Semantic Web Technologies for Aerospace

Paul Kogut
Lockheed Martin Management & Data Systems
P.O. Box 8048
Philadelphia, PA 19101
610-354-3524
paul.a.kogut@lmco.com

Jeff Heflin
Department of Computer Science and Engineering
Lehigh University
19 Memorial Drive West
Bethlehem, PA 18015
610-758-6533
heflin@cse.lehigh.edu

Abstract—Emerging Semantic Web technology such as the DARPA Agent Markup Language (DAML) will support advanced semantic interoperability in the next generation of aerospace architectures. The basic idea of DAML is to mark up artifacts (e.g., documents, sensors, databases, legacy software) so that software agents can interpret and reason with the information. DAML will support the representation of ontologies (which include taxonomies of terms and semantic relations) via extensions to XML. XML alone is not sufficient for agents because it provides only syntactic interoperability that depends on implicit semantic agreements. DAML is the official starting point for the Web Ontology Language, an emerging standard from the World Wide Web Consortium. This paper will cover promising aerospace applications and significant challenges for Semantic Web technologies. Potential applications include higher-level information fusion, collaboration in both operational and engineering environments and rapid systems integration. The challenges that will be discussed include the complexity of ontology development, automation of markup, semantic mismatch between current object-oriented models and Semantic Web ontologies, scalability issues related to reasoning with large knowledge bases and technology transition issues. The paper will explain ongoing research that is focused on addressing these challenges.

TABLE OF CONTENTS

1. INTRODUCTION

Agent-based systems have been promoted as panaceas for people like military commanders, intelligence analysts and DoD engineers who are suffering from acute information overload. Agents are supposed to help find answers more quickly, automate routine tasks, and allow the human to focus on the difficult decisions rather than mundane knowledge management tasks. Agents and the other trendy technology *web services* are being touted as the solution for legacy system interoperability. The problem is that agents and web services will not live up to their promise unless something is done about semantic interoperability. Agents will need to talk to other agents and web services that were built by different organizations for use in unanticipated contexts. Without a way to discover the meaning of terms and relationships, this vision of dynamic adaptable systems is doomed. Semantic interoperability problems have a variety of causes such as polysemy (a word has multiple meanings), synonymy (different words have similar meanings), and inconsistent model assumptions and granularity. Fortunately, DARPA and the World Wide Web Consortium (W3C) realized this and organized the Semantic Web initiative to focus on machine-readable ontologies that are accessible to agents on the web [1]. Ontologies are explicit semantic models, which include taxonomies of terms and semantic relations that help interpret queries and reason with knowledge. Semantic Web technology solves the polysemy and model assumption/granularity problems by allowing a developer to mark up a document, sensor, database schema, or legacy software interface by linking concepts (i.e., classes, relationships, instances) to other concepts defined in an ontology. Each concept is referenced by means of a unique Uniform Resource Identifier (URI). Semantic Web technology solves the synonymy problem by allowing explicit declarations that term X in an ontology or markup is equivalent to term Y in another ontology or markup.

Emerging Semantic Web technology such as the DARPA Agent Markup Language (DAML) [2][3] will support advanced semantic interoperability in the next generation of aerospace architectures. The basic idea of DAML is to mark up various artifacts so that software agents can interpret and reason with the information as shown in Figure 1. DAML supports the representation of ontologies via extensions to recent web technologies including XML and the Resource Description Framework (RDF)[4].

