Problems and Prospects for Formally Representing and Reasoning about Enemy Courses of Action

Brian Ulicny, Christopher J. Matheus
VIStology, Inc.
Framingham, MA USA.
{bulicny,cmatheus}@vistology.com

Mieczyslaw M. Kokar
Northeastern University
and Vistology, Inc.
Boston, MA USA
mkokar@ece.neu.edu

Gerald M. Powell
U.S. Army RDECOM CERDEC
I2WD
Fort Monmouth, NJ, U.S.A
gerald.m.powell@us.army.mil

Abstract* – *Enemy Courses of Action (ECOAs) play a central role in the process of situation development in military decision-making. In order to reason about ECOAs, it would be necessary to adequately represent them in a formalism that allows for automatic reasoning. In this paper, we examine the benefits and drawbacks of representing ECOAs within several frameworks that have been encoded as OWL ontologies.

Keywords: information fusion; situation development; action; situation theory; JC3IEDM

1 Introduction

Situation development is a process that takes information about the enemy, weather and terrain, both geophysical and human, and outputs enemy courses of action (ECOAs), ranked by likelihood and level of threat.

If we intend to have computer applications that can help generate, process or reason about ECOAs, we need a formal representation for the ECOAs. In this paper, we survey various options for representing and reasoning about ECOAs formally (here, in OWL, the Web Ontology Language) and address their benefits and drawbacks.

The structure of the formal Intelligence Estimate (INTEST) provides a good context for understanding the components of situation as understood within the military intelligence process of situation development. An Intelligence Estimate consists of the following elements. Although a formal, written Intelligence Estimate is not always produced, these elements are crucial to the commander's understanding of the current situation.

I. MISSION STATEMENT.
II. AREA OF OPERATIONS.
  Weather.
  Terrain.
  Other characteristics.
  Effect on friendly/enemy COAs.
III. ENEMY SITUATION.
  Disposition, Composition, Strength
  Recent Significant Acts.
IV. ENEMY CAPABILITIES.
  Enumeration of Enemy COAs
  (what, where, when, with what strength);
  Indications of COA adoption/rejection.
V. CONCLUSIONS.
  Most probable Enemy COAs;
  Enemy Vulnerabilities.

In essence, an Intelligence Estimate presents and summarizes the reasons why it is believed that certain Enemy Courses of Action are most likely, given the preceding elements of the INTEST.

According to FM 34-130 "Intelligence Preparation of the Battlefield", each ECOA must answer five questions ([16] pp 2-44,2-45):

- WHAT - the type of operation, such as attack, defend, reinforce, or conduct retrograde.
- WHEN – the (earliest) time the action will begin.
- WHERE - the sectors, zones, axis of attack, avenues of approach, and objectives that make up the COA.
- HOW - the method by which the threat will employ his assets, such as dispositions, location of main effort, the scheme of maneuver, and how it will be supported.
- WHY - the objective or end state the threat intends to accomplish.

Thus, a fully automated situation development system would infer all of the most probable ECOAs, including all these elements, on the basis of facts about the current Mission, Area of Operations, Enemy Situation and Capabilities, as outlined above.

Typically, in maneuver-centric conventional operations, ECOAs are accompanied by a Situation Template (SITEMP) that depicts the ECOA graphically, showing how the enemy will act in the battlespace as described by the Modified Combined Obstacle Overlay. Time Phase Lines further detail how the action will proceed over time. A Situation Matrix depicts the progress of enemy activity over time across several ECOAs, and a set of Indicators specifies how this ECOA can be distinguished from others. Named Areas of Interest are identified as crucial areas to observe with respect to each ECOA.

A US Naval War College training document [3] provides the following as an example ECOA narrative:

```
ECOA 1: REDLAND initially conducts
joint operations to disrupt JTF [Joint
Task Force] Blue Sword forced entry
operations, and upon establishment of
the JTF Blue Sword in REDLAND, the
REDLAND armed forces disperse into
small-unit formations in the mountains
and cities and initiate insurgency
operations to defeat the JTF ground
forces.
```

We will focus on this ECOA as a running example. Is it possible to encode the What, When, Why, How and When of these ECOAs adequately in a formal OWL representation?[1] Would the representation support the inferences that are intuitively required?

