Ontology Language Standardisation Efforts

Editor: Sean Bechhofer
Information Management Group
Department of Computer Science
University of Manchester
Oxford Road
Manchester M13 9PL
UK
seanb@cs.man.ac.uk
OntoWeb Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2000-29243.

Vrije Universiteit Amsterdam (VU)-Coordinator
Faculty of Sciences,
Division of Mathematics and Computer Science
De Boelelaan 1081a
1081 HV Amsterdam, the Netherlands
Fax and Answering machine: +31-(0)84-872 27 22
Mobil phone: +31-(0)6-51850619
Contactperson: Dieter Fensel
E-mail: dieter@cs.vu.nl
## Revision Information

<table>
<thead>
<tr>
<th>Revision date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>29-01-2002</td>
<td>1</td>
<td>Final Version</td>
</tr>
</tbody>
</table>
**Introduction**

We are on the brink of a new generation of World Wide Web (WWW) which, in his recent book "Weaving the Web" [Ber99], Tim Berners-Lee calls the Semantic Web. Unlike the existing WWW, where data content is primarily intended for human consumption, the Semantic Web will provide data whose content is also machine processable. This will enable a wide range of intelligent services such as information brokers, search agents, information filters etc., a process that Berners-Lee describes as “Bringing the Web to its full potential”.

The development of ontologies will be central to this effort. Ontologies are metadata schemas, providing a controlled vocabulary of terms, each with an explicitly defined and machine processable semantics. By defining shared and common domain theories, ontologies help both people and machines to communicate more effectively. They will therefore have a crucial rôle in enabling content-based access, interoperability and communication across the Web, providing it with a qualitatively new level of service: the Semantic Web.

In order for ontologies to fulfill their rôle in the semantic integration of the Web, there will need to be some standardisation of Web ontology languages.

This deliverable provides a brief overview of existing standards and standardisation efforts in the area of ontology languages.

This report is intended as a resource for the OntoWeb Portal. To that end, the references and pointers provided within the document are primarily in the form of URLs, referencing electronically available resources.

The content of the report has been gleaned from a number of other resources and documents.

**Important Notes**

Note that we are not here concerned with the issue of ontology content standardisation, for example providing standardisations of upper level ontologies. Such initiatives (for example the Standard Upper Ontology Working Group [SUO]) are, of course, vital to the production of usable ontologies which can be exchanged and used within applications, but are not the focus of our attention. Instead we concentrate on the underlying languages which will be used for representation of such content.

Nor is this intended as a rigorous evaluation of ontology language standards. Rather, we present the various proposals for the purpose of information. In some cases, however, comments have been made on the appropriateness or otherwise of language proposals.

**Ontologies**

An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms [Usch98].

Gruber [Gru93] defines an ontology as "the specification of conceptualisations, used to help programs and humans share knowledge". The conceptualisation is the couching of knowledge about the world in terms of entities (things, the relationships they hold and the constraints between them). The specification is the representation of this conceptualisation in a concrete form. One step in this specification is the encoding of the conceptualisation in a knowledge representation language.

Guarino [Gua98] defines an ontology as "an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality." A formal ontology has some underlying logical structure which allows us to reason about the concepts in the ontology.
The goal is to create an agreed-upon vocabulary and semantic structure for exchanging information about that domain.

**Standardisation Approaches**

In general there are two strategies for achieving a standard:

- Defining a "small" set of modeling primitives that are common across the community, and defining a proper semantics for them;

- Defining a "large" set of modeling primitives that are present in some of the approaches in a community and glue them together.

Both may lead to success. The first approach can be illustrated with HTML. Its first version was very simple and limited but therefore allowed the Web to catch on and become a worldwide standard. Meanwhile we have HTML version 5, XHTML, and XML. So, beginning with a core set, and successively refining and extending it, has proven to be a successful strategy. The second approach has been taken by the UML community by designing a model that is broad enough to cover all its modeling concepts. This leads to ambiguity and redundancy in modeling primitives and sometimes a precise semantic definition is lacking. However, UML has been adopted by the software industry as one of the major approaches in the meantime and is therefore also a success. Obviously, these two opposite approaches to standardization may both work successfully.

**Existing Standards and Standardisation Efforts**

There are a (large) number of existing standards and standardisation efforts which are either directly or indirectly involved in the provision of ontology language standards.

