Ontologies in OWL for Rapid Enterprise Integration

Suzette Stoutenburg[1], Leo Obrst[2], Deborah Nichols[2], Paul Franklin[1], Ken Samuel[2], Michael Prausa[1]

[1] The MITRE Corporation, 1155 Academy Park Loop, Colorado Springs, CO 80910
[2] The MITRE Corporation, 7515 Colshire Drive, McLean, VA 22102-7508
{suzette,lobrst,dnichols, pfranklin,ksamuel,mprausa}@mitre.org

Abstract. Ontologies enable explicit expression of collective concepts and support Machine-to-Machine (M2M) interactions at the semantic level. Ontologies expressed in a standard language, such as the Web Ontology Language (OWL) and exposed on a network offer the potential for unprecedented interoperability solutions since they are semantically rich, computer interpretable and inherently extensible. In this paper, we describe how we applied ontologies in OWL for rapid enterprise integration of heterogeneous data sources. We found that once a robust foundational domain ontology is established, it is easy and quick to integrate new data sources and therefore rapidly provide new system capabilities.

Keywords: Ontology, Web Ontology Language, Semantic Web, Enterprise integration, Agile systems, Service-oriented architecture.

1 Introduction

Increasingly, Command and Control (C2) systems require the ability to respond to rapidly changing environments. C2 systems must be agile, able to integrate new sources of information rapidly for enhanced situational awareness and response to real-time events. Data from varied sources across the world must be integrated and transformed into knowledge that can be leveraged. Machine-to-machine capabilities are also increasingly necessary to accomplish mission goals. To this end, MITRE has researched the use of semantic technology, in particular semantic models expressed in ontologies and rules, to address emerging mission needs. While the original focus of our effort was on semantic rules, we have found that ontological models offer a powerful tool for rapid enterprise integration. In fact, using ontologies, we were able to integrate new sources of data within hours, instead of weeks or months, using traditional software development methods. This work will be showcased at the Joint Expeditionary Force Experiment (JEFX) 2008. We also found that the OWL-Description Logic (DL) language offers rich capabilities, though there are a few capabilities we would like to see added. The purpose of this paper is to describe the use case, the ontologies used to model the use case and how they supported rapid, enterprise integration. We will share our recommendations for additions to OWL. We will also provide some detail on the prototype application. In addition, we acknowledge previous related research in using ontologies for situational awareness and information fusion such as [4], [5]. In addition, there has been interesting non-ontological work concerning convoy movement (the basis of our use case, below) issues [6], [7].

2 Use Case

Our initial research focused on a military command and control domain with a supply convoy moving through an unsecured area. Figure 1 presents an example situation, in which a convoy is moving north along a primary route, approaching the location where intelligence has reported an enemy sniper is stationed. New information can become available at any time, such as the discovery of a new enemy object in theater, change in weather, etc. Through the ontological model and associated semantic rules, the system provides alerts and recommendations to the convoy commander that enhance situational awareness and can cause the commander to change route or call for support. For example, in the situation shown in Figure 1, an enemy unit is within the convoy's region of interest (the circle surrounding the convoy), so the system might generate the following: "Alert: Intelligence report of enemy sniper in the vicinity." and "Recommendation: Take alternate route." For a more detailed description of the mission use case, see [1].

