NoE InterOp (IST-508011)

WP8, subtask 8.2

State of the Art Report
Ontology Management and Engineering

Classification	Restricted
Project Responsible :	UNITILB
Authors :	Manfred Jeusfeld (jeusfeld@uvt.nl)
Contributors	members of WP8
Task	State of the Art Ontology Management and Engineering
Version	Version 1.0
Date	15-Nov-2004

History

No.	Date	Description
0.1	07 Sep, 2004	First draft
0.2	17 Sep, 2004	Second draft; included introduction
0.2a	29 Sep, 2004	Extended section on ontology learning (Paola Velardi)
0.2b	6 Oct, 2004	Added to section on validation (Alessandro Cucchiarelli)
0.2c	6 Oct, 2004	Sections 5 and 6 Revised and extended (Nacer Boudjlida and Cheng Dong); minor typo corrections; Refernce list extension according to the section revision and extension.
0.3	7 Oct, 2004	Added more core domain ontologies (Mounira Harzallah); added more on enterprise evoution (Manfred Jeusfeld); added more on ontology tools (Manfred Jeusfeld)
0.3a	8 Oct, 2004	Added material on ontolgy vs metadata (Manfred Jeusfeld)
0.4	11 Oct, 2004	Added material on ontology creation and maintenance (Enrico Franconi)
0.4a	14 Oct, 2004	Added some more core domain ontologies (Manfred Jeusfeld, Maria Bergholtz)
1.0	15 Nov, 2004	Processed comments by peer reviewers

Purpose: This report has the goal to create an account of what is available in the area of ontology engineering and management and relate the start of the art to the goals of enterprise modeling and enterprise application integration.

Note: This deliverable subsumes a section on core domain ontologies which replaces the deliverable on subtask 7.4 ("WP7-related ontologies and domains")!

1. INTRODUCTION

Ontologies are an application-independent way to represent knowledge about the entities of an enterprise. In this report, we present an overview of the aspect of managing ontologies and engineering their content by a group of experts. Hence, rather than reporting on languages and (reasoning) methods (topics addressed in Chapter 1 of this report), this report concentrates on

- how to build an ontology
- what tools are available to support the building process
- how to reach consensus about disputed elements in an ontology
- which workflows are available to create ontologies
- how to maintain ontology versions and control their evolution
- how to learn ontologies automatically from sources
- what are the specificity of ontologies for enterprise application integration

The last item is expected to be the more difficult one since the use of ontologies for enterprise application integration is a relatively new idea. Furthermore, this topic will be extensively addressed in Chapter 4 of this State of the Art. Section 2 is devoted to the state of the art in ontology building and authoring. Section 3 is addressing ontology learning tools. Section 4 gives an overview on core domain ontologies. Section 5 investigates the issues of merging and integration. Section 6 presents the state of the art in validating ontologies. Section 7 addresses evolution and versioning and section 8 multi-lingual and multi-cultural aspects in ontology engineering. Finally, section 9 presents social and cultural challenges that might be an obstacle to the creation of ontologies.

2. BUILDING/AUTHORING

2.1 State of the art on creation and maintenance of ontologies

Ontologies are seen as "explicit specifications of a conceptualisation" (e.g., Wache et al. 2001); in this sense, ontologies are used in the integration task to describe the information source semantics, expliciting their content, so that they can be used for the identification of semantic interrelationships between concepts in the different contexts. In this section the focus is on tools to support users to design ontologies and integrate them. Therefore, we concentrate on the role of ontologies in supporting high-level user-oriented tasks; on the one hand the creation and maintenance of the knowledge by knowledge engineers (our ontology engineers), and on the other the knowledge exploration by both knowledge engineers and final (non expert) users as well (our end-users). We focus here on the creation and maintenance task. In the next section we propose a brief survey on methodologies and methods for creating and maintaining ontologies; after that we show the main systems supporting such an activity.

2.1.1 Methodologies and methods for creating and maintaining ontologies

Different methodologies for building ontologies exist and we can classify (Gomez and Lopez et al 2002) them as follows:

- Methods and methodologies for building ontologies starting from scratch;
- Methods for reengineering ontologies;
- Methods for cooperative ontology construction.
- Ontology merge methods.

For each methodology we will present a brief description and the main research issues.

Methods and Methodologies for building ontologies from scratch. A bunch of approaches have been reported to build ontologies. (Lenat and Guha 1990) published the general steps and some interesting points related to the CYC development process.

The *CYC methodology* consists of the following steps: first, one have to extract, by hand, common sense knowledge that is implicit in different sources. Next, once enough knowledge in the ontology is vailable, new common sense knowledge can be acquired either using nural language or machine learning tools. Some years later (Uschold and King 1995) published the main steps followed in the development of the Enterprise Ontology. The ethod proposes some general steps to develop ontologies, which are:

1. to identify the purpose;
2. to capture the concepts and the relationships among these concepts and the terms used to denote both of them;
3. to codify the ontology.

The ontology has to be documented and evaluated. Other ontologies can be used to build the new one. Michael Gruninger and Mark Fox reported the methodology used for building the *TOVE* (TOronto Virtual Enterprise) ontology in the domain of enterprise modelling (TOVE 1999). One year later, (Uschold and Gruninger 1996) proposed some methodological approach for building ontologies. First, they propose to identify intuitively the main scenarios (possible applications in which the ontology will be used). Later, a set of natural language questions, called *competency questions*, are used to determine the scope of the ontology, that is, the questions that could be answered using the ontology. These questions are used to extract the main concepts, their properties, relationships and axioms, which are formally defined in **Prolog**. Therefore, this is a very formal methodology that takes advantage of the robustness of classic logic, and can be used as a guide to transform informal scenarios in computable models.

(Bernaras at al 1996) presented a method used to build an ontology in the domain of electrical networks as part of the Esprit *KACTUS* project. The ontology is built on the basis of an application knowledge base (KB), by means of an abstraction process. The more the applications, the more general the ontology. In other words, they propose to start the process building a KB for a specific application and, when a new knowledge base in a similar domain is needed, to generalise the first KB into an ontology and adapt it for both applications. Applying this method recursively allows for capturing within the ontology the consensual knowledge needed in all the applications.

Methontology (Fernandez et al 1997) appeared at the same time and was extended a few years later. It is a methodology for building ontologies either from scratch, or by a re-engineering process. The

Methontology framework enables the construction of ontologies at the knowledge level. It includes: identification of the main ontology development process activities (i.e., evaluation, configuration management, conceptualisation, integration, implementation, etc.); a life cycle based on evolving prototypes; and the methodology itself, which specifies the steps to be taken to perform each activity, the techniques used, the output products and the way to evaluate them.

In 1997, a methodology has been proposed for building ontologies based on the *Sensus* ontology (Swartout et al 1997). The proposed method is a top-down approach for deriving domain specific ontologies from huge ones. The authors propose to identify a set of terms that are relevant to a particular domain. Such terms are linked manually to a broad-coverage ontology (in that case, the Sensus ontology, which contains more than 50,000 concepts). They select automatically the relevant terms for describing the domain and pruning the Sensus ontology. Consequently, the algorithm delivers the hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a KB.

Methods for ontologies re-engineering. Ontological reengineering is the process of retrieving and mapping a conceptual model of an implemented ontology to another, more suitable, conceptual model, which is re-implemented. The Ontology Group of Artificial Intelligence Laboratory at UPM presented a method for reengineering ontologies that adapts Chikofsky's software reengineering schema to the ontology domain (Gomez-Perez and Rojas 1999). Three main activities were identified: reverse engineering, restructuring, and forward engineering.

Methods for cooperative ontology construction. An ontology is a shared and common understanding of a domain. Right now, emphasis has been put on the ontology content consensus, i.e., on the collegial agreement on the formal specification of the concepts, relationships, attributes, and axioms the ontology provides. However, the problem of to jointly construct an ontology in a distributed environment is still unsolved. (Euzenat 1995,1996) identified the following problems:

1. concerning collaborative construction of ontologies: management of the interaction and communication among people;
2. data access control;
3. recognition of a moral right about the knowledge (attribution);
4. error detection and management;
5. concurrent management.

Two are the main detailed proposals about how to collaboratively build ontologies, CO4 (Euzenat 1996) proposed for collaborative construction of KBs at INRIA and KA (Euzenat 1995) used in ontologies building at the Knowledge Annotation Initiative of the Knowledge Acquisition Community. CO4 is a protocol to reach consensus among several KBs and it is based on the main idea that people can discuss and commit about the knowledge introduced in the KBs. These KBs are built to be shared, and they have consensual knowledge, hence they can be considered ontologies. The experimentation has been done, above all, in the molecular genetic domain. According to Euzenat's proposal, the KBs are organised in a tree. The leaves are called user KBs, and the intermediate nodes, group KBs. On the one hand, it is not mandatory that the user KBs share consensual knowledge. On the other, each group KB represents the consensual knowledge among its sons (called subscriber KBs). The goal of the Knowledge Annotation Initiative of the Knowledge Acquisition community, also acknowledged as the KA initiative (Decker et al 1999), is to model the knowledge-acquisition community using ontologies developed in a joint effort by a group of people at different locations using the same templates and language.

To make the process of building the Research-Topic ontology easier, the ontology coordinating agent distributes a template among the Ontopic agents, which used e-mail in their intra-communication and also to send their results to the coordinating agents (experts in different topics). The ontology is generated from the knowledge introduced by the template. Once the ontology coordinating agents got all the portions of the ontologies from the Ontopic agents, they integrated them, activity that benefits from the presence of a common pattern

2.2 Ontology Management Systems

In this section, we will provide a broad overview of some of the available tools and environments - not already mentioned above - that can be used for building ontologies, either from scratch or reusing other existing ones. These tools usually provide a graphical user interface allowing for creating ontologies without using a formal specification language. We will provide a brief description of most important proposals.

- *Apollo* (Apollo 2003) is a user-friendly knowledge modelling application. Modelling is based on the basic primitives, such as classes, instances, functions, relationships, etc.. The internal model is build as a frame system according to the internal model of the OKBC protocol. Apollo performs a full consistency check while editing. The application is not bound to any representation language and can be adapted to support different storage formats through I/O plug-in. The user interface has an open architecture (view based) and allows for implementing additional views of the knowledge base. The software is written in Java and is available for the download.

- *JOE* (Java Ontology Editor) is a tool for building and viewing ontologies developed in the Centre for Information Technology, University of South Carolina (JOE 2003). The JOE basic idea is to provide a knowledge management tool that supports multiple cooperative users and distributed, heterogeneous operating environments. Ontologies are represented using a frame-based approach and can be viewed in three different formats: as ER diagrams, as a hierarchies akin to Microsoft Windows file manager, or as a graphical tree structures. JOE allows for writing queries by using a visual representation of the ontology and a point-and-click approach for adding query conditions.

- *OntoEdit* (OntoEdit 2003) is an ontology engineering environment developed at the Knowledge Management Group (AIFB) of Karlsruhe University. It is a stand-alone application that provides a graphical ontology editing environment (that enables inspecting, browsing, codifying, and modifying ontologies, supporting in this way the ontology development and maintenance task) and an extensible architecture for adding new plug-in. The conceptual model of an ontology is internally stored using a powerful ontology model, which can be mapped onto different, concrete representation languages. Ontologies are stored in relational databases (in its commercial version) and can be implemented in XML, FLogic, RDF(S) and DAML+OIL.

- *Protégé-2000* (Protege 2003) is a graphical and interactive ontology-design and knowledge-acquisition environment that is being developed by the Stanford Medical Informatics group (SMI) at Stanford University. It is an open source, stand alone application that provides a graphical ontology editing environment and an extensible architecture for the creation of customized knowledge-based tools. Its knowledge model is OKBC-compatible. Its component-based architecture enables system builders to add new functionalities by creating appropriate plug-in. The Protégé plug-in library contains plug-in for graphical visualisation of knowledge bases, inference-engines for verification of first-order logic constraints, acquisition of information from remote sources such as UMLS and WordNet, semi-automatic ontology merging, etc. It also provides translators to FLogic, OIL, Ontolingua and RDF(S), and can store ontologies in any JDBC-compatible relational database. Plug-in, applications, and ontologies, which have been developed both by the Protégé group and other Protégé users, are available in the Protégé Contributions Library.

- *Chimaera* (Chimera 2003) is a tool mainly intended for merging knowledge base (KB) fragments, which also supports users in creating and maintaining distributed ontologies on the Web. Its two major functions are merging multiple ontologies and diagnosing individual or multiple ontologies. It supports users in reorganizing taxonomies, resolving name conflicts, browsing ontologies, editing terms, etc. The process of KB merging typically involves activities as resolving name conflicts and aligning the taxonomy. This tool has special support for finding name conflicts and for pointing out interesting places in the merged taxonomy.

- *OILEd* is a graphical ontology editor developed by the University of Manchester (Bechhofer et al 2001) that allows the user to build ontologies using DAML+OIL. The knowledge model of OILEd is based on that of DAML+OIL, although this is extended by the use of a frame-like presentation for modelling. Thus OILEd offers a familiar frame-like paradigm for modelling while still supporting the rich expressiveness of DAML+OIL where required. The main task that OILEd is targeted at is that of editing ontologies or schemas, as opposed to knowledge acquisition or the construction of large knowledge bases of instances. A key aspect of OILEd behaviour is the use of the FaCT reasoner (Horrocks 1998) to classify ontologies and check consistency via a translation from DAML+OIL to the SHIQ description logic (Horrocks et al 1999). This allows the user to describe their ontology classes and have the reasoner determine the appropriate place in the hierarchy for the definition. The DAML+OIL RDF Schema (March 2001) is used for loading and storing ontologies.

- *ICOM* is a tool supporting the conceptual design phase of an information system, and in particular of an integration information system -- such as a data warehouse. The tool is an evolution of part of the conceptual modelling demonstrators suite (Jarke et al 2000) developed within the European ESPRIT Long Term Research Data Warehouse Quality (DWQ) project (Jarke et al 1999). ICOM adopts an extended Entity-Relationship (EER) conceptual data model, enriched with multidimensional aggregations and interschema constraints. ICOM is fully integrated with a very powerful description logics reasoning server which acts as a background inference engine. The tool supports multiple schemas with interschema constraints but it turned out to be extremely useful also in supporting the conceptual modelling of "classical" databases involving a single rich schema with integrity constraints, and in designing ontologies for various purposes. ICOM reasons with (multiple) diagrams by encoding them in a single description logic knowledge base, and shows the result of any deductions such as inferred links, new or stricter constraints, and inconsistent entities or relationships. Theoretical results from the DWQ project guarantee the correctness and the completeness of the reasoning process: the system uses the SHIQ description logic, as a mean to provide the core expressivity of the DLR logic developed by (Calvanese et al 1998). ICOM has been installed in more than 1,200 locations.
- The *MOMIS* methodology (Bergamaschi et al 2001) follows a semantic approach to information integration based on the object-oriented logical schema of the information sources. In order to create a global virtual schema of involved sources, MOMIS generates a common thesaurus of terminological intensional and extensional relationships describing intra and inter-schema knowledge about classes and attributes of the source schemata. On the

basis of the common thesaurus content, MOMIS evaluates affinity between intra and inter-sources classes and groups similar classes together in clusters using hierarchical clustering techniques. A global class, that becomes representative of all the classes belonging to the cluster, is defined for each cluster. The global view for the involved source data consists of all the global classes. A graphical tool, the Source Integration Designer, SI-Designer, supports the MOMIS methodology (Bergamaschi et al 1999).