Some of the languages and efforts are listed below along with a brief description of the language or effort.

Note that this list contains a number of languages or language specifications that are not actual standards. For example, OIL and DAML+OIL are language specifications that have not been accepted as standards by a recognised body. However, such languages have been used in much recent ontology work, and have taken on the mantle of *de facto* standards and will heavily influence any further development or standardisation. For this reason they are included here.

<table>
<thead>
<tr>
<th>CKML</th>
<th>CLIPS</th>
<th>Conceptual-Graphs</th>
<th>CycL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAML+OIL</td>
<td>DAML-ONT</td>
<td>ebXML</td>
<td>FIPA</td>
</tr>
<tr>
<td>F-Logic</td>
<td>GRAIL</td>
<td>KIF</td>
<td>OCML</td>
</tr>
<tr>
<td>OIL</td>
<td>OKBC</td>
<td>OML</td>
<td>Ontolingua</td>
</tr>
<tr>
<td>Protege</td>
<td>RDF and RDF(S)</td>
<td>RuleML</td>
<td>SHOE</td>
</tr>
<tr>
<td>Thesauri</td>
<td>Topic Maps</td>
<td>UML</td>
<td>UPML</td>
</tr>
<tr>
<td>WebOnt</td>
<td>XML-Schema</td>
<td>XOL</td>
<td></td>
</tr>
</tbody>
</table>

**CKML & OML**

The Ontology Markup Language [OML] is an ontology specification language based on [Conceptual Graphs]. It allows the representation of concepts organized in taxonomies, relations and axioms in first order logic. Conceptual Knowledge Markup Language [CKML] is an extension of OML.

**Resources**

CLIPS

CLIPS is a productive development and delivery expert system tool which provides a complete environment for the construction of rule and/or object based expert systems. CLIPS is used throughout the public and private community including: all NASA sites and branches of the military, numerous federal bureaus, government contractors, universities, and many companies. CLIPS provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented and procedural. Rule-based programming allows knowledge to be represented as heuristics, or "rules of thumb," which specify a set of actions to be performed for a given situation. Object-oriented programming allows complex systems to be modeled as modular components (which can be easily reused to model other systems or to create new components). The procedural programming capabilities provided by CLIPS are similar to capabilities found in languages such as C, Pascal, Ada, and LISP.

Resources


Conceptual Graphs

Conceptual graphs [CGs] are a system of logic based on the existential graphs of Charles Sanders Peirce and the semantic networks of artificial intelligence. They express meaning in a form that is logically precise, humanly readable, and computationally tractable. With a direct mapping to language, conceptual graphs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages. With their graphic representation, they serve as a readable, but formal design and specification language. CGs have been implemented in a variety of projects for information retrieval, database design, expert systems, and natural language processing. A proposal for a [CG standard] has been developed, based on work from the [NCITS.T2] Committee on Information Interchange and Interpretation. In addition, a more up to date working draft of an [ISO CG Standard] Standard has also been produced.

Resources


CycL

The [CycL] language has been developed within the [Cyc] project. CycL is a formal language whose syntax derives from first-order predicate calculus (the language of formal logic). In order to express real-world expertise and common sense knowledge, however, it goes far beyond first order logic. The vocabulary of CycL consists of terms: semantic constants, non-atomic terms (NATs), variables, numbers, strings, etc. Terms are combined into meaningful CycL expressions, ultimately forming meaningful closed CycL sentences (with no free variables.) A set of CycL sentences forms a knowledge base.

Resources

[Cyc] http://www.cyc.com

DAML+OIL

DAML+OIL builds on work from both the OIL and DAML-ONT initiatives. It provides modelling primitives commonly found in frame-based languages (such as an asserted subsumption hierarchy and the description or definition of classes through slot fillers) and has a clean and well defined semantics. The latest
version of the schema [March 2001] extends DAML+OIL (December 2000) with values from [XML Schema] datatypes. DAML+OIL is effectively an alternative presentation syntax for a Description Logic (SHIQ with the addition of concrete datatypes) with an underlying RDFS based delivery mechanism.

The presence of the well-defined semantics in terms of SHIQ allow the use of description logic reasoners such as [FaCT] or [RACER], in particular to support the tasks of classification and inconsistency detection.