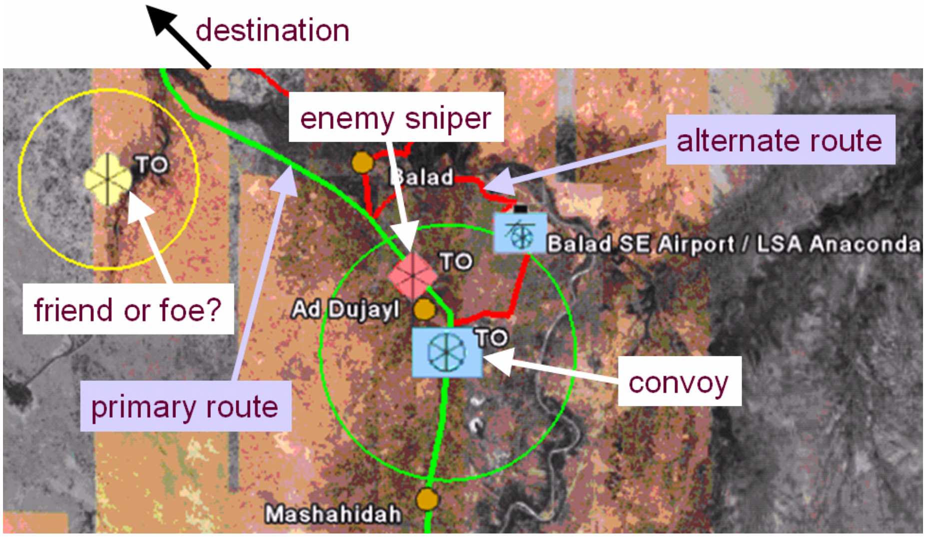

Figure 1. Google Earth view of the basic use case that was initially supported, showing key elements modeled in the ontology. The initial ontology supported the concepts of convoy and other entities in theater, routes, regions of interest (shown in green), etc.

Once we demonstrated how ground position and intelligence data could be integrated using ontologies, we were asked to extend our prototype by adding the additional event types.

1. Space events, such as a degradation in satellite communication
2. Live satellite positions
3. Ship movement

We added in these events in just hours into our framework. As an example, Figure 2 shows a pilot entering into an area in which satellite communication is degraded. Figure 3 shows a constellation of satellite positions. We will describe how we integrated these new event types in the following section.

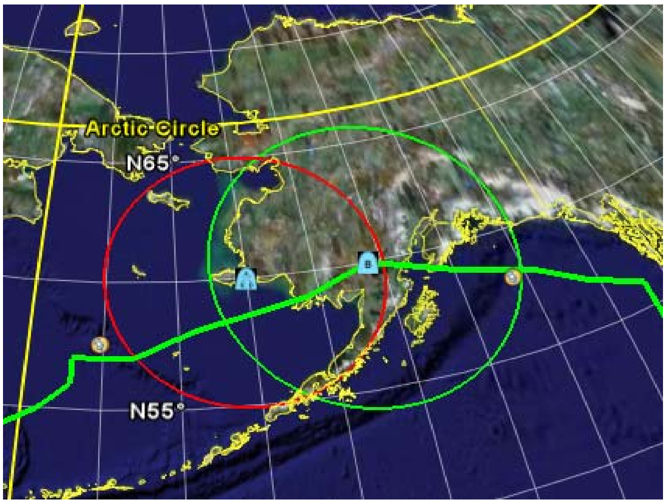

Figure 2. View of Google Earth client, showing new ontological elements; in this case, a pilot entering an area impacted by degraded satellite communication. The region of interest in red shows the projection of the satellite coverage area onto the earth.

3 Ontology Design

To model the objects and events described in Section 2, we chose to construct five topical ontologies, as follows.

- TheaterObject – to describe objects in the battlefield and reports about them.
- RegionOfInterest – to describe regions of interest on the battlefield.
- Convoy – to describe the convoy, its mission, components, etc.
- Convoy Routes – to describe routes the convoy might take.

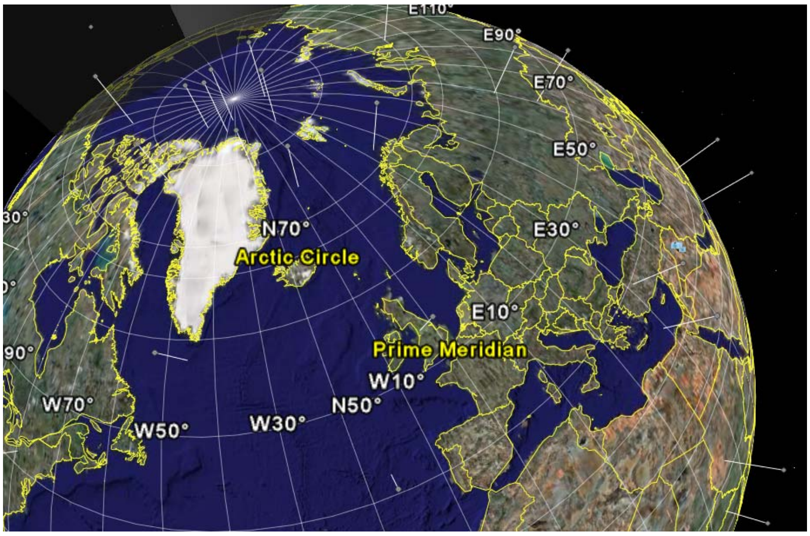

Figure 3. View of Google Earth client, showing constellation of satellites in real time. These satellite positions were obtained from the WWW.