2 ECOAs in JC3IEDM

The Multilateral Interoperability Programme (MIP) is a long-standing, NATO-supported program intended to foster international interoperability of command and control information systems through the development of standard data models and exchange mechanisms. The data model was first released in the mid-1990s as the Generic Hub (GH) Data Model. In its current form, it is called the Joint Consultation, Command and Control Information Exchange Data Model 3.1 (hereinafter, JC3IEDM) [13]. It captures information about 271 entities, 372 relationships between entities, 753 entity attributes and over 10,000 value codes.

Several projects currently envision using JC3IEDM as the basis for automatically encoding and exchanging battlespace information, such as the German Sokrates project [14], an automatic battlespace report analysis tool, and SISO's Coalition Battlefield Management Language research program (C-BML) [2].

A high-level overview of JC3IEDM is shown in Figure 1 with the main entities shaded in gray. The entities near the bottom of the diagram that focus around OBJECT-ITEM, OBJECT-TYPE and LOCATION tend to be used to represent situational awareness, i.e., what objects there are, what qualities they have, where they are located and how they are related to one another. Near the top of the diagram are entities concerned with describing ACTIONs, both planned and observed; these tend to be dynamic and are used to describe capabilities, their use and effect.

JC3IEDM treats ACTIONs as first-class entities alongside physical objects, locations, times, reports, and so on. ACTIONs are further subclassified as ACTION-EVENTS and ACTION-TASKs, the distinction being that ACTION-TASKs are known to be planned. Taking ACTIONs as primitive members of the ontology places the JC3IEDM approach within the tradition initiated by philosopher Donald Davidson [4] who argued that

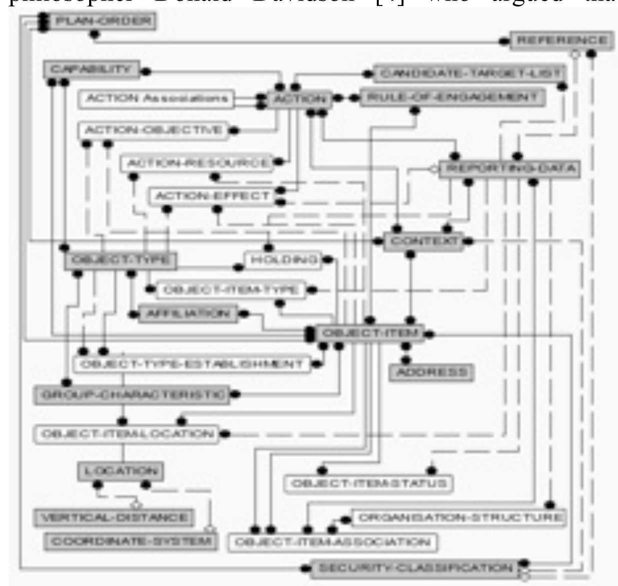

Figure 1 Basic JC3IEDM Elements

events are particulars that constitute a fundamental ontological category over which quantification is necessary for a first-order model-theoretic semantics of natural language.

In JC3IEDM, an ACTION has several possible entities that optionally further characterize it beyond its type. An ACTION has an agent (*who*) specified through an ORGANISATION-ACTION-ASSOCIATION. An ACTION- LOCATION specifies *where* the ACTION takes place. An ACTION-RESOURCE specifies any tools or instruments that are used to perform the action (*with what strength*). An ACTION-OBJECTIVE specifies the focus of the ACTION, the thing that is acted upon. An ACTION-TEMPORAL-ASSOCIATION specifies *when* the ACTION takes place, either absolutely or relative to other ACTIONS. An ACTION-FUNCTIONAL-ASSOCIATION specifies non-temporal relations among ACTIONs. One important such functional relation is the relation of sub-ACTION, encoding a mereology of events. Specifying one ACTION as a sub-ACTION of another is a way to specify *how* an ACTION is to be accomplished [15]. For example, an enemy might disrupt an election by bombing a polling place. The bombing would here be a sub-ACTION of the disrupting. In addition, the bombing might be specified as occurring *in-order-that* the disruption occurs. In this way, JC3IEDM allows one to express the means (*how*) of an ACTION as well.