- SymOntoX (Missikofff and Taglino 2003) is an ontology management system that makes use of a web-based interface and targets specifically the management of ontologies for the eletronic business domain. SymOntoX allows the management for multipole ontologies, supports access rights for different user profiles, and supports multiple languages. It provides for a set of feature types to decribe concepts: similar concepts, narrower concepts, part-of concepts, and attributes of concepts. By means of access rights, certain roles can be defined for the creation of ontologies. For example, only a super user can propose new terms while ordinary users are restricted to browsing.

The book Ontological Engineering (Gómez-Pérez et al, 2004) provides a comprehensive list of ontology tools. These include

- Ontolingua: a web-based server that allows to enter ontology definitions using a large number of possible slots; no reasoning capabilities; reasoning rather done by external tools; LISP-like syntax with the ability to define logical axioms
- OntoSaurus: browser for LOOM-based ontologies; LOOM is a variant of description logics supporting concept classification and instance matching
- WebOnto: a front-end to the OCML ontology definition language; largely compatible with Ontolingua; graphical user inferface integrated into web browsers via Java applets; OCML supports rules and definitions for relations and functions; checks consistency constraints and correct typing of attribute values
- WebOde: extensible ontology engineering suite with ports to different external formals (DAML-OIL, RDF(S), F-Logic, etc.); includes an editor and tools for merging and integration of ontologies; includes editor for first-order axioms; querying via its internal concept definition formaty OKBC or via Prolog; several plug-ins have been developed, e.g. ODEClean for checking consistency of concept taxonmies
- KAON: ontology engineering suite based on extension of RDF(S); graphical user interface; links to pre-defined core ontologies accessible via HTTP; support for multi-lingual labels
- PROMPT (Noy and Musen 2000), is an algorithm that provides a semi-automatic approach to ontology merging and alignment. PROMPT performs some tasks automatically and guides the user in performing other tasks for which his intervention is required. PROMPT also determines possible ontology inconsistencies and suggests ways to fix them. PROMPT is based on an extremely general knowledge model and therefore can be applied across various platforms.
- The method of FCA-Merge (Kietz et al 2000) is guided by application-specific instances of the given source ontologies that are to be merged. Natural language processing and formal concept analysis techniques are applied, in order to derive a lattice of concepts. The generated result is then explored and transformedinto the merged ontology with human interaction.
- GLUE: machine learning algorithm for finding similar concepts

- OBSERVER: a merging tool scanning for synonyms and hyponyms; can deal with heterogeneous ontology sources

ONIONS (ONtological Integration Of Naive Sources) is a methodology for conceptual analysis and ontological ntegration or merging of terminologies (Gangemi et at 1999). ONIONS aims to provide extensive axiomatisation, clear semantics, and ontological depth in he domain terminologies that are to be integrated or merged. Extensive axiomatisation is obtained through a careful conceptual analysis of the terminological sources and their representation in a logical language with a rigorous semantics. Analysis ontological depth is obtained by reusing a library of foundational ontologies, on which the axiomatisation depends. Such a library may include multiple choices among partially incompatible ontologies. Consequently, the tools that implement ONIONS (Calvanese et al 2001) should support enough expressivity and classification services.

When comparing the import/export compatibility of the above-mentioned tools, WebODE supports the most ontology dialects and has ports to most plug-in tools. Next follower is Protégé. The main difference is that WebOde favors DAML-OIL (now called OWL-DL by (W3C, 2004)) whereas Protégé is following a first-order axioms representation (now called OWL-Full).

Starting 2003, the ontology community has started to structurally compare ontology tools (Sure and Corcho, 2003) and to develop metrics aboput the expressiveness and performance of ontology tools. The metrics are based on a DAML library of concept definitions. Experiments on ontology parsers and validators (A. Gómez-Perez et al, 2003) showed that RDF(S)-based validation tools could read DAML-OIL ontologies but not detect all types ot errors, whereas DAML-OIL-based tools could read and validate RDF(S)-based ontologies. In general, the report showes that import/export are not yet fully understood.

The survey (Ribiere and Charlton, 2001) lists besides the well-known ontology languages KIF, OKBC, RDF(S), XOL, OIL, DAML+OIL the ontology management tools Protege2000 and OntoEdit.

The papers (Cui et al 2001, Cui and O'Brian 2000) discuss a so-called Domain Ontology Management Environment for creating and mapping between ontologies, browsing and constructing queries. The DOME prototype was developed mainly for integrating e-commerce applications and to bridge the semantic gap in this domain. Their method combines both top-down and buttom-up developement of ontologies. However, not many details are given how that is actually achieved.

The survey (Denny 2002) lists editing tools available and links to building ontology and comparison of languages. It lists more than 50 ontology editing tools. The survey shows that the ontology tools are quite diverse on the ontology language they support. The main languages supported are: XML, RDF, UML, DAML, LOOM, F-Logic, Prolog, CycL. Exchange formats include XML, RDF, ERD, DAML OIL, and KIF. OntoBuilder and OntoLingua have the broadest support for import/export. Very fiew porvide graphical views of the ontologies. If they do, they rely on UML or ER-like representations. The tools based on a formal representation language also tend to provide support for consistency checks. The majority of tools does not support multiple users. Details are listed in appendix 2.

This paper by (Klein et al 2002a) addresses how the ontologies need to be stored, sometimes aligned and also how it is possible to manage their evolution. It provides an introduction on these themes and addresses also ontology mapping, proposing the RDFT (RDF Transformation) mapping meta-ontology, that specifies a small ontology for mapping XML DTDs to/and RDF Schemas and is built on top of RDF Schema. According to (Stojanovic 2002) ontology editors are main tools for ontology development. Ontologies must be able to evolve for a number of reasons including changing user needs and changes in the application domain. An ontology editor thus has to support ontology evolution. A more recent survey on ontology editors is contained in (Sure, Corcho 2003). In one paper

(Troncy et al. 2003), the DOE tool and its methodology is presented. The tool relies on XSLT style cheets for exchaniging ontology models. Two transformation techniques for ontology mappings are investigated. One uses RDFS, the other OWL. The OpenGalen system (Rector et al, 2001) has been used extensively in the medical domain, in particular for dealing with SNOMED. It follows an elaborated workflow.

The report by (Noy et al 2002) considers the very complex domain of human anatomy and discusses what sorts of logical formalisms are required. There is a discussion of the connection between "frame based" systems and FOL, and a discussion of the implementation in Protege.

The paper of (Kim 2001) extensively discusses and presents a methodology for ontology development (combining both data driven and process driven approaches) called BPD/D Ontology Engineering Methodology.

2.3 Consensus Systems and Groupware for Ontology Communities

The OntoWeb project (OntoWeb 2002) has created a survey of the most relevant methodologies and methods used for building, maintaining, evaluating and reengineering ontologies. It addresses also some of the main ontology merging methods and tools (ONIONS method, PROMPT, FCA-Merge, IFOM, and MOMIS).

The work by (von Buol 1999) presents a detailed process model for coordinating the work of ontology builders. First, she proposes to define so-called *reference models* that define which types of ontological concepts are allowed in which relations are pre-defined. These reference models also can contain specific constraints. Second, she defines *quality models* both for the products (the ontologies) as for the production process (building the ontologies). Other than for the constraints in the reference models, the quality models measure properties by metrics. In case of low quality, a change process is started.Thirdly, an *execution model* based on events is proposed that defines which experts are to be involved in which situations during the ontology building. The approach has been investigated in the area of medical ontologies. She indentifies the following steps for building ontologies:

1. *Goal definition*: specifies the goal of ontology building project. It also defines the type of the ontology to be built. This step can not be revised.
2. *Team forming*: the members of the ontology building project are invited
3. *Project milestones*: the team defines improtant milestones in their calendar; criteria for milestone fulfillment are defined
4. Role assignement: depending on the expertise of team members, responsibilities for certain sub-sections of the ontology are assigned
5. Document acquisition: literature about the domain is collected
6. Concept extraction: experts extract the important concepts out of the documents
7. Concept grouping: the extracted concepts are grouped into hierarchies
8. Concept relations: experts create links between the concepts dependent on the type of the ontology (the type defines the allowed links)
9. Quality check on hierarchy: experts evaluate whether the concepts are classified into the rights position in the hierarchy; they also check whether tere are missing concepts
10. Quality check on redundancy: in particular synonyms and polysems
11. Variant check: if there are multiple definitions for a concept (e.g. proposed by different experts), these variants are resolved, e.g. by voting
12. Detailed concept definition: the detailed definition of a given concept is created
13. Decision on naming: the preferred name and possibly alternative names (synonyms) are decided

14. Quality check on naming and detailed concept definition

The DOE editor by (Bachimont et al 2002) realizes a tool-supported methododology for ontology building. In particular it helps users to organize concepts in subsumption hierarchies.

2.4 Limitations of the Current Approaches

Today, the key role of ontologies in information management in general and the Semantic Web in particular, has led to the rapid development of a large number of ontologies. These ontologies have, however, usually been developed in an ad-hoc manner by domain experts, often with only a limited understanding of the semantics of ontology languages. The result is that many ontologies are of low quality - they make poor use of the languages in which they are written and do not accurately capture the author's rich knowledge of the domain. To the best of our knowledge, an ontology building methodology which uses the support provided by state-of-the-art logic based ontology languages and also helps the user to take advantage of the full expressive power of such a language providing a positive and constructive end user interaction experience is still missing.

Such a methodology is indeed needed within an ontology design tool, in order to build high-quality ontologies since (1) domain experts are, in general, not experts in the ontology language they are using, (2) formalising one's knowledge in any kind of formalism is, in general, a highly complicated task which requires a lot of discipline, perseverance, and/or support. These problems are often aggravated by the fact that one ontology is built by several domain experts, and that ontologies need to be maintained, extended, and most importantly integrated/interoperated over the time.

Other limitations of the current methodologies arise from the current status of the reasoning techniques employed by the state-of-the-art ontology systems. First of all, the expressive power required for representing high-quality ontologies is still not fully provided. Secondly, the sheer size of realistic ontologies pose problems that may require new optimisation techniques for reasoning. Thirdly, the fact that ontologies need to be integrated and interoperated requires the investigation of novel paradigms (such as a mixed global-as-view and local-as-view approach) and of reasoning problems tailored to supporting these tasks.

Poor quality ontologies usually require localised "tuning" in order to achieve the desired results within applications. This leads to further degradation in their overall quality (e.g., their interpretability), increases the brittleness of the applications that use them, and makes interoperability and re-use difficult or impossible. These considerations will be of crucial importance in the Semantic Web projects, where it is expected that (very large numbers of) web pages will be marked up using multiple ontologies, without the authors of the ontologies having a precise idea of what would be the effects of inserting the authored ontology in the larger integrated context generated by the brokering agents.

In particular, ontologists need clear and measurable quality criteria, and tools that take them into account when building, maintaining, and inter-operating ontologies. To "take into account" means both to measure the quality of an ontology w.r.t. specific quality criteria as well as to support the user when operating an ontology. The latter point requires that the user, when operating an ontology, can state her quality requirements, which are then taken into account by the system supporting this operation.

The OntoClean method (Guarino and Welty 2002) is about annotating concepts (or its properties) of an ontology by concepts of a backbone ontology. The backbone ontology is about concepts like essence (how rigid is a concept) and identity (how to tackle the problem of changes to an object over time). The goal of OntoClean is the correct use of relationship types (like 'isA') between concepts.

3. AUTOMATIC TECHNIQUES FOR ONTOLOGY LEARNING (E.G., TEXT MINING)

Comprehensive ontology construction and learning has been an active research field in the past few years. Several workshops (ECAI, 2000) (ECAI, 2002) (ECAI, 2004) and (IJCAI, 2001) have been dedicated to ontology learning and related issues.

The majority of papers in this area propose methods to extend an existing ontology with new words, using natural language processing, statistical and machine learning techniques. Very often, the existing ontology is WordNet, a linguistic general purpose ontology, or MeSH, a taxonomically organized thesaurus in medicine. The motivation is that these resources, though partly or scarcely compliant with the computer science formal view of an ontology, have a large coverage and are widely available.

In (Agirre at al, 2000) a method is proposed to enrich WordNet with topic signatures extracted from on-line documents. In (Alfonseca and Manandhar 2002) an algorithm is presented to enrich Wordnet with unknown concepts on the basis of hyponymy patterns. For example, the pattern: *hypernim(N2,N1):-appositive(N2,N1)* captures an hyponymy relations between *Shakespeare* and *poet* in the appositive NP: "Shakespeare, the poet...".

(Berland and Charniak, 1999) propose a method to extract whole-part relations from corpora and enrich an ontology with this information. A method to extract taxonomic relations from heterogeneous evidences (extracted from corpora and web sites) is proposed in (Cimiano et al., 2004). Both these paper do not rely on a specific existing ontology.

A few papers propose methods to extensively enrich an ontology with domain terminology. For example (Vossen, 2001) use statistical methods and string inclusion to create syntactic trees, as we do (see Figure 4). However, no semantic disambiguation of terms is performed. Very often, in fact, ontology learning papers regard domain *terms* as *concepts*. (Navigli and Velardi, 2004) propose a method for assigning a *sense* to complex multiword expressions based on compositional interpretation and a novel algorithm for semantic disambiguation. The method is applied to several domains (e.g. tourism, ecomomy, computer networks). An integrated ontology management and learning architecture, based on the system extensively presented in (Navigli and Velardi, 2004) is proposed in (Missikoff et al., 2002).

A statistical classifier for automatic identification of semantic roles between terms is presented in (Gildea and Jurafsky, 2001). In order to tag texts with the appropriate semantic role they use a training set of 50,000 sentences manually annotated within the FrameNet semantic labeling project.

(Missikoff et al 2002) have developed a software environment, centered around the OntoLearn tool, that can build and assess a domain ontology for intelligent information integration within a virtual community.

Finally, in (Maedche and Staab, 2000 and 2001) an architecture is presented to help ontology engineers in the difficult task of creating an ontology. The main contribution of this work is in the area of ontology engineering, although machine learning methods are also proposed to automatically enrich the ontology with semantic relations.

4. Core Domain Ontologies

4.1 WordNet

WordNet is an extremely large and freely available online English lexical database (WordNet 2003). The database is divided by part of speech into nouns, verbs, adjectives, and adverbs, The nouns are organized as a hierarchy of nodes. In version 2.0 of WordNet, there are 141690 noun synsets, 24632 verb synsets, 31015 adjectives and 5808 adverbs. WordNet is continually updated, and several versions of the database currently used in Information Retrieval and Natural Language Processing applications.

