Assessment Report

Sean Bechhofer & Ian Horrocks
University of Manchester
Kilburn Building
Oxford Road
Manchester M13 9PL
email: {seanb,horrocks}@cs.man.ac.uk

Identifier	Del 23
Class	Deliverable
Version	1.0
Date	14-06-2004
Status	Final
Distribution	Public
Lead Partner	VUM

WonderWeb Project

This document forms part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2001-33052.

For further information about WonderWeb, please contact the project co-ordinator:

Ian Horrocks
The Victoria University of Manchester
Department of Computer Science
Kilburn Building
Oxford Road
Manchester M13 9PL
Tel: +44 161 275 6154
Fax: +44 161 275 6236
Email: wonderweb-info@lists.man.ac.uk

Contents

Executive Summary

This document provides an overview of the use of WonderWeb technology, in particular the use of ontology languages, in real world applications.

1 Introduction

The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. The Semantic Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for the Semantic Web is an ontology language that can formally describe the meaning of terminology used in Web documents. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema. In February 2004, the W3C released the Web Ontology Language OWL as a Recommendation [11]. OWL is used to publish and share ontologies, supporting advanced Web search, software agents and knowledge management.

OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is a revision of the DAML+OIL [6] web ontology language incorporating lessons learned from the design and application of DAML+OIL.

The definition of OWL was motivated by a number of Use Cases (detailed in the OWL Use Cases and Requirements Document [8], which also provides more details on ontologies, and formulates design goals, requirements and objectives for OWL. OWL has been designed to meet the need for a Web Ontology Language and is part of the growing stack of W3C recommendations related to the Semantic Web.

- XML provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
- XML Schema is a language for restricting the structure of XML documents and also extends XML with datatypes.
- RDF is a datamodel for objects ("resources") and relations between them, provides a simple semantics for this datamodel, and these datamodels can be represented in an XML syntax.
- RDF Schema is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes.
- OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

1.1 OWL Impact

OWL was released as a W3C Recommendation in February 2004, although the OWL specification was reasonably stable some time before this. A number of research projects (including some of those discussed below) based their early implementations on DAML+OIL as a representation language. The fact that OWL builds on DAML+OIL, however, means that migration paths for applications using the earlier language specification are available and relatively straightforward – as a result, there are a number of applications that are now seen to be using OWL as a representation language for ontologies.

In addition, we are beginning to see real interest in the application of Semantic Web technologies such as OWL in the commercial and business world. For example, a recent panel session at the Thirteenth International World Wide Web Conference[1] held in New York in May 2004 included presentations from companies including Adobe, Brandsoft, IBM, Network Inference and Hewlett Packard. All of these companies (and others) are exploring solutions that make use of Semantic Web technology including OWL. This is discussed in more detail in Section 3.

A number of projects in the Bioinformatics domain are now using OWL – Bioinformatics has a history as an early adopter of computer science and web technology – although there are also examples in other fields such as fluid dynamics and mathematics. These projects include:

- myGrid. A UK e-science project that makes use of ontologies for describing services and workflows, and for the description of provenance information.
- AKT. A multi-million pound, six-year collaboration between research groups in the UK from the Universities of Aberdeen, Edinburgh, the Open University, Sheffield.
- GEODISE. A UK project which aims to provide a Grid-based generic integration framework for computation and data intensive multidisciplinary design optimisation tasks while maintaining the autonomy of each individual domain expert. The GEODISE demonstrator is intended to carry out engineering design search and optimisation involving fluid dynamics, and is bringing together the collective skills of engineers and computer scientists. GEODISE makes use of ontologies to:
- GONG. The Gene Ontology Next Generation (GONG) project is investigating the use of OWL to remodel a large vocabulary – the Gene Ontology (GO). The use of OWL allows the capture of implicit knowledge explicitly, facilitating error checking and the production of coherent taxonomies within the vocabulary.
- GOAT. The Gene Ontology Annotation Tool (GOAT) aims to create an application that will guides users, especially biomedical researchers, in the annotation of gene products with terms from the Gene Ontology. This will makes use of the enriched version of the Gene Ontology that is the result of the GONG project.