This paper will suggest promising aerospace applications and discuss significant challenges for Semantic Web technologies. This paper will discuss two example applications: higher-level information fusion and rapid
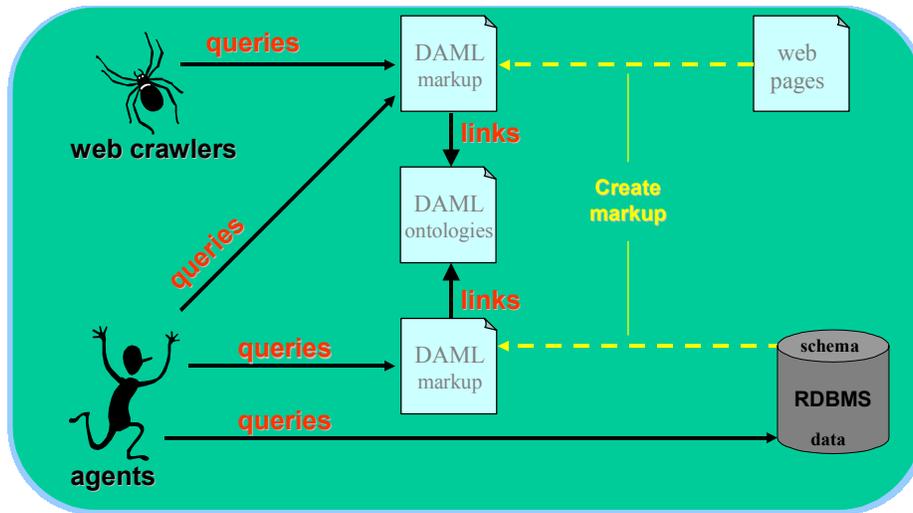
Figure 1 – DAML markup for artifacts such as web pages, database schemas, legacy software interfaces, and sensor outputs.

systems integration. The challenges include the complexity of ontology development, automation of markup, semantic mismatch between current object-oriented models and Semantic Web ontologies, scalability issues related to reasoning with large knowledge bases and technology transition issues. The paper will explain ongoing research that is focused on addressing these challenges.

2. SEMANTIC WEB LANGUAGES

Before we discuss applications and challenges we need to explain some examples of ontologies and markup. We will use the current DAML+OIL language to illustrate the concepts [5]. DAML+OIL is the official starting point for the Web Ontology Language (OWL), an emerging standard from the W3C [6]. DAML+OIL and OWL contain modeling concepts similar to class diagrams in the Unified Modeling Language (UML) plus other modeling formalisms that originated in Artificial Intelligence knowledge representation languages [7]. DAML+OIL is designed for efficient reasoning with ontological knowledge. DAML+OIL will eventually be extended to represent more first-order logic style rules and axioms.

Ontology languages like DAML+OIL have more formally defined semantics than catalogs and glossaries to ensure that agents interpret ontologies and markups in a consistent manner [8]. A simple DAML ontology is shown below. The first 4 lines declare the XML namespaces followed by an ontology definition that states that this ontology imports concepts from another ontology. This is followed by a series of class definitions. The restriction on the *person* class states that only an *organization* can be in a *PersToOrg* relationship with a *person*. In DAML+OIL, there are object properties and datatype properties. Object properties represent class-to-

class relationships. Datatype properties are for representing relationships to things like string and integer attributes. The example shows that any class with the *age* property can only have one unique value for *age*.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
>
<daml:Ontology rdf:about="">
   <daml:imports
rdf:resource="http://www.daml.org/2001/03/daml+oil#"/>
   <daml:label>AeroDamlOntology</daml:label>
 </daml:Ontology>

<daml:Class rdf:ID="AIRCRAFT">
   <daml:subClassOf rdf:resource="#VEHICLE"/>
</daml:Class>

<daml:Class rdf:ID="ORGANIZATION"/>

<daml:Class rdf:ID="PERSON">
 <rdfs:subClassOf>
   <daml:Restriction>
     <daml:onProperty rdf:resource="# PersToOrg "/>
     <daml:toClass rdf:resource="#ORGANIZATION"/>
   </daml:Restriction>
 </rdfs:subClassOf>
</daml:Class>

<daml:ObjectProperty rdf:ID="OrgToLoc"/>
<daml:ObjectProperty rdf:ID="PersToOrg"/>

<daml:DatatypeProperty rdf:ID="age">
```

```
    <rdf:type
rdf:resource="http://www.daml.org/2001/03/daml+oil#Uniq
ueProperty"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#no
nNegativeInteger"/>
</daml:DatatypeProperty>