JC3IEDM also provides a way to represent the fact that other artifacts may provide further information about the ACTION encoded in the database. These artifacts would include SITEMPS, Situation Matrices, and so on. In an ACTION's optional associated ACTION-REFERENCE element, one can specify, for example, that a particular SITEMP or SITMATRIX provides

---

[1] OWL-DL for the purposes of this discussion.

further details about the ACTION described. This, of course, would cause difficulties for automating inference of ECOAs, since crucial information might be represented in these artifacts in a non-formal way, as graphics or unstructured text.

Every piece of information in JC3IEDM has a mandatory associated REPORTING-DATA element that specifies when the information was reported, by whom, and specifies other elements of its pedigree: how certain the report was, how reliable the reporter, how likely the information reported is to be true, and so on.

In JC3IEDM, therefore, ECOAs would be represented as complex ACTIONs committed by hostile forces and predicted to occur with various likelihoods (*possible*, *probable*, *improbable*, etc.). In JC3IEDM, an ECOA's status as a prediction is reflected in the REPORTING-DATA category code *predicted*. Actions, as we have said, will be represented as having an internal structure, with sub-ACTIONS bearing temporal, causal and other relations to one another.

Nevertheless, because JC3IEDM is purely a relational data model, there are some ECOAs that can't be completely captured.

For ECOA 1 (above), it can be represented in JC3IEDM that the ECOA predicts an ATTACK ACTION by REDLANDS units immediately following *(Starts-after-end-of*) an invasion (INVASI) ACTION by the JTF Blue Sword. These ACTIONS are to be followed by redeployment (REDEPL) ACTION-TASKs and more ATTACK ACTION-TASKs.

What can't be represented in JC3IEDM is quantification. The data model is one of purely first-order relations without quantification. The first part of the ECOA says that all of the REDLAND forces (in the area) will participate in the attack. Then, however, it says that these forces will disperse into smaller units and redeploy to mountains and cities. This can be paraphrased using explicit quantification as: for every unit that is a component of the REDLANDS forces (within the specified area), there exists some mountain or city to which it will redeploy for further attacks. (It would be incorrect to specify merely that the REDLANDS forces as a whole will redeploy to a mountain or a city, since this would entail that all of the units would wind up in the same mountain or the same city.) Lacking quantification, we must simply enumerate all of the sub-units as redeploying.

Disjunction is similarly inexpressible: there is no way to express that a unit will redeploy to either a mountain or a city without specifying which. JC3IEDM does allow one to say that every unit will redeploy to a mountain, but, as a provisional sub-ACTION, it will redeploy to a city (or vice versa). Note that OBJECT-TYPEs (here, "mountain" or "city") can be specified as ACTION objectives in JC3IEDM as well as individuals.

JC3IEDM can represent different types of participation in an event: an organization may initiate, control, reinforce, or support an ACTION. Some actions require joint actors and some actions are distributed. For example, if Jack and Jill went up a hill, then Jack went up a hill and Jill went up a hill. But if ten ships blockaded a harbor, it doesn't follow that any one of the ships individually blockaded the harbor. Blockading a harbor (usually) necessitates joint action. JC3IEDM allows one to roll up units into an ORGANISATION via an ORGANISATION-STRUCTURE entity that would allow one to make a distinction between joint and distributed actions: joint actions are done by the hierarchically constituted group; distributed actions are done by each of several participants. Thus, one could represent that a convoy of ships blockaded the harbor and block any inference that a member of that convoy blockaded the harbor by means of this convention.

JC3IEDM also allows one to represent ACTIONS as *feints*, i.e. false attacks designed to mislead or distract. Therefore, an ACTION in the database that is marked as a feint is one that is said not to (completely) happen. It is important to check for the feint qualification on every ACTION to make accurate assessments of the situation. However, there is no straightforward way to represent an ACTION as not occurring at all, now or in the future.