[1] `http://www.www2004.org`

- MONET. An EU funded project, concerned with the development of a framework for the description and provision of web-based mathematical services. MONET uses an OWL ontology for the description of services.
- S-MOBY. MOBY is a project to design, test, and implement a distributed data retrieval and analysis architecture operational on the World Wide Web. The MOBY Semantic Architecture (S-MOBY) is a development effort within the larger MOBY project to explore the use of the Semantic Web and RDF (Resource Description Framework) as a respective conceptual framework and technology to realize this vision. S-MOBY makes use of OWL for representing ontologies.
- NCI. The NCI thesaurus has been represented using OWL
- GEON. Infrastructure to support geoscientists.
- BioPAX. An initiative developing a common exchange format for biological pathways data.
- Protege. Perrhaps the best known ontology editor is now supporting OWL.
- Industrial and Commercial Applications. Industrial and commercial uses of OWL are now coming on-stream.

In the following sections, we highlight the contributions made by the WonderWeb project and describe in more detail the use of ontologies, and OWL in particular, within some of the projects mentioned. This should not be taken as a comprehensive list of OWL usage – there are many other uses of OWL which are not mentioned here.

2 Project Contributions

The WonderWeb project has made a number of contributions.

A key aspect of the W3C recommendation process is the provision of implementation experience – it must be demonstrated that the recommendations are implementable. Implementation experience gained during WonderWeb (for example experience in the parsing and validation process [2, 3], and the provision of efficient DL reasoning) provided valuable input to this process, and helped convince the W3C membership (both academic and industrial) that OWL implementation *is* feasible.

In addition, the implementations and solutions developed during the project (including the OWL API work [4], editors and the KAON infrastructure [13]) are all available as open-source projects. For example, as discussed in Section 3, the production of standard protocols for reasoning facilitated the use of Description Logic reasoners in third-party developments such as Jena.

WonderWeb has also contributed work on Foundational Ontologies. This includes a clarification of the role of foundational ontologies for the Semantic Web, the definition of an architecture for the WonderWeb library of foundational ontologies, and the development of a first reference module for an ontology library — DOLCE: a Descriptive Ontology for Linguistic and Cognitive Engineering. In addition, extensions of DOLCE have been provided (see Section 4.12.

3 Industrial Takeup

There is now an increasing interest and takeup of OWL from Industry. With the release of OWL as W3C Recommendation, this is likely to accelerate. A number of companies have already announced their support for the standard, and testimonials providing support for the recommendation can be seen at the W3C web site[2]. Organisations providing positive testimonials include Adobe, Agfa-Gevaert, Boeing, Fujitsu, Hewlett Packard, Nokia and Sun Microsystems Ltd. Such industry support for the standard is encouraging.

Contributions in terms of infrastructure and tools are also vital if applications are to be supported. We provide a brief overview of some industrial activity – further information regarding experience of OWL implementations, including industrial (as opposed to academic research) contributions can be found again on the W3C site[3].

3.1 Industrial Advisory Board

During the course of the WonderWeb project, two workshops were held for members of the Industrial Advisory Board, one in Manchester in September 2002, one in Sanibel Island in October 2003. At both workshops, the project progress was presented along with demonstrations of tools. At the first workshop, IAB members also presented up to date information about their own related research activities and there was an extended technical discussion, with particular emphasis on language architecture and the core API and component based architecture being developed in WP2.

Several members of the IAB, including Boeing, IBM and SUN, are evaluating tools and methodologies developed in the project by using them in internal R&D projects.

3.2 Boeing

Boeing have been active members of the Industrial Advisory Board, sending a representative to both meetings. This collaboration led to the use of WonderWeb technology, in particular OilEd and FaCT, in a project in Boeing. The resulting application is described in [12], and Boeing are continuing to investigate the further use of WonderWeb technology.

3.3 Hewlett Packard

Jena [10] is a Java framework for building Semantic Web applications. It is an open source project that has grown out of work from HP Labs Semantic Web Programme and is widely used. Jena provides comprehensive support for RDF, RDF(S) and OWL. Functionality includes:

- APIs for representing RDF models, RDF Schema and OWL ontologies;
- Readers (parsers) and writers for a number of different concrete serializations;

[2] http://www.w3.org/2004/01/sws-testimonial
[3] http://www.w3.org/2001/sw/WebOnt/impls

- In-memory and persistent storage mechanisms;
- Query mechanisms using RDQL;
- A rule-based inference engine;
- Connectors to use external Description Logic based inference. This is supplied through use of the DIG [5] protocol, the development of which was supported by the WonderWeb project.

