔ `utteranceSituation` are used to link instances of `UtteranceSituation` with instances of `FocalSituation`.
- The properties `resourceSituation` ↔ `referringSituation` are used to link instances of `UtteranceSituation` with instances of `ResourceSituation`.

Moreover, we introduce the `relevantIndividual` property whose domain is `Situation` and range is `Individual` along with its inverse `inSituation`. These properties capture links between situations and individuals that are part of the situation in one direction, and individuals that are in a situation, respectively. `focalIndividual` is a subproperty of `relevantIndividual`.

In Fig. 5, we show the three situation types, and a partial view of the properties of STO that are related to the three situation types as well as to the notion of *relevant individual*. An instance of `UtteranceSituation` is required to have at least one `focalIndividual` (i.e., it must have a value for property `focalIndividual`) and at least one `focalSituation`. An instance of `FocalSituation` is required to be associated with an `UtteranceSituation`. A situation is qualified as a `ResourceSituation` only if it is related to another situation by the `referringSituation` property.

We interpret Devlin's discussion about focal situations to be that `focalIndividual` is a subproperty of `relevantIndividual`; in other words, all focal individuals of a situation are also relevant. Our intentions go one step further – in applications of the ontology based approach to situation awareness we would like to be able to derive all of the other individuals that are relevant to a situation. This derivation is based on the knowledge of focal individuals, as well as other knowledge that can be extracted either from a query or from background knowledge.

5. Example

In this section, we present some examples of situations. First, we present their descriptions in natural language. In our discussion we will refer to Fig. 6, a World that gives rise to a number of situations. This figure shows some objects involved in these situations. One of these is that the dog, Rex, is chasing the cat, Fluffy. Jim is the owner of Fluffy and thus is involved in watching the chase due to his concern about Fluffy's well being, if chasing means that Rex is threatening Fluffy, but not if Rex is just playing with Fluffy. There are also two other objects, a mouse (Mickey) and a flower (we call it Tulip1), which are not involved in the chasing or watching situations but which might be involved in some other situations. Nevertheless, the chasing situation is of interest to Mickey, since it knows that it is safe from the cat for as long as the cat is being threatened by a dog. Now we list some of the situations related to the World shown in Fig. 6 that are of interest to us in this paper.

RexThreatenFluffy In this situation, Rex chases Fluffy. Relevant relations include `chases` and `threaten`. Both are binary relations. The first parameter represents the chaser, or threatener, and the second the one being chased, or the victim. Attributes of this situation may include the

Fig. 6. An example of a World.

location of the situation and the time associated with the occurrence of the chase. Objects may have attributes of their own, like location or velocity. The situation may include an attribute of 'rate of approach', i.e., the difference between the velocities of the pursuer and the pursued.

RexPlayingWithFluffy This is similar to the previous situation except that `playWith` rather than `threaten` is relevant. One can distinguish this situation from the previous only if one knows whether the chaser is friendly toward the other animal.

JimWatching Jim, the owner of Fluffy, is involved in this situation. He is a focal object of this situation and so is Fluffy. The relevant relations include the `watch` relation in which the watcher is the first argument and the object being watched is the second. Rex is in this situation too, although his belonging to this situation comes through the fact that Jim is interested in the RexThreatenFluffy situation because Fluffy is involved in that situation. If another dog were threatening Fluffy, that dog would be in this situation. The RexThreatenFluffy situation plays the role of a resource situation for the JimWatching situation.

SafeFromCat From the point of view of Mickey (the mouse), safety from a cat is one of the main concerns. The relevant relation is `safeFromCat` with the first argument being the subject (in this case Mickey) and second argument being the object (in this case Fluffy). The RexThreatenFluffy situation again plays the role of a resource situation. Since Fluffy is preoccupied with the possibility of being attacked, Mickey is safe from the cat.

6. Representation of situations in STO

In this section, we describe how the situations discussed in Section 5 are captured in a formal language.

First, in order to distinguish general terminology for the STO from the specific terminology of the examples, we introduce two *namespaces*. The `sto:` prefix will be used for distinguishing terminology in the STO namespace (i.e., `http://vistology.com/ont/2006/STO/STO.owl`). The `rex:` prefix will be used for terminology specific to the examples (i.e., `http://vistology.com/ont/2006/STO/Rex.owl`).