</rdf:RDF>
```

The next example shows markup that corresponds to the ontology above. Markup contains instances found on a web page that are linked to classes in the ontology declared in the daml:imports statement. For example, *Pierre Lortie* is a *person* who has a relationship with an *organization* named *Bombardier Regional Aircraft* and *Pierre Lortie* is the same person as *Lortie* mentioned in the document.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:aac="http://ubot.lockheedmartin.com/ubot/2001/08/
extract.daml #">

<rdf:Description about="">
  <daml:imports  resource="http://ubot.lockheedmartin.com/
ubot/2001/08/extract.daml" />
</rdf:Description>

<aac:AIRCRAFT rdf:about="Dash8Series400">
  <daml:label>Dash 8 Series 400</daml:label>
</aac:AIRCRAFT>

<aac:ORGANIZATION
rdf:about="BombardierRegionalAircraft ">
  <daml:label>Bombardier Regional Aircraft
  </daml:label>
</aac:ORGANIZATION>

<aac:PERSON rdf:about="PierreLortie">
  <aac:PersToOrg
rdf:resource="BombardierRegionalAircraft"/>
  <daml:sameIndividualAs rdf:resource="Lortie"/>
  <daml:label>Pierre Lortie</daml:label>
 </aac:PERSON>
```

The next ontology fragment shown below contains a more complex class definition[2]. It says that an *animal_lover* is the intersection of a *person* and an entity that has at least 3 pets. This illustrates the expressiveness of DAML+OIL, which can be used to model real-world phenomena. As is apparent in the examples above, DAML+OIL is designed to be machine-readable. Graphical tools are needed to display DAML+OIL in human-readable form.

---

[2] An excerpt from an ontology developed at the University of Manchester for the OIL-Ed project.

```
<daml:Class rdf:ID="animal_lover">
 <daml:sameClassAs>
  <rdfs:Class>
   <daml:intersectionOf rdf:parseType="daml:collection">
     <daml:Class rdf:about="#person"/>
     <daml:Restriction daml:minCardinalityQ="3">
       <daml:onProperty rdf:resource="#has_pet"/>
       <daml:hasClassQ>
         <daml:Class rdf:about="#animal"/>
       </daml:hasClassQ>
     </daml:Restriction>
   </daml:intersectionOf>
  </rdfs:Class>
 </daml:sameClassAs>
</daml:Class>
```

3. SEMANTIC WEB APPLICATIONS

There are numerous Semantic Web applications that would be beneficial to operational users and engineers in the aerospace community, such as intelligent notification [9], information retrieval [10], C2 decision support [11] and data analysis workflow for solar science [12]. We will discuss two example applications relevant to aerospace: higher-level information fusion and rapid systems integration.

A common activity in command and control, computers, communication, intelligence, surveillance and reconnaissance (C4ISR) is the fusion of information from multiple heterogeneous sources such as military messages, annotations attached to imagery by analysts and automatically processed sensor data. Figure 2 shows a generic architecture for fusion of battlefield or intelligence data. Level 1 fusion includes processing of raw sensor data such as correlating tracks and parsing messages. In level 2 fusion, numeric and symbolic data from multiple sources is combined to produce situation assessments. In level 3 fusion, all data is combined to assess the threat and analyze enemy intentions. There are three key ideas in this ontology-based approach to fusion. First, the data from all sources is semantically grounded by creating markup that references one or more ontologies. When data from various sources produced at different times are semantically marked up against the same set of ontologies then the collection of markup becomes a virtual knowledge base for fusion reasoning. The second key idea is that facts that are not explicitly stated in any single source can be derived automatically from a reasoner based on knowledge in an ontology and other knowledge bases. The third key idea is that static reference knowledge can be leveraged for fusion. DAML ontologies like OpenCyc and DAML knowledge bases like the CIA World Factbook are becoming available to support this type of fusion reasoning. This approach explicitly stated in any single source can be derived automatically from a reasoner based on knowledge in an ontology and other knowledge bases. The third key idea is that static reference knowledge can be leveraged for fusion. DAML ontologies like OpenCyc and DAML knowledge
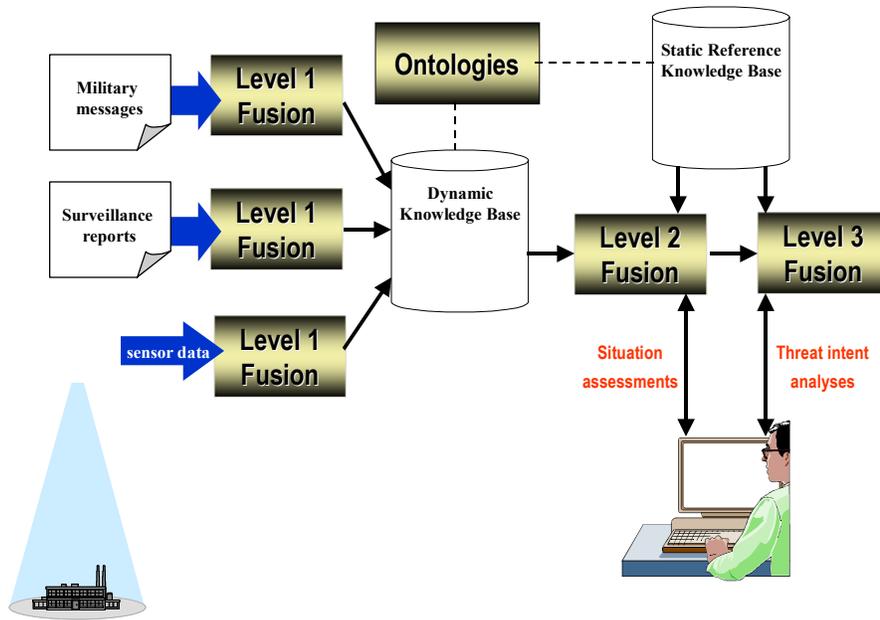
Figure 2: Ontology-based Fusion Architecture