3.4 Mondeca

Mondeca has chosen to use OWL to organize the ontology level in its Knowledge Management software Intelligent Topic Manager. The ontology level in ITM includes definition of topic classes, association types, role types, data types, and constraints tying them together. The use of OWL by Mondeca is intended to be modular, the ontologies in Mondeca namespace being imported in specific customers' workspaces, and/or in customers' domain ontologies. Further information is available from the Mondeca Web site[4].

3.5 IBM

IBM's SNOBASE[5] is a framework for loading, creating and modifying ontologies. The framework provides an inference engine, persistent store and query mechanisms. Support is provided for language standards such as RDF(S) and OWL.

3.6 Network Inference

Network Inference[6] has developed a set of tools around their Cerebra Inference engine that allow for the development and use of OWL ontologies. These tools are already seeing use in commercial applications – in particular Clinical Support Technology, a healthcare provider, have made use of technology based on OWL in order to manage rapidly changing classifications of patient knowledge. A Fortune 500 electronics manufacturer is also making use of an OWL-drive platform for the allocations of unit sales and application of business rules. In both of these cases, the expressive power of OWL, coupled with the well-founded semantics and ability to apply reasoning is crucial.

4 Research

A number of research projects, situated in both academia and industry are seeing increased use of OWL and ontology technology.

[4] http://www.mondeca.com/owl/
[5] http://www.alphaworks.ibm.com/tech/snobase
[6] http://www.networkinference.com

4.1 myGrid

myGrid[7] is a UK e-Science project involving five UK universities, the European Bioinformatics Institute and many industrial collaborators. The myGrid project aims to exploit the growing interest in Grid technology, with an emphasis on the Information Grid, and provide middleware layers that make it appropriate for the immediate needs of bioinformatics.

Specifically, myGrid is building high level services for data and application resource integration such as resource discovery, workflow enactment and distributed query processing. However, these services merely enable experiments to be formed and executed. Additional services are needed to support the e-based scientific method and best practice found at the bench but often neglected at the workstation, notably provenance management, change notification and personalisation.

Within myGrid, ontologies are used to help describe services and workflows. These service descriptions can then be queried by the scientist in order to facilitate service and workflow discovery. In addition, ontological terms are used to provide provenance information associated with data. Such provenance is particularly important within myGrid's application domain – the ability to capture who produced data items, when, and how are crucial.

Ontologies are used to help describe services and workflows in the following way:

1. Semantic descriptions of services (or workflows) are written using vocabulary from the ontologies. These descriptions are published into a registry view (aka Service Directory).

2. Descriptions are forwarded to the find service for classification based indexing.

3. The indexing is achieved using generic semantic web components including an instance store and a description logic reasoner.

4. The reasoner is able to calculate the position of a service or workflow in an index by comparing the description assigned to it with those already present in the service classification.

5. Services can then be browsed using a user tool that can display the index of available services and workflows. Services and workflows are both presented in a classification organised by the semantic type of inputs and outputs, and the tasks each service or workflow performs.

4.1.1 Ontologies

The myGrid ontology is actually made up of a suite of modules to describe molecular biology concepts, bioinformatics concepts, service concepts, publishing concepts and organisation concepts. The ontology provides a controlled vocabulary of concepts with which to describe the kinds of data being manipulated and the nature of the services and workflows which are performing the manipulation. For example, it provides the concept

[7] http://www.mygrid.org.uk/

DNA sequence data along with which is specified in a machine interpretable way that this is kind of data that ”encodes the sequence of a deoxyribonucleotide molecule which is a kind of nucleotide molecule”.

Kinds of data are the key concepts described. They in turn contribute the description of services and workflows by specifying the types of input allowed and output produced. Services and workflows are further described by a simple vocabulary of tasks that can be performed, resources that can be used, and methods that can be employed. With this vocabulary of concepts, myGrid is able to annotate actual data, workflows and services, and index them using a classification.

4.1.2 Service/ workflow discovery

The myGrid find service is responsible for gathering ontology based descriptions of resources (currently workflow definitions and services, gathered from the registry view) and indexing them based on the myGrid ontologies. It provides the main interface between the ontological information, the backend semantic technology, such as the instance store, and the rest of myGrid. The service browser provides and interface which displays services and workflows registered in the view, and organises them based on an ontological classification – which is itself produced using a reasoner.

