Ontology Versioning Support: The DIONE Project[*]

Brian Kettler, Peter Haglich, Jason Losco, Kevin Spivey, Jim Starz, and Bonnie Swart
ISX Corporation
4301 N. Fairfax Dr., Suite 370, Arlington, VA 22203
bkettler@isx.com

1. Introduction

The Semantic Web vision describes a web of content that machines, as well as humans, can exploit. This in turn requires rich, semantic representations for content that make its meaning explicit and machine-processable. Languages such as the W3C's Web Ontology Language (OWL) use ontologies to provide these semantics. These ontologies specify concepts, relationships among concepts, and rules. Humans, software agents, and other applications can use ontologies to markup, organize, retrieve, and process data. As ontologies proliferate, the number of dependencies involving them grows. Ontologies published on the web will spread in use and many of these dependencies will be unknown. Techniques, tools, and methodologies are needed to manage the inevitable evolution of these ontologies in response to changes in domains they represent, the conceptualization of those domains, and the formal, machine-processable specifications of those conceptualizations. Without this management, ontology changes will break applications and render marked up data incomprehensible.

Building on database schema versioning work, a number of researchers have begun to investigate the problem of ontology versioning and evolution. The **DAML Integrated Ontology Evolution (DIONE) project**, sponsored by DARPA DAML, has begun to apply the results of this research to support pilot semantic web efforts for the Intelligence/DOD communities. DIONE has developed a number of open source tools for ontology versioning.[1]

2. DIONE Tools

DIONE tools are addressing two major aspects of ontology versioning. The **VPI (Virtual Perspective Interface)** computes and applies ontology perspectives to access knowledge in a knowledge base (KB) derived from OWL markup using different ontology versions. Unlike an approach involve bulk translation of markup to migrate to new ontology versions, this supports multiple KB clients using differing versions. VPI is usable with various kinds of OWL (and RDF) knowledge bases. VPI and the theory of perspectives is described elsewhere.[2]

The **ONTADEPT (Ontology Administration and Deployment Toolkit) tools** support a knowledge engineer in managing dependent schemas including (OWL) ontologies; relational database and XML schemas for data to import and store in a repository (knowledge base) of OWL instance data; mappings from ontologies based on the those schemas to the knowledge base ontologies; and markup templates for the Semantic Markup Tool.[3] These tools are described below.

Figure 1 shows a flow in which a change to an ontology version (O.1) triggers a process of analyzing, executing, and propagating changes to a number of kinds of schemas including dependent ontologies (which import O.1), mappings from ontologies to O.1, markup templates that reference O.1, etc. This is complicated by a number of factors such as:

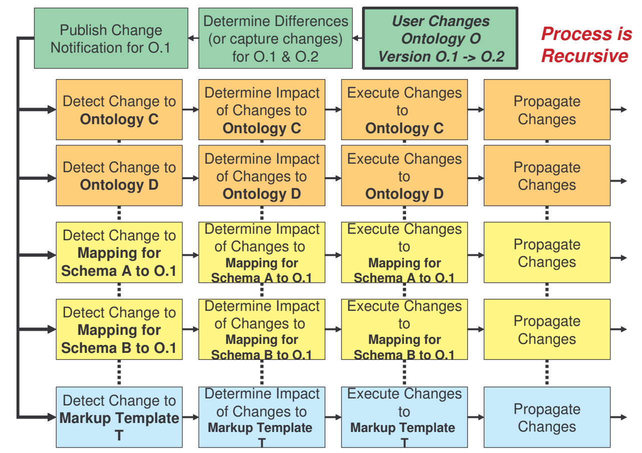
Publish Change Notification for O.1
Determine Differences (or capture changes) for O.1 & O.2
User Changes Ontology O Version O.1 -> O.2
Process is Recursive
Detect Change to Ontology C
Determine Impact of Changes to Ontology C
Execute Changes to Ontology C
Propagate Changes
Detect Change to Ontology D
Determine Impact of Changes to Ontology D
Execute Changes to Ontology D
Propagate Changes
Detect Change to Mapping for Schema A to O.1
Determine Impact of Changes to Mapping for Schema A to O.1
Execute Changes to Mapping for Schema A to O.1
Propagate Changes
Detect Change to Mapping for Schema B to O.1
Determine Impact of Changes to Mapping for Schema B to O.1
Execute Changes to Mapping for Schema B to O.1
Propagate Changes
Detect Change to Markup Template T
Determine Impact of Changes to Markup Template T
Execute Changes to Markup Template T
Propagate Changes

Figure 1. The Versioning Process

[*] The DIONE work has been supported by DARPA through the Space and Naval Warfare Systems Center. We wish to thank the former and current DAML Program Managers – Dr. Mark Greaves and Dr. Robert Popp – and our partners in this work, Lehigh University (Prof. Jeff Heflin).

[1] www.semwebcentral.org

[2] e.g., J. Heflin, Z. Pan. A Model Theoretic Semantics for Ontology Versioning. *Third International Semantic Web Conference*, Hiroshima, Japan, LNCS 3298 Springer, 2004, pp. 62-76.

[3] B. Kettler, J. Starz, W. Miller, and P. Haglich. Template-based Markup Tool for Semantic Web Content. To appear in *Proceedings of the 2005 International Semantic Web Conference.*

- the process is recursive (e.g., a change to a dependent ontology may trigger a "sub-invocation" of the process)
- the process may involve multiple, (spatially and temporally) distributed knowledge engineers
- in the Web, many downstream dependencies on an ontology or other type of schema may be unknown
- changes are not generally logged to ontologies, especially not in machine-understandable form
- automated change detection assumes access to the prior version of an ontology to determine the delta.[4]

To support the above, ONTADEPT includes tools ("vertical services") designed to support specific types of schemas (e.g., ontology mappings and SMT templates) and generic tools ("horizontal services") such as the ONTADEPT Change Manager described below.