- ConditionsAndAlerts – to model how the knowledge base builds, resulting in conditions and alerts that affect the convoy.

Figure 4 shows the high level relationships between its ontology and its major concepts. The TheaterObject class is the heart of the ontology. This is the class of entities that represent objects in theater. Some types of TheaterObjects include MilitaryUnit, Sniper, RoadObstacle, Facility. TheaterObjects have a location, and may have a speed, heading, and combatIntent, among other features. The property combatIntent is used to represent whether an object in theater is friendly, hostile or unknown.

To distinguish the entity in theater from reports about it, we specified the class of ObservationArtifacts, which is the class of reports about objects in the theater. Instances of ObservationArtifact have properties such as timeOfObservation, locationOfObservation (of thing observed, not sensor/observer platform), speedObservation, etc. We found the distinction between theater object and observation to be very important, as it allows inferencing over multiple reports about the same object in theater. This built the foundation for using rules to fuse information from multiple sources [4], [5].

The RegionOfInterest (ROI) ontology models the class of geospatial areas of special interest surrounding TheaterObjects. We modeled TheaterObject as the focal object of a DynamicROI, since most TheaterObjects move on the battlefield. The location of an ROI is centered on the position of its focal object. An ROI has shape, dimensions and area, the dimension and area of which depend on the type of threat or

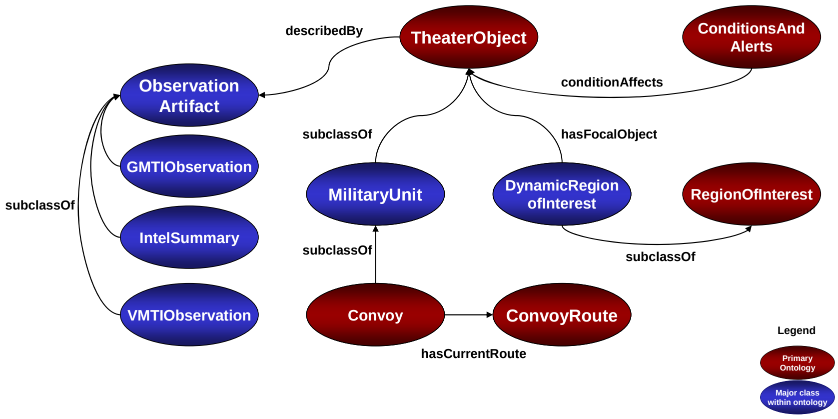

Figure 4. High level ontology design for the initial use case.

interest. ROIs are used to define a "safety zone" around a convoy which must not be violated by hostile or suspicious objects. Also, ROI is used to define the area around a reported hostile track that defines the potential strike area of the threat.

The Convoy ontology models the class of organized blue forces moving on the ground. This ontology allows specification of the mission, components and personnel on the Convoy. Convoy is a subclass of TheaterObject. This ontology was based primarily on [2].

The ConvoyRoute ontology provides a representation of paths of a convoy, including critical points (CPs) for primary and alternate routes. Recommended routes can change based on application of rules.

The ConditionsAndAlerts ontology provides a description of situations on the battlefield based on aggregations of events and actions of theater objects. As the knowledge base grows, a set of conditions is constructed based on events on the battlefield, which can result in alerts and recommendations to Blue Forces. Conditions, alerts and recommendations are generated through the application of rules.

These ontologies were developed in OWL-DL, which we found to offer robust descriptive capabilities. OWL-DL supported almost all of the features of the scenario. We would like to see a few new features should be added to OWL to fully support the needs of the mission use case, including uncertainty with respect to class membership and existence of properties between classes and individuals. The uncertainty should be user-defined, with perhaps a link to a set of rules for how to calculate the probability. We also needed n-ary predicates and individual-denoting functions[1]. We were able to achieve n-ary predicates through rules by extending SWRL.