Use of a controlled vocabulary structured as an ontology enables more sophisticated discovery of resources. Instead of using a text search we can ask for services which ”accept this kind of data”, ”accept something more specific than this kind of data”, or ”accept a component of this data record”.

In bioinformatics, data which is semantically the same such as nucleotide sequence data is carried by a myriad of different data formats. In the future myGrid aims to use the more explicit description of the semantic type of data and data formats to assist in the manipulation of data between services that use different formats.

4.2 AKT

AKT (Advanced Knowledge Technologies)[8] is a multi-million pound, six-year collaboration between research groups in the UK from the Universities of Aberdeen, Edinburgh, the Open University, Sheffield and Southampton. AKT provides a number of component technologies, intended to support the different challenges encountered though the various stages of the knowledge lifecycle.

AKT is making use of OWL as an ontology language – the AKT Reference Ontology has been developed by the AKT partners to represent the knowledge used in the CS AKTive Portal testbed and is represented using OWL.

4.3 GEODISE

Geodise (Grid Enabled Optimisation and Design Search for Engineering) [9] is a grid-based system that allows a user to combine together sets of Computational Fluid Dynamics

[8] http://www.aktors.org/akt
[9] http://www.geodise.org/

(CFD) packages in order to do design optimisation in areas such as aircraft wing design. The Geodise system is based on Matlab, which is the development environment most commonly used by engineers working in the field.

Everything in the Geodise environment is a Matlab function: either a standard Matlab function, or a Geodise-specific function. The latter category includes both low-level functions for performing various tasks including grid access, and higher-level functions for invoking standard CFD packages to run within the Geodise grid environment.

The CFD engineer can use combinations of both high- and low-level functions to construct workflows for performing particular optimisation tasks. This can be done manually by creating a Matlab script, or else via the Geodise Workflow Construction Environment (WCE), which allows the workflow scripts to be created visually using a drag-and-drop GUI.

A key aspect of Geodise is that the use of a Description Logic-based representation allows users to retrieve functions and workflows as a result of queries involving DL-based reasoning. They can thus form complex queries of a type that could not be framed using normal DBs or RDF.

4.3.1 Geodise Knowledge Architecture

The knowledge architecture aids the Geodise user by providing tools for the semantic annotation of both functions and workflows, and a semantic query mechanism for the retrieval functions and workflows. There is also a Workflow Construction Advisor (WCA) that uses semantic querying to advise the user during the process of workflow construction, and which can run either from the WCE, or from a special Matlab-aware text-editor.

4.3.2 Ontology View Framework

The Geodise knowledge architecture is built on top of the Ontology View Framework (OVF), which allows an ontology to be accessed and manipulated, via a simplified ”view”. The view consists of a set of relatively simple entities that map to more complex constructs in the underlying ontology. The manner in which the entities in a particular view map to the constructs in the underlying ontology, is determined by a specifically created ”view configuration”. The framework provides an API, upon which are built the following set of GUI-based tools:

- View configuration GUI: Used by knowledge engineers to configure ontology views.
- Ontology editing GUI: Used by knowledge engineers or domain experts to edit ontologies, and maintain sets of instances, via specific views.
- Query GUI: Used by the end-user to formulate and execute queries.

4.3.3 Geodise Ontology Development and Instance-Store Maintenance

The creation of the Geodise ontology and the initial population of the instance store, have been carried out by a ”Domain Expert”, who is actually a member of the engineering side

of the Geodise team. This work has been done using the OVF ontology editing GUI, with a view configuration created by members of the Geodise knowledge management team. In addition to the generic ontology editing facilities provided by the OVF, there is also a Geodise- specific Function Annotation Tool (FAT), which is built on top of the OVF API. This tool allows Geodise users to annotate their own functions, and incorporate them into the Geodise system.

4.4 MONET

MONET (Mathematics On the NET)[10] was an EU funded project, concerned with the development of a framework for the description and provision of web-based mathematical services. MONET used an OWL ontology for the description of services. The principal objective of MONET was to develop a framework for the description and provision of web-based mathematical services. The key to such a framework is the ability to discover services dynamically based on published descriptions which describe both their mathematical and non-mathematical attributes. Discovery and subsequent interaction can then be mediated by software agents which are capable of recognising the criteria which should determine how particular kinds of problems are solved, and extracting them from the user's problem description.

