

EFFECTIVE KNOWLEDGE MANAGEMENT USING AN ONTOLOGY-BASED APPROACH

Kamal Gella, Concurrent Technologies Corporation, gella@ctc.com
J. Wesley Regian, Concurrent Technologies Corporation, regianj@ctc.com
Laurie Waisel, Concurrent Technologies Corporation, waisel@ctc.com
Paul J. Kovacs, Robert Morris University, kovacs@rmu.edu

ABSTRACT

Effective management of knowledge requires the seamless creation and exchange of information. Using artificial intelligence techniques, ontologies were developed to increase software reuse and knowledge sharing among application users. This paper describes a group of freely available technologies and tools that the authors have used to develop and field knowledge-based applications. The authors then describe the Comprehensive OWL Military Ontology (COMO): a military operations ontology that has been developed using these technologies and tools. Finally a brief outline is provided of two of the applications that have been implemented using COMO. These are the Military Analogical Reasoning System (MARS) and the Planning for Urban Terrain Operations (PLUTO) system.

Keywords: Knowledge Management, Ontology, Semantic Reasoning, Descriptive Logic, and WWW

INTRODUCTION

The word “ontology” has a specific meaning in artificial intelligence (AI). In the domain of philosophy, ontology referred to the subject of existence. In AI, however, an ontology is a formal description of a set of concepts and relationships. Since the publication of Gruber’s seminal paper on ontologies [5], these hierarchical formalized descriptions have become indispensable in AI research and development. One instance of using ontologies to support very large-scale knowledge sharing is the Semantic Web.

The Semantic Web is the next generation World Wide Web (W3), in which information has meaning associated with it (via metadata) in order to enable more sophisticated machine processing than was previously possible [1]. The W3 Consortium has proposed guidelines for the standardization of ontologies and ontology tools that support the Semantic Web. Two Semantic Web technologies,

Resource Description Framework (RDF) and Web Ontology Language (OWL), were approved in 2004 by the W3 consortium. RDF and OWL are used to share and reuse knowledge among different entities.

In order to support very large-scale knowledge sharing, the Semantic Web requires ontologies with a significant degree of structure. Thus it was necessary to specify descriptions for the following kinds of concepts [6]:

- The classes (general things) in many domains of interest.
- The relationships that can exist among classes.
- The properties (or attributes) those classes may have.

ONTOLOGY APPLICATIONS

Semantic Web technologies are used to develop web-enabled knowledge applications. These applications include:

- *Web Portals.* Web applications that changes for different types of users. Users type in key words, which are processed by the Semantic Web to produce results.
- *Metadata-Driven Applications.* Metadata is data about data, including but not limited to, the time and data the data were collected, the location from which they were obtained, and a description of the data. In metadata-driven applications, documents can be retrieved not only by a keyword search, but also by searching the metadata fields.
- *Intelligent Agent Applications.* A search agent that looks for specific information on the web is an example of an intelligent agent. An intelligent agent is a piece of software that is capable of autonomous behavior, responds to changes in its environment, and persistently pursues goals [11]. Ontologies frequently are used in combination with intelligent agents.

Web-Based Ontologies

The Web Ontology Language (OWL) is a family of knowledge representation languages that can be used to develop ontology applications and to enhance knowledge sharing on the Semantic Web. OWL has three sublanguages that are used for different purposes. They are OWL Lite, OWL DL, and OWL Full.

- *OWL Lite* is used when minimal language specification is needed. It is an easy-to-use language designed for the development of simple taxonomies.
- *OWL DL* is used when more expressive constructs are required. It includes all OWL language constructs and supports parent-child relationships in a class hierarchy. RDF syntax is used for OWL DL's description logic.
- *OWL Full* extends the construction of OWL Lite by supporting instantiation of classes, also called enumerated classes. Complex class descriptions that include enumerated classes, property restrictions, and boolean combinations are possible with OWL Full.

Ontologies conforming to OWL standards allow navigation of the internet based on semantic concepts. This approach integrates heterogeneous distributed knowledge and supports semantic querying based on the meaning of concepts rather than searching for text string matches, i.e., keywords.

ONTOLOGY TOOLS

Protégé

Protégé was developed by the Knowledge Modeling Group at Stanford Medical Informatics. Protégé is an ontology editor that is designed for the creation of large knowledge bases [13]. Protégé-Frames is used for constructing frame-based domain ontologies, and Protégé-OWL is used for constructing OWL-compliant ontologies. The ontologies created by Protégé, which is implemented in Java, can be exported to a different format such as RDF, OWL, and XML schema. Other applications can access the knowledge base using the `getKnowledgeBase()` API. Protégé is available as an open source application and has a strong user community, the academic and government users in that community develop diverse applications ranging from modeling and simulation to geospatial tools.