---

[1] Individual-denoting functions are functions which, when applied to another term, form a complex term that denotes an individual in a descriptive way; for example, in "MotherOfFn(OsamaBinLaden)" MotherOfFn is applied to the single argument

The ontologies were applied in an overall framework for situational awareness, described in detail in Section 4. We built rules that operated over the ontologies as part of an integrated knowledge base that could be queried dynamically. For example, we built queries such as, "Provide all the TheaterObjects within the region of interest of a particular convoy" or "Provide all threats within the convoy's region of interest." The dynamic nature of semantic queries offers agile capabilities in and of themselves, since we can query any portion of the knowledge base. Furthermore, when we added new classes and properties to the basic set of ontologies, we were able to take advantage of the overall situational awareness framework, thus effecting new capabilities very quickly. For example, by adding satellite positions and maritime events (Satellite and Ship classes, respectively) to the TheaterObject ontology, instances of those classes are automatically retrieved when a query is launched. So, a query such as "Provide all TheaterObjects" will retrieve all instances of satellites and ships with no changes to the software framework. (Changes are required to recognize new message types, of course, but no changes to the services that query the ontology were necessary.) We found this to be quite powerful, a function of the fact that ontologies and semantic rules are expressed in data and can be rapidly changed; also, a function of the formal semantics and query mechanisms offered with OWL in conjunction with other technologies. We were able to show that new types of information could be integrated in just hours; less than an hour for small changes to the ontology, and just a few hours to a day or two at most to connect to new services and new sources of data. The simple changes we made to the ontology are shown in Figure 5.

Therefore, to integrate additional data sources, we simply added new classes and made them subclasses of TheaterObject. We added satellite, ship and degradation event locations. Obviously, in the future, we would seek to map our ontologies to other ontologies in development, so as to take advantage of a complete set of information on satellites and ships. Our goal was to show how quickly new sources could be integrated into an overall situational awareness framework.

4 Prototype Design

We integrated the ontologies and rules that model C2 scenarios into a loosely couple service-oriented architecture that operates over XML-based messages. We call the application the "Semantic Environment for Enterprise Reasoning (SEER)." The high-

OsamaBinLaden, to refer to Osama's mother, even though we may not know her name. Another example: AreaOfFn(ROI_4213), or, more complexly, AreaOfFn(ROIFn(Convoy1))). Some problems may arise with these function constructs if they are created for relations that are not in fact functional. For example, the referent of UncleFn(OsamaBinLaden) is not necessarily a determinate individual, since Osama may have more than one uncle. Where they can be used, they are a convenient shorthand and simplification for constructing references to individuals whose names are unknown and may not be of interest. Another use of functions is to denote physical units of measure, e.g., (Meter 1000) = (Kilometer 1).

level design of the application is shown in Figure 6. The components of the system include the following.

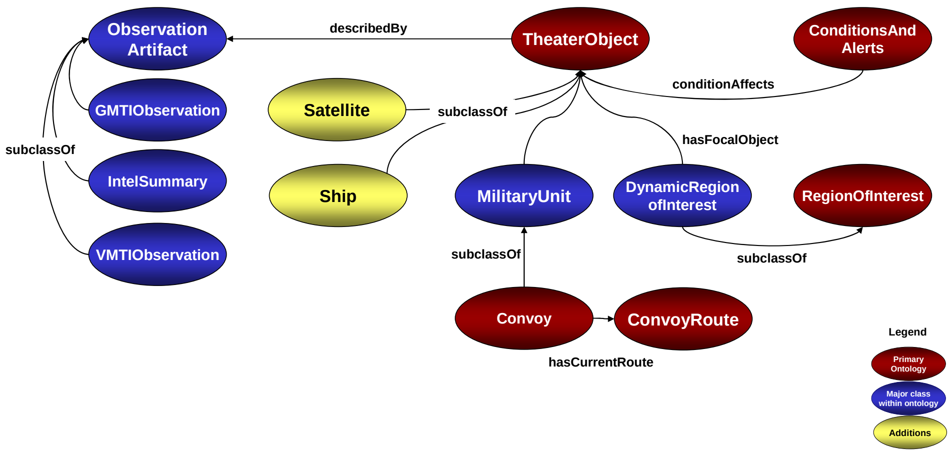

Figure 5. Modified ontology. By simply adding a few new classes, we were able to leverage the overall situational awareness framework, described in Section 4.

- Enterprise Service Bus (ESB)
- Google Earth[2] Client
- Reasoner, implemented in AMZI! Prolog Logic Server[3]
- Knowledge Base, composed of ontologies in OWL, rules in SWRL and instances in Prolog
- Situational Awareness Service
- Event Mediation Services
- Adaptors
- Message Simulator

We selected Mule[4] as the ESB solution to manage the interactions of the components in our solution. The ESB provides an abstraction layer over disparate messaging technologies, allowing interaction between components with minimal code development. Mule provides support for transport and transformation of publisher/subscriber pairs; in our architecture, applying the XSLTs of the Adaptors when appropriate. Mule also allowed us to detect events, including trigger events that cause the swapping of knowledge bases. Our selection of an ESB proved very useful as we were able to integrate sources for satellite information and other events, simply by creating a Mule endpoint.