The development of DAML+OIL was the responsibility of the Joint US/EU ad hoc Agent Markup Language Committee [DAML Joint Committee]. Many members of that committee are now part of the WebOnt Committee, so it is likely that further development will take place within that forum.

Resources

[DAML Joint Committee] http://www.daml.org/committee/
[FaCT] http://www.cs.man.ac.uk/fact

DAML-ONT

The original language proposal from the DAML program, now superceded by DAML+OIL. The latest release of [DAML-ONT] was October 2000.

Again, as with OIL, DAML-ONT was a proposal for a language specification rather than a ratified standard.

Resources


ebXML

The mission of [ebXML] is to provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties. This is to be achieved through a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. EbXML will provide a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes.

EbXML is sponsored by [OASIS] and [UN/CEFACT]. [OASIS], the Organization for the Advancement of Structured Information Standards, is a non-profit, international consortium that creates interoperable industry specifications based on public standards such as XML and SGML, as well as others that are related to structured information processing. [UN/CEFACT] is the United Nations body whose mandate covers worldwide policy and technical development in the area of trade facilitation and electronic business. Headquartered in Geneva, it has developed and promoted many tools for the facilitation of global business processes including UN/EDIFACT, the international EDI standard. Its current work programme includes such topics as Simpl-edi and Object Oriented edi and it strongly supports the development and implementation of open interoperable, global standards and specifications for electronic business.

Resources

FIPA

The Foundation for Intelligent Physical Agents [FIPA] is a non-profit organisation aimed at producing standards for the interoperation of heterogeneous software agents.

FIPA98 Specification Part 12 [Ontology Service] deals with technologies enabling agents to manage explicit, declaratively represented ontologies. It specifies an ontology service provided to a community of agents by a dedicated Ontology Agent. The FIPA 98 specification is now considered to be obsolete. A more recent Ontology Service specification has been produced.

The model of agent communication in FIPA is based on the assumption that two agents, who wish to converse, share a common ontology for the domain of discourse. It ensures that the agents ascribe the same meaning to the symbols used in the message. For a given domain, designers may decide to use ontologies that are explicit, declaratively represented (and stored somewhere) or, alternatively, ontologies that are implicitly encoded with the actual software implementation of the agent themselves and thus are not formally published to an ontology service.

The FIPA specification deals with technologies enabling agents to manage explicit, declaratively represented ontologies. An ontology service for a community of agents is specified for this purpose. It is required that the service be provided by a dedicated agent, called an Ontology Agent (OA), whose role in the community is to provide some or all of the following services:

- discovery of public ontologies in order to access them,
- maintain (for example, register with the DF, upload, download, and modify) a set of public ontologies,
- translate expressions between different ontologies and/or different content languages,
- respond to query for relationships between terms or between ontologies, and,
- facilitate the identification of a shared ontology for communication between two agents.

As with the earlier specification, this deals only with the communicative interface to such a service while internal implementation and capabilities are left to developers. The interface is specified at the agent communication level as opposed to a computational API. Therefore, the specification defines the interaction protocols, the communicative acts and, in general, the vocabulary that agents must adopt when using this service.

Resources


F-Logic

F-Logic is a deductive, object oriented database language which combines the declarative semantics and expressiveness of deductive database languages with the rich data modelling capabilities supported by the object oriented data model. The basic constructs of F-Logic are objects. Objects model real world entities and are internally represented by object identifiers which are independent of their properties. Following the object oriented paradigm, objects may be organized in classes. Furthermore, methods represent relationships between objects. F-Logic forms the core of systems such as [Ontobroker].
**GRAIL**

The [GRAIL](http://www.opengalen.org/open/CRM/) formalism was a product of the European GALEN project, now the responsibility of the [OpenGALEN](http://www.opengalen.org) organisation. GRAIL is effectively a description logic with a primitive role hierarchy, transitive roles and restricted forms of concept inclusion axioms. The GALEN Terminology architecture also includes functionality to support lexical operations and multilingual support. The language has primarily been used to produce models of medical terminology (the GALEN Common Reference Model [CRM]).

A suite of modelling tools known as the [OpenKnoME](http://www.topthing.com/) are available for GRAIL modelling.