OntoEdit

OntoEdit is an Ontology Engineering Environment developed by the Institute of Applied Informatics and Formal Description Methods at University of Karlsruhe [10]. The graphical interface allows users to develop ontologies that can be exported to object-relational database schema and data transformation services. OntoEdit allows the user to directly edit the class hierarchy. It also provides a cut-and-paste feature for moving parts of one ontology to another. OntoEdit supports the development of knowledge intensive applications such as multilingual ontologies.

WebOnto

WebOnto was developed at the Open University in Milton Keynes. The main objective of the tool was to provide a web-ontology editor that can support browsing, visualization, and editing [16]. The tool is implemented as a Java Applet that provides coarse grained and fine grained browsing. Client APIs allow the retrieval of ontologies built using WebOnt. WebOnto is used to develop e-commerce application that requires rich collaborative support for browsing, creating, and editing ontologies.

OilEd

OilEd is a simple ontology editor developed at the University of Manchester. OilEd allows the user to build ontologies, use a FaCT (Fast Classification of Terminologies) reasoner to check the consistency of ontologies and add implicit subclassOf relations, and export ontologies in a number of formats including both OIL-RDF and DAML-RDF [9] OIL defines classes and organize them in a hierarchy. Applications that require information to be processed by machines are developed using OilEd. Examples include search engines, e-commerce applications, and Intranets.

REASONING TOOLS

Ontologies are simply knowledge representations. They do not include reasoning logic and thus do not actually do anything useful until paired with semantic reasoning logic. A variety of open source tools support developing efficient constructing logic for reasoning over ontologies. These tools number in the hundreds and vary greatly in both quality of implementation and in their intended application. In selecting a reasoning tool, one should look for a tool that is intended to do what you want it to do and that has an active user community. It is also a good idea to select a tool that is implemented in a programming language you are familiar with. Here are four

reasoning tools that have been found to be useful. These are all well supported by active user and developer communities.

Jena

Jena is a programming environment for RDF, RDFS and OWL [7], and includes a rule-based inference engine. Jena is implemented in Java.

Pellet

Pellet is an open-source OWL descriptive logic reasoned [12] that is implemented in Java. Pellet can be downloaded and included in other applications.

FaCT++

FaCT++ is a descriptive logic reasoned [2] that can reason over OWL DL or OWL 2 ontologies. It is implemented in C++.

ONTOLOGY EXAMPLE

COMO

The Comprehensive OWL Military Ontology (COMO) was developed at Concurrent Technologies Corporation (CTC) (using the tools and techniques described above) and supports a diverse variety of knowledge-based military machine-reasoning applications. CTC has already developed applications for analogical reasoning across military stories, automated derivation of lessons learned from prior military scenarios, application of lessons learned to current military scenarios, probabilistic success evaluation of possible ingress and egress routes to accomplish mission objectives, and prediction of possible scenario outcomes based on prior known outcomes.

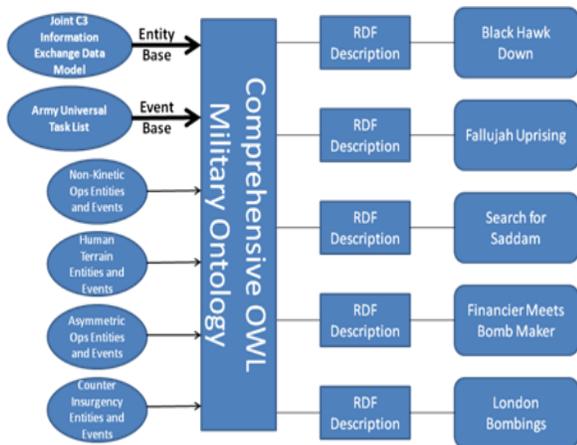


Figure 1: COMO Framework

COMO is a 1,600-concept ontology optimized for the modern military domain. Written in the OWL Web Ontology Language, this ontology covers a broad spectrum of military entities, tasks, and events, with extensive cross-referencing to NATO’s Joint Command, Control, and Coalition Information Exchange Data Model (JC3IEDM) and the Army Universal Task List (AUTL). COMO uses JC3IEDM and AUTL as its main object/entity and task/event base (respectively). Thus, COMO-based systems are able to easily interact with other information systems that adhere to NATO and U.S. information processing standards. Furthermore, in addition to supporting these important aforementioned standards, COMO also contains events and entities that we added, which are not based on JC3IEDM or AUTL. Both JC3IEDM and AUTL reflect a focus on traditional force on force military operations, training, and logistics. In order to support representation and reasoning about many of the kinds of operations of interest to the U.S. Military today, it is necessary to add events and entities that are required to represent scenarios about non-kinetic, human terrain, asymmetric, and counterinsurgency operations.