Second, we extend STO to accommodate the specifics of the world and of the situations. Towards this aim, a number of classes and relations are introduced. The classes include `Person`, `Cat`, `Dog`, `Mouse`, and `Flower`. These classes are subclasses of `Individual`.

Third, we assert a number of facts about the world (shown in Fig. 6) in terms of the extended ontology. In particular, we assert that `Jim`, `Fluffy`, `Rex`, `Mickey`, and `Tulipl` be instances of `Person`, `Cat`, `Dog`, `Mouse`, and `Flower`, respectively. Each of these individuals have attributes of `Location`. In our example `Jim` has location `L_Jim`, `Fluffy` has location `L_Fluffy` and so on. Each of the attributes has a `dimensionality`. All attributes of type `Location` in our representation have dimensionality `[m]` (for simplicity), and all velocity attributes have dimensionality `[mps]`. Additionally, we introduce relations `watch`, `chases`, and `safeFromCat` with arguments of appropriate types. Using the usual mathematical notation, the arguments of these relations are as in `watch(Person, Cat)`, `chases(Dog,Cat)`, and `safeFromCat(Mouse, Cat)`. All these relations are defined as instances of `Relation`.

Having done this, we can now discuss various situations that can be considered in this world. Each of the situations starts with an utterance situation. Since we are not going to deal with natural language processing in this paper, and moreover, some of the agents in this world don't use natural language at all, we will focus on the utterance situations rather than the utterances that give rise to them.

6.1. Representation of 'RexThreatenFluffy'

In this situation, Rex and Fluffy are both moving at high speed in the apartment. This is shown by the location and velocity infons. It is also known that Rex is not friendly toward Fluffy. These are called the *observed* infons because they are determined by sensors or some other source of information. They are also called *asserted* infons because they must be explicitly given to the rule engine. The other infons that are supported by the situation are derived from the observed infons or from other derived infons. For example, one can use the locations and velocities of Rex and Fluffy to conclude that Rex is moving in the direction of Fluffy. Similarly, one can infer that Rex is near Fluffy. The fact that Rex is chasing Fluffy is derived from other derived infons, and the fact that Rex is threatening Fluffy is derived from both a derived infon and an observed infon. Mathematically the last three infons just mentioned are the relation tuples `near(Rex,Fluffy)`, `chases(Rex, Fluffy)` and `threaten(Rex,Fluffy)`. The infons mentioned above are written as follows in the notation of situation theory:

$$\texttt{RexThreatenFluffy} \models \ll \texttt{location,Rex,L\_Rex,1} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{velocity,Rex,V\_Rex,1} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{location,Fluffy,L\_Fluffy,1} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{velocity,Fluffy,V\_Fluffy,1} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{near,Rex,Fluffy,1} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{chases,Rex,Fluffy,1} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{isFriendlyToward,Rex,Fluffy,0} \gg$$
$$\texttt{RexThreatenFluffy} \models \ll \texttt{threaten,Rex,Fluffy,1} \gg$$

To illustrate what these infons would look like when written in the OWL Abstract Notation, we show the last two infons as follows:

```
Individual(sto:RexThreatenFluffy
    type(sto:ResourceSituation)
    value(sto:supportedInfon Individual(_ type(sto:Ele-
    mentaryInfon)
```

value(sto:anchor1 rex:Rex) value(sto:relation rex:isFriendlyToward)
value(sto:anchor2 rex:Fluffy) value(sto:polarity sto:_0)))
value(sto:supportedInfon Individual(_ type(sto:ElementaryInfon)
value(sto:anchor1 rex:Rex) value(sto:relation rex:threaten)
value(sto:anchor2 rex:Fluffy) value(sto:polarity sto:_1))))

The main difference between situation theory notation and OWL notation is that the implicit features in situation theory notation are explicit in OWL notation. For example, the double angle brackets in situation theory are explicitly labeled with `type(sto:ElementaryInfon)` in OWL notation. Similarly, the meaning of one of the slots in situation theory notation is implicitly determined by its position in the infon, while in OWL notation the purpose of each slot is explicitly labeled, and the slots may appear in any order. For example, the first slot of an infon in situation theory notation is the name of the relation, while in the OWL notation it is explicitly labeled as in `value(sto:relation sto:threaten)` and does not have to occur first in the infon.