A simple example of the kind of interaction that MONET is intended to support is as follows. A domain-specific software package (e.g. CAD, financial analysis etc.) might need to solve a set of differential equations. It would look to see which services offering this kind of solution were currently available on the web by contacting one or more brokers. Selecting a service to use would of course involve matching its mathematical capabilities to the problem in hand, but also various non-mathematical issues such as the user's preferences for particular kinds or brands of software, whether the user is permitted to use the service (because he is in a particular domain, or has paid a subscription), whether the service has enough resources available to solve the user's problem in the time available and so on. Having negotiated access to a service the client would send it the problem and receive the result back, which it would return to the user.

4.4.1 Ontologies

MONET uses a number of sub-ontologies which are then collected together to form the entire MONET ontology. These include models of hardware, software, algorithms and so on. More precise details of the MONET ontologies can be found at the MONET web site.

4.4.2 Service Registration and Matching

When registering services, they can be supplied with a description providing the characteristics of the service. Due to the use of OWL as the representation language, these descriptions need not be enumerated pre-hoc, but can be arbitrary compositions of the available vocabulary terms. Reasoning can then be used during service discovery in order to find matching services.

[10] `http://monet.nag.co.uk`

4.4.3 Storing Instances

MONET uses an approach known as the Instance Store[11] [9] to store service descriptions. The Instance Store allows descriptions of individuals using OWL expressions. By limiting the expressivity of the assertions supported (relations between instances are not allowed), a combination of database and reasoning can be used to provide efficient retrieval over large numbers ($> 500,000$) of individuals. Although this approach places restrictions on the expressivity of the information that can be represented, such limited expressivity can be seen to be sufficient to meet the requirements of projects such as MONET.

The Instance Store approach has also been used with other project making use of OWL such as GONG and GOAT. WonderWeb contributed towards the development of the Instance Store – the current implementation makes use of the re-engineered FaCT++ reasoner (see project Deliverable 14), allowing the support of large taxonomies.

4.5 GONG

There now exist many biological databases containing enormous quantities of entries of genes and gene products along with descriptions and data about a wide variety of their functional properties. However, the synonymy and polysemy of the descriptive terms and the lack of explicit relationships among them hampers consistent, reliable querying of and interoperability between these databases. In response to this, the Gene Ontology (GO) [12], a structured controlled vocabulary of nearly 17,000 terms, has been (and is being) developed to be used to functionally describe the gene products of various organisms. GO is divided into three subontologies of terms (most of which also have natural-language definitions) which may be used to annotate gene products in terms of the molecular functions they possess, the higher-level biological processes in which they are involved, and the cellular locations in which they are active. Each term of each of these subontologies is related to each respective parent term via an is-a or a part-of relationship. GO is represented as a Directed Acyclic Graph (DAG) which encodes both is-a and part-of hierarchies.

The resulting, publicly available GO has become the defacto standard used to provide 250,000 annotations for entries in at least 14 major bioinformatics databases. GO has been successful in supporting the needs of molecular biologists due to its comprehensive coverage in a relatively simple but consistent structure acceptable to the biological communities. However, its growing success and size now leads to several challenges for ongoing manual curation. GO is represented as a Directed Acyclic Graph (DAG) which encodes both is-a and part-of hierarchies. The intention of the GONG project [14] is to evolve GO from this simple DAG-based representation to a richer one using a Description Logic based representation.

Description Logic based representations such as OWL are powerful languages for formally representing ontologies. However, it is not necessary to use all the expressive power: they are light-weight enough to represent simple taxonomies or frame-like ontologies, of which The Gene Ontology is an example, without obstruction. They are also, however, expressive enough to present richer logic-based models such as the TAMBIS

[11] `http://instancestore.man.ac.uk`
[12] `http://www.geneontology.org/`

Ontology [ref], and all points in between. Some parts of the ontology can be simple, others complex; moreover, the language offers the ontologist an evolutionary development path whereby they can progressively introduce more expressive constructs.