**KIF**

Knowledge Interchange Format [KIF](http://logic.stanford.edu/kif/kif.html) provides a declarative language for describing knowledge and for the interchange of knowledge among disparate programs.

KIF has a declarative semantics (i.e. the meaning of expressions in the representation can be understood without appeal to an interpreter for manipulating those expressions). It is logically comprehensive i.e. it provides for the expression of arbitrary sentences in the first-order predicate calculus, and it provides for the representation of knowledge about knowledge.

KIF is not intended as a primary language for interaction with human users (though it can be used for this purpose). Different programs can interact with their users in whatever forms are most appropriate to their applications (for example frames, graphs, charts, tables, diagrams, natural language, and so forth). As a pure specification language, KIF does not include commands for knowledge base query or manipulation. [OKBC](http://www.ai.sri.com/~okbc/) is complementary to such language specifications.

There is a "draft proposed American National Standard" [dpANS](http://logic.stanford.edu/kif/dpans.html) for KIF. The IEEE Standard Upper Ontology (SUO) Study Group Knowledge Interchange Format group [SUO Study Group](http://suo.ieee.org/KIF/) are also working towards a standardisation of KIF in order to support the SUO project.

**OCML**

The Operational Conceptual Modelling Language [OCML](http://www.ai.sri.com/~okbc/) is a modelling language from the Open University's [OU](http://www.open.ac.uk) Knowledge Media Institute [KMi]. It supports the construction of knowledge models by means of several types of constructs. It allows the specification and operationalization of functions, relations,
classes, instances and rules. It also includes mechanisms for defining ontologies and problem solving methods, the main technologies developed in the knowledge modelling area. About a dozen projects in the [KMi] are currently using OCML to provide modelling support for applications in areas such as knowledge management, ontology development, e-commerce and knowledge based system development. OCML modelling is also supported by a large library of reusable models, providing a useful resource for the knowledge modelling community. This library can be accessed through the [WebOnto] editor.

Resources

- [OU] http://www.open.ac.uk
- [KMi] http://kmi.open.ac.uk

OIL

The Ontology Inference Layer OIL [OIL] was a proposal for a web-based representation and inference layer for ontologies, which combined the widely used modelling primitives from frame-based languages with the formal semantics and reasoning services provided by description logics. It is compatible with RDF Schema [RDFS], and includes a precise semantics [OIL Semantics] for describing term meanings (and thus also for describing implied information).

OIL presents a layered approach to a standard ontology language. Each additional layer adds functionality and complexity to the previous layer. This is done such that agents (humans or machines) who can only process a lower layer can still partially understand ontologies that are expressed in any of the higher layers.

Although not an official standard endorsed by a standards body, OIL has proved a major influence on the development of DAML+OIL, with many of the ideas developed in OIL being present in the DAML+OIL language. The latest language definition (as an RDF Schema definition) dates from 10th November, 2000 [OIL Schema]. It is unlikely that any further work will be done on OIL itself, although features of the language will undoubtedly be present in any ontology language standardisation proposal produced by the W3C's [WebOnt] activity.

Resources

- [RDFS] http://www.w3.org/TR/rdf-schema/

OKBC

Open Knowledge Base Connectivity [OKBC] is not an ontology representation language but is an application programming interface for accessing knowledge bases stored in knowledge representation systems (KRSs).

OKBC is a successor of Generic Frame Protocol [GFP] which was primarily aimed at systems that can be viewed as frame representation systems and was jointly developed by Artificial Intelligence Center of [SRI] International and the Knowledge Systems Laboratory [KSL] of Stanford University.

OKBC provides a uniform model of KRSs based on a common conceptualization of classes, individuals, slots, facets, and inheritance. OKBC is defined in a programming language independent fashion, and has existing implementations in Common Lisp, Java, and C. The protocol transparently supports networked as well as direct access to KRSs and knowledge bases.
OKBC consists of a set of operations that provide a generic interface to underlying KRSs. This interface isolates an application from many of the idiosyncrasies of a specific KRS and enables the development of tools (e.g., graphical browsers, frame editors, analysis tools, inference tools) that operate on many KRSs. It has been successfully used in several ongoing projects at SRI and Stanford University.

The development of OKBC is being overseen by a working group, chaired by Richard Fikes.