Yet another representation of these infons is shown in Fig. 7 using OntoVIZ. Because infons have no explicit labels, Protégé introduced labels for them (i.e., `@_:A21` and `@_:A26`).

6.2. Representation of 'RexPlayingWithFluffy'

The Playing situation is nearly the same as the Threaten situation, except that Rex is friendly toward Fluffy, and thus the inferences are different. Relevant derived infons in the situation include that Rex is near Fluffy, Rex is chasing Fluffy and Rex is playing with Fluffy. Mathematically these three relations are `near( Rex,Fluffy)`, `chases(Rex,Fluffy)` and `playingWith(Rex,Fluffy)`. The infons for this situation are the same as the infons for the `RexThreatenFluffy` situation except for these:

$$\texttt{RexPlayingWithFluffy} \models \ll \texttt{isFriendlyToward}, \texttt{Rex}, \texttt{Fluffy}, 1 \gg$$

$$\texttt{RexPlayingWithFluffy} \models \ll \texttt{playWith}, \texttt{Rex}, \texttt{Fluffy}, 1 \gg$$

6.3. Representation of 'JimWatching'

In the `JimWatching` situation we presume that the background resource situation is the `RexThreatenFluffy` situation. Most of the infons in this situation are derived from the background resource situation. The additional infon for this situation is the following:

$$\texttt{JimWatching} \models \ll \texttt{inDanger}, \texttt{Fluffy}, 1 \gg$$

In other words, Jim concludes that his cat Fluffy is in danger.

6.4. Representation of 'SafeFromCat'

The `SafeFromCat` situation is similar to the JimWatching situation, but it is from the point of view of Mickey who comes to a different conclusion:

$$\texttt{SafeFromCat} \models \ll \texttt{safeFromCat}, \texttt{Mickey}, \texttt{Fluffy}, 1 \gg$$

In other words, Mickey concludes that it is safe from the cat.

7. Using formal representations

In this section, we discuss some of the possible uses of formal representations of situations and advantages from such an ontology-based approach.

7.1. Inferring facts about situations

One of the great advantages of having situations represented in a formal language is that facts that are not explicitly stated can be derived using an inference engine. In this section, we consider how one derives facts within a single situation, given that some other facts are known. Inference in OWL includes a form of reasoning called *subsumption reasoning* that it is based on subclass relationships. We would like to express the rule that a dog that is chasing a cat and that is also unfriendly with the cat is threatening the cat. More specifically, we would like to state this rule in the case of Rex. To express the rule using subsumption one uses classes for each of the three parts of the rule. Inference can be invoked when such classes are defined and facts about instances of the classes are known. In this particular case, it is assumed that the following classes are already defined in the ontology:

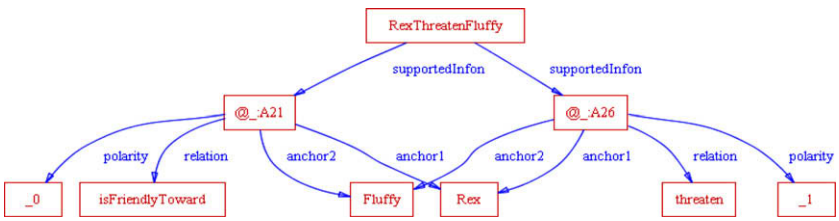

Fig. 7. Two infons of the RexThreatenFluffy situation.

```
SubClassOf(intersectionOf(sto:RexUnfriendlyCat sto:RexChasingCat) sto:RexThreatenCat)
```

Fig. 8. Subsumption rule for threatening.

1. RexUnfriendlyCat, the class of cats with which Rex is unfriendly;
2. RexChasingCat, the class of cats that Rex is chasing; and
3. RexThreatenCat, the class of cats that Rex is threatening.

The three definitions are shown at http://vistology.com/ont/2006/STO/Rex.owl.