Below is an example of a non-traditional military situation scenario represented as an instance in COMO. The scenario represented here (Financier Meets Bomb-Maker) is that a terror financier met with a bomb-maker at the Camel Dog Café. The terror financier gave the bomb-maker some money. Here is the RDF code to represent this instance.

```

</Bombmaker>
</takes_object_filler_of>
<takes_agent_filler_of>
<Terror_Financier
rdf:ID="Terror_Financier_AAAA001">
  <fills_agent_slot_of
rdf:resource="#sAAAA_e0001"/>
  <fills_agent_slot_of
rdf:resource="#sAAAA_e0000"/>
</Terror_Financier>
</takes_agent_filler_of>
<takes_action_filler_of>
<Meeting rdf:ID="Meeting_AAAA001">
  <fills_action_slot_of
rdf:resource="#sAAAA_e0000"/>
</Meeting>
</takes_action_filler_of>
<takes_place_filler_of
rdf:resource="#Camel_Dog_Cafe"/>
</Story_Event>
</fills_place_slot_of>
</Cafe>
</takes_place_filler_of>
<takes_agent_filler_of
rdf:resource="#Terror_Financier_AAAA001"/>
<takes_receiver_filler_of
rdf:resource="#Bombmaker_AAAA001"/>

```

```
<is_preceded_immediately_by
rdf:resource="#sAAAA_e0000"/>
<takes_action_filler_of>
<Giving rdf:ID="Giving_AAAA001">
  <fills_action_slot_of rdf:resource="#sAAAA_e0001"/>
</Giving>
</takes_action_filler_of>
</Story_Event>
```

EXAMPLE APPLICATIONS

COMO is designed to be a general knowledge representation within the military domain, and makes no attempt to provide knowledge representation in support of reasoning outside the military domain. The intention is that within the military domain, COMO should be able to express all important knowledge, and support many kinds of reasoning applications, problem-solving solutions, decision support systems, and other human performance support methodologies. This section describes two very different reasoning applications that have developed, both based on the COMO military knowledge representation ontology.

MARS

The Military Analogical Reasoning System (MARS) is a prototype performance support system and decision aid for commanders in Tactical Operations Centers. MARS enhances and supports the innate human ability for using stories to reason about tactical goals, plans, situations, and outcomes.

MARS operates by comparing many instances of stored tactical stories, determining which include analogous situations and thus related lessons learned, and then returning a description of the relevant lessons learned. The returned description of the lessons learned is at a level of abstraction that can be generalized to a range of tactical situations. The machine-understandable story representations are based on a military operations data model and are represented in COMO as instances of unfolding tactical situations.

Analogical reasoning appears to be a cognitively easy task for human beings, yet is known to be very difficult to automate, or to accomplish in software. MARS implements two kinds of automated analogical reasoning. In the first kind, the authors would like to compare two stories (two structured representations, which are refer to, , as base and target [4]). The question to be answered is “What does this story tell me about that one?” We refer to this as simple story comparison. It is simple in that it compares only two stories, both stored rather than live. In the second kind of automated analogical reasoning, we search through a (potentially large) corpus of stories and answer the questions “Which among these many stories can tell me something useful about this unfolding target story?” and then “What do these few selected base stories tell me about this unfolding target story?” We refer to these two examples of the second kind of automated reasoning as story selection and advanced story comparison. Advanced story comparison differs from simple story comparison in that it compares multiple stories, and the target story is live and unfolding rather than stored. We consider both kinds of reasoning analogical, and we accomplish them by drawing analogies from structured, formal representations of stories.

Our analogical reasoning algorithm is based on Structure Mapping Theory [3]. Consider the following two stories. In the first, a U.S. Platoon in Vietnam diverts around a minefield and subsequently comes under ambush from a large hill overlooking their new position. In the second, a U.S. Task Force in Iraq diverts around a biochemical hazard and subsequently comes under ambush from the roof of an abandoned building. MARS recognizes these stories as analogical, and derives the following abstraction: When enemy-placed obstacles force us into an unplanned route, beware of ambush from elevation or concealment.