This is a serious deficiency since it is important to be able to represent that an ACTION did not take place in order to encode reports such as: Observer *O* reports that unit *U* did not destroy bridge *B*. Such a report is different from a report that observer *O* did not observe the bridge destroyed or being destroyed. The latter requires that the bridge not be destroyed while the observers are observing it; the former only requires no observations of a destroyed bridge. Either is consistent with the bridge's destruction at the time of the report.

VIStology has developed a set of transformations to automatically translate the evolving JC3IEDM ERWIN specification into an OWL ontology comprising over 7900 elements (OWL classes, properties and their instances) [12]. A great deal of the semantics of the model remains trapped in text descriptions of the entities and relations, and we have not captured the JC3IEDM business rules for valid combinations of values in the ontology. However, it is possible to encode JC3IEDM ECOAs in a format that, at least in theory, supports formal reasoning. The parallel OBJECT-ITEM and OBJECT-TYPE hierarchy in JC3IEDM makes straightforward inferences about super- and subclasses of event participants impossible, however.

3 ECOAs in Situation Theory

Jon Barwise and John Perry developed Situation Theory at Stanford in the 1980s as an alternative to the possible-worlds semantics that had been introduced to solve puzzles in the interpretation of modal logic. Situation Theory 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". [Situation Theorists] call those limited parts of the world *situations*.

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

$\ll R, a_1, \ldots, a_n, 0/1 \gg$ (elementary infon)

where R is an n-place relation and $a_1, \ldots, a_n$ are objects appropriate for R, with the last two slots denoting the time and place of the situation. Since Situation Theory is multi-sorted, "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). Infons may be recursively combined to form compound infons by using conjunction, disjunction and situation-bounded quantification. We call basic, uncombined infons *elementary infons*.

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

$s \models \sigma$

The relationship 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.

In Situation Theory, ECOA 1 would be represented as a set of infons supported by the (predicted) situation.

s |= (<<invade, JTF Blue Sword, loc(Redland), t, 1>> ∧ <<attack, Redland Forces, JTF Blue Sword, loc(JTF Blue Sword), t' > t, 1>>)

The quantificational part of the ECOA would then be represented as:

∀(u : u ∈ Redland Forces) ∃(x : x ∈ city(x) v mountain(x)) {<< redeploy, u, x : city(x) c mountain(x), loc(x), t'' > t', 1>> ∧ <<attack, u, JTF Blue Sword, loc(JTF Blue Sword), t''' > t'', 1>>

Situation Theory by itself doesn't necessarily provide an explicit mechanism for representing joint rather than distributed actions. A Situation-Theoretic representation of this would therefore require the introduction of an operator to distinguish joint actors from distributed actors. Thus, we might say that the join of ten boats, represented as:

⊕{boat1, boat2, … boat10}

blocked the harbor is a supported infon, but it does not entail any infon formed by substituting a proper subset of

{boat1, boat2, … boat10}

in the same position.. That is, in a particular situation,

s |= <<blockade, ⊕{boat1, boat2, … boat10}, SomeHarbor,t,1>>

but not

s |= <<blockade, {w: w ⊂ {boat1, boat2, … boat10}}, SomeHarbor,t,1>>

Further, a feint would be represented as a situation in which:

s |= <<feint(blockade), …, 1>>

from which we might infer (by means of a rule):

s |= <<blockade, …, 0>>

Finally, with respect to negative events, if observer *O* sees unit *u* not stop at point *p*, then

s |= << sees, o, s', …, 1>> (o sees situation s')

and s' |= <<stop, unit u, …p, t, 0>>

where s' supports the negative infon.

These examples of negative polarity represent capabilities that are not present in JC3IEDM natively and which may be necessary for representing some ECOAs. As implemented within a relational database that assumes a closed world and negation-as-failure, every statement not represented as true is false. It is not possible to represent an ACTION as non-occurrent directly within JC3IEDM itself, however, nor in JC3IEDM-OWL.