Figure 4.1: Links between words in WordNet

English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing

Table 1

Illustrating the Concept of a Lexical Matrix:

F_1 and F_2 are synonyms; F_2 is polysemous

Word Meanings	Word Forms F_1	F_2	F_3	. . .	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2		$E_{2,2}$			
M_3			$E_{3,3}$		
⋮				. . .	
M_m					$E_{m,n}$

one underlying lexical concept. Different relations link the synonym sets (Fig. 4.2).

Figure 4.1 illustrates links between word signification and forms and synonym words. This allows to make a clear difference between WordNet and a classic dictionaries. In fact, the relationships (is-a, part-of, synonymie, etc.) offer the possibility to find the signification of concepts and their relationships with other concepts. For example, if we want to know the components of a taxi, we can't find them in a dictionary but we can find them in Wordnet.

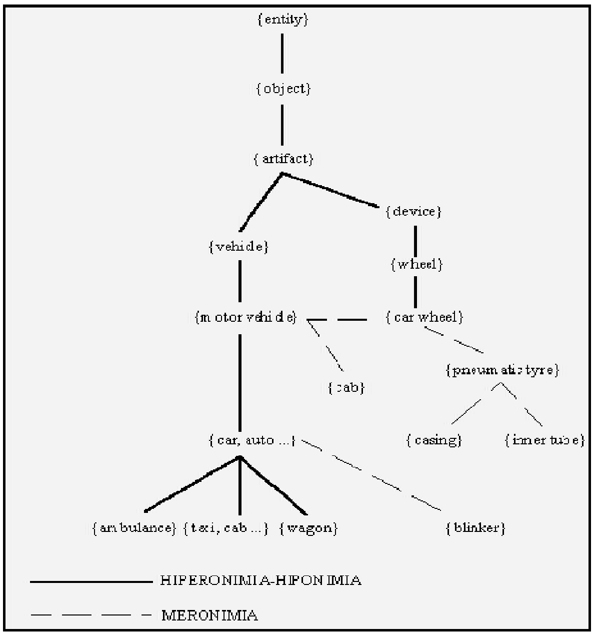


Figure 4.2. Example of word structuring in WordNet

EuroWordNet is a multilingual database with WordNets for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian) (Vossen Piek 1999). The wordnets are structured in the same way as the WordNet in terms of synsets (sets of synonymous words) with basic semantic relations between them. Each wordnet represents a unique language-internal system of lexicalizations.

4.2 Upper Cyc

The Upper Cyc ontology captures the most general concepts of human consensus reality. It includes more than 3000 terms and assertions which relate those terms. This ontology is divided into many (currently thousands of) "microtheories", each of which is essentially a bundle of assertions that share a common set of assumptions. CycL, the Cyc representation language, is a large and extraordinarily flexible knowledge representation language. It is essentially an augmentation of first-order predicate calculus (FOPC) (CycL 2004). This ontology is used for natural language processing.

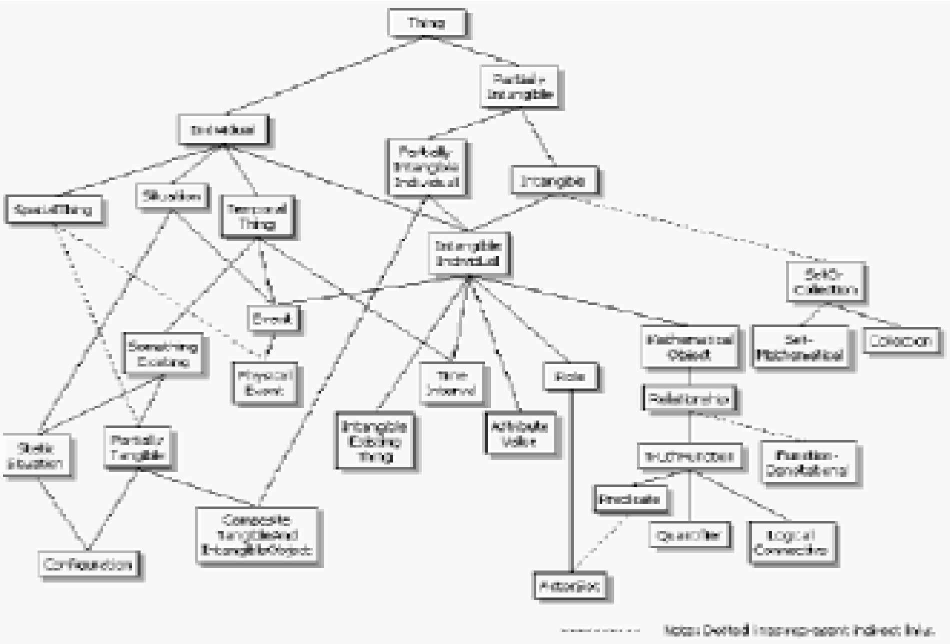


Figure 4.3. The Upper Cyc ontology

4.3 Sowa's Ontology

Sowa's ontology is for representing knowledge. It is based on his book "Knowledge Representation". The upper level of this ontology is presented by Fig. 4.4. In this ontology, T is the universal type and ⊥ is the type absurd (interpreted by the empty set). The structure of this ontology is the trellis. The trellis (or lattice) is a structure that has a relationship of order and has two operators ($\cap$ and $\cup$) with the following proprieties:

- $a \cap b \leq a$ et $a \cap b \leq b$.
- If c is an element of L for which $c \leq a$ and $c \leq b$, then $c \leq a \cap b$.
- $a \leq a \cup b$ and $b \leq a \cup b$.
- If c is an element of L for which $a \leq c$ and $b \leq c$, then $a \cup b \leq c$.

The nodes of the trellis can be concepts or attributes. This ontology uses as structure of representation the trellis and the FCA – « Formal Concept Analysis » to do improvements [Sowa01].

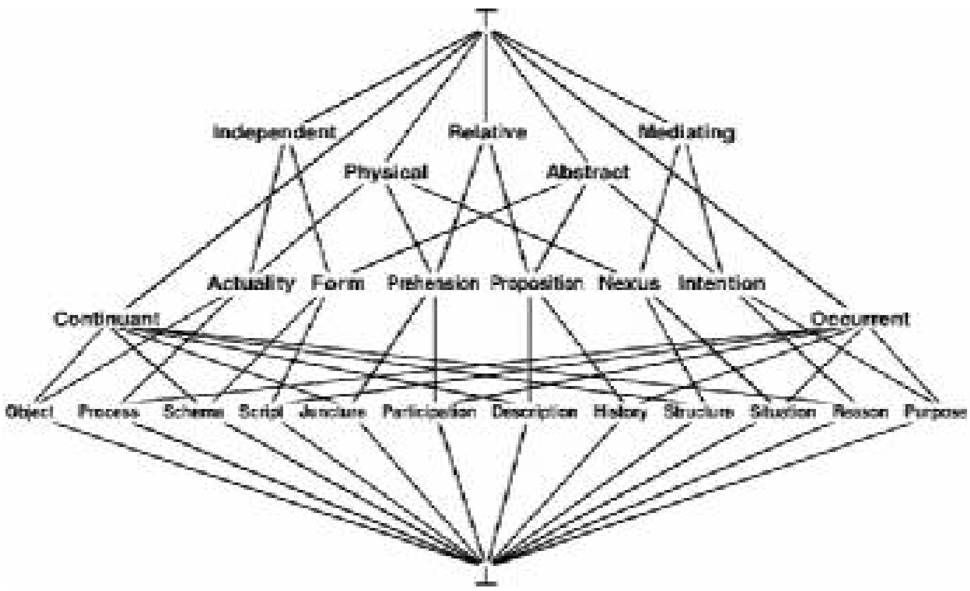


Figure 4.4. The upper level of Sowa's ontology

4.4 Sensus

Sensus is a taxonomy of approximately 50,000 symbols that represent the semantic meanings conveyed in translations, for natural language processing. It is being constructed at USC/ISI by extracting knowledge from a variety of sources (the most important one is Wordnet).

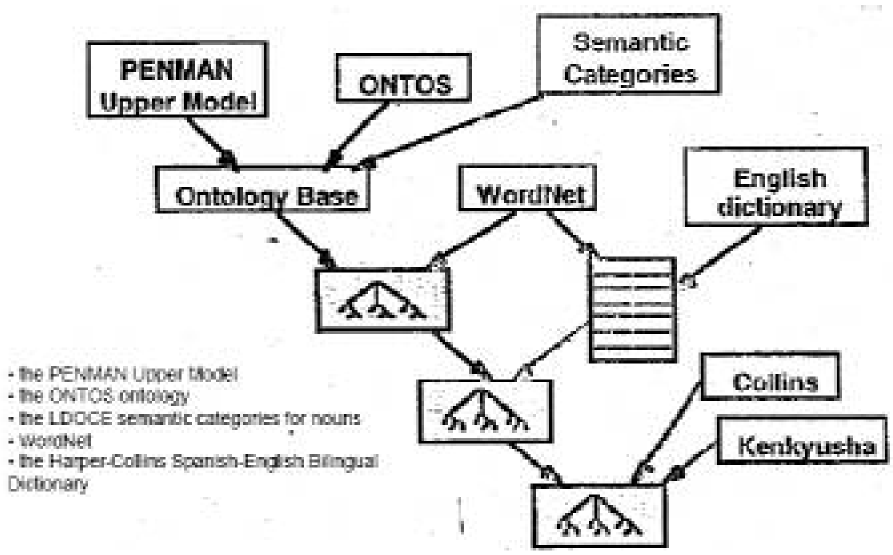


Figure 4.5: The sources of the ontology Sensus

The architecture of Sensus and its sources are presented by Figure 4.5. It is represented in Loom, FrameKit, and Prolog (Swartout 1997).

4.5 The Enterprise Ontology

The Enterprise Ontology (Enterprise Ontology 2003) is a collection of terms and definitions concerning the business domain. This ontology was developed by "Artificial Intelligence Applications Institute" of the University of Edinburgh in order to improve the enterprise activity. A best definition of concepts of enterprise and relationships between them allows a best modeling and analyze of enterprise. This ontology is devised on several parties: Activities and processes, Organization, Strategy and Marketing.

Developed on Ontolingua, it includes 92 classes, 68 relationships, 483 axioms, 10 individuals and 7 functions. In the top of hierarchy, Classes are Eo-Entity, Role-Class, Segmentation-Variable, Set-Class, State-Of-Affairs. Classes, slots and axioms are constructed automatically on KIF. Fig 4.6 represents the graphic interface of Ontolingua. This figure presents the class Eo-Entity as seen on Ontolingua browser. We can observe classes to which it belongs (for example Thing, Relation, etc.) (1) and information about this class (2). Also, in this page, we can find sub*classes (for example Activity-or-Spec, For-Sale, Good-Service-or-Money, Legal-Entity, Market, Misc-Special-Detail, Need, Special-Actor, Potential-Sale, Sale), super-classes (for example Individual-Thing, Individual, Thing) and axioms.

The Enterprise Ontology has been developed to provide a method and a set of computer tools for enterprise modelling. It is intended:

1. to ensure non ambiguous communication between enterprise parties (from business managers to software engineers),
2. to help in information exchange between users, tasks and systems,

3. to improve interoperability thanks to the development of translators that convert terms used by a tool or an IT system to be integrated.

The proposed EO is not a final one nor an exhaustive one. It needs to be extended to include more details related to to specific business applications. Indeed, the idea is that the EO contains most of the general terms related to an enterprise (e.g. sales, strategy, etc.) and that the applications extend the EO with their own terms (e.g. bid analysis, project portfolio analysis, etc.). In this work, EO has various forms as shown on figure 4.7.

Figure 4.6:Class Eo-Entity of Enterprise Ontology - on Ontolingua

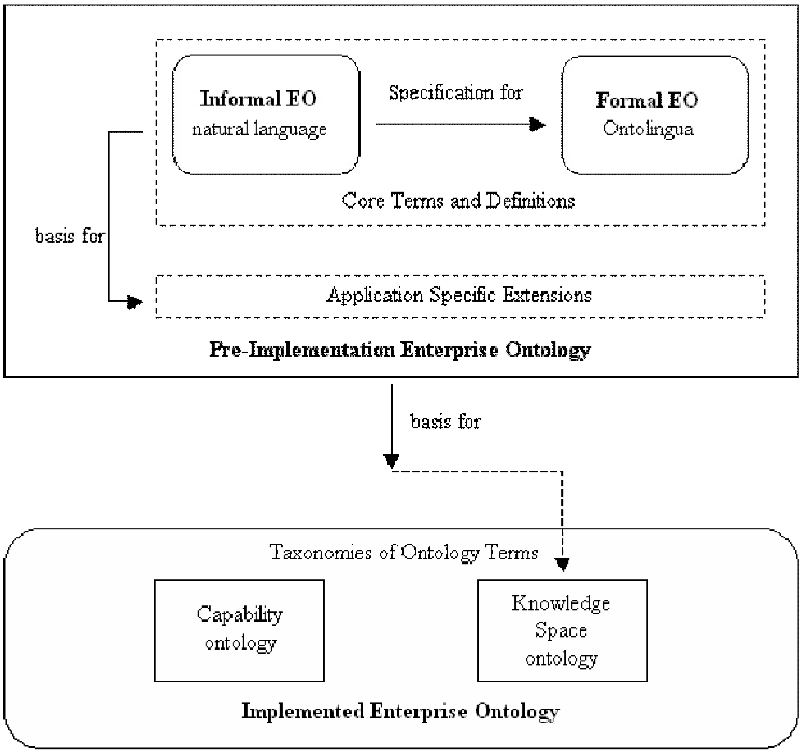


Figure 4.7. Forms of the Enterprise Ontology

Further, the EO definition process was performed thanks to the following steps:

1. Identify the scope and the boundaries for the ontology,
2. Choose the terms,
3. Provide a definition to the choosed terms. These definitions may rely on a a restricted number of building blocks (like Entity, Relationship, State of Affairs, Time and Roles) referred to as a *"meta-onotology"*: Entities pays roles ine relationships and States of Affairs designate situations characterized by a combination of entities being in relationships with one another.

Figure 4.8 sumarizes the content of the proposed EO (see (Uschold and al 1997) for the actual definition of the terms) where the columns:

- ACTIVITY contains terms related to processes and planing (like activity, planning, authority, Resource allocation),
- ORGANISATION contains terms that concern an organization structure (like person, organizational unit, legal entity),
- STRATEGY includes terms relevant to high level planning (like purpose, mission, decision, success factors),
- MARKETING contains terms related to marketing and selling of goods and services (like sale, customer, produc and price)
- TIME includes terms that concern time (like time interval) together with terms that constitue the *meta-ontology*.

Together with the unformal definition of the terms, a translation of the unformal EO into a formal EO is proposed where axioms may be introduced to formally specify termes properties and constraints.