As the definitions are similar to one another we explain just the first one. To define this class, we first define another class, called RexUnfriendlyInfon, consisting of the elementary infons that express the fact, in the situation, that Rex is unfriendly toward a cat. The class of such infons is the intersection of five classes:

1. The class of elementary infons, ElementaryInfon.
2. The class of infons supported by the RexThreatenFluffy situation. Mathematically, this is the set $\{x|\mathtt{supportedBySituation}(x, \mathtt{RexThreatenFluffy})\}$.
3. The class of infons for which the relation is isFriendlyToward, or $\{x|\mathtt{relation}\ (x, \mathtt{isFriendlyToward})\}$.
4. The class of infons for which the first slot of the isFriendlyToward relation has value Rex, or $\{x|\mathtt{anchor1}(x, \mathtt{Rex})\}$.
5. The class of infons for which the polarity is 0, or $\{x|\mathtt{polarity}(x, 0)\}$.

In Situation Theory this class of infons is written:

$$\mathtt{RexThreatenFluffy} \models \ll \mathtt{isFriendlyToward}, \mathtt{Rex}, c, 0 \gg$$

where c is a parameter that can have any cat as its value. This is expressed in OWL by leaving the second slot of the isFriendlyToward relation unconstrained in the definition of the class RexUnfriendlyInfon.

The class RexUnfriendlyCat is defined as the intersection of two classes:

1. The class of cats, Cat.
2. The class of objects which are in the second slot (i.e., anchor2) of an infon in RexUnfriendlyInfon. Mathematically, this is the set $\{x|\exists i \in$ RexUnfriendlyInfon s.t. $\mathtt{anchor2}(i) = x\}$. To express this in OWL, one must use the relation anchor2inverse which is the *inverse* of anchor2, i.e., the property for which the subject and object have been reversed. In other words, one must express the set mathematically as $\{x|\mathtt{anchor2inverse}(x) \cap \mathtt{RexUnfriendlyInfon} \neq \emptyset\}$.

Once one has defined the three classes, one can represent the rule by asserting that the third class is a subclass of the intersection of the first two as shown in Fig. 8. In other words, if a cat is both in RexUnfriendlyCat and in RexChasingCat, then this cat is in RexThreatenCat.

In the RexThreatenFluffy situation, the infon

$$\ll \mathtt{isFriendlyToward}, \mathtt{Rex}, \mathtt{Fluffy}, 0 \gg$$

was asserted. This infon can be expressed mathematically by stating that $\mathtt{Fluffy} \in \mathtt{RexUnfriendlyCat}$. In OWL/XML it is written as

$$\langle \mathtt{RexUnfriendlyCat\ rdf:about} = \text{``\#Fluffy''} / \rangle$$

The infon

$$\ll \mathtt{chases}, \mathtt{Rex}, \mathtt{Fluffy}, 1 \gg$$

was derived by rules similar to the one explained here, the result being that $\mathtt{Fluffy} \in \mathtt{RexChasingCat}$. The subsumption rule above then allows one to conclude that $\mathtt{Fluffy} \in \mathtt{RexThreatenCat}$. Stated in terms of Situation Theory, this is the infon:

$$\ll \mathtt{threaten}, \mathtt{Rex}, \mathtt{Fluffy}, 1 \gg$$

The advantage of the form of inference described in this section is that it is situation-specific. Different situations will not only have different infons, but also different inference rules. This can be very useful when customized situation-specific rules are needed. However, many rules apply to more than one situation, and so it is useful to have a mechanism for stating rules that have more general applicability. This is the subject of Section 7.3.

7.2. Inference using rules

While the above example was accomplished completely within the scope of OWL, there are many cases for which OWL is not sufficiently expressive to capture all of the desired concepts. In particular, it is not possible to construct a complex property defined as the composition of other properties. This limitation derives from OWL's lack of variables and the inability to define joins [34]. Consider for example the notion of chasing. As discussed in Section 6.1, this can be inferred from the relations near and inDirectionOf. The following simple Horn clause rule expresses this inference for arbitrary objects:

```
If
  near(X,Y) and inDirectionOf(X,Y)
Then
  chases(X,Y)
```

This rule has two advantages over the rule in Fig. 8. First, it is more general, applying to arbitrary dogs, not just

Rex. Second, it is much more succinct. To give another example of the power of rules, consider another way that chasing might be inferred. If a species preys on another species, one can infer that a particular animal of a predator species will chase a particular animal of the prey species if the former animal sees the latter animal. Using a Horn clause this can be expressed as follows:

```
If
  belongsToSpecies(X,S)
  and belongsToSpecies(Y,T)
  and preysOn(S,T) and sees(X,Y)
Then
  chases(X,Y)
```