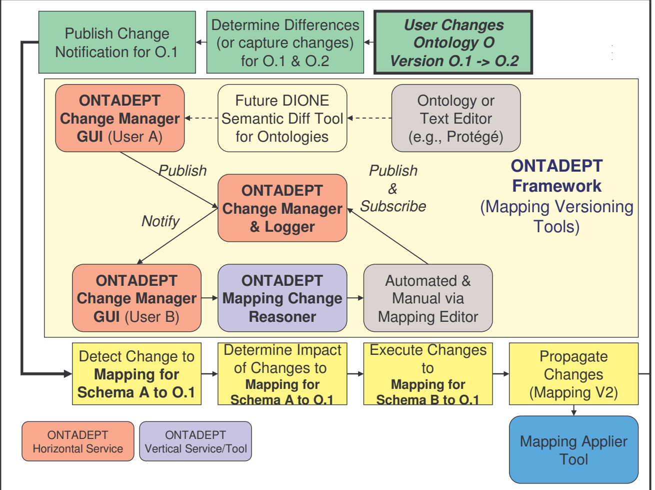
Publish Change Notification for O.1
Determine Differences (or capture changes) for O.1 & O.2
User Changes Ontology O Version O.1 -> O.2
ONTADEPT Change Manager GUI (User A)
Future DIONE Semantic Diff Tool for Ontologies
Ontology or Text Editor (e.g., Protégé)
Publish
Publish & Subscribe
ONTADEPT Framework (Mapping Versioning Tools)
Notify
ONTADEPT Change Manager & Logger
ONTADEPT Change Manager GUI (User B)
ONTADEPT Mapping Change Reasoner
Automated & Manual via Mapping Editor
Detect Change to Mapping for Schema A to O.1
Determine Impact of Changes to Mapping for Schema A to O.1
Execute Changes to Mapping for Schema B to O.1
Propagate Changes (Mapping V2)
Mapping Applier Tool
ONTADEPT Horizontal Service
ONTADEPT Vertical Service/Tool

Figure 2. ONTADEPT Tools for Ontology Mappings

Figure 2 shows the use of ONTADEPT to support change detection, change determination, change impact analysis, and change propagation/execution for mappings between ontologies. These mappings are manually created between OWL ontologies used in a knowledge base (KB) of instance data for a semantic web pilot application and ontologies describing reference data for this application, including the CIA Factbook (OWL) data and Factiva notable persons data.[5] Mappings between these ontologies are files of SWRL rules (using the SWRL XML syntax). Our **Mapping Applier Tool** translates SWRL rules into rules for Jess.[6] These rules are then executed on the instance data to be translated by Drexel University's OWLJessKB into the target ontologies used by the application KB. The facts asserted by the Jess rules are then serialized out as OWL instance data, which stored later in the KB as reference data.

If a data source ontology or KB ontology changes, these mappings may need to be modified using following ONTADEPT tools and process, shown in Figure 2:

- the **ONTADEPT Change Manager (OCM)** supports the distributed management of dependencies through a publish-and-subscribe mechanism for change notifications, change proposals, and change requests. Users and client tools can publish changes and subscribe to them at varying levels of granularity, supported by an extensible OWL ontology of changes. Subscriptions are generally at the file level. Implicit in these subscriptions are dependencies between files that are manually and automatically determined depending on the type of schemas involved. Currently ontology changes must be triggered by the manual entry via the OCM's GUI of a change notification by the ontology author.[7]
- The **OCM** server receives and processes change publications, matching them to subscriptions (using an ontology of changes). Notifications are sent to subscribers via email and are also viewable by the OCM's GUI.
- The **OCM's Change Logger** maintains a history (in OWL) of schema versions and specific changes made
- The receipt of a change notification by the OCM server might trigger a schema-specific reasoner that will augment the change notification with details about downstream changes to be made and perhaps execute those changes automatically. Here the **Mapping Change Reasoner (MCR)** is triggered by notifications to subscriptions for changes to ontologies referenced in mapping files. The MCR reasons about the impact of ontology changes to the mapping using a modified version of BBN's SWRL Validator[8] for detecting broken mappings (e.g., due to a deleted class) and additional logic that suggests modifications to the mapping (e.g., due to an added ontology class). The MCR obtains information on ontology changes from the **DIONE Semantic Diff Tool** currently under development. The impact of the ontology change is then handled automatically where possible: e.g., if an ontology class has been renamed in the newer version, that name is changed in the mapping file. Change Reasoners have been implemented for other schema types: e.g., the Semantic Markup Tool templates, which reference ontologies.

3. Conclusion

The DIONE ONTADEPT tools are extensible and have been shown to be effective on a pilot semantic web application. Further work is needed to extend these to other kinds of schema types, augment their automated reasoning, and to better integrate them with schema authoring tools.

[4] Under DIONE, Lehigh is exploring using DL reasoner to augment a structural comparison tool such as Stanford's PROMPT tool to perform a "semantic diff" on ontologies.

[5] We use BBN's OWL ontology for the CIA Factbook (http://www.daml.org/2003/09/factbook/.) The Factiva XML data (from www.factiva.com) is converted to OWL.

[6] This approach is based on work by Mike Dean at http://www.daml.org/2004/05/swrl-translation/Overview.html

[7] The GUI supports creating, viewing, and managing subscriptions and alerts. A web-based GUI exists as well as a GUI plug-in to Eclipse (www.eclipse.org). We are experimenting with a plug-in to Stanford's Protégé ontology editor to send ontology change notifications to the OCM automatically.

[8] http://owl.bbn.com/