Resources


Ontolingua

[Ontolingua] provides a distributed collaborative environment to browse, create, edit, modify, and use ontologies. The ontology server architecture provides access to a library of ontologies, translators to other languages, and an editor to create and browse ontologies. Remote editors can browse and edit ontologies. Applications can access any of the ontologies in the ontology library using [OKBC].

The original Ontolingua language, as described by Gruber, was designed to support the design and specification of ontologies with a clear logical semantics and built on [KIF]. KIF was extended with additional syntax to capture intuitive bundling of axioms into definitional forms with ontological significance; and a Frame Ontology to define object-oriented and frame-language terms. The Ontolingua Server has extended the original language in two ways. First, it provides explicit support for building ontological modules that can be assembled, extended, and refined in a new ontology. Second, it makes an explicit separation between an ontology’s presentation and representation.

Resources


Protégé-2000 toolset:

[Protégé-2000] provides an integrated knowledge-base editing environment and an extensible architecture for the creation of customized knowledge-based tools.

Protégé's model is essentially frame-based and offers compatibility with [OKBC]. Protégé also has a flexible metaclass architecture which provides configurable templates for new classes in the knowledge base. The use of metaclasses makes Protégé easily extensible and enables its use with other knowledge models. Protégé also supplies extensive functionality to support the knowledge acquisition process and can thus be used to support knowledge base construction (i.e. models populated with instances) as well as the construction of ontologies.

Resources


RDF and RDF(S)

RDF (Resource Description Framework) [RDF] is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web. RDF
emphasizes facilities to enable automated processing of Web resources. The broad goal of RDF is to define a mechanism for describing resources that makes no assumptions about a particular application domain, nor defines (a priori) the semantics of any application domain. The definition of the mechanism should be domain neutral, yet the mechanism should be suitable for describing information about any domain.

As a result of many communities coming together and agreeing on basic principles of metadata representation and transport, RDF has drawn influence from several different sources. The main influences have come from the Web standardization community itself in the form of HTML metadata and [PICS], the library community, the structured document community in the form of [SGML] and more importantly [XML], and also the knowledge representation (KR) community. There are also other areas of technology that contributed to the RDF design; these include object-oriented programming and modeling languages, as well as databases. While RDF draws from the KR community, readers familiar with that field are cautioned that RDF does not specify a mechanism for reasoning. RDF can be characterized as a simple frame system. A reasoning mechanism could be built on top of this frame system (as is the case with, for example, [DAML+OIL]).

The foundation of RDF is a model for representing named properties and property values. The RDF model draws on well-established principles from various data representation communities. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources and an RDF model can therefore resemble an entity-relationship diagram. (More precisely, RDF Schemas — which are themselves instances of RDF data models — are ER diagrams.) In object-oriented design terminology, resources correspond to objects and properties correspond to instance variables.

The RDF data model is a syntax-neutral way of representing RDF expressions. The RDF data model, as specified in [RDFMS], defines a simple model for describing interrelationships among resources in terms of named properties and values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. As such, the RDF data model can therefore resemble an entity-relationship diagram. The RDF data model, however, provides no mechanisms for declaring these properties, nor does it provide any mechanisms for defining the relationships between these properties and other resources. That is the role of [RDF Schema].

Resource description communities require the ability to say certain things about certain kinds of resources. For describing bibliographic resources, for example, descriptive attributes including "author", "title", and "subject" are common. For digital certification, attributes such as "checksum" and "authorization" are often required. The declaration of these properties (attributes) and their corresponding semantics are defined in the context of RDF as an RDF schema. A schema defines not only the properties of the resource (e.g., title, author, subject, size, color, etc.) but may also define the kinds of resources being described (books, Web pages, people, companies, etc.).

The RDFS specification does not specify a vocabulary of descriptive elements such as "author". Instead, it specifies the mechanisms needed to define such elements, to define the classes of resources they may be used with, to restrict possible combinations of classes and relationships, and to detect violations of those restrictions. More succinctly, the RDF Schema mechanism provides a basic type system for use in RDF models. It defines resources and properties such as [rdfs:Class] and [rdfs:subClassOf] that are used in specifying application-specific schemas.