In Situation Theory, however, there is no corresponding notion of functional relations between events that can be used to encode the idea that a specified action will be accomplished by performing some other set or sequence of actions. Situation-Theoretic accounts may encode spatio-temporal co-occurrence of infons, but that is all. That is, if the election is to be disrupted by means of the bombing, Situation-Theoretic accounts can represent the two relevant infons as co-occurrent, but not that one event is the means of accomplishing the other, although Situation Theorists like Devlin introduce relations among situations (or situation types) including 'involves' and 'causes' ([5], pp. 91,184).

Situation Theory Ontology

Now we show how Situation Theory can be formalized as an OWL ontology; we call it the Situation Theory Ontology (STO). Details of this ontology are presented in [9]. A graphical representation of a small part of STO is shown in Figure 2.

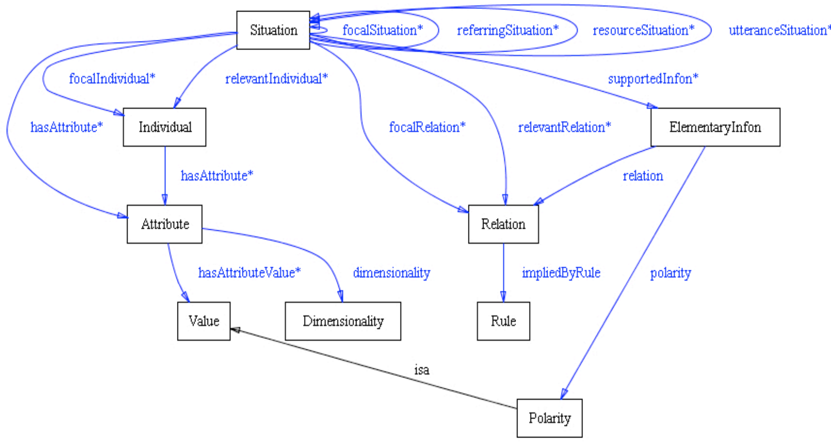

Figure 2: STO Ontology

The STO ontology is here visualized using the Protégé plugin 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 arrow is the range of the relation[2].

`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 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 in OWL using the subClassOf property. In OntoVIZ notation, this is depicted by the *isa* ("is a") 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`.

We introduce a class `ElementaryInfon` for elementary infons, and we use OWL class constructors and rules to deal with compound infons. 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 (elementary 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 in OWL, but in STO, it can have additional slots, such as the time when the property holds for two individuals. The slots a_1, . . . , a_n can be filled with: individuals, relations, location (spatial and temporal), situations, and types of all of the above.

Thus STO is a step towards encoding Situation Theory in a formalism that supports automatic reasoning. Since STO doesn't fully support quantification, it would not be possible to represent ECOAs that necessarily involve quantification, such as the quantificational part of ECOA 1, for example.

The SAW-CORE Ontology

In previous work [11], we have outlined a Situation Awareness Core (SAW-CORE) ontology, encoded in OWL, that is also inspired by Situation Theory. Figure 3, below, is a UML diagram of the SAW-CORE ontology from our Situation Awareness project, in which rectangles represent classes and connecting lines indicate inter-class relationships or properties.

The `Situation` class (upper right) defines a

---
2 http://vistology.com/onto/STO.owl

situation to be a collection of Goal, SituationObjects and Relations. SituationObjects are entities in a situation -- both physical and abstract -- that can have characteristics (i.e., Attributes) and can participate in relationships with other objects (i.e., Relations). Attributes define values of specific object characteristics, such as position, weight or color. A PhysicalObject is a special type of SituationObject that necessarily has the attributes of Volume, Position and Velocity. Relations characterize subsets of the Cartesian product of ordered sets of SituationObjects.