bases like the CIA World Factbook are becoming available to support this type of fusion reasoning. This approach works well in a human-in-the-loop environment where a human can confirm suggestions derived by the reasoner. There is ongoing work in ontology-based fusion for level 2 [13] and level 3 [14].

Semantic interoperability between legacy systems is also a major recurring problem with joint and coalition force operations. Web services are being advocated by industry as an advantageous approach for integrating existing enterprise systems. However, the web service approach still suffers from the problem of semantics buried in procedural code. To overcome this problem, significant efforts are under way to apply Semantic Web concepts to web services. The long-term goal of these efforts are the development of self-integrating systems. The basic idea is to ground the semantics of interfaces by linking the terms into ontologies and leverage these semantics to automatically find and execute appropriate services on demand. A group of researchers has developed a set of DAML ontologies for web services called DAML-S [15]. The design of DAML-S was influenced by AI planning research. DAML-S automates:

- Web service discovery
  - o *Find me a sensor that has the most recent imagery of Kabul.*
- Web service selection and composition
  - o *Plan mission and deconflict the air space*
- Web service invocation*Reconfigure the sensor.*Web service execution monitoring
  - o *Has the sensor been reconfigured yet?*

DAML-S consists of a service profile, a process model and a service grounding. The service profile describes the capabilities the service provides and enables inference-based matchmaking. The process model describes how the service works (i.e., inputs, outputs, preconditions and effects) and the control flow. The service grounding describes how the service can be accessed. The grounding is implementation specific (e.g., SOAP, HTTP, CORBA IDL, Java RMI). DAML-S is designed to work with the lower level W3C Web Services Description Language (WSDL). DAML-S is an important step towards self-integrating systems.

The Lockheed Martin DAML team has built a prototype of a system that uses DAML-S to integrate intelligence, surveillance and reconnaissance (ISR) services for intelligence preparation of the battlefield. The architecture is shown in Figure 3. An imagery agent and a Synthetic Aperture Radar (SAR) agent register advertisements for currently available services with the DAML-S matchmaker. The imagery agent registers inputs that it accepts such as latitude/longitude and image resolution in meters. The SAR agent registers additional required inputs such as frequency band. The Semantic Operational Net Assessment Tool [11] determines a course of action (e.g., destroy target X) and requests supporting imagery. The ISR agent communicates with the matchmaker to identify appropriate services and to invoke services. The CMU DAML team developed a prototype DAML-S matchmaker for use on the web [16]. An advertisement matches a request when the advertisement describes a service that is sufficiently similar to the service
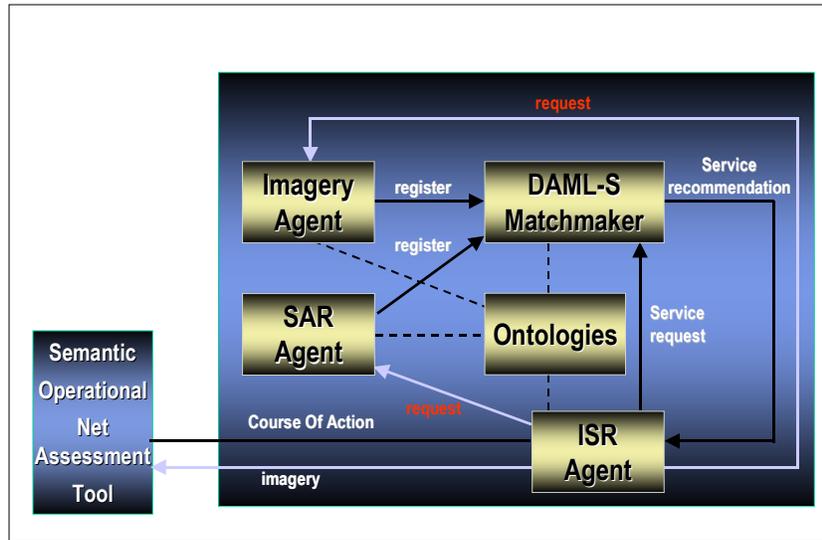
Figure 3: Prototype DAML-S Architecture