RDF(S) itself, however, lacks the necessary rich concept forming operators that one would expect in an ontology representation language, and is thus not seen as a candidate for such a language. However, richer languages can be built on RDF and RDF(S) (as demonstrated by the [OIL] and [DAML+OIL] proposals). Indeed the layered nature (see Section 5.4) of this approach can provide a benefit. Applications that are aware or “understand” RDF(S) can still process, for example, DAML+OIL ontologies. Although the application may not be able to extract and use all the information within the ontology (such as being able to make inferences about complex definitions), basic information about class hierarchies may be available to such an application.

Resources

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

Rules in (and for) the Web have become a mainstream topic since inference rules were marked up for E-Commerce and were identified as a Design Issue of the Semantic Web, and since transformation rules were put to practice for document generation from a central XML repository. Rules have also continued to play an important role in Intelligent Agents and AI shells for knowledge-based systems, which need a Web interchange format, too. The Rule Markup Initiative has taken initial steps towards defining a shared Rule Markup Language [RuleML], permitting both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential-transformational tasks.

Resources

[RuleML] http://www.dfki.uni-kl.de/ruleml

SHOE

The [SHOE] project, from The University of Maryland's Parallel Understanding Systems Group [PLUS], provided an extension to HTML which allowed the incorporation of machine-readable semantic knowledge in HTML or other World-Wide Web documents. In SHOE, ontologies were fairly simple structures, with an ISA hierarchy of classes/categories, plus a set of atomic relations between these categories, and a set of inferential rules in the form of simplified horn clauses. Categories inherit relations defined for parent categories.

Resources


Thesauri

There are a number of standards pertaining to Thesauri from [ANSI], [ISO] and [NISO]. Although these standards deal partly with content (in that they describe the way that, for instance, compound terms should be used within the thesaurus), they also discuss the kinds of relationship that one would encounter in a thesaurus, e.g. Broader Term, Narrower Term, Related Term etc.

[ISO] 2788:1986, Documentation--Guidelines for the establishment and development of monolingual thesauri is intended "to ensure consistent practice within a single indexing agency, or between different agencies (for example, members of a network)." They are not limited to either pre- or post-coordinate indexing methods, but focus on the use of "preferred terms" for "indexing collections of documents listed in catalogues or bibliographies." They are not intended for "back-of-the-book" indexing, "although many of [the] recommended procedures may be useful for that purpose." The general principles in ISO 2788 are considered language- and culture-independent. As a result, [ISO] 5964:1985, Documentation--Guidelines for the establishment and development of multilingual thesauri, refers to ISO 2788 and uses it as a point of departure for dealing with the specific requirements that emerge when a single thesaurus attempts to express "conceptual equivalencies" among terms selected from more than one natural language.
[NISO] Z39.19 shows how to formulate descriptors, establish relationships among terms, and present the information in print and on a screen. Included are thesaurus maintenance procedures and recommended features for thesaurus management systems.

Resources


**Topic Maps**

Topic Maps provide a formalisation of the notion of a back-of-book index. There is an ISO standard [ISO/IEC 13250] for topic maps, and the claim is that they provide a mechanism for describing knowledge structures and associating them with resources. Topic Maps are based around three notions, that of Topic, Association and Occurrence.

A topic represents any “thing” whatsoever – a person, entity, concept – regardless of whether it exists or has any specific characteristics. The notion of topic types allows the specification of classes of topics – thus topics represent both classes and instances (in the traditional frame-based or description logic senses).

An occurrence of a topic is an information resource that is deemed to be relevant to the topic in some way. Topic associations allow the representation of relationships between topics.

Topic maps are strongly connected to notions of indexing. As such, they may prove to be a useful mechanism when employed as indexing structures within the Semantic Web. However, their lack of built in concept forming operators or constructs with detailed semantics suggests that they may not provide an appropriate formalism for the representation of rich ontologies.

[TopicMaps.Org] is an independent consortium of parties interested in developing the applicability of Topic Maps to the Web. Their work includes the development of version 1.0 of an XML grammar for interchanging Web-based Topic Maps, called XML Topic Maps [XTM] Version 1.0, written by the Topicmaps.Org Authoring Group. All versions of the XTM Specification will be permanently licensed to the public.

Resources


**UML**

The Unified Modeling Language [UML] is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system. UML fuses the concepts of Booch, OMT andOOSE, resulting in a single, common, and widely usable modeling language for users of these and other methods.