Futhermore, to enable effective use of the EO and tool integration, an agent-based architecture for an Enterprise Tool Set has been designed. This includes:

1. An *Agent Toolkit* that transforms a tool into an agent and registers that agent in the Tool Set,
2. a *Procedure Builder* that generates process models and makes them available to the Tool Set,
3. a *Task Manager* that enacts process models and runs end users applications.

The Tool Set also includes facilities to edit and browse a hierarchy of ontology terms.

Finally, the defined EO has been experimented in various applications, including bid analysis, market analysis and continuos process improvement.

ACTIVITY, etc.	*ORGANISATION*	*STRATEGY*	*MARKETING*	*TIME*
Activity	Person	Purpose	Sale	Time Line
Activity Specification	Machine	Hold Purpose	Potential Sale	Time Interval
Execute	Corporation	Intended Purpose	For Sale	Time Point
Executed Activity Specification	Partnership	Purpose-Holder	Sale Offer	
T-Begin	Partner	Strategic Purpose	Vendor	
T-End	Legal Entity	Objective	Actual Customer	
Pre-Condition	Organizational Unit	Vision	Potential Customer	
Effect	Manage	Mission	Customer	
Doer	Delegate	Goal	Reseller	
Sub-Activity	Management Link	Help Achieve	Product	
Authority	Legal Ownership	Strategy	Asking Price	
Activity Owner	Non-Legal Ownership	Strategic Planning	Sale Price	
Event	Ownership	Strategic Action	Market	
Plan	Owner	Decision	Segmentation Variable	
Sub-Plan	Asset	Assumption	Market Segment	
Planning	Stakeholder	Critical Assumption	Market Research	
Process Specification	Employment Contract	Non-Critical Assumption	Brand	
Capability	Share	Influence Factor	Image	
Skill	Shareholder	Critical Influence Factor	Feature	
Resource		Non-Critical Influence Factor	Need	
Resource Allocation		Critical Success Factor	Market Need	
Resource Substitute		Risk	Promotion	
			Competitor	

Figure 4.8. *Overview of Enterprise Ontology*

4.6 TOVE

The TOVE Ontology (Toronto Virtual Enterprise) is set of integrated ontologies to create a model of data having the following characteristics:

- It provides a shared terminology for enterprise, that can be understand and used by each agent
- It defines the sense of each term.
- It constructs a set of axioms to describe the semantic with which it is possible to answer to many questions.
- It defines modality of representing terms and concepts graphically

Entities of TOVE are objects with proprieties and relationships. Objects are structured on taxonomies and the definitions of objets, attributes and relationships are specified on the logic of the first order. This ontology is built as follows: Objects are identified and represented by constants and variables. Then, proprieties of objets are identified and represented by predicates. A set of axioms isconstituted on a "microtheory" that provides a specification for activities to model. These "microtheories" should contain a set of axioms to resolve problems.

The first ontology is the ontology of the time and actions. It is based on the relationship "<" that represents ordering between time/action instances. The second is the ontology of states (Fig. 4.9).

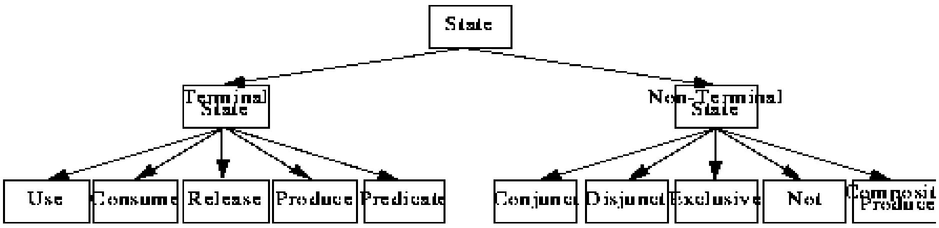


Figure 4.9. TOVE ontology of states

The third one is to represent resources. Resources are considered as roles and entities implied on activities. The others are the ontologies of product, organization, and the management of the cost. All these ontologies are put together to define the model of the enterprise (TOVE 1999).

Another important business ontology is contained in the Process Handbook of MIT (MIT Process Handbook 2003). It defines about 3000 business activities subsumed under 'buying', 'making' and 'selling'. The business concepts are defined textually with some ability to define related concepts and properties of concepts. The concepts (activities) defined oin the process handbook are linked to a generic business activity model which relates the key activities. Activities are decomposed into sub-activities.

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) (Masolo et al 2003) is reference point to clarify the assumptions underlying an ontology. It follows the philosphical roots of ontologies by distinguishing natures of entities, e.g. whether they are enduring (objects that do not change) are perduring (objects who acquire new properties over time). The DOLCVE ontology has been used as founding ontology for domain ontologies.

4.7 Other Related Work

Product ontologies are used to unify the reference to product groups. They appear in form of classification (or coding systems). The E-class classification system (E-class 2004) is a comprehensive effort to provide a four-level classification system for products regardless of the industrial sector. E-class is a commercial effort from German industry and is directed to e-procurement. It supports multi-lingual access. At the lowest level, describing attrributes can be attached to the concepts. A similar proposal (Jeusfeld 2004) has been developed by the Esprit **MEMO** project (Mediating and monitoring electronic commerce). It designed multi-lingual product ontologies (multiple versions for different roles on a veretival market) where concepts and concept attributes (size, color, fire resistance etc.) are treated uniformly. The goal of the MEMO product ontologies is to provide access to source product databases regardless of the structure of the source databases.

The **REA** (Resource-Event-Agent) framework builds on Sowas ontology to provide an enterprise ontology, i.e. an ontology of the core concepts in an enterprise (McCarthy 1979,1982; Geerts and McCarthy 2000). It extends Sowas ontology by concepts for value chains and commitments. Its goal is to describe the interaction of agents in an enterprise and to formulate axioms for these interactions. REA distinguishes operational infrastructure (tasks, processes, scripts) from the knowledge infrastructure (process types, exchange types). The latter can be seen as the type lavel for the former. It is also used for expressing enterprose policies.

SUMO (Suggested Upper Merged Ontology) was promoted by the IEEE Standard Upper Ontology effort (SUMO 2003). The SUMO was created by merging publicly available ontological content into a single, comprehensive, and cohesive structure. As of February 2003, the ontology contains 1000 terms and 4000 assertions. It is implemented in DAML+OIL.

Many ontologies are built for different domains. In this section, several ontologies classified according the domain considered are presented. Some of them have been discussed before.

Linguistic domain

- Wordnet is a lexical database for the English language
- Upper Cyc is an ontology concerning a database of generic knowledge.
- Sensus is a composition of many sources of knowledge
- SUMO – "Suggested Upper Merged Ontology" is an ontology for the upper level proposed by IEEE.
- GUM – "Generalized Upper Model" is an ontology for organizing information to express them on the natural language. It is developed on LOOM.
- EuroWordNet, Germanet – it is an ontology similar to Wordnet.

Medical domain

- UMLS (Unified Medical Language System) is an ontology that contains more than 1000000 concepts and 2500000 words that define concepts. A concept can have denominations in many languages. Eleven types of different relationships can be existed between concepts (UMLS 2001).
- TAMBIS – is an ontology to describe biologic information. It is developed on DAML+OIL (Tambis 2001).

Geographic domain

- CIA World Factbook is an ontology built by CIA to describe the state of the world. This ontology is modified each year. It provides social and geographic information (CIA 2001). An example of a country description can be found in: http://www.cia.gov/cia/publications/factbook/geos/fr.html

Enterprise domain

- BMPO (Business Process Management Ontology) – is an ontology that provides a platform to define business processes. This ontology uses the language BMPN (Business Process Modeling Notation). It has as elements: entities, tasks and context.
- TOVE – "Toronto Virtual Enterprise" is a set of integrated ontologies for enterprise.
- Enterprise Ontology is a collection of terms concerning the business domain.
- PH: Process Handbook developed by Massachussets Institute of Technology (MIT Process Handbook 2003)

The paper by (Söderström 2002) addresses ontologies in the e-business domain and finds that even the concept of 'standard' is hard to agree upon.

The paper (Hoy 2001) lists a few reference ontologies such as CYC (general domain), SENSUS (general domain), MIKROKOSMOS (natural langue translation).

5. MERGING, INTEGRATION

5.1 Mapping, Merging, Integration: a tentative definition

Ontology mapping is an activity that attempts to relate the vocabulary of two ontologies that share the same domain of discourse. There is lot of work originating from different communities that express some relevance to ontology mapping. For example, terms and work encountered in the literature include *alignment, merging, articulating, fusion, integration, morphism* and so on.

(Kalfoglou and Schorlemmer 2003) gives a state of the art report on ontology mapping including theoretic frameworks, methods, and tools. They also provide experiments comparing some tools.

They adopt an algebraic approach where an ontology is a pair ***O=(S, A)*** where ***S*** denotes the *ontological signature* (it describes the vocabulary) and ***A*** denotes a set of *ontological axioms* (roughly speaking, ***A*** expresses the intended interpretation of ***S***). The mapping is then defined as a function that preserves the mathematical structure of ontological signatures and their intended interpretations, as specified by the ontological axioms i.e. mapping considered as ontology *morphisms* (like for instance, a morphism of posets i.e. a function ***f*** that preserves a partial order: $a \le b$ implies $f(a) \le f(b)$). Ontology mappings are then characterized as morphisms of ontological signatures and the mapping may be total or partial.

(Pinto and al 99) have a less formal approach in separating ontology integration into three cases: *integration, merge* and *use.* In the following, we rely on their definitions.

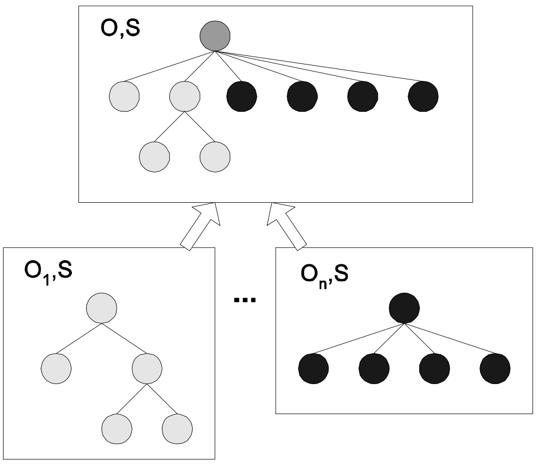


Figure 5.1. Integration

Integration when building a new ontology reusing other available ontologies. In integration we have one or more ontologies ($O_1, O_2, \ldots, O_n$,in figure 5.1) that exist, and a new one (***O***, in the figure 5.2) is created as a result from the integration process. Each ontology integrated in the resulting ontology usually is about a different domain either from the resulting ontology (***D***) or the various integrated ontologies domains ($D_1, D_2, \ldots, D_k$). The resulting ontology usually encompasses whole or parts of the ontologies that are used in the integration process. It may also include additional knowledge.

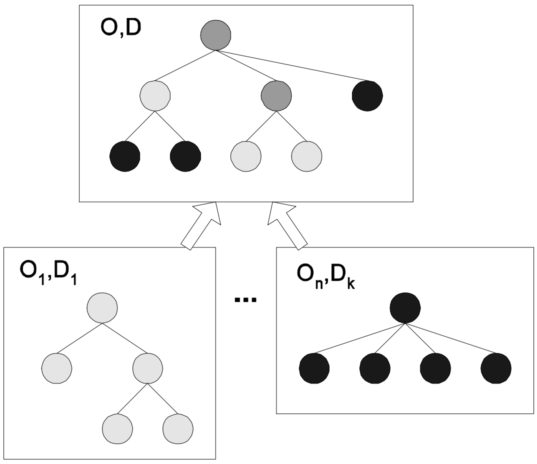


Figure 5.2. Merge

Merge when building an ontology by merging several ontologies (at least two) into a single one that unifies all of them. In figure 11, ***O_1,O_2,… ,O_n*** are merged into ***O***. The goal is to make a more general ontology about a subject (***S***) by gathering knowledge from several other ontologies in that same subject. The subject of both the merged and the resulting ontologies are usually the same (***S***).

Use when building an application using one or more ontologies. In *use*, there are one or more ontologies involved (***O_1,O_2,… ,O_n***) and there is no resulting ontology. One cannot draw any conclusions as to the architecture of the resulting application because that depends on the application itself.

5.2 Some approaches to Ontology integration and merge

(Klein et al 2002) argue that ontologies need to be stored, sometimes aligned and also require facilities to manage their evolution. They provides an introduction on these themes and addresses also ontology mapping, proposing the RDFT (RDF Transformation) mapping meta-ontology, that specifies a small ontology for mapping XML DTDs to/and RDF Schemas and is built on top of RDF Schema.

(Stumme and Maedche 2003) specifically address ontology merging, namely the FCA-Merge method. This method is exploiting representations of examples of ontology concepts, natural-language processing, and formnal concept analysis. It is not automatic but iassists the human expert in her merging activity.

The paper by (Hovy 2001) proposes a concept-by-concept approach to compare two ontologies. By that one can also find justifications for the presence in one ontology by its presence in the other ontology. The work is motivated by the search for a standard reference ontology. The relative match between two ontology concepts (stemming from different ontologies) is expressed by a score.

Another approach to mapping focused of the study on *lexical and figurative meaning* (Ahrens and al 2003). Some online electronic dictionaries were developed for assisting the processing of natural languages. They support some relationships between the terms (name of concept), as *name equality, synonyms, homonyms, hyponyms, abbreviations* and so on. Some of these studies are briefly reviewed hereafter.

Additional information on ontology merging is contained in the survey by (OntoWeb 2002). Let us now turn our attention to ontology integration as compared to database schema integration.

5.3 Database Schema Integration vs Ontology Integration

Some work performed in the database schema integration domain might also be applicable to ontology merging and integration. See for example (Parent and Spaccapietra 2000) and the very wide literature about view and schema integration in database domain[1]. Most of recent approaches map local schemas to a global (or canonical) data model. The purpose of such an integration may be, for instance, a uniform access to distributed databases or database evolution.

Even though from the ontology domain some authors argue that ontology evolution is not the same as schema evolution (Noy and Klein 2002), in the following we elaborate on some approaches for schema matching that use artificial intelligence techniques.

Schema matching is a basic problem in many database application domains, such as heterogeneous database integration, E-commerce, data warehousing, and semantic query processing. Most work on schema match has been motivated by schema integration. Given a set of independently developed schemas, construct a global view has been investigated (Spaccapietra and Parent 1998). In an artificial intelligence setting, this is the problem of integrating independently developed ontologies into a single one.

(Rahm and Bernstein 2001) presents an analysis of seven published prototype implementations that uses AI techniques for schema and data sources matching.

The **SemInt** (Clifton and Li 2000) match prototype creates a mapping between individual attributes of two schemas (i.e. its match cardinality is 1:1). SemInt uses neural networks to determine match candidates. This approach requires similar attributes of the first input schema to be clustered together. SemInt represents a powerful and flexible approach to hybrid matching, since multiple match criteria can be selected and evaluated together.