The two Horn clause rules above have been implemented in a rule language supported by the BaseVISor inference engine [35]. BaseVISor is optimized for the processing of RDF/OWL triples and implements the semantics for RDF/RDFS and a subset of OWL called *R-Entailment* [36]. When the chase rule is submitted to BaseVISor along with the facts from the `RexChasesFluffy` situation and the additional fact that Rex "sees" Fluffy, the fact that Rex chases Fluffy is automatically derived and added to the collection of facts about the situation. In addition to deriving this fact, BaseVISor also infers other facts about the situation using the RDF/OWL semantics defined by the R-Entailment axioms; one such inferred fact is that the situation is an instance of the class `RexChasesFluffySituation` discussed in the next section.

7.3. Inferring situation types

One of the important types of inference about situations is the classification of a given situation to a situation type. This form of class definition differs from that in Section 7.1 in that it classifies situations where some behavior has occurred, rather than classifies objects within a situation.

Consider, for example, the notion of chasing. In any situation, a dog is chasing a cat if the dog is near the cat and moving in the direction of the cat. We would like to express this as a subsumption rule. As we are specifically interested in Rex and Fluffy, we will express the rule in terms of them. Mathematically, this rule can be expressed as a subset relation between classes of situations:

$$S_1 \cap S_2 \subseteq S_3$$

where

$$S_1 = \{s | s \models \ll \texttt{near}, \texttt{Rex}, \texttt{Fluffy}, 1 \gg\}$$
$$S_2 = \{s | s \models \ll \texttt{inDirectionOf}, \texttt{Rex}, \texttt{Fluffy}, 1 \gg\}$$
$$S_3 = \{s | s \models \ll \texttt{chases}, \texttt{Rex}, \texttt{Fluffy}, 1 \gg\}$$

In situation theory this relationship between the situation types would actually be written $S_1 \cap S_2 \Rightarrow S_3$, and the relationship between the situation types is referred to as reflexive involvement as discussed in Section 3.5. In the STO, the situation types are shown in `http://vistology.com/ont/2006/STO/Rex.owl`, where they are called `RexNearFluffySituation`, `RexInDirectionOfFluffySituation`, and `RexChasesFluffySituation`, respectively. Fig. 9 shows how the rule is written in OWL/XML.

7.4. Querying formal representations

After one has expressed all of the rules that apply in general for any situation and also the ones that apply for either just the particular situation or a type of situation, one can ask questions about the situation by using *queries*. The difference between queries to a database and queries to an OWL knowledge base is that the answer to a knowledge base query may include facts that are inferred as well as facts that have been explicitly asserted. We now give some examples of queries that may be used in the situations we have described above. We use the SparQL language [37] to express these queries.

Consider first how Jim would determine whether Fluffy is being threatened by Rex. The query for this is:

```
BASE     <http://example.org/jwf/>
PREFIX    sto: <http://vistology.com/onto/
STO/>
ASK
```

```
<sto:SIT>
  <owl:intersectionOf rdf:parseType="Collection">
    <sto:SIT rdf:about="#RexNearFluffySituation"/>
    <sto:SIT rdf:about="#RexInDirectionOfFluffySituation"/>
  </owl:intersectionOf>
  <rdfs:subClassOf>
    <sto:SIT rdf:about="#RexChasesFluffySituation"/>
  </rdfs:subClassOf>
</sto:SIT>
```

Fig. 9. Subsumption rule for chasing situations.

```
FROM NAMED <http://example.org/jwf/>
WHERE     {GRAPH <http://example.org/jwf/>
          {?infon sto:relation <threaten>;
            sto:anchorl ⟨Rex⟩;
            sto:anchor2 <Fluffy>;
            sto:polarity sto:_1 .
          }
        }
```

The BASE is the URI for the unqualified terms in the query. These are the ones in the specific situation to be queried. The PREFIX is the prefix for terms in the STO. The FROM NAMED clause specifies the source for the data to be queried; namely, the facts in the situation. In this case, only one situation is being considered, but queries can involve several situations at the same time. The WHERE clause specifies a graph pattern that is matched against the knowledge base. ASK signifies that the result of the query is only whether there is at least one match, not what the matches might be.