UML focuses on a standard modeling language, not a standard process. Although the UML must be applied in the context of a process, experience has shown that different organizations and problem domains require different processes. (For example, the development process for shrink-wrapped software is an interesting one, but building shrink-wrapped software is vastly different from building hard-real-time avionics systems upon which lives depend.) Therefore, the efforts concentrated first on a common metamodel (which unifies semantics) and second on a common notation (which provides a human rendering of these semantics). The
UML authors promote a development process that is use-case driven, architecture centric, and iterative and incremental.

The [UML Specification] is provided by the the Object Management Group [OMG], an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. OMG’s membership roster, about 800 strong, includes virtually every large company in the computer industry, and hundreds of smaller ones. There are a large number of tools supporting UML and the creation of UML models.

UML includes mechanisms such as Use Case, Sequence and Activity Diagrams that allow the description of behaviour as well as structure. In this way, UML provides more than just an ontology representation language. UML class diagrams are more or less in correspondence with ER schemas, having classes, attributes and relations, with is-a links and cardinality constraints. Moreover, a full FOL additional constraint language [OCL] can be used. A drawback of UML, however, is its lack of agreed upon formal semantics. Languages like UML have a great advantage in that they provide graphical representations for ontologies, and have a wealth of tools to support the creation and manipulation of those models. The price to pay, however is that they are usually less expressive than languages such as [DAML+OIL].

Resources

UPML

The Unified Problem-solving Method description Language [UPML] is a language resulting from the [IBROW] project. UPML is a framework for developing knowledge-intensive reasoning systems based on libraries of generic problem-solving components. UPML provides a [Protége] based editor for writing specifications.

Resources

W3C Web Ontology Working Group (WebOnt)

The W3C Web Ontology Working Group [WebOnt], part of W3Cs Semantic Web Activity, will focus on the development of a language to extend the semantic reach of current [XML] and [RDF] meta-data efforts.

The working group will focus on building the ontological layer necessary for developing applications that depend on an understanding of logical content, not just human-readable presentation, and the formal underpinnings thereof.

Specifically, the Web Ontology Working Group is chartered to design the following component:

- A Web ontology language, that builds on current Web languages that allow the specification of classes and subclasses, properties and subproperties (such as [RDFS]), but which extends these constructs to allow more complex relationships between entities including: means to limit the properties of classes with respect to number and type, means to infer that items with various properties are members of a particular class, a well-defined model of property inheritance, and similar semantic extensions to the base languages.

Furthermore, the following general requirements must be met by the work produced by this Working Group:

- The products of the WebOnt group should not presuppose any particular approach to either ontology design or ontology use. In addition, the language must support the development and linking of
ontologies together, in a web-like manner.

- The products of this working group must be supported by a formal semantics allowing language
designers, tool builders, and other "experts" to be able to precisely understand the meaning and
"legal" inferences for expressions in the language.

- The language will use the [XML] syntax and datatypes wherever possible, and will be designed for
maximum compatibility with XML and RDF language conventions.

The Working Group is chaired by Jim Hendler (Univ of Maryland).

**OntoWeb Note** There are close links between WebOnt and OntoWeb, with key members of the OntoWeb
project being involved with WebOnt. OntoWeb is in a prime position to supply requirements and case
studies and thus influence the direction of the Working group.

**Resources**


**XML schema**

The Document Type Definition (DTD) of [XML] 1.0 provides a mechanism for declaring constraints on the
use of markup. Automated processing of XML documents, however, requires more rigorous and
comprehensive facilities in this area. Requirements are for constraints on how the component parts of an
application fit together, the document structure, attributes, data-typing, and so on. [XML Schemas] express
shared vocabularies and allow machines to carry out rules made by people. They provide a means for
defining the structure, content and semantics of XML documents.

XML Schema also provides a facility for defining [datatypes] which has been used by other language
specifications (e.g. [DAML+OIL]).

**Resources**

/XML/ http://www.w3.org/XML/

/XML Schema Part 0: Primer] http://www.w3.org/TR/xmlschema-0/


**XOL**

An early proposal for an XML based ontology exchange language, designed in response to a study of
languages from the [BioOntology Core Group] was the language [XOL]. The semantics of XOL were based
on OKBC-Lite (a simplified form of [OKBC]. XOL was an influence on early drafts of [OIL], but is no
longer being actively developed.

**Resources**


References and Web Resources