The **LSD** (Learning Source Descriptions) system uses machine-learning techniques to match a new data source against a previously determined global schema, producing a 1:1 atomic-level mapping (Doan and al 2001). It represents a composite match scheme with an automatic combination of match results. A global matcher that uses the same machine-learning technology is used to merge the lists into a combined list of match candidates for each schema element. It can take self-describing input, such as XML, and make its matching decisions by focusing on the schema tags while ignoring the data instance values.

The **SKART** (Semantic Knowledge Articulation Tool) prototype follows a rule-base approach to semi-automatically determine matches between two ontologies (Mitra and al 1999). Rules are formulated in first-order logic to express match and mismatch relationships and methods are defined to derive new matches. SKAT is used within the NION (Mitra and al 2000) architecture for ontology integration. In ONION, ontologies are transformed into a graph-based object-oriented database model.

The **TransScm** prototype (Milo and Zohar 1998) uses schema matching to derive an automatic data translation between schema instances. The matching is performed node by node (element-level, 1:1) starting at the top and presumes a high degree of similarity between the schemas. In (Palopoli and Sacc 1999,Sacc and Ursino 1998), Palopoli et al. propose algorithms to automatically determine synonym and inclusion (is-a, hypernym) relationships between objects of different entity-relationship schemas. The algorithms are based on a set of user-specified synonym, homonym, and inclusion properties that include a numerical "plausibility factor" (between 0 and 1) about the certainty that the relationship is expected to hold. This algorithms are embodied in the ***DIKE*** system.

---

[1] In the database domain, work on this topics has been initiated a long time ago: see, for example, (Batini and al. 1986) and (Sheth and Kashyap 1993.

ARTEMIS is a schema integration tool (Castano 2001). It first computes "affinities" in the range 0 to 1 between attributes, which is a match-like step. It then completes the schema integration by clustering attributes based on those affinities and then construction views based on the clusters. ARTEMIS is used as component of a heterogeneous database mediator, called MOMIS (Mediator envirOment for Multiple Information Sources).

Cupid is a hybrid matcher based on both element and structure level matching (Madhavan and al 2001). It is intended to be generic across data models and has been applied to XML and relational examples. It uses auxiliary information sources for synonyms, abbreviations, and acronyms.

6. VALIDATION (WRT USAGE)

A white paper by (Angele and Sure 2002) offers some thoughts on how to validate ontologies. Among others, they investigate the tool OntoGenerator to generate artificial ontologies that can then be used to validate tools. Some metrics like depths of nesting are shortly discussed but not further elaborated. The approach is further followed in the EON workshop series (Sure and Corcho 2003) which however focuses more on comparing ontology tools rather than on ontologies.

Many interesting papers have recently been published on the related topic of ontology evaluation. They propose different methods for quantitative and qualitative evaluation, the former being based on formal characteristics of ontology representation, the latter involving the user or the expert in rating the defined ontology.

An example of the first approach is the paper of (Gòmes-Pérez and. Suárez-Figueroa, 2004), where an in depth evaluation of parser and platforms using the most popular RDF(S), DAML+OIL and OWL formalisms for ontology representation has been conducted. The evaluation metric is based on circularity problems, partition errors and grammatical redundancy problems detection capabilities of the investigated tools, and has been applied to ten widely used ontology parsers and validators.

On the qualitative side, in order to classify the possible approaches, (Brewster et al., 2004) suggested to focus the evaluation on the principles used in ontology construction, on the effectiveness of ontology in the context of an application, or on the congruence (fit) between an ontology and a domain of knowledge.

In the same paper they propose a data driven method able to estimate the ontological fit, as a measure of vocabulary overlap between the concepts contained in a given ontology and the terms extracted from a corpus of texts related to the domain. Their ontology evaluation in view of a corpus is based on three steps: identification of keywords/terms (based on latent semantic analysis and a clustering algorithm), query expansion (by using a two-step hypernyms WordNet look up) and ontology mapping.

An example of evaluation made in the context of an application is given by (Porzel and Malaka, 2004). They define a schema able to test an ontology on a given task on three basic levels: the scope (or fit) of the vocabulary, the wellness (fit) of the taxonomy, i.e. the generalization or *isa* hierarchy and the adequacy of the non-taxonomic relations, i.e. the fit of the semantic relations. The method has been applied to the task of tagging the ontological relations that hold between ontologically marked-up entities, showing good results regarding the evaluation of the qualities of each individual level of the ontological model with respect to a gold standard.

Finally, in (Navigli et al., 2004), an original method for qualitative evaluation of linguistic ontologies is presented. The authors moved from the consideration that ontology construction, apart from the technical aspects of a knowledge representation task (i.e. choice of representation languages, consistency and correctness with respect to axioms, etc.), is a *consensus building* process, one that implies long and often harsh discussions among the specialists of a given domain. To facilitate this process, they define an algorithm for automatic generation of textual explanations (glosses) for the multi-word expression defined in an ontology. Such description, expressed in natural language, can be used by domain experts to compare their intuition of a domain with its description, as provided by the ontology concepts, thus facilitating a *qualitative* per-concept evaluation of the congruence between the ontology and the modelled domain.

7. EVOLUTION, VERSIONING

(Klein et al. 2002a) investigated the requirements for ontology management systems (OMS_ and found that versioning would be very important but that almost no system (except SHOE) supports it! More work exists on evolution (change) since this is a core task of any OMS. The distiguish types of changes like change of a natural language text, structural (logical) change, addition of definitions, deletion of definition etc.

(Noy and Klein 2003) argue that much can be learned from schema evolution but that ontology evolution also has some peculiarities: namely, different semantics, different usage paradigms. They see no difference between evolution and versioning since an ontology should contains its complete history, i.e. all its versions.

The paper (Klein et al 2002b) distinguishes **conceptual changes** (the way a domain is interpreted) from **explication changes** (the way how concepts are specified). For example, adding a new slot to an existing concept is an explication change when the interpretation of the concept is not affected. The difference can however not ne detected by analyzing the ontology or the ontology change since it is a categorization of the change before it is submitted to the ontology. The paper also presents the versioning system of OntoView, which is inspired by the CVS versioning system. Currently, OntoView supports versioning when importing external ontology sources into its repository, much like adding a new module to CVS. At import time, the user specifies some meta data about the imported ontology source. In the future, explicit changes to the ontology shall be supported as well. Thes changes can be:

- non-logical changes: for example, changes to the natural language description of a concept
- logical changes: changes in the definition of a concept that affects its formal semantics
- name change: changing the identifying label of a concept
- addition of a concept
- removal of a concept

Logical changes can be specified by a production rules that control which (property) changes are admissable. The rules are formulated on RDF triples and remind very much of integrity rules in (deductive) databases.

In (Klein and Noy 2003), changes to an ontology are seen as sequences of individual update operations like a log file of a database system. They discuss minimal transformations between two given ontology states, i.e. how to go from one state to the other with the smallest set of individual updates and how to construct complex update operators from sequences of individual updates (represented as minimal transformations). These update operations can themselves be organized as an ontology and offered to the user in a menu.

8. MULTI-LINGUAL, MULTI-CULTURAL, META-DATA

There is little research done on multi-ligual and multi-cultural aspects of ontology development. One of the fe exceptions is the MACS project (MACS 2004). MACS aims to provide multilingual subject access to library catalogues. MACS enables users to simultaneously search the catalogues of the project's partner libraries in the language of their choice (English, French, German). The partners are: the Swiss National Library (SNL), project leader, the Bibliothèque nationale de France (BnF), The British Library (BL) and Die Deutsche Bibliothek (DDB).

The IST project MKBEEM (Multilingual Knowledge Based European Electronic Marketplace) develops a mediation system which adapts the language and the trading conditions of an Internet sales point according to its international customership. MKBEEM has developed a multi-lingual ontology for an e-marketplace broker (Leger et al 2002).

Another example of a multi-lingual ontology is EUROWORDNET (Vossen Piek 1999). It is the multi-lingual version of WordNet covering about half a dozen European languages.

In the medical domain, a huge effort is being made every decade to translate the American SNOMED taxonomy (www.snomed.org). SNOMED covers more than 350.000 clinical terms along the dimensions clinical findings (diseases), procedure/therapy, observable entity, body structure, organisms, substances etc. Besides these terms, there are roughly 1 million synonyms and 1,5 million relationships between concepts. Translation projects exist for English to Spanish and English to German. Each translation effort takes several years and is carefully planned.

Multi-lingual product catalogs were investigated in (Jeusfeld 2004). The idea is to separate catalog data structures from product ontologies and classify elements in product catalogs to concepts in product ontologies. These concepts then have multi-lingual labels allowing multi-lingual accesses to the heterogeneous product catalogs. Besides supporting multiple languages, the approach also allows managing several parallel product ontologies - each specialized for a certain market. The approach is implemented by the ConceptBase system (www.conceptbase.cc). It has also been used for the ontology system developed by (vBuol 2000).

The paper by (Brase and Nejdl 2003) discusses ontologies and meta-data. The application domain is E-learning. It builds on the standard Dublin Core for document meta-data (providing fields for author, creation, subject, etc.). The link to ontologies is made via the 'subject' field, in their case linking to the ACM Computer Classification System and the SWEBOK ontology. The querying is done via Datalog engines ranging over regular meta-data but also the ontologies linked to the documents.

9. SOCIAL & CULTURAL CHALLENGES (E.G., DEDICATED GW)

The appears to be little work on this in the ontology domain. Some work from software engineering (Wangler and Persson 2002, Wangler et al 2003) draws on intentions of stakeholders in the software development process. They demand certain skills of people involved in a software development project like listening skills. Another interesting aspect is to make goals explicit. That is state of the art in software engineering literature but appears to be under-developed in the field of ontology engineering. For a counter example see (vBuol 2000). She lists goal definition as one of the first steps on ontology engineering projects.

10. CONCLUSIONS

The overall conclusion of this report is that there is a rather good selection of ontology management tools but there is little known on how to create 'good' ontologies in a multi-lingual and multi-cultural setting.

Concerning ontology managament tools, there is basically the choice between tools originating from the US and tools originating from Europe. A prominent example the first category is OntoLingua. It provides a the schema for representing concepts but leaves reasoning services basically to external tools such as Prolog. The second category is characterized by relying on a strong ontology language foundation, namely description logic, and provides integrated services for reasoning within ontologies, e.g. to find out whether one concept is more special than another concept.

Given this state of the art, we recommend investigating the following questions:

1. What are the ingredients of an ontology language for enterprise application integration? Specifically, shall we represent processes, organizational units, business goals, business entities?
2. How to separate ontology engineering from enterprise application modeling (WP 5)? We must avoid to duplicate work and define the precise border and link between an enterprise model and an enterprise ontology.
3. How to cope with multi-linguality and multiple cultures? One goal of an ontology is to increase the level of understanding between different people, potentially with different cultural background.
4. Which role can ontologies play in an evolving collection of inter-related enterprise applications? Can ontologies an integral part of enterprise applications in the sense of self-aware software systems?

11. BIBLIOGRAPHY

Ahrens, Kathleen , Fong Chung, Siaw and Huang, Chu-Ren (2003): *Conceptual metaphors: Ontology-based representation and corpora driven mapping principles*. In Mark Lee John Barnden, Sheila Glasbey and Alan Wallington, editors, Proceedings of the ACL 2003 Workshop on the Lexicon and Figurative Language, pages 35-41, 2003.

Batini, C., Lenzerini, M. and Navathe, S.B. : A Comparative Analysis of Methodologies for Databases Schema Integration. ACM Cmputing Surveys, Vol.18(4), December 1986, pp.323-364.

S. Bechhofer, I. Horrocks, C. Goble, R. Stevens (2002): "OilEd: A Reason-able Ontology Editor for the Semantic Web. Proceedings Description Logics 2001, http://ceur-ws.org/Vol-49.

S. Bergamaschi, S. Castano, M. Vincini (1999): Semantic Integration of Semistructured and Structured Data Sources SIGMOD Record 28(1).

S. Bergamaschi, S. Castano, D. Beneventano, M. Vincini (2001): Semantic Integration of Heterogeneous Information Sources. Data & Knowledge Engineering 36(1), pp. 215-249.

A. Bernaras, I. Laresgoiti and J. Corera (1996): Building and Reusing Ontologies for Electrical Network Applications. Proceedings European Conference on Artificial Intelligence (ECAI'96).

Jan Brase, Wolfgang Nejdl (2003): Ontologies and Metadata for eLearning. In "Handbook on Ontologies", Springer-Verlag 2003, ISBN 1-58603-345-X.

D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, R. Rosati (1998): Description Logic Framework for Information Integration. In Proceedings KR-1998, Morgan Kaufman, pp. 2-13.

D. Calvanese, G. De Giacomo, M. Lenzerini (2001): A Framework for Ontology Integration, Proceedings of 2001 Int. Semantic Web Working Symposium.

Clifton, Christopher W. and Li, Wen-Syan (2000): Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data and Knowledge Engineering, 33(1), April 2000.

V. De Antonellis and S. Castano (2001): Global viewing of heterogeneous data sources. IEEE Transcation on Knowledge and Data Engineering, 13(2):277-297, April 2001.

S. Decker, M. Erdmann, D. Fensel and Rudi Studer (1999): Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In "Semantic Issues in Multimedia Systems", Proceedings of DS-8, Kluwer Academic Publisher, 1999, pp. 351-369.

A. Doan, P. Domingos and A.Y. Halevy (2001): Reconciling schemas of disparate data sources: A machine-learning approach. In SIGMOD Conference, 2001.

J. Euzenat (1995): Building consensual knowledge bases: context and architecture. In "Building and sharing large knowledge bases", IOP Press, pp. 143-155.

J. Euzenat (1996): Corporative memory through cooperative creation of knowledge bases and hyper-documents. Proceedings 10th KAW, Banff (Canada).

M. Fernandez, A. Gomez-Perez and N. Juristo (1997): METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Symposium on Ontological Engineering of AAAI, Stanford, California, 1997, pp. 33-40.

A. Gangemi, D.M. Pisanelli, G. Steve (1999): An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. Data & Knowledge Engineering 31, pp. 183-220.

G.L. Geerts, W.E. McCarthy (2000): The Ontological Foundation of REA Enterprise Information Systems. Working paper, Michigan State University (August 2000).

A. Gomez-Perez and M.D. Rojas (1999): Ontological Reengineering and Reuse. European Knowledge Acquisition Workshop (EKAW) 1999.

I. Horrocks (1998): Using an Expressive Description Logic: FaCT or Fiction? Proceedings KR-98, pp. 636-647.

I. Horrocks, U. Sattler, S. Tobies (1999): Practical Reasoning for Expressive Description Logics. In Proceedings LPAR 1999, LNAI 1705, Springer-Verlag, 1999, pp. 161-180.

M. Jarke et al (1999): Fundamentals of Data Warehousing. Springer-Verlag, 1999.

M. Jarke et al. (2000): Concept Based Design of Data Warehouses: The DWQ Demonstrators. In Proceedings SIGMOD 2000.

M.A. Jeusfeld: Integrating Product Catalogs via Multi-language Ontologies. Proceedings EAI 2004, http:/ceur-ws.org/Vol-93.