---

[2] http://earth.google.com/
[3] http://www.amzi.com/
[4] http://mule.codehaus.org/

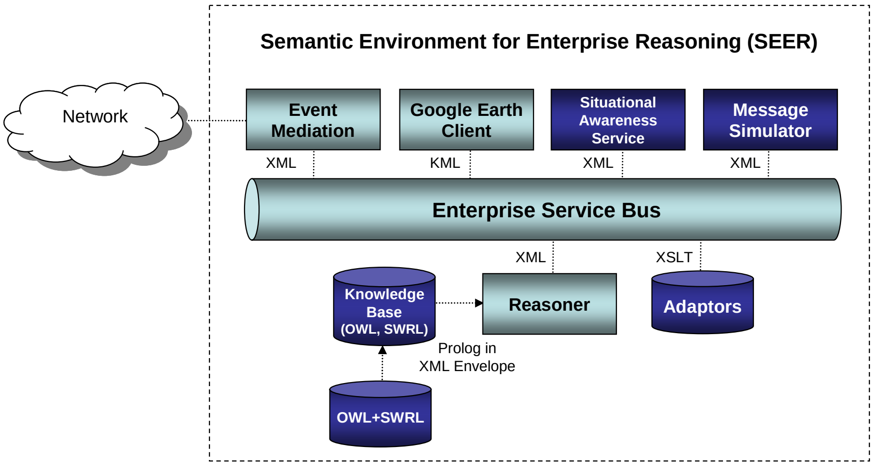

Figure 6. Architecture of the Semantic Environment for Enterprise Reasoning

We chose Google Earth as the SEER client, since it offers seamless integration of multiple data sources via its Keyhole Markup Language (KML). We were able to show that structured data from heterogeneous sources can be translated to KML and easily rendered, thus offering the potential for dynamic, user-defined displays. Google Earth also provides excellent maps and zoom in capabilities, enhancing the prototype demonstration. We did find some limitations however, such as Google Earth's ability to render custom shapes. For example, to draw a circle to represent a region of interest, we had to build code to render 360 points.

The commercial vendor AMZI! and its Prolog Logic Server was selected as the platform on which to host the integrated ontologies and rule base. Prolog was selected because it is based on logic programming, a syntactic restriction of formal logic, thus supporting the notion of proof, and a known efficient reasoning formalism. To date, AMZI! has performed well.

The Knowledge Base consists of integrated ontologies, rules and instances. Ontologies were developed in the OWL, the rules in the SWRL and they were translated to a single knowledge base in Prolog. For details on our approach to transforming and amalgamating Description Logic based languages into Prolog, please see [3].

We constructed the Situational Awareness Service, a software application that detects events (message exchanges over the ESB), consults the knowledge base, and delivers appropriate alerts and recommendations to the convoy commander via Google Earth clients. Events can be object movement, changes in weather, changes in alert conditions, etc. The service can dynamically query the knowledge base.

We built a set of Event Mediation Services which handle different types of service communication, including SOAP synchronous request/response, SOAP pub/sub, polling and REST. So it is through the Event Mediation services that SEER interacts with outside message sources.

The Adaptors are a set of XSLTs that are invoked by the ESB to translate messages to the appropriate format as they move between components. Events are initiated in an XML format that contains the AMZI! command format. These events are then asserted to both knowledge bases and translated to KML for display on Google Earth. The active knowledge base generates alarms and recommendations (when queried by the Conductor) and these messages are translated to KML for display as well.

We reused a Message Injector, developed at MITRE, which sends messages over the ESB to simulate events on the battlefield.

The SEER application works as follows. First, messages are received on the ESB, either from network sources or by the Message Injector. The ESB applies the appropriate XLSTs of the Adaptor and commits the new information to the knowledge base and sends KML to Google Earth.

6 Conclusions and Next Steps

We successfully showed that ontologies can be applied for rapid enterprise integration, allowing us to deliver new capabilities in hours, instead of weeks or months. By simply adding new classes to an existing, robust ontology, we were able to leverage the power of the situational awareness framework and deliver new information and capabilities rapidly. Therefore, we believe ontologies in OWL should be transitioned to support heterogeneous data integration in operational systems. We look forward to learning a great deal as we demonstrate this capability at JEFX 08.