Similarly, Mickey can query whether it is safe from the cat. A somewhat more complicated query is a request for all solutions to the pattern rather than just whether there is a solution. For example, Jim might ask which animals are chasing each other in the following query:

```
BASE      <http://example.org/jwf/>
PREFIX    sto: <http://vistology.com/onto/
STO/>
SELECT    ?dog ?cat
FROM NAMED <http://example.org/jwf/>
WHERE     {GRAPH <http://example.org/jwf/>
          {?infon sto:relation <chases>;

             sto:anchorl ?dog;
             sto:anchor2 ?cat;
             sto:polarity sto:_1.

          }
       }
```

7.5. Communicating situations

Once a situation is captured in a formal language, it can be communicated to another agent and the receiving agent will interpret the situation in exactly the same way as the sending agent. In the JimWatching situation, Jim must be able to distinguish whether Rex is threatening or playing. If he has been told by the neighbor who owns Rex that Rex is friendly toward Fluffy, then Jim would not infer that Fluffy is in danger. This information could be conveyed to Jim as the infon:

```
JimWatching ⊨
            ≪ isFriendlyToward,Rex,Fluffy,1 ≫
```

In STO this could be expressed as follows as discussed in Section 7.1:

```
<RexFriendlyCat rdf:about="#Fluffy"/>
```

provided the definition of RexFriendlyCat is available in the JimWatching situation.

Alternatively, and more in the spirit of situation theory, the information could be conveyed using situation types. Namely, there was a situation in which Rex and Fluffy were previously observed to be playing, with no harm to either. From this one can infer that Rex is friendly toward Fluffy.

This may seem to be a lot of information to exchange, but if the two agents have already exchanged information about the situation and some new piece of relevant information about the situation becomes available to one of the agents, the agent can send only this new piece to the other agent with a reference to the resource representing the situation. For example, Jim could communicate with his neighbor about the current situation, and the neighbor would add useful information which would enable Jim to determine whether Fluffy is being threatened. For such a communication to be useful, Jim and his neighbor must share a common understanding about dogs, cats, and their behavior toward one another, in general, as well as about Rex and Fluffy in particular.

8. Conclusions and further research

The main contribution of this paper is to provide a computer-processable semantics for situation theory which is compatible both with the situation theory of Barwise and with Endsley's model of human situation awareness. To achieve this we expressed situation theory as a formal ontology in OWL. The advantage of an ontology based approach to situation awareness is that once facts about the world are stated in terms of the ontology, other facts can be inferred using an inference engine. This is particularly important for situation awareness since it heavily relies on the knowledge of relations. Because there are so many possible relations, it is impractical to expect that procedures could be written for all potential relations.

In this paper we introduced a formalization of the basic components of a situation awareness ontology (STO). In particular, we formalized all the basic types, classes and relations of situation theory. We have also introduced a number of new types, classes and relations. It is our hope that the STO will be used as a starting point for the information fusion community to achieve consensus on an ontology for situation awareness. We hope that STO can play the role of a basis for a unifying theory of computer-based situation awareness.

In the future we plan to give more examples of the use of the STO in practice. We are also developing tools that can make it easier for an end user to use situation theory.

However, the most important task is to develop a comprehensive situation awareness ontology through a community-wide effort so that the developed ontology can become a de facto standard ontology.

Acknowledgements

This material is based upon work supported by the United States Air Force under Contract No. FA9550-06-C-0025. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