J. Kietz, A. Maedche, R. Volz (2000): A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. Workshop on Ontologies and Texts, http://ceur-ws.org/Vol-51/

C. Masolo et al (2003). The WonderWeb Library of Foundational Ontologies. WonderWeb Deliberable D17. http://www.loa-cnr.it/Papers/DOLCE2.1-FOL.doc

W.E. McCarthy (1979): An Entity-Relationship View Of Accounting Models. The Accounting Review (October): 667-86.

W.E. McCarthy (1982): The REA Accounting Model: A Generalized Framework for Accounting Systems in A Shared Data Environment. The Accounting Review (July), pp. 554-578.

Mitra P.; Wiederhold G.; Jannink J (1999): Semi-automatic integration of knowledge sources. Proceedings of Fusion '99, July 1999, (9), July 1999.

Mitra Prasenjit, Wiederhold Gio, and Kersten, Martin (2000): A graph-oriented model for articulation of ontology interdependencies. Lecture Notes in Computer Science, 1777:86+, 2000.

Madhavan, Jayant, Bernstein, Philip A. and Rahm, Erhard (2001): Generic schema matching with cupid. In The VLDB Journal, pages 49-58, 2001.

Miller, George A. and al (2003): Wordnet 2.0, 2003. http://www.cogsci.princeton.edu/ wn/.

Milo, Tova and Zohar, Sagit (1998): Using schema matching to simplify heterogeneous data translation. In Proceedings of the 24rd International Conference on Very Large Data Bases, pages 122-133. Morgan Kaufmann Publishers Inc., 1998.

Spaccapietra, S. and Parent, C. (1998): Issues and approaches of database integration. CACM, 41(5):166-178, 1998.

Piek. Vossen, Eurowordnet (1999): A multilingual database with lexical semantic networks. EuroWordNet General Document, 3, July 19 1999.

Rahm, Erhard and Bernstein, Philip A. (2001): A survey of approaches to automatic schema matching. VLDB Journal: Very Large Data Bases, 10(4):334-350, September 2001.

Sheth, A.P. and Kashyap, V (1993): So far (schematically) Yet so near (semantically). In Interoperable Database Systems, Elsevier Science, North Holland, D.K. Hsiao, E-J. Neuhold and R. Sack-Davis Eds, 1993.

Palopol, L., Sacc, D. and D. Ursino (1998): Semi-automatic, semantic discovery of properties from database schemes. In Proceedings of the 1998 International Symposium on Database Engineering & Applications, page 244. IEEE Computer Society, 1998.

Von Buol, B. (2000): Qualitätsgestützte, kooperative Terminologiearbeit. Shaker-Verlag, Aachen, Germany, 2000, ISBN 3-8265-7096-0

Dehli,, Einar, Smith-Meyer Henrik and Lillehagen Frank, July 2003, Metis LEARN, Enterprise Architecture paper presented at CE 2003, Madeira, Portugal; see http://www.ce2003.org/.

Camarinho-Matos, Luis and all, June 2004; Virtual Organization systems and Practices, Book to be published, see chapter 1.4 Ontology and Knowledge Management.

Fox, Mark and Grueninger M., September 1997, The TOVE Ontologies, presentations at the Globeman 21 project workshop, Japan, see also http://www.eil.utoronto.ca/enterprise-modelling/tove/.

Gavrilova, Tatiana and Gorovoi, Vladimir, October 2003; "Ontological Engineering for Knowledge Portal Design", Slides presented at the PROVE 2003 conference in Lugano, Italy.

A. Gómes-Pérez, M.C. Suárez-Figueroa (2003): Results of Taxonomic Evaluation of RDF(S) and DAML+OIL ontologies using RDF(S) and DAML+OIL Validation Tools and Ontology Platforms import services. In Proceedings EON2003, http://ceur-ws.org/Vol-87/EON2003_Figueroa.pdf

A. Gómes-Pérez, M. Fernández-López, O. Corcho: Ontollogical Engineering. Springer-Verlag, 2004.

A. Gómes-Pérez, M. Fernández-López, O. Corcho (2002): Technical Roadmap D.1.1.2. OntoWeb Consortium.

D.B. Lenat and R.V. Guha (1990): Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Addison Wesley.

Lillehagen, Frank, Computas, 1997 and 2002, The Foundations of the AKM technology, last presented at the CE 2004 conference, Madeira July, 2003, see also www.akmii.net.

Lillehagen, Frank and others, July 2003, IDEAS report D 2.4: Visions of 2010, for IDEAS project, see also www.ideas-roadmap.net.

Lillehagen, Frank:, February 2004; Enterprise Modeling Tutorial, presented at the Athena project kick-off meeting; Valencia, Spain. Go to: www.athena-ip.com.

Lillehagen, Frank, Solheim Helge and Tinella, Stefano, Enterprise Knowledge Spaces, providing scenes for new approaches to SE and KM, Paper to be presented.

McElroy, Mark and Firestone Joe, The New Knowledge Management, Book published by Butterworth Heineman, ISBN: 0-7506-7608-6.

Noy, Natalia F. and McGuinness, Deborah L., Summer 2003, Ontology Development 101: A Guide to Creating Your First Ontology http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

Petit, Michael et al., April 2003, State-of-the-art in Enterprise Modeling. UEML project report and presentation; see www.UEML.org.

Rohlenko, Oleg, Spring 2002, Ontology Seminar, Given at the Interop NoE project Kick-off; Bordeaux, France.

Ross, Hugh, " SADT – Structural Analyses and Development Tool", presentation at IFIP seminar Røros, Norway, 1969.

York Sure, Oscar Corcho (2003): Evaluation of ontology-based tools. Proceedinds EON2003, http://ceur-ws.org/Vol-87

W3C (204): OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/

Zhan Cui, Paul O'Brien (2002): Domain Ontology Management Environment, Proceedings of Business to Business E-Commerce, HICSS33, January 2000, http://more.btexact.com/projects/ibsr/papers/b2bCommerce.doc

Zhan Cui, Dean Jones, Paul O'Brien (2001): Issues in Ontology-based Information Integration, Proc. IJCAI01 Workshop on E-Business and the Intelligent Web, Seattle, USA, August 5, 2001.

M. Denny (2002): Ontology Building: A Survey of Editing Tools, http://www.xml.com/pub/a/2002/11/06/ontologies.html, 2002.

Michel Klein, Ying Ding, Dieter Fensel, Borys Omelayenko (2002): Ontology management - Storing, aligning and maintaining ontologies, http://homepage.uibk.ac.at/~c703205/download/Chap5-final.pdf, 2002.

L. Stojanovic, B. Motik (2002): Ontology Evolution with Ontology Editors, http://km.aifb.uni-karlsruhe.de/eon2002/EON2002_Stojanovic.pdf, 2002.

York Sure, Oscar Corcho (2003): Evaluation of Ontology-based Tools, Proceedings of the 3rd Workshop ADIS-2002, El Escorial, Madrid, Spain, November 18, 2002, http://CEUR-WS.org/Vol-87/

Raphael Troncy, Antoine Isaac, and Veronique Malaise (2003): Using XSLT for Interoperability: DOE and The Travelling Domain Experiment, Proceedings EON 2003, http://CEUR-WS.org/Vol-87/EON2003_Troncy.pdf

Alan L Rector, Chris Wroe et al. (2001): Untangling Taxonomies and Relationships:Personal and Practical Problems in Loosely coupled development of large ontologies, http://www.cs.man.ac.uk/~rector/papers/rector-k-cap-untangling-taxonomies-web.pdf.

N. F. Noy; M. A. Musen; J. L.V. Mejino; Jr., C. Rosse (2002): Pushing the Envelope: Challenges in a Frame-Based Representation of Human Anatomy, http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2002-0925.html.

Ontoweb Project Partners (2002): A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies, Ontoweb Project (del 1.4), http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D1.4-v1.0.pdf.

M. Missikoff, R. Navigli, P. Velardi (2002): Integrated approach to web ontology learning and engineering, IEEE Computer, 2002, http://www.dsi.uniroma1.it/~velardi/IEEE_C.pdf

M. Missikoff, F. Taglino (2003): SymOntoX: A Web-Ontology Tool for eBusiness Domains. Proceedings Fourth International Conference on Web Information Systems Engineering (WISE'03) December 10 - 12, 2003, Roma, Italy, pp. 343 ff.

B. Bachimont, A. Isaac A. and R. Troncy R. (2002): Semantic Commitment for Designing Ontologies: A Tool Proposal, 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2002).

Eva Söderström (2002): Standardising the Business Vocabulary of Standards, The ACM Symposium on Applied Computing (SAC 2002).

Michel Klein, Ying Ding, Dieter Fensel, Borys Omelayenko (2002a): Ontology management - Storing, aligning and maintaining ontologies, OntoKnowledge Project Report, http://homepage.uibk.ac.at/~c703205/download/Chap5-final.pdf

Michel Klein, Atanas Kiryakov, Damyan Ognyanov, and Dieter Fensel (2002b): Ontology Versioning and Change Detection on the Web. In 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), Sigüenza, Spain, October 1-4, 2002.

Michel Klein and Natasha F. Noy (2003): A component-based framework for ontology evolution. In Proceedings of the Workshop on Ontologies and Distributed Systems, IJCAI '03, Acapulco, Mexico, 2003.

C. Parent and S. Spaccapietra (2000): Database integration: The key to data interoperability, in Advances in Object–Oriented Data Modeling, M. Papazoglou, S. Spaccapietra, Z. Tari (Eds.), MIT Press.

G. Stumme and A.Maedche (2003): Ontology Merging for Federated Ontologies on the Semantic Web, Proceedings IJCAI 2003, www.tzi.de/buster/IJCAIwp/Finals/stumme.pdf.

Y. Kalfoglou and M. Schorlemmer (2003): Ontology mapping: the state of the art, The Knowledge Engineering Review, Vol. 18:1, 1–31.

Hoy (2001): Comparing sets of semantic relations in ontologies, Kluwer, 2001.

J. Angele and Y. Sure (2002): Evaluation of Ontology-based Tools, Whitepaper, www.ontoweb.org.

N.F. Noy and M. Klein (2002): Ontology Evolution: Not the Same as Schema Evolution, submitted to Knowledge and Information Systems; also available as Smi-2002-0926, University of Stanford, Stanford medical Informatics.

N.F. Noy M.A. Musen (2000): PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000).

B. Wangler and A. Persson (2002): Capturing Collective Intentionality in Software Development, n Fujita, H. and Johannesson, P. (eds.), New Trends in Software Methodologies, Tools and Techniques, IOS Press, Amsterdam, Netherlands, pp 262-270.

B. Wangler, A. Persson, P. Johannesson, L. Ekenberg (2003): Bridging High-Level Enterprise Models to Implementation-Oriented Models, The Lyee Workshop, Stockholm, 2003.

Macs (2004): Multilingual access to subjects. Online http://infolab.uvt.nl/prj/macs/, Sep. 2004.

Alfonseca E. and Manandhar S. (2002): Improving an Ontology Refinement Method with Hyponymy Patterns. Language Resources and Evaluation (LREC-2002), Las Palmas, Spain, May 2002.

Agirre E., Ansa O., Hovy E. and Martinez D. (2000): Enriching very large ontologies using the WWW, in ECAI 2000 Ontology Learning Workshop http://ol2000.aifb.uni-karlsruhe.de.

M. Berland and E. Charniak. (1999): Finding Parts in Very Large Corpora. In Proceedings of the the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), pp 57-64

Cimiano P., Pivk A., Schmidt-Thieme L. and Staab S. : Learning taxonomic relations from heterogeneous evidence, in ECAI-2004 Workshop on Ontology Learning and Population (http://olp.dfki.de/ecai04/cfp.htm)

Gildea D. and D. Jurafsky, (2001): Automatic Labeling of Semantic Roles, Computational Linguistics, vol. 99 n.9.

N. Guarino, C. Welty (2002). Evaluating ontological decisions with OntoClean. CACM 45(2):61-65.

Jeusfeld, M.A. (2004): Integrating Product Catalogs via Multi-language Ontologies. Proceedings Workshop Enterprise Application Integration, Oldenburg, Germany, 2004.

Léger, A., Michel, G., Barret P. et al (2000): Ontology Domain Modeling Support for Multi-lingual services in E-Commerce, Proceedings Workshop on Applications of Ontologies and Problem solving Methods at ECAI'00, Berlin, Germany, August 20-25, 2000.

Maedche A. and Staab S. (2000) : Semi-automatic Engineering of Ontologies from Text Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE'2000).

Maedche A. and Staab S. (2001): Ontology learning for the Semantic Web, IEEE Intelligent Systems, 16(2), 2001

R. Navigli ,Velardi P.,. (2004).: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites". Computational Linguistics. vol. 50 (2), 2004.

M. Missikoff, R. Navigli, Velardi P. (2002): An integrated approach for Web ontology learning and engineering. IEEE-Computer., Nov. 2002.

Vossen P. (2001) : Extending, Trimming and Fusing WordNet for technical Documents*, NAACL 2001 workshop on WordNet and Other Lexical Resources*, Pittsbourgh, July 2001.

C. Brewster, H. Alani, S. Dasmahaptra, Y. Wilks (2004): Data Driven Ontology Evaluation, Proceedings of the 4th Language Resources and Evaluation Conference (LREC2004), Lisbon, Portugal, May 2004.

A. Gòmes-Pérez, M.C. Suárez-Figueroa (2004): Ontology evaluation Functionalities of RDF(S)m DAML+OIL, and OWL Parsers and Ontology Platforms, Proceedings of the 4th Language Resources and Evaluation Conference (LREC2004), Lisbon, Portugal, May 2004.

R. Navigli, P. Velardi, A. Cucchiarelli, F. Neri (2004): Quantitative and Qualitative Evaluation of the OntoLearn Ontology Learning System, Proceedings of the 20th International Conference on Computational Linguistics (COLING2004), Geneva, CH, 2004.

R. Porzel, R.Malaka (2004): A Task-Based Approach for Ontology Evaluation, Proceedings of the ECAI 2004 Workshop on Ontology Learning and Population, 1-6, Valencia, Spain, 2004.

B. Swartout, R. Patil, K. Knight, T. Russ (1997): Toward Distributed Use of Large-Scale Ontologies, Symposium on Ontological Engineering of AAAI, Stanford, California, 1997. http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff_96_final_2.html

M. Denny (2002): Ontology Building: A Survey of Editing Tools, http://www.xml.com/pub/a/2002/11/06/ontologies.html

M. Uschold and M. Gruninger (1996): Ontologies: Principles Methods and Applications. Knowledge Engineering Review 2, pp. 93-155.

M. Uschold and M. King (1995): Towards a methodology for building ontologies. Proceedings IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada.

H. Wache, T. Vogele, U. Visser et al. (2001): Ontology-Based Integration of Information A Survey of Existing approaches. IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, pp. 108-117.

Harmoconsys Team. Harmonise Deliverable 3D.1 - Solution Transfer and Adaptation of a Co-operative System for Consensus Process HarmoConSys: A Consensus System for Collaborative Ontology Building, Laboratory for Enterprise Knowledge and Systems (LEKS) IASI-CNR, Italy.