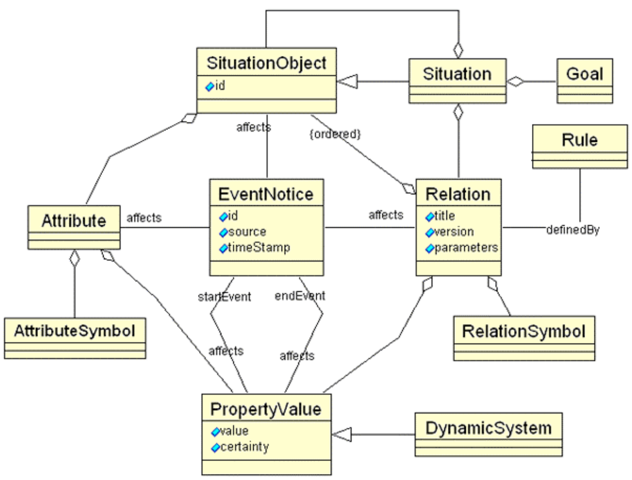

Figure 3 SAW-CORE Ontology

An important aspect of Attributes and Relations is that they need to be associated with values that can change over time. To accomplish this Attributes/Relations are associated with zero or more PropertyValues each of which defines two time dependant functions, one for the actual value and the other for the certainty assigned to that value. A new PropertyValue is created for an Attribute/Relation whenever an EventNotice arrives (from a sensor) that reports that Attribute/Relation. The value of an Attribute/Relation at a particular point in time (either current, past or future) can be determined by accessing the value function of the PropertyValue instance that is in effect at the prescribed time. Every situation has one or more Relations that constitute the Goal of the Situation.

The ontology permits a PropertyValue to be implemented as a DynamicSystem. This means that the value and certainty functions are dynamically modeled and therefore they cause the PropertyValue to change even in the absence of new EventNotices (normally, values are inertial between sensor readings). To illustrate the need for a DynamicSystem implementation of PropertyValues, consider the Position attribute of a PhysicalObject. The object's Position attribute's value at time t+1 is related to the object's Velocity (a vector providing speed and direction) at time t. Even if no new EventNotice affecting the position is received at time t+1, it is reasonable to assume that the object's position has changed. In the absence of additional information (e.g., acceleration, trajectory) it might be reasonable to assume that the object continues to move with its last noted speed and direction until informed otherwise, albeit with increasing uncertainty as time goes on. To be able to make such projections in the absence of explicit sensory information requires predictive models. It is for this reason that the SAW-CORE ontology employs DynamicSystems as a way of implementing PropertyValues.

This aspect of SAW-CORE illustrates an important capability that diverges from JC3IEDM and STO in its current formulation. The incorporation of dynamic models (DynamicSystems) as providing various attribute values over time goes beyond JC3IEDM in representing ECOAs as *courses of action*, i.e. dynamic entities that unfold over time, but, of course, only in situations in which the action can be specified mathematically. In such cases, it would be possible to infer quantities such as the acceleration of objects and distance traveled, by calculation. In JC3IEDM, while it is possible to associate an ACTION with an ACTION-TASK-STATUS that can be used to specify a completeness ratio (e.g. one can assert that a task will be some percent complete at a particular time), this is far less information about systems and actions that can be modeled dynamically.

It is worth noting that Fernando [6] provides a model of events as sets of sequences of (potential) observations, like SAW-CORE's EventNotices, expressed as regular expressions over observation frames. Fernando's idea is notable in providing a model-theoretic account of entailment relations between event statements as relations between sets of observation sequences described by regular expressions. In addition, his account provides a way to compose expressions semantically, e.g. deriving the appropriate semantics for 'walked a mile' from those 'walk' and 'for a mile'. Fernando's account relates inertial markers on observations to Kleene star operators on observation frames, encoding the idea that if one is observed to be doing certain things at t, one will normally continue to be doing those things until something intervenes or an end state is reached. As such, it is possible to specify, for example, all sequences of observations consistent with the "Yale Shooting Problem" in this finite-state formalism [8]. Fernando's event semantics provides only for the linear ordering of event components, unlike the hierarchical notion of plans embodied in JC3IEDM (e.g. means clauses) and contemporary plan recognition algorithms [7]. Finally, Fernando's model-theoretic account of entailment is different from that of OWL, and it is not clear how they could be combined.