The philosophy within GONG is therefore to incrementally migrate in situ the primitive GO expressions to defined concept expressions placing more emphasis on relationships and definitional descriptions. The original GO codes migrate with the concepts as they are transformed. The methodology employed is as follows:

- Translate GO into an ontology in the appropriate language (e.g. OWL). During this stage, the is-a relationships of the DAG are translated to subclass relationships and part-of relationships are translated to existential quantifications over a part-of property. Reasoning can then be used to group related components based on part-of relationships specified in the current GO.

- Programmatically create partial class descriptions from existing structured information in bioinformatics databases, enabling the grouping of existing terms under new abstractions. There are numerous bioinformatics resources available that contain structured information – for example characterizing various aspects of enzymes. Other resources used here include the inclusion of chemical ontologies translated from sources such as Medical Subject Headings MeSH[13].

- Manual checks can then be made on the partial descriptions of step 2 to enable the reasoner to check the consistency of the existing hierarchy and detect missing is-a relationships.

- Feedback results to GO curators.

A key aspect of the use of DL-based languages here is the ability to use a reasoner to organise the concept hierarchy. Using reasoning, we can spot new inferred subsumption relationships in the taxonomy. Thus missing or erroneous is-a relationships can be discovered and communicated back to the GO curators, supporting the production of a more consistent taxonomy. The implicit semantics which were present in the names or rubrics attached to terms and their asserted subsumption relationships have been replaced with explicit semantics, encoded using rich ontological descriptions.

4.6 GOAT

The Gene Ontology (see above) has been a success in that its terms are being used to functionally annotate genes and gene products in a number of prominent biological databases. However, as GO continues to increase in size, users find it increasingly difficult to find the terms they wish to use for annotation. Furthermore, although a large vocabulary is provided, the terms have no links to each other apart from those relationships that form the three taxonomic/partonomic hierarchies.

Thus, beyond this hierarchical information, there are no constraints within GO that can be used to indicate which terms should or should not be used together in the annotation

[13] http://www.nlm.nih.gov/mesh/

of a given gene product. It is possible (though unlikely) that an annotator, in describing a protein, could associate the terms ”viral life cycle”, ”amino-acid biosynthesis”, and ”extracellular matrix” to that protein; it is more likely that he would accidentally do so. In either case, this is likely to be biologically nonsensical. Good annotation relies upon the domain expertise of the annotator and the usability of the annotation tool. We seek to improve upon the latter by creating formal relationships between pairs of GO terms (as well as between GO terms and gene-product types) mined from biological databases and building an application that, relying upon these relationships, can dynamically retrieve and present those GO terms that are most likely to be applicable for a given gene product based on the GO terms and the gene-product type already entered by the user for that gene product.

Thus, if an annotator has already selected ”viral life cycle” as a biological-process term and then indicated that she wanted to add a molecular-function term, she would be presented with those molecular-function terms that have been used as annotating terms along with ”viral life cycle” (as well as those terms’ descendants).

The aim of the GOAT project [1] is to create an application that will guide users, especially biomedical researchers, in the annotation of gene products with terms from the Gene Ontology. This will directly use the enriched representation of the Gene Ontology that is the result of the GONG project. GOAT also makes use of the Instance Store as discussed above.

4.7 S-MOBY

The BioMOBY Project[14] is a project to develop a web services architecture for bioinformatics. Within the MOBY project, S-MOBY (Semantic MOBY) aims to address a number of problems:

- fatal mutability of traditional interfaces - the problem where if providers change their interface, client code depending on that signature fails en masse. This has the undesirable property that the more clients engage an interface (i.e., the more widely adopted it is), the less flexibility providers have in evolving it.

- rigidity and fragility of static classification schemes (hierarchies) - the problem where changing the properties of a class near the root of an inheritance hierarchy simultaneously affects the entire sub-tree. This has the undesirable property that in an open-world, where independent parties are arbitrarily adding ISA relationships, the nodes near the root, which where added when the system was least evolved, are the least able to change without causing failure en masse.

- confounding structure and content - the problem where access to the content information of the data-the ”data” itself-is entangled with the presentation layer and/or implicit behaviors of the presentation software. HTML and idiosyncratic XML both suffer from this type of entanglement. This has the undesirable property that the data content of value to the client may be difficult or essentially impossible to

[14] http://www.biomoby.org

parse from the data presentation of an arbitrary provider, thereby crippling machine-automated, semantic determination.