11.1 Ontology Links

Apollo (2003): http://apollo.open.ac.uk

Chimera (2003): http://www.ksl.stanford.edu/software/chimaera

CIA (2001): CIA World Factbook http://www.cia.gov/cia/publications/factbook/geos/fr.html

CycL (2004]) Ontological Engineer's Handbook, http://www.cyc.com/doc/handbook/oe/06-el-hl-and-the-canonicalizer.html

E-Class (2004): http://www.eclass.de/

Enterprise Ontology (2003): Enterprise Ontology Project, http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html

JOE (2003): http://www.engr.sc.edu/research/CIT/demos/java/joe

MIT Process Handbook (2003): http://ccs.mit.edu/ph/

OntoEdit (2003): http://ontoserver.aifb.uni-karlsruhe.de/ontoedit

Protege (2003): http://protege.stanford.edu

Snomed (2004). http://www.snomed.org

Sowa J.F. (2001): Sowa Ontology, http://www.jfsowa.com/ontology/

SUMO (2004): Suggested Upper Merged Ontology – http://suo.ieee.org/

SUO(2003): The ieee standard upper ontology web site, 2003. http://suo.ieee.org.

Tambis (2001): TAMBIS, http://imgproj.cs.man.ac.uk/tambis/

TOVE (1999): TOVE Manual, http://www.eil.utoronto.ca/tove/ontoTOC.html

UMLS (2001): Unified Medical Language System, http://www.nlm.nih.gov/research/umls/

11.2 Ontology learning workshops

(ECAI, 2000) 1st ECAI-2000 Workshop on *Ontology Learning* (http://ol2000.aifb.uni-karlsruhe.de)

(IJCAI, 2001) 2nd IJCAI-2001 Workshop on *Ontology Learning* (http://ol2001.aifb.uni-karlsruhe.de)

(ECAI, 2002) ECAI-2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (http://www-sop.inria.fr/acacia/WORKSHOPS/ECAI2002-OLT/)

(ECAI, 2004) ECAI-2004 Workshop on Ontology Learning and Population (http://olp.dfki.de/ecai04/cfp.htm)

12. APPENNDIX 1: ONTOLOGY AND KNOWLEDGE MANAGEMENT

by F. Lillehagen, COMPUTAS

This chapter gives an overview of ontology and knowledge management approaches as applied to Virtual Organizations. It focuses on research efforts from the mid-nineties, and briefly describes approaches, implementations, and innovative achievements.

The dominant approach is for VO projects to develop application systems and tools and to integrate them through common databases and ICT architectures. Ontology tools and ontology structures, knowledge management systems and knowledge elements are developed and managed disjoint from other systems.

However, recent model-based approaches, creating ontology and knowledge services as reusable tasks, are indicating a paradigm-shift in the approach.

12.1 Background Work

Research on ontology and knowledge engineering and management as technologies contributing towards developing quality virtual organizations and extended enterprises goes back to the provision of the first web-servers in the early nineties.

Both areas and technologies have been researched as potential contributors to new industrial approaches to systems development and engineering, process and product design, organizational development and solutions delivery. The recent development of many new innovative technologies is adding to the visions of an entirely new approach to industrial computing, solutions design and collaboration.

The technologies that give rise to the coming paradigm-shift involving Model Designed Solutions and Model Generated Workplaces are:

Web technologies with services, repositories and standards

Agent technologies with plug-and-play ICT architectures

Portal-based Ontology and Knowledge Engineering Environments

The Active Knowledge Modeling technology and visual knowledge spaces

The first generation of Knowledge Engineering (KE) tools targeted the reconstruction of semantic space of human expertise, and repertory grid-centered tools like the Expertise Transfer System (ETS) [Boose, 1986], AQUINAS [Boose, Shema, Bradshaw, 1989] and others. The second generation KE tools - visual knowledge engineering - provides ideas of CASE technology to AI [Aussenac-Gilles, Natta, 1993;]. These early tools did not fit into any sustainable approach or architecture so they were bound to fail, but they contributed to our understanding.

In the mid-nineties industry was flooded with Knowledge Management Systems (KMS). They were, as many other systems, sold with the promise to support creative work, and were dramatically oversold. The knowledge entities or nuggets as some chose to call the knowledge elements were stored as entities in traditional databases. Storage and retrieval of content depended on carefully defined identity schemes and categorization structures, and the context for reuse was not adequately captured. These KMS systems were nothing other than advanced information and file management systems, but they provided novel methods for navigation of contents.

The new generation of KE tools enables knowledge capture, structuring and reuse, and helps cut down the revise and review cycle-times. They promise to refine, structure and test human knowledge and expertise in the form of ontology [PROTEGE, WebOnto, OntoEdit, 2001-2003], but they are standalone tools.

Enterprise Modeling has also provided approaches capturing, expressing and sharing enterprise knowledge. Some projects have created prototypes that demonstrate the possibility of achieving model designed solutions, and models for execution. Knowledge-model driven approaches are becoming available from leading tool vendors [Computas AS, Ptech Inc., and MEGA, 2000-2003], and some of them also provide methodology and infrastructure to support reuse.

12.1.1 The role of Ontology and Knowledge Engineering

The role of ontology and knowledge management in VO development and operations is to express and share knowledge that can be extended and adapted to support model-designed solutions and operations. Capturing situated knowledge, achieving interoperable solutions, and providing interactive views are important for increased stakeholder participation. The ICT and KE architecture, transforming IT component and semantics, must be dynamically extended. Capabilities needed are:

- Semantics mediation and transformation
- Capturing, expressing and storing knowledge
- Separating the architectural layer and their respective constructs

Ontology and knowledge modeling must provide integrated services to enable model extension and adaptation. Models are reused from enterprise knowledge repositories. Abstract enterprise models are pure ontology structures, but there are many types and kinds of ontology structures that must be aligned, re-designed and combined.

12.2 Approaches and Implementations

The dominant approach for VO projects so far has been to develop application systems and tools and to integrate them towards common databases and ICT architectures. Ontology tools and ontology structures, knowledge management systems and knowledge elements are developed and managed disjoint from other application systems and tools. Most approaches are based on fairly traditional client-server architectures, but some novel web-based software architectures have been prototyped. Moving to the web and exploiting web-standards some projects have developed more comprehensive architectures. These projects have been able to demonstrate plug-and play capabilities of software components [e-Colleeg, 2001- 2003]. One project [EXTERNAL, 2000-2002] has experimented with a layered architecture, and a model-driven approach to systems development and engineering. The layered architecture is no doubt a break-through, separating business behavior and execution from enterprise knowledge processing and from ICT –architectures. The layered architecture is further described in section 1.4.

12.2.1 State-of-the-art

There is a wide spectrum of approaches, methodologies and achieved prototypes that have been researched. However, the practical solutions are based on standardized IT architectures and interfaces, rigidly integrating existing applications, databases and tools. The most frequently met tools cover process modeling and simulation, enterprise knowledge modeling and management (including information management), and more recently ontology tools manifested as topic-maps and concept maps.

Knowledge processing and management is dependent on the encoder's proximity to actions and events, on the medium, method and language of encoding, and on concurrently capturing knowledge, designing and performing work. Designing and developing working environments, workplaces, rules and habits, and generating operational solutions are interactive tasks performed by users while executing work. Knowledge engineering should focus on company core competencies and activities, that is strategy, business, human capital, architectures, platforms, operational solution, and infrastructure and services.

12.2.1.1 Use of ontology to design VOs

Ontology defines the basic terminology and relations comprising the structured vocabulary of a topic area. Ontology has to do with naming conventions for industrial nomenclature of all types of core knowledge structures of any enterprise. This is illustrated in figure 1.

"Ontology is an explicit specification of a conceptualization or a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base"(Gruber, 1993).

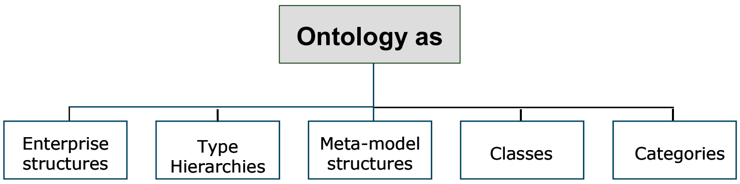


Figure 1 An ontology is a structure relating concepts and artefacts by their identity, enhancing the total meaning.

Formattato: Inglese (Regno Unito)

Formattato: Inglese (Regno Unito)

Codice campo modificato

Eliminato: 1

Formattato: Inglese (Regno Unito)

Ontology tools are becoming available as web services, but the structures to be transformed are not easily imported and exported out of the systems and databases out there.

12.2.2 Model-driven solutions

OMG and many other institutions are pushing model-driven approaches to systems architecture and development. The models referred to are UML diagramming models. When executing these models there is no feedback or services to update the models based on execution and performance experiences. This cyclic behavior of model-generated task solutions, and task execution-driven model development is enabled by the AKM technology [Fox, 1997]. By this approach a closed loop is created between visual modeling, execution, value-creation and experience gathering, thus implementing learning by doing.

12.2.2.1 Knowledge Spaces and AKM technology

Active Knowledge Modeling (AKM) technology [Lillehagen, 2003] is an innovative way of expressing, representing and continuously developing an interactive layered enterprise architecture, separating business operations and models from enterprise knowledge architecture and constructs, and from ICT architecture and components.

The AKM technology is built on the paradigm-shifting concept that enterprises and working organizations are cascaded knowledge spaces implemented as pro-active visual scenes.

12.3 Ontology tools

Ontology tools and ontology structures created are not currently part of the operational enterprise systems, except for ontology structures embedded in entity relationship diagrams and modeling languages. There is no support for interactive development and adaptation of ontology structures. Enterprise structures, flows and working rules must be modeled and embedded in active models driving operational architectures and solutions, integrating all enterprise knowledge dimension.

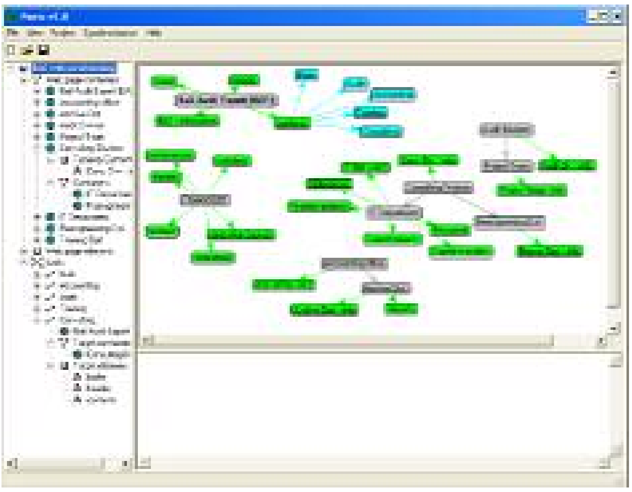


Figure 2 the Porto BAE portal ontology:

Some of the most well-known tools are: OntoEdit (www.ontoprise.de),

Apollo (www.apollo.open.ac.uk), WebOnto (kmi.open.ac.uk),

CAKE/VITA (www.csa.ru/ailab/) and PROTEGE (www.protege.stanford.edu).

2.3 KM systems and KE tools

Knowledge engineering and management cannot be achieved by introducing KM systems and ontology tools alone. Knowledge engineers, workers and systems analysts, designers of information systems in the different subject domains, must be directly engaged in expressing, embedding and executing their knowledge.

12.4 Scientific Foundations

Ontology and knowledge structures and processes are created, used and supported by most cognitive sciences. It is therefore an integral part of all our sciences and solutions. So expressing knowledge and building the enterprise and VO knowledge architectures are major challenges faced by IT providers and users when attempting to converge and integrate dispersed enterprise knowledge, and when exploiting the new digital multi-media. The ontology of the physical world or what exists in other media encoded by use of other languages does not necessarily readily apply.

12.4.1 Pedagogic and Epistemological Views

Ontology is not only a philosophical (epistemological, about knowing and knowledge evolution) discipline studying the being. It is about knowledge artifact naming, identity, identification, coding schemes, categorization, classification, and thesauruses, taxonomies and glossary. It is about the entire enterprise logical description, the meta- and operational views of the enterprise. The challenge is manifold when we integrate across enterprises to form VOs, and when we start expressing and representing the meta-knowledge of the enterprise practisioners.

Figure 3 defines four zones of knowledge perception and representation, each delimited by a perceptive horizon or border, defining knowledge existence, form and value as dependent on proximity to action and method of knowledge representation and diffusion. The innermost zone exists only in collaborative scenes.

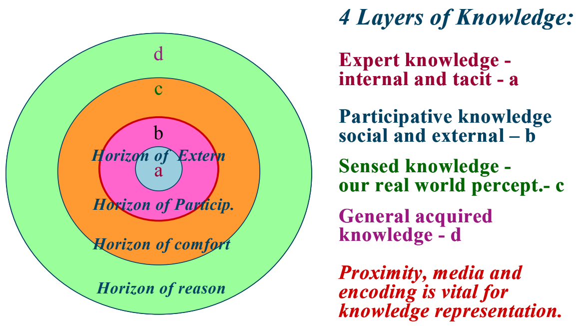
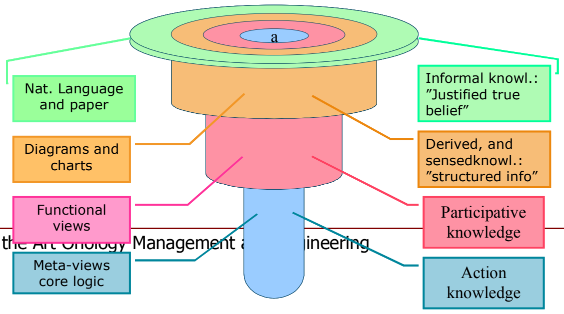


Figure 3 Existence of knowledge as defined by pedagogy and epistemology.

12.4.1.1 Knowledge Spaces

Visual organization of subjective space [Fox, 1997], the perception and evaluation of its coordinates, colors and shapes are different in the right and left halves of the human brain. The right provides isomorphic reflection of objects and scenes, while the left introduces an object into generalized classes of phenomena and episodes, and provides for logical context and operations. Therefore visual techniques and graphical approaches, activating right-half functions, work as versatile cognitive tools for more transparent and effective data/knowledge base design procedures.

The AKM technology and way of thinking [Lillehagen, 2003] opens up for model designed



approaches, methodologies, intelligent infrastructures and dynamically generated solutions. This implies that in all enterprise activity there are four major knowledge dimensions involved: Approach, Methodology, Infrastructure and Solution. (AMIS).

Figure 4 The value of knowledge according to existence and form.

These four-dimensional spaces are the source of the four logical flows of process as discovered by Hugh Ross as far back as 1969 [Dehli et al, 2003].