References

[1] A.N. Steinberg, C.L. Bowman, F.E. White, Revisions to the JDL data fusion model, in: Proceedings of the SPIE. Sensor Fusion: Architectures, Algorithms and Applications, SPIE, 1999, pp. 430–441.
[2] A.N. Steinberg, C.L. Bowman, Revision to the JDL data fusion model, in: D.L. Hall, J. Llinas (Eds.), Handbook of Multisensor Data Fusion, CRC Press, 2001, pp. 2-1–2-19.
[3] M.R. Endsley, Theoretical underpinnings of situation awareness: a critical review, in: Situation Awareness Analysis and Measurement, Lawrence Erlbaum Associates, Mahawah, NJ, USA, 2000.
[4] C.J. Matheus, M.M. Kokar, K. Baclawski, A core ontology for situation awareness, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 545–552.
[5] A.-C. Boury-Brisset, Ontology-based approach for information fusion, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 522–529.
[6] T. Horney, E. Jungert, M. Folkesson. An ontology controlled data fusion process for a query language, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 530–537.
[7] K. Sycara, M. Paolucci, M. Lewis. Information discovery and fusion: semantics on the battlefield, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 538–544.
[8] E.P. Blasch, S. Plano, Ontological issues in higher levels of information fusion: User refinement of the fusion process, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 634–641.
[9] A.I. Chao, B.C. Krikeles, Lusignan A.E., E.Starczewski, An extensible, ontology-based, distributed information system architecture, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 642–649.
[10] D.L. McGuinness, Ontologies for information fusion, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 650–656.
[11] C. Nowak. On ontologies for high-level information fusion, in: Proceedings of the Sixth International Conference on Information Fusion, 2003, pp. 657–664.
[12] C.J. Matheus, K.P. Baclawski, 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, SPIE, 2003, pp. 298–309.
[13] M.M. Kokar, J. Wang. Using ontologies for recognition: an example, in: Proceedings of the Fifth International Conference on Information Fusion, 2002, pp. 1324–1343.
[14] M.M. Kokar, J. Wang, An example of using ontologies and symbolic information in automatic target recognition, in: Sensor Fusion: Architectures, Algorithms, and Applications VI, SPIE, 2002, pp. 40–50.
[15] V. Akman, M. Surav, Steps toward formalizing context, AI Magazine 17 (3) (1996) 55–72.
[16] J. McCarthy, Generality in artificial intelligence, Communications of the ACM 30 (12) (1987) 1030–1035.
[17] F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole, 2000.
[18] F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, 1984.
[19] J.F. Sowa, Conceptual Graphs. Proposal for an ISO Standard. Reference number of working document: ISO/JTC1/SC 32/WG2 N 000, 2001. <http://www.jfsowa.com/cg/cgstand.htm>.
[20] J. Barwise, Scenes and other situations, Journal of Philosophy 78 (7) (1981) 369–397.
[21] J. Barwise, J. Perry, Situations and Attitudes, MIT Press, Cambridge, MA, 1983.
[22] J. Barwise, The Situation in Logic, CSLI Lecture Notes No. 17, Stanford University, California, 1989.
[23] K. Devlin, Logic and Information, Cambridge University Press, Cambridge, UK, 1991.
[24] W3C. Semantic Web Activity, 2006. <http://www.w3.org/2001/sw/>.
[25] RDF. Resource description framework (RDF) model and syntax specification, Feburary 1999. http://www.w3.org/TR/REC-rdf-syntax>.
[26] W3C. Web Ontology Language Reference OWL, 2004. <http://www.w3.org/2004/OWL/>.
[27] Data fusion lexicon. Technical report, The Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, 1991.
[28] A.N. Steinberg, C.L. Bowman, F.E. White. Revisions to the JDL data fusion model. In The Joint NATO/IRIS Conference, 1998.
[29] Merriam-Webster Online, 2004. <http://www.m-w.com/>.
[30] L. von Mises, Human Action: A Treatise on Economics, Fox & Wilkes, 1997.
[31] M. Bunge, Treatise on Basic Philosophy. III: Ontology: The Furniture of the World, Reidel, Dodrecht, 1977.
[32] K. Devlin, Situation theory and situation semantics, in: J. Woods et al. (Eds.), Handbook of the History of Logic, vol. 7, Elsevier, 2006, pp. 601–664.
[33] F. Baader, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), The Description Logic Handbook, Cambridge University Press, 2003.
[34] I. Horrocks, F. Patel-Schneider, A proposal for an OWL Rules Language, in: Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004), ACM, 2004, pp. 723–731.
[35] C.J. Matheus, K. Baclawski, M.M. Kokar, Newblock BaseVISor: A Triples-Based Inference Engine Outfitted to Process Rule ML & R-Entailment Rules, in: RuleML-2006, Rules and Markup Languages for the Semantic Web, Second International Conference, 2006.
[36] H. ter Horst, Combining RDF and Part of OWL with rules: semantics, decidability, complexity, in: Proceedings of the Semantic Integration Workshop of the Fourth International Semantic Web Conference (ISWC-05), 2005, pp. 668–684.
[37] W3C. SPARQL Query Language for RDF. W3C Candidate Recommendation, 2006. <http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/>.