requested. The matching algorithm is described in detail in [17]. This prototype system demonstrates the feasibility of the DAML-S semantic interoperability approach. Further experiments are needed to assess how well this approach scales up to real world environments.

4. SEMANTIC WEB CHALLENGES AND RESEARCH

This section will discuss some technical and transition challenges that have been identified in the DAML program and explain ongoing research that is focused on addressing these challenges. These challenges include ontology engineering, markup creation, scalability and ontology mapping/translation.

Ontology Engineering

Semantic Web technology sounds like a silver bullet – what hurdles do I have to overcome to make the magic happen? The development and maintenance of ontologies is fundamentally difficult. Ontologies that take advantage of the expressiveness of Semantic Web languages become complex quickly. Another problem is that ontologies evolve and markup refers to specific versions of ontologies, so there needs to be a systematic way to manage change on the Semantic Web [18].

There are two main approaches that have been investigated to overcome the problems of the complexity of ontologies. The first approach is to develop tools that check the consistency of the knowledge represented in ontologies. The Lockheed Martin DAML team has developed ConsVISor, which takes DAML ontologies as input and reasons with the

formal semantics of DAML to identify contradictory declarations in ontologies [19] [20]. This type of tool is especially useful for the Semantic Web where ontologies import other ontologies and ontologies are developed by extending existing ontologies.

The second main approach has been to investigate visualization techniques for ontologies. Most of the visualization research has focused on developing Unified Modeling Language (UML) front-ends for DAML [21]. There are many advantages to using UML for ontology development:

- UML is a graphical notation based on many years of experience in software analysis and design by a variety of companies in a wide spectrum of industries and domains.
- UML is an open standard maintained by the Object Management Group (OMG).
- UML has standard mechanisms for defining extensions for specific application contexts such as ontology modeling.
- UML is widely adopted in industry and taught in many university courses.
- UML is supported by widely-adopted CASE tools.
- Real-world industrial agent-based systems need to interact with legacy enterprise systems, which often have existing UML models.

The main problem with using UML for DAML is that there are significant semantic incompatibilities between the two modeling languages. For example, UML does not have a

first-class concept of an "association" (analogous to a "property" in DAML). Associations in UML can only exist in the context of two or more classes. Properties in DAML are first-class elements that can be defined in an ontology without reference to classes. If a DAML ontology states that "company owns vehicle" and "person owns dog", "owns" is the same property, whereas they would be different associations in a UML model. We expect that future standardization efforts will resolve these semantic incompatibility issues.

Finally, another problem is that many mainstream software practitioners think procedurally rather than declaratively and do not have the necessary education in knowledge representation and logic. Typical programmers are trained to think about algorithms (i.e., the concrete solution space) and many have trouble assimilating object-oriented analysis (OOA), which focuses on the abstract problem space. Ontology engineering requires declarative thinking like OOA and an understanding of the reasoning that will make use of the knowledge in the ontologies and markup. This problem can only be solved by education and training. Good practitioner-oriented textbooks are clearly needed to support the mindset change from procedural to declarative thinking.