We found that OWL is expressive and meets the majority of identified requirements. However, we would like to see OWL extended to support uncertainty, n-ary predicates and individual-denoting functions. Extensions to represent uncertainty and probability are needed for modeling real-world information and data sources, and to support decision making. Support for predicates of arity greater than two would enable the representation of important relations (e.g., "*x* is between *y* and *z*") which now require awkward workarounds. Individual-denoting functions offer a convenient way of abbreviating key types of data (e.g., "(Meter 1000)"). We believe that standard eXtensivble Markup Language (XML)-based languages with formal semantics, such as OWL and SWRL, will form the basis for a new tier in future n-tiered architectures. These languages are inherently extensible because they are standards. Just as the database tier emerged in the 1980s, freeing software developers from writing their own database interfaces, we believe the semantic tier is emerging as part of a robust, agile architecture.

Of course, the construction of the original set of ontologies was a manual process, and therefore expensive and time-consuming. Therefore, we are proposing additional research at MITRE to investigate, enhance and/or develop approaches for automatic ontology generation and mapping. In addition, we believe that eventually semantic interoperability requirements across multiply developed ontologies and their supporting data will require embedding these ontologies within common super domain, and upper/middle/foundational ontologies [8], [9], [10].

References

1. Stoutenburg, S.; Obrst, L.; Nichols D.; Samuel, K.; and Franklin, P. 2006. Applying Semantic Rules to Achieve Dynamic Service Oriented Architectures. *Second International Conference on Rules and Rule Markup Languages for the Semantic Web*, co-located with *ISWC 2006*, Athens, GA, November 10-11, 2006. In: *Service-Oriented Computing – ICSOC 2006*, Lecture Notes in Computer Science Volume 4294, 2006, pp. 581-590. Heidelberg: Springer-Verlag, Also: MITRE Technical Report MTR 06B0000014, March 2006. http://www.mitre.org/work/tech_papers/tech_papers_06/06_0904/index.html.

2. Convoy Operations Battle Book. 2005. Tactics, Techniques, and Procedures for Training, Planning and Executing Convoy Operations. Tactical Training and Exercise Control Group. Marine Aviation Weapons and Tactics Squadron One.

3. Samuel, K.; Obrst, L; Stoutenburg, S.; Fox, K.; Franklin, P.; Johnson, A.; Laskey, K.; Nichols, D.; Lopez, S.; and Peterson J. 2006. Applying Prolog to Semantic Web Ontologies & Rules: Moving Toward Description Logic Programs. *ALPSWS: Applications of Logic Programming in the Semantic Web and Semantic Web Services*, Aug. 16, 2006. *International Conference on Logic Programming*, pp. 112-113. *Federated Logic Conference 2006*, Seattle, WA. Poster presentation and extended abstract. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-196/alpsws2006-poster5.pdf.

4. Matheus, C. J.; K. P. Baclawski; and Kokar, M. M.. 2003. Derivation of Ontological Relations Using Formal Methods in a Situation Awareness Scenario. *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2003*, pages 298–309. SPIE.

5. Matheus, C. J.; Kokar, M. M.; and Baclawski, K. 2003. A Core Ontology for Situation Awareness. *Proceedings of FUSION'03*, Cairns, Queensland, Australia.

6. Chardaire, P.; McKeown, G. P.; Harrison, S. A.; and Richardson, S. B. 2005. Solving a time-space network formulation for the convoy movement problem. *Operations Research* 53(2): 219–230.

7. Chardaire, P.; McKeown, G.P.; Harrison, S. A.; and Richardson, S. B. 2001. Convoy planning in a digitized battlespace. *Journal of Defence Science*, 6(2):168–175.

8. Semy, S. K.; Pulvermacher, M. K.; and Obrst, L. J. 2004. Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains: An Evaluation. http://www.mitre.org/work/tech_papers/tech_papers_05/04_1175/04_1175.pdf.

9. Pulvermacher, M.K.; Stoutenburg, S.; and Semy, S. K. 2004. Netcentric Semantic Linking: An Approach for Enterprise Semantic Interoperability. http://www.mitre.org/work/tech_papers/tech_papers_04/04_1174/04_1174.pdf.

10. Upper Ontology Summit Joint Communiqué. 2006. http://ontolog.cim3.net/cgi-bin/wiki.pl?UpperOntologySummit/UosJointCommunique.