A Graphical Environment for Ontology Development

Luigi Ceccaroni
Fujitsu Laboratories of America
595 Lawrence Expressway
Sunnyvale, CA 94085, USA
luigi.ceccaroni@lycos.com

Elisa Kendall
Sandpiper Software, Inc.
2053 Grant Road, #162
Los Altos, CA 94024, USA
+1 650 960 2456
ekendall@sandsoft.com

ABSTRACT
This paper presents current research on a semantically rich, graphical representation of ontologies and its utility for collaborative construction based on requirements outlined by the Agentcities initiative. A new tool, called the Visual Ontology Modeler, is described and evaluated in the context of Agentcities. Its distinguishing qualities include: ease of use, multi-user configuration management, integrated consistency and completeness checking, automated export of DAML+OIL code. The application domain is an open, dynamic test-bed for agent deployment; the ontologies are encoded in DAML+OIL and explicitly designed to be shared by several agent-based services within this environment.

Categories and Subject Descriptors
I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods – frames and scripts, representation languages, representations, semantic networks.

General Terms
Design, Experimentation, Standardization, Languages.

Keywords
Ontologies, Agentcities, DAML+OIL, Web Ontology Language (OWL), Unified Modeling Language (UML).

1. INTRODUCTION
We describe a new tool for semantically rich, graphical representation of ontologies. Using the tool, substantial productivity gains are expected in (1) development time, (2) consistency and correctness of the DAML+OIL code, and (3) ease in maintenance when changes are introduced. This tool extends Rational Rose and enables ontology development with user-friendly wizards, automatically creating the logical model and related diagrams. The Agentcities.RTD project [http://www.agentcities.org/EURTD/] provides a backdrop for evaluation purposes. Agentcities is part of a worldwide initiative intended to realize and advance the potential of agent-based applications by constructing an open, distributed network of platforms hosting diverse agents and services.

2. ONTOLOGY REQUIREMENTS
In early 2002, several publicly available ontology editors were considered for use in Agentcities, but none inherently supported DAML+OIL or provided a logically consistent, graphical notation for representing ontologies. By mid-year, several important new options emerged that directly support the OWL or DAML+OIL languages for ontology development. In some cases, methodologies for object-oriented design, such as the Object Management Group’s Unified Modeling Language (UML), have been applied or extended to support ontology development [3]. Most UML tools are multi-user and many are integrated with commercial-grade configuration management capabilities. Of the tools reviewed, however, few addressed Agentcities’ requirements for large-scale, collaborative ontology-development: (1) support for OWL or DAML+OIL modeling and/or production, (2) an easy to use, easy to convey graphical representation, (3) multi-user development, (4) configuration management capabilities (e.g., version control, with parallel management of multiple versions of the same ontology), (5) the ability to analyze and compose multiple models at a time, (6) integrated consistency and completeness checking, and (7) automated import/export of OWL. Deliverable 1.3.1 of the OntoWeb project [http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip] provides a broad survey of numerous tools.

Until recently, partners in large-scale agent-systems development exchanged the source code for ontologies and a corresponding graphical view created in Visio, PowerPoint or Rational Rose. This approach is limited: there is no automated logical relation between the two, and the modification of such ontologies often leads to confusion and inaccuracy. Collaborative construction and reuse is essential in large-scale agent systems projects due to a need for a high degree of interoperability and sharing of a consistent, visual representation of the ontologies among team members. Text-based ontologies can be hard to read, particularly if there is no agreement on development methodologies or code format. A formal, graphical representation is not only more readable, but provides a much more consistent vehicle for conveying ontological concepts and for sharing them with other domain experts not versed in the representation language. Automated target-language generation would ensure consistency and correctness in the encoding and result in substantial time-savings, particularly in maintaining large algorithms.

3. VISUAL ONTOLOGY MODELER
Here we introduce a new tool, Sandpiper’s Visual Ontology Modeler (VOM), and detail its performance against Agentcities requirements. The VOM is a UML-based development environment, supporting automated generation of DAML+OIL. It
extends Rational Rose and enables ontology construction through user-friendly wizards, automatically creating the required UML model elements and related diagrams. The VOM enables graphical model development in a standards-based notation and exports most required constructs for large-scale agent systems’ ontologies (e.g., disjointWith, complementOf, sameClassAs). We implemented several existing ontologies used in such systems without modification to evaluate the tool’s capabilities as well as the DAML+OIL code produced. The VOM’s performance with respect to stated requirements leads us to recommend it for the creation of all new ontologies as well as for maintenance of existing ones.

In Figure 1, at the right side, the graphical representation of an ontology is shown in UML notation. A similar graphical view is achievable with a number of UML modeling tools. The left side of Figure 1 provides a logical view of the same ontology. While a number of ontology tools can model the logical or hierarchical view of an ontology, few have implemented the underlying semantics to produce both logical and graphical perspectives and only VOM can generate the code associated with both representations. Additionally, the VOM makes it possible to automatically publish fully browseable HTML versions of these representations (see Figure 1) for review by all distributed team members. We believe the functionality provided by the VOM will be increasingly important for agent-system engineers who need tight interoperability among services and are members of distributed teams or constructing multiple, interrelated ontologies for agent use.

Enhancements in work include import of DAML+OIL, OWL and RDFS ontologies and tighter integration with configuration management capabilities. Sandpiper currently provides a library of base ontologies with the tool as a starting point for ontology development, and is working with beta users to identify high priority ontologies for inclusion, such as those that define concepts specified by ISO metadata standards. Planned research includes additional collaboration experiments with distributed teams as well as methodology development.

4. ACKNOWLEDGEMENTS

The authors wish to extend their thanks to Jonathan Dale and Dominic Greenwood (Fujitsu) and Susan Root (Sandpiper) for the feedback after having read a preliminary version of this paper. The research described is partly supported by the EC project Agentcities.RTD (IST-2000-28385). The opinions expressed in this paper are those of the authors and are not necessarily those of the EU Agentcities.RTD partners.

5. REFERENCES