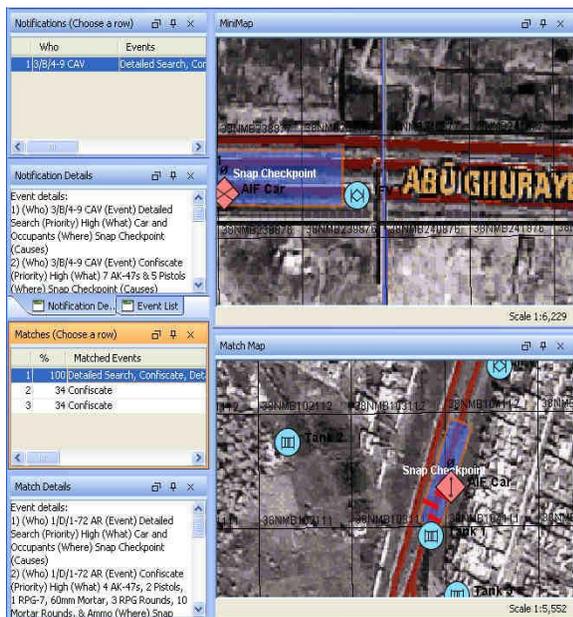


Figure 2: Military Analogical Reasoning System

PLUTO

The Planning for Urban Terrain Operations (PLUTO) software system leverages 3D models of urban areas to conduct geospatial reasoning about objects, locations, lines of sight, and vulnerabilities in support of urban mission plan analysis.

The 3D area models used by PLUTO are pre-interpreted such that important entities, objects, and locations have been manually tagged by intelligence analysts – with those tags being labels that correspond to military entities in COMO. These tagged models are submitted to PLUTO along with a plan specifying (a) formal mission objectives and parameters, (b) one or more proposed ingress and egress routes from a safe area to the target location and back again, and (c) the modes of transport to be used along segments of those route(s).



Figure 3: Planning for Urban Terrain Operations

The system returns quantified indices reflecting the quality of the plan, and also identifies danger areas along the routes, specifying why they are dangerous. Reasoning about the plan is informed by the most recent version of COMO, which (in addition to knowledge of military entities, tasks, and events) also knows about geospatial object shapes, locations, and classifications. Much of the low-level reasoning in PLUTO is implemented using Pellet (an OWL reasoning and query system), but more complex reasoning logic is implemented in Java.

CONCLUSIONS

Ontology-based applications increase knowledge sharing by combining machine learning, knowledge extraction, and linguistic techniques. After years working in the field of AI, the authors are surprised to find that many of the best tools available today are free, well-supported by large communities of well-qualified users, fully web-enabled, and run on inexpensive hardware. This is in stark contrast to our early work in AI, which required expensive special-purpose stand-alone hardware and software environments. Today’s AI technologies and tools facilitate effective real time knowledge management by allowing large-scale machine-mediated web-enabled knowledge sharing and reuse.

REFERENCES

- Daly, J., Forgue, M., & Hiraakawa, Y. (2004). World Wide Web consortium issues RDF and OWL recommendations. Available: <http://www.w3.org/2004/01/sws-pressrelease>
- Fact++. Available: <http://owl.man.ac.uk/factplusplus/>
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, Vol. 7, 155-170.
- Gentner, D. and Forbus K. (1991). MAC/FAC: A model of similarity-based retrieval. *Proceedings of the Cognitive Science Society*.
- Gruber, T.R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220. Available: <http://tomgruber.org/writing/ontolingua-kaj-1993.pdf>.
- Heflin, J. (2004). OWL Web ontology language use cases and requirements. Available: <http://www.w3.org/TR/webont-req/>
- Jena, Available: <http://jena.sourceforge.net/>
- McGuinness, D., & Harmelen, F. (2004). OWL Web ontology language overview. Available: <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.3>
- OilED, Available: <http://xml.coverpages.org/oilEdANn20001204.html>
- Ontoedit, Available: <http://www.ontoknowledge.org/tools/ontoedit.shtml>
- Padgham, L. & Winikoff, M. (2004) Agents and multi-agent systems. In: *Developing intelligent agent systems: a practical guide* (Ed, Woolridge, M.), pp. 1-6. John Wiley & Sons, West Sussex, UK.
- Pellet, Available: <http://clarkparsia.com/pellet/>

13. Protégé, Available: <http://protege.stanford.edu/>
14. Regian, J. W. Denny, M., Desmedt, W., Waisel, L. (2007). Automated Reasoning Across Tactical Stories to Derive Lessons Learned. *Proceedings of The 5th International Conference on Computing, Communications and Control Technologies (CCCT)*, Orlando, Florida, July 12-15.
15. Regian, J. W. Denny, M., DeSmedt, W., Gella, K. and Waisel, L. (2008). Geospatial Reasoning about Military Operations on Urban. Submitted to *11th International Conference on Knowledge Representation and Reasoning*, Sydney, Australia.
16. WebOnto, Available: <http://www.aktors.org/technologies/webonto/>