S-MOBY's design uses a single, canonical structure (an OWL-DL graph) to embed everything needed for how a provider's service is described, how a client's request is made to the discovery service to find a provider, how that request is satisfied by the discovery service, how the client's query is made to the provider, how the provider's answer is returned to the client, and how the client parses that response; in short, the description is the query is the answer.

4.8 NCI Cancer Thesaurus

The NCI Thesaurus is a public domain description logic-based terminology produced by the National Cancer Institute, distributed as a component of the NCI Center for Bioinformatics caCORE distribution. It is deep and complex compared to most broad clinical vocabularies, implementing rich semantic interrelationships between the nodes of its taxonomies. The semantic relationships in the thesaurus are intended to facilitate translational research and to support the bioinformatics infrastructure of the Institute. Topics described in the ontology include diseases, drugs, chemicals, diagnoses, genes, treatments, anatomy, organisms, and proteins. The NCI Thesaurus evolved from the NCI Metathesaurus, which is based on the National Library of Medicine Unified Medical Language System (UMLS) Metathesaurus. The NCI Metathesaurus has been operational since 1999. The Mindswap group at the University of Maryland Institute for Advanced Computer Studies[15] have provided an OWL translation of the NCI thesaurus[16]. This particular ontology is large (approximately 500,000 RDF triples) and proves a useful benchmark for testing applications.

4.9 GEON

To give geoscientists broader views of the Earth, researchers in the Data and Knowledge Systems (DAKS) program at the San Diego Supercomputer Center (SDSC) are collaborating with geoscientists who study the solid Earth to build a prototype national Geosciences Cyberinfrastructure Network – GEON[17]. GEON is intended to help weave the separate strands of the solid Earth sciences disciplines and data into a unified fabric, giving the geosciences an 'IT head start' for viewing the complex dynamics of the Earth system as an interrelated whole.

GEON is being designed as a scientist-centered cyberinfrastructure, freeing researchers to think and be creative by relieving them of onerous data management tasks. Through a scalable and interoperable network, the project will provide scientists with a growing array of tools they can use without having to be IT experts. These include data integration mechanisms, as well as computational resources and integrated software for analysis, modeling, and visualization. In this way, GEON will bridge traditional

[15] http://www.mindswap.org
[16] http://www.mindswap.org/2003/CancerOntology/
[17] http://www.geongrid.org

disciplines-an indispensable step in understanding the Earth as a unified system. GEON is using OWL for the representation of ontologies.

4.10 BioPAX

The goal of the Biological Pathways Exchange or BioPAX[18] group is to develop a common exchange format for biological pathways data. The BioPAX project began at the Fourth BioPathways Consortium Meeting, a satellite of the ISMB'02 Conference held in Edmonton, Canada in August 2002. There it was decided that the creation of a standard data exchange format for pathway information was not only a good first step toward building an open source pathway information resource, but also that such an exchange format would be a desirable end in itself as it would facilitate sharing of pathways information between existing databases, both public and private. OWL is being used to represent the BioPAX ontology.

4.11 Protege

Protege[19] is a tool supporting the construction of ontologies. It also provides an application platform for knowledge based systems and libraries for application building. Protege has a large and active user community. Recent work has produced an OWL plugin for Protege[20], supporting the editing of OWL ontologies and the Collaborative Open Ontology Development Environment CO-ODE[21] project is investigating support for users modelling in OWL. Protege already has a large user base and it is likely that the addition of OWL support within Protege will help to promote OWL usage within a larger community.

Early prototypes of the Protege OWL plugin used the WonderWeb OWL API to provide access to reasoning functionality.

4.12 Foundational Ontologies

WonderWeb has supported the development of DOLCE: a Descriptive Ontology for Linguistic and Cognitive Engineering. DOLCE is a rich, carefully axiomatized top-level ontology, which despite its clear cognitive bias (especially appropriate for the semantic web) has been designed in such a way to avoid hidden ontological assumptions, by relying on a rich axiomatization. Positive feedback on DOLCE has been expressed by distinguished researchers involved in major ontology projects.[22] In an extended version, it is being used by ISTC-CNR in ontology-related projects involving various application domains.