4 Discussion

In this paper we have presented three approaches to encoding ECOAs in formalisms that are amenable to automatic reasoning: JC3IEDM-OWL, and two Situation-Theoretic approaches, STO, and SAW-CORE.

The purpose of encoding these representations in OWL is to facilitate formal reasoning about them. OWL representations natively enable at least the following kinds of reasoning:

Subsumption reasoning – allows the inference that one class is a subclass of another. This inference is based upon the intentional definitions of the classes. For instance, if ECOA class A is defined using various properties that the class must have, it is possible to infer that a proposed description of a ECOA Class B, is a subclass of A.

Satisfiability reasoning - allows one to infer whether a proposed ECOA type is satisfiable, i.e., whether it can be instantiated concretely.

Instance retrieval - infer which of the instances are instances of a particular class.

Type inference (instantiation) - what are the classes that a given thing is an instance of?

The task of Situation Development with which we are primarily concerned is a different sort of inference: an inference from present conditions to the most probable future ECOAs. It goes beyond pure OWL inferencing.

In [9], some illustrations of reasoning about situations represented in STO were presented. The relations between classes of actions were represented in the ontology, and, given the well-known expressive limitations of OWL, in rules layered on top of the OWL ontology. OWL for example, lacks property restrictions and joins that would be needed to encode concepts such as that uncleOf(X,Y) is true iff for some Z, brotherOf(X,Z) and parentOf(Z,Y). It seems clear that ECOA's would require similar expressive power, and therefore rules as well.

The difference between the JC3IEDM representation of an ECOA and the Situation Theoretic representations (STO, and to a lesser extent, SAW-CORE) involves a difference in metaphysics. In JC3IEDM, the ontology is basically an ontology of Davidsonian events. JC3IEDM ACTIONs are individuals that fall into certain types. Every JC3IEDM ECOA must be constructed out of the 445 ACTION-TASK and 346 ACTION-EVENT types that have been provided by the JC3IEDM vocabulary.

Events, in contrast to situations, have particular identity conditions, with what Kratzer calls 'minimality' [10]; they contain their own parts, but expanding (or contracting) the spatio-temporal boundaries of one event does not (necessarily) delimit a second event. By contrast, redrawing the spatio-temporal boundaries of one (real) situation always delimits another situation.

Situation Theory does not break down what happens into a set of types of events; rather, situations *contain* events. For Barwise and Perry, situations are metaphysically basic, and events are logical constructions built from them.[3] In Situation-Theoretic approaches, events are not taken to be particulars. Rather statements describing what happened are explicated in terms of infons that are supported or made true by situations. A (concrete) situation in Situation Theory is basically a region of space-time that carries certain positive and negative information. Certain elements of a Situation-Theoretic representation of ECOAs would be useful in augmenting a JC3IEDM event-based encoding of ECOAs, particularly the idea of negative infons.

We enumerate the differences in these approaches to encoding and reasoning about ECOAs in Table 1 below.

5 Conclusion

We have shown that there are advantages and disadvantages to embodying both the JC3IEDM and Situation-Theoretic approaches within an OWL formalism for automatic reasoning with respect to representing and reasoning about enemy courses of action, a crucial part of situation development.

Acknowledgments

This material is based upon work supported by the United States Army under Contract No. W15P7T-06-C-T202 and 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 Army or Air Force.

References

[1] J. Barwise and J. Perry. *Situations and Attitudes*, Cambridge, MA: MIT Press. 1983.

[2] C. Blais, K Galvin, M Hieb. Coalition Battle Management Language Study Group Report. *Fall Simulation Interoperability Workshop*, Orlando, FL. 2005.

[3] Commander's Estimate of the Situation, NWC 4111G Instructional Workbook. Newport, RI: Naval War College. 2004.

[4] Donald Davidson. The logical form of action sentences, 1967. Reprinted in *Essays on actions and events*. Oxford: Clarendon Press (1980).

[5] Keith Devlin. *Logic and Information*, Cambridge, UK: Cambridge University Press, 1991

[3] Specificially, for Barwise and Perry([1], pp. 8-9), events (or what they call "courses of events") are sets of partial functions from spatio-temporal regions to "situation-types" defined by a tuple of objects standing or failing to stand in a certain relation. In order to be dynamic, these regions must have a temporal dimension.