Markup Creation

How much is it going to cost to do markup? The creation of markup from unstructured text sources such as web pages is tedious and time-consuming. Anyone who produces documents on a regular basis (e.g., intelligence analyst, commander) or has a large quantity of legacy documents needs some form of automated markup assistance. The Lockheed Martin DAML team has experimented with the application of information extraction technology to reduce the effort required for markup. They have built a tool called AeroDAML, which automatically generates markup for a number of common domain-independent classes and properties [22] [23]. The author can do markup additions and corrections to the output of AeroDAML via a drag and drop markup tool such as OntoMat [24]. The processing of raw text is very difficult but sufficient levels of precision and recall are being attained to make this automated assistance approach worthwhile. AeroDAML can also be customized for domain-specific markup generation.

The current AeroDAML can generate markup that consists of words (entities) linked to ontologies as instances of classes and relationships that are linked to ontologies as instances of properties. A list of typical examples is shown below:

- Proper nouns – example: *Japan* instanceOf *nation*
- Common nouns – example: *gun* instanceOf *weapon*
- Co-references – example: *Clinton* sameIndividualAs *Bill Clinton* instanceOf *person*
- Measure – example: *22 inches* instanceOf *measure*
- Money – example: *$200* instanceOf *money*

- Absolute date – example: *December 19, 1997* instanceOf *absolute date*
- Organization to location – example: *Tyrolean Airways* to *Austria*
- Person to organization – example: *Bill Gates* to *Microsoft*

AeroDAML also links instances of nations to instances of countries in the DAML CIA World Factbook knowledge-base. This provides query access to extensive data about countries.

Scalability

Will this Semantic Web technology scale up to real-world problems? This is an open question for researchers. Reasoning with complex ontologies and large collections of DAML markup in a knowledge base is known to be computationally intensive. DAML query answering involves both data retrieval and reasoning with ontologies to infer additional information. The question of scalability is best understood as a tradeoff between query answering capability and query execution time. For example, relational databases can answer queries efficiently with large knowledge bases but do little reasoning to infer information that is not explicitly stored. On the other end of the spectrum there are first-order logic theorem provers that are capable of extensive reasoning but they may run for long periods of time and may never terminate execution. There are other systems that fall in the range between these extreme points such as PARKA [25], XSB [26] and FaCT [27]. Researchers at Lehigh University are currently developing a framework to understand these tradeoffs and conduct experiments which will help answer practical questions like "What query answering infrastructure should I choose based on the characteristics of my application?".

Ontology Mapping and Translation

Will the inevitable proliferation of ontologies really solve the semantic interoperability problem? The answer is clearly no. The widespread adoption of ontologies only gets us half-way to semantic interoperability nirvana by forcing the use of explicit semantics. The other major challenge is mapping from one agent's ontology to another agent's ontology. The approaches to solve this problem range from static manually created ontology mappings to dynamic on-demand agent-based negotiation of ontology mappings.

AeroDAML provides a drag-and-drop tool to create static ontology mappings and also includes predefined mappings to popular ontologies. These AeroDAML tools support the generation of markup based on a target ontology that is chosen by the user. Our experience with ontology mapping shows that it is difficult task. Often classes from one ontology are not exact matches for classes in another ontology. Finding good property mappings is even harder.

An alternative general approach is to map ontologies that you control to large domain-independent upper ontologies such as OpenCyc [28] or SUMO [29]. The theory is that if a critical mass of ontologies is mapped to these reference ontologies then automatic translation will be facilitated. Another approach is to store mappings in a library that can be chained together by an ontology translation server [30]. Other DAML researchers have developed a rule-based approach to map ontologies [31]. Robust real-world dynamic ontology translation appears to be feasible but it is still a long-term research goal.

5. CONCLUSIONS

In this paper we have briefly described Semantic Web technologies and discussed potential applications to aerospace. The prototype system discussed in section 3 demonstrates the feasibility of the DAML-S semantic interoperability approach. We explained challenges and corresponding research. We believe that Semantic Web technology will become mainstream information technology and that it will be gradually adopted by the aerospace community. Despite the research challenges presented in this paper, we believe it is worth beginning to adopt this technology now.

REFERENCES

[1] T. Berner-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Scientific American*, May 2001.

[2] J.Hendler and D. McGuinness, "The DARPA Agent Markup Language," *IEEE Intelligent Systems* 15, No. 6:67-73, 2000.