[18] `http://www.biopax.org/`

[19] `http://protege.stanford.edu`

[20] `http://protege.stanford.edu/plugins/owl/index.html`

[21] `http://co-ode.man.ac.uk/`

[22] Including: Christiane Fellbaum, WordNet, Princeton University; Tony Cohn, University of Leeds; Barry Smith, IFOMIS, University of Leipzig; Chris Welty, IBM Watson Research Center; Bill Andersen, OntologyWorks; Werner Ceusters, Language and Computing; Peter Eklund, WebKB, University of Queensland; Joost Breuker, University of Amsterdam.

A second strand of work has been the development of the WonderWeb Foundational Ontology Library (WFOL) (See Deliverables 17 and 18). The final version of the library includes:

- Axiomatic characterizations of reference modules (called visions): OCHRE, BFO and DOLCE.
- Two new extensions of DOLCE: the ontology of "Descriptions and Situations" (D&S), and a minimal ontology of plans (PO).
- An ontology of web services (WSO) based on D&S and PO.
- A mapping between DOLCE+D&S+PO and the English version of WordNet. This is particularly relevant for two reasons: it provides a bridge between ontologies and natural languages which is particularly useful in applications, and it contributes to the improvement of the ontological structure of lexical resources.

Organisations including the UN/FAO are investigating the use of the WonderWeb foundational ontologies.

The DOLCE ontology, together with other foundational ontologies, is also being officially considered among the input resources for a proposed W3C "Semantic-Web Best Practices and Deployment" (SWBPD) Working Group[23].

5 Conclusions

Although OWL has only been a Recommendation since February 2004, it is clear that a significant number of research projects are using OWL. Industrial interest is also increasing, with companies providing public backing for the recommendations. The WonderWeb project has made important contributions to this work, both in terms of technical input to the standardisation process, and in the provision of implementations supporting the use of OWL.

References

[1] Michael Bada, Robin McEntire, Chris Wroe, and Robert Stevens. GOAT: The Gene Ontology Annotation Tool. In *Proc UK e-Science programme All Hands Conference*, Nottingham, UK, September 2003.

[2] Sean Bechhofer. OWL Web Ontology Language: Parsing OWL in RDF/XML. W3C Working Group Note, World Wide Web Consortium, January 2004.

[3] Sean Bechhofer and Jeremy J. Carroll. Parsing OWL DL: Trees or Triples? In Marc Najork and Craig Wills, editors, *Proceedings of World Wide Web Conference, WWW2004*. ACM Press, May 2004.

[23] http://www.w3.org/2001/sw/BestPractices/

[4] Sean Bechhofer, Phillip Lord, and Raphael Volz. Cooking the Semantic Web with the OWL API. In Fensel et al. [7].

[5] Sean Bechhofer, Ralf Möller, and Peter Crowther. The DIG Description Logic Interface. In *Proceedings of DL2003 International Workshop on Description Logics*, Rome, September 2003.

[6] DAML+OIL. `http://www.daml.org/language`.

[7] Dieter Fensel, Katia Sycara, and John Myolopoulos, editors. *Proceedings of the 2nd International Semantic Web Conference, ISWC2003*, volume 2870 of *Lecture Notes in Computer Science*, Sanibel Island, Florida, October 2003. Springer.

[8] Jeff Heflin. OWL Web Ontology Language Use Cases and Requirements. Recommendation, World Wide Web Consortium, 2004.

[9] Ian Horrocks, Lei Li, Daniele Turi, and Sean Bechhofer. The Instance Store: DL Reasoning with Large Numbers of Individuals. In *2004 International Workshop on Description Logics*, Whistler, CA, 2004.

[10] Jena – A Semantic Web Framework for Java. `http://jena.sourceforge.net/`.

[11] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. Recommendation, World Wide Web Consortium, 2004.

[12] Michael Uschold, Peter Clark, Fred Dickey, Casey Fung, Sonia Smith, Stephen Uczekaj, Michael Wilke, Sean Bechhofer, and Ian Horrocks. A Semantic Infosphere. In Fensel et al. [7].

[13] Raphael Volz, Daniel Oberle, Steffen Staab, and Boris Motik. KAON SERVER - A Semantic Web Management System. In *Proc. of WWW-2003*, Budapest, Hungary, 05 2003.

[14] C.J. Wroe, R.D. Stevens, C.A. Goble, and M. A Ashburner. A Methodology to Migrate the Gene Ontology to a Description Logic Environment Using DAML+OIL. In *8th Pacific Symposium on Biocomputing (PSB)*, pages 624–636, 2003.