[6] Tim Fernando. A finite-state approach to events in natural language semantics. *J. Logic and Computation* 14(1): 79—92. 2004.

[7] Christopher W. Geib and Mark Steedman. On Natural Language Processing and Plan Recognition. *IJCAI* 2007: 1612-1617. 2007.

[8] Lauri Karttunen, The Yale Shooting Problem. http://www.stanford.edu/~laurik/fsmbook/examples/YaleShooting.html. 2005.

[9] Mieczyslaw M. Kokar, Christopher J. Matheus and Kenneth Baclawski Ontology-based situation awareness. *Information Fusion*, In Press.

[10] Angelika Kratzer, 2007. "Situations in Natural Language Semantics", The Stanford Encyclopedia of Edward N. Zalta (ed.), http://plato.stanford.edu/archives/spr2007/entries/ situations-semantics/. 2007.

[11] Christopher J. Matheus, Mieczyslaw M. Kokar, and K. Baclawski. A core ontology for situation awareness. FUSION 2003.

[12] Christopher J. Matheus and Brian Ulicny. On the Automatic Generation of an OWL Ontology based on the Joint C3 Information Exchange Data Model. 12th ICCRTS, Newport, RI, June 19-21. 2007.

[13] MIP, Joint C3 Information Exchange Data Model (JC3IEDM Main), Greding, Germany, December 2005.

[14] Ulrich Shade. Automatic Report Processing. In Proceedings of the 9th ICCRTS. 2004.

[15] Judith Jarvis Thomson. *Acts and Other Events*, Ithaca, NY: Cornell University Press. 1977

[16] US Army. FM 34-130: *Intelligence Preparation of the Battlefield.* Washington, DC. 1994.

[17] Øystein Linnebo, "Plural Quantification", The Stanford Encyclopedia of Philosophy (Spring 2008 Edition), Edward N. Zalta (ed.), URL = http://plato.stanford.edu/archives/spr2008/entries/plural-quant/. 2008.

Table 1 ECOA elements by Ontology

ECOA Elements	JC3IEDM	SAW-CORE	Situation Theory	STO
Who (aggregate agents)	JC3IEDM ORGANISATION with specified role; supports aggregate agents for joint action;	Any instance of OWL:Thing; Situation Object can be aggregate of multiple objects.	Both aggregates and their members are of type IND; membership is a relation.	Aggregates are a subtype of STO:INDIVIDUAL; membership specified as property.
What (feints; negative events, quantified events; closed/open world (non-)distributive actions; extrinsic references)	One of specified ACTION EVENT or ACTION-TASK types; supports feints, but no negative events; Closed world. No quantification; no distributivity qualifier on actions; info can be provided extrinsically	Any binary relation of OWL:Things, possibly specified as a dynamic system; limited quantification; no feints; no negative infons; Open world; distributivity requires rules; extrinsic information via rdfs:seeAlso	Any n-ary relation of individuals. Negative infons; no feints; Partial world; Full quantification. Distributivity rquires subtyping relations and *involves* relation; Extrinsic information could be specified as a relation.	Any binary relation of OWL:Things, possibly specified as a dynamic system; limited quantification; no feints; negative infons but limited inferences; Open world; distributivity requires rules; Extrinsic info via rdfs:seeAlso
When (Absolute/relative)	Absolute and relative time w/respect to ACTIONS and their stages	SAW-CORE:Attribute - Absolute time expressed in OWL	Any element of type TIM	Absolute time instant expressed in OWL as e.g. STO:Time
Where	JC3IEDM LOCATION; Geophysical points and regions	SAW-CORE:Attribute - Location expressed in OWL	Any element of type LOC	Geophysical point STO:Location expressed in OWL
How/Why (means clauses; purposes clauses)	Sub-Actions and functional relations; ACTION-OBJECTIVE	No means clauses; Specified Goal Relation(s).	No means clauses; Goals require representing intentional states.	Unspecified.