[3] http://www.daml.org

[4] http://www.w3.org/RDF/http://www.w3.org/RDF/

[5] http://www.daml.org/2001/03/daml+oil-index.html

[6] http://www.w3.org/2001/sw/WebOnt

[7] John Sowa, *Knowledge Representation*, Pacific Grove, CA: Brooks/Cole, 2000

[8] D. McGuinness, "Ontologies Come of Age," in Fensel, Hendler, Lieberman, Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential*, Cambridge, MA: MIT Press, 2002.

[9] R. S. Cost, T. Finin, A. Joshi, Y. Peng, F. Perich, C. Nicholas, H. Chen, L. Kagal, Y. Zou, and S. Tolia, "ITTALKS: A Case Study in how the Semantic Web Helps", International Semantic Web Workshop, July 2001, Stanford.

[10] U. Shah, T. Finin and J. Mayfield, "Information Retrieval on the Semantic Web", unpublished, available at http://daml.umbc.edu/papers/.

[11] J. Flynn and M. Dean, "DAML Experiment Plan," unpublished, June 2002.

[12] http://java1.lmsal.com/

[13] M. Kokar, K. Baclawski, and R. Saha, "Situation Awareness and DAML: An Example", unpublished, available at http://vis.home.mindspring.com/index.html.

[14] B. Bell, E. Santos, and S. Brown, "Making Adversary Decision Modeling Tractable with Intent Inference and Information Fusion," in *Proceedings of the 11th Conference on Computer Generated Forces and Behavioral Representation*, May 2002, Orlando, FL.

[15] http://www.daml.org/services/

[16] http://www.damlsmm.ri.cmu.edu/index.html

[17] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities", in *Proceedings of The First International Semantic Web Conference (ISWC)*, June 2002.

[18] J. Heflin, "Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment", Ph.D. Thesis, University of Maryland, College Park, 2001.

[19] http://vis.home.mindspring.com/index.html

[20] K. Baclawski, M. Kokar, R. Waldinger, and P. Kogut, "Consistency Checking of Semantic Web Ontologies," in *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, Sardinia, Italy, June 2002

[21] P. Kogut, S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. Kokar, and J. Smith, "UML for Ontology Development", in *Knowledge Engineering Review Journal Special Issue on Ontologies in Agent Systems*, vol. 17 no. 1, March 2002, pp 61-64.

[22] http://ubot.lockheedmartin.com/ubot/hotdaml/aerodaml.html

[23] P. Kogut and W. Holmes, "AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages", in *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001) Workshop on Knowledge Markup and Semantic Annotation*, Victoria, B.C., October 2001.

[24] http://annotation.semanticweb.org/ontomat.html

[25] http://www.cs.umd.edu/projects/plus/Parka/parka-db.html

[26] http://xsb.sourceforge.net/

[27] http://www.cs.man.ac.uk/~horrocks/FaCT/

[28] http://www.opencyc.org

[29] http://ontology.teknowledge.com/

[30] http://phd1.cs.yale.edu:4040/ontoMerge.html

[31] http://codip.grci.com/Tools/ArtiServicePage.html

BIOGRAPHY

Paul Kogut *is the principal investigator for the Lockheed Martin team in the DARPA DAML program. He has participated in a variety of research projects related to agents, natural language processing and knowledge representation. He was a resident affiliate at the Carnegie Mellon University Software Engineering Institute in 1994. As an employee of the U.S. Army CECOM in the 1980s, he provided software engineering support for Army, Air Force and Marine C2 systems. Dr. Kogut is an adjunct professor at Penn State Great Valley where he teaches an AI course that emphasizes multi-agent systems. He has a Ph.D. in Computer Science from Lehigh University.*

Jeff Heflin *is an assistant professor at Lehigh University. He has been conducting Semantic Web research for over five years, and was one of the developers of SHOE, the first Web ontology language. Dr. Heflin is also a member of the committee that designed the DAML+OIL language. His research focuses on the scalability of semantic web systems, the applications of ontology languages to agents, and the integration of information given dynamic, distributed ontologies such as those needed by the Semantic Web. He received his M.S. and Ph.D. in Computer Science from the University of Maryland.*