Enterprises have many knowledge spaces; the innovative, the operational, the strategic and the social. This defines the foundation of the AKM technology. [Lillehagen, 1991 – 2004]. The core knowledge of these knowledge spaces is the knowledge within the two inner zones of figure 4. Action knowledge typically has a set of intrinsic properties; reflective views, recursive processes, repetitive tasks and replicable meta-views, that do not exist in the outer zones. Expressing this knowledge is still a challenge for Enterprise Modeling languages and tools.

12.5 Enterprise Architectures

Enterprise model-designed architectures supporting execution will play an increasingly important role in future VOs. They will be a source of precise knowledge for managers and users as well as systems engineers and providers as regards all aspects of user model-designed solutions, support and maintenance.

In figure 5 there are three layers that are glued together by an intelligent infrastructure visualized as barrels and the customizable EKA box.. The architecture of the infrastructure is, contrary to the remaining architectures, independent of industrial sectors, applications and user solutions, so it is generic. The two barrels are services that are stored in the process repository as repeatable tasks. These services can be re-activated and adapted to any given platform and solution.

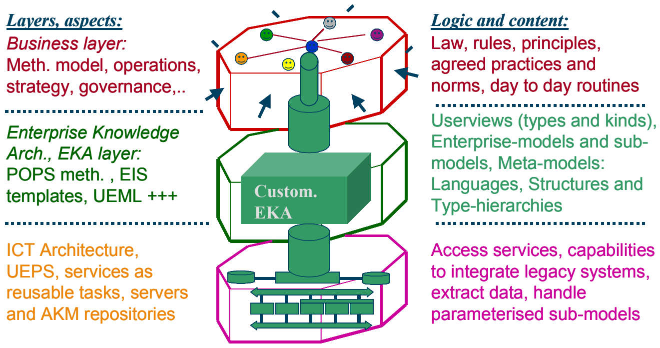


Figure 5 Layers of Enterprise Architecture – integrated by an Intelligent Infrastructure.

12.5.1 The Business Model Architecture (BMA)

The concepts at the core of the Business Model Architecture concern the integrated representation of law, business rules, project norms, project and working rules and habits, and new rules of collaborative practices. Support to define and assess the adherence to these rules and regulations must be available in an e-Business network. Law, contracted values, rules and obligations are controls on business processes. Business rules are typically agreements on value and risk sharing, division of tasks and responsibilities, and agreements on liabilities and life-cycle support. These mutual expectations and controls must be expressed in a separate business model and correlated with program and project performance indicators. Being able to predict and track performance, and assessing the degree of adherence to contract and key performance indicators, are important properties for successful collaboration. These capabilities all warrant a separate set of models where rules and regulations of working together apart must be continuously refined and managed, and where the rules of collaboration are continuously changed to fit local practices and habits.

12.5.2 The Enterprise Knowledge Architecture (EKA)

The Enterprise Knowledge Architecture is a set of inter-dependent knowledge dimensions that allow us to de-couple, engineer and manage enterprise knowledge constructs, methodologies and operational solutions. The six major enterprise knowledge dimensions are composed of these structures and constructs:

1. User enterprise views (types and kinds), the kinds of views are comparable to many Extended Enterprises frameworks (Zachman, Cimosa, and GERAM), but the types are not described and explained by these frameworks,
2. Enterprise models and sub-models, and structures of integrated solution models, supporting distributed working,
3. Partial meta-model of many types and supporting many kinds of services,
4. Language, core visual constructs form the basis for modeling languages,
5. Structures of meta-model objects and constructs, and finally,

12.6 Type-hierarchies representing standardized industrial knowledge.

These enterprise knowledge structures are vital for the formation, integration and operation of intelligent enterprises and smart virtual organizations, and must be visually editable and manageable in order to harvest the full benefits of models of knowledge spaces as visual scenes. The EKA is the most important and common knowledge asset of any enterprise, integrating mental models and augmenting the minds of all employees, for which the EKA is an active visual support system. The EKA represents the nervous centre and the communication central of the enterprise. It is the main source of manageable Enterprise Intellectual Capital.

12.6.1 The ICT Architecture

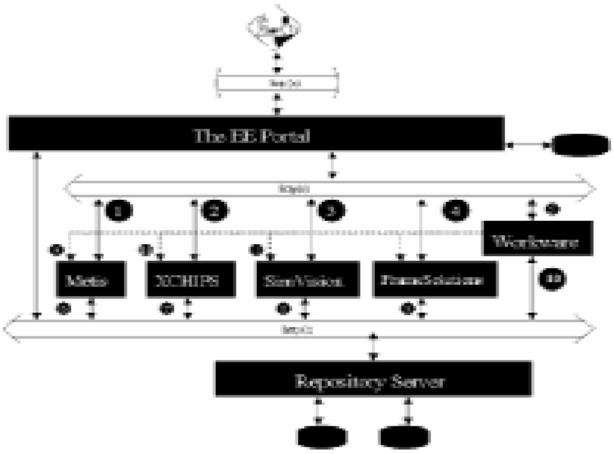


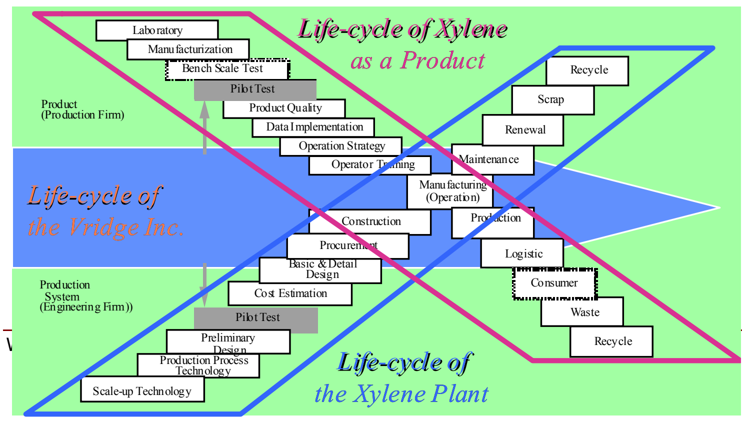
Figure 6 The ICT Architecture layer has four tiers

Figure 6 illustrates the ICT architecture as implemented in the EXTERNAL project. It shows that the user meets the infrastructure through a portal-based project engineering and management environment. This provides a set of web services to support modelling, model designed solutions engineering, work execution, value-metrics and experience gathering. The project environment acts as an integrator and as a platform to plug in and perform all kinds of services: software, tasks and applications. Among the services provided are services to build knowledge models (Metis), services to cooperate and collaborate (XCHIPS), services to do project simulation (SimVision), services to do work management (Workware) and services to perform work, enactment and execution (FrameSolutions). The portal-based technology used was prototyped in the project. As a front end, it has the main objective of generating a dynamic and personalized user interfaces, bridging the gaps among the available systems, tools and functions and the users.

Combining the Intelligent Infrastructure with techniques enabling dynamic generation of user interfaces, we are able to describe, implement, and deploy software platforms capable of generating operational solutions, driven and managed by knowledge models.

12.7 Supporting Design and Engineering

The AMIS knowledge space of four main extensively dependent dimensions was discovered by industrial practitioners and not by researchers. The figure below, termed the "Railway-crossing Diagram" is from the Globeman 21 project. Japanese industry partners directed our attention to the



fact that the project, the plant, the production and the building site (infrastructure) were actually designed, engineered and evolving in parallel, but that current methods and systems are not able to adequately support concurrent engineering of these flows. With enterprise knowledge spaces and the AKM technology, implementing the spaces as visual scenes, performing concurrent knowledge engineering work will be possible.

Figure 7 The four knowledge dimensions of construction industry.

The AKM technology and the adapting AMIS approach have the potential to support concurrent knowledge engineering and execution of project-oriented work.

12.7.1 Active Knowledge Spaces

AKM technology will enable model-designed solutions, and work-driven active model development and knowledge engineering. This means that ontology and Enterprise Modeling tools must be integrated in the ICT architecture and described as tasks to work on the structures of the EKA, in order that semantic ambiguities and flaws may be detected and corrected. Ontology engineering is an integral service of active knowledge modeling. This means that most Enterprise Modeling approaches also have to open up, and recognize the true nature of enterprise knowledge. Enterprise Modeling has to capture and engineer structures and support work execution in knowledge spaces. "Enterprise Modeling is externalizing situated knowledge (4th horizon KM) and building the Enterprise Knowledge Architecture."

Interoperability, integration and reuse are facilitated by the layered architecture, and the services supported and provided by the EKA. Situated knowledge always involve four major inter-dependent knowledge dimensions. This has long been accepted in problem-solving and design theories.

By this approach all systems are federated and integrated systems, embedded in the VO knowledge space and all its cascaded knowledge spaces for innovation, business operation, strategy governance and balanced values assessment.

12.8 The Future of Ontology and Knowledge Management

Work to develop the three layers of the described Enterprise Architecture integrated by the Intelligent Infrastructure and its services is underway [x].

12.8.1 Model Designed Solutions

Knowledge models should have web components and user environments defined in their meta-models. Then the knowledge that is externalized and managed in models can be reused to generate truly dynamic user interfaces.

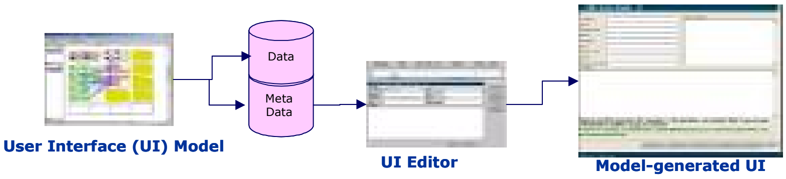


Figure 7 Model-generated workplaces and task execution views.

Once a user environment has been built and its behavior defined, we still need to guide the users' performance of their task assignments.

12.8.2 The powers of knowledge spaces and visual scenes

These capabilities are not obtainable in other settings than visual scenes:

- Continuous knowledge development and capture, modeling processes and views, executing tasks, and aggregating values and experiences,
- Knowledge creation and processing, meta-modeling, executing and monitoring task execution as adaptive services,
- Knowledge use and value creation, performing model interrogations and task and method executions, and providing value metrics,
- Knowledge management, developing meta-models, reflective meta-model views, handling architectures, and generating governance views,
- Cognitive and Generative Learning, preparing knowledge for practical use as competence and skill profiles.

There are properties, such as pro-active learning, that are only possessed by scenes.

12.9 Recommendations

Enterprise Interoperability concerns how to express, share and represent enterprise knowledge. Software is nothing but knowledge encoded for the digital multi-media to execute and manage, so our challenges are to de-couple the knowledge that must be computed, from that needing interactive enhancement by users, from that which has to be simple but yet powerful in supporting on-demand business opportunities.

Understanding the importance of continuously being able to express and represent knowledge and for stakeholders to be able to continuously interact with their knowledge through lifecycles has been and still is a major challenge.

Enterprise Architecture (EA) development is still only able to capture descriptive legacy views and limited dependencies, but there are initiatives towards also supporting operational solution architectures,

Conceptual knowledge drives work and actions, work yields values, values yield experiences and behavior, and experiences drive enhanced knowledge, so this human learning cycle will be closed. The degree of interaction with the contents of the enterprise knowledge architecture will vary across sectors, type of knowledge work, and platforms, and the ambition of the foreseen solutions.

12.10 Future research directions

Business Models, Business Architectures and Enterprise Knowledge Architectures still require a lot of research and experimentation to become viable industrial solutions. To get business people involved in modelling, the user interface must be truly simple to use and the value contributed by the models and the platform supporting them must be substantial. Business case analyses must be easily performed. Enterprise Modelling must be progressed to support the capturing of knowledge in enterprise knowledge spaces, and to exploit the services of intelligent infrastructures turning models into executable workplaces and tasks.

Semi-automatic Knowledge management

Extending the services of the Enterprise Knowledge Architecture it is possible to start researching the objective of automating knowledge management in the area of design and knowledge engineering. Intelligent Infrastructures act as a ”reflective enterprise visual memory”, supporting the intrinsic properties of knowledge, and will allow us to also collapse the temporal dimension. This means cycles are represented by task patterns and we may be relieved of revisions and nested versioning.

Integrating Enterprise Modeling and Ontology

Integrating ontology tools and services to: - edit ontology structures of the enterprise knowledge architecture, - transform ontology structures between cultures and many global enterprise architectures, - compare ontology structures to validate changes and incompliance, and propose corrective actions, and finally to manage ontology structures as part of the enterprise Knowledge Architecture and its life-cycles.

Work Execution and Management

Project engineering and management environments (PEME), supporting model generated workplaces and tasks execution, are needed to move knowledge work to the web. Web services will be of at least four types: software services, dynamic agents, adaptive work process tasks and application capabilities,

Enterprise Visual Scenes (EVS) hold the key to a new style of computing that will enhance creativity, reduce complexity, and augment human capacity.

12.10.1.1 Enterprise Knowledge Architectures

Structures of coherent meta-models, the EKA and its services, seamlessly integrate enterprises; creating situated semantics, syntax and work action context,

Systems Development

Systems Engineering is transforming to Solutions Engineering, separating business, enterprise and IT specific logic and views. The solutions will be model designed, and all solutions will be embedded in knowledge spaces.

12.10.1.2 *Support for Governing Networked Organizations*

In any research project use-cases must be modeled as knowledge spaces as early as possible. VO research needs true industrial user involvement and good user environments for concurrent modeling, meta-modeling, work execution and monitoring. Some key R&D results:

- Each use-case meta-knowledge integrates and validates the results,
- Specific enterprise knowledge architectures yield interoperable solutions,
- Knowledge models and architectures are major results of the project,
- Services of many types and kinds will have to be developed.
- Enterprise Modeling as services adapts and extends the generic architectures.

Knowledge Management means building knowledge architecture solutions. Semantics alone is not the key to interoperability. Patterns of work in reproducing contexts will be needed. Interoperability is all about knowledge architectures and services to extending and adapting models to capture the situation as we execute work.

13. APPENDIX2: ONTOLOGY EDITOR SURVEY RESULTS

Excerpted from (Denny 2002).

Tool	Version	Release Date	Source	Modeling Features/Limitations	Base Language	Web Support & {Use}	Import/Export Formats	Graph View	Consistency Checks	Multi-user Support	Merging	Lexical Support	Information Extraction	Comments
Apollo	1.0 beta 5	9/20/02	Knowledge Media Institute of Open University (UK)	Classes with slots plus relations; functions; hierarchical views.	OKBC model	No, but {server is planned}.	CLOS; OCML	No, but planned.	Yes	No	No	No	No	None
CIRCA Taxonomy Administrator	1.1	3/1/02	Applied Semantics, Inc.	Maps designed taxonomies to built-in general lexical ontology using weighted concept clusters ("gist"). No definable relations.	Proprietary	No	(RDFS planned)	Browsing of ontology (not for editing).	Yes, limited.	No	Yes, via common mapping.	Yes	Via other CIRCA tools.	Part of CIRCA Auto-Categorizer. A future (4Q'02) product may support relations and RDF import/export.
CoGITaNT	5.1.0	9/14/02	LIRMM CNRS (France)	Conceptual graph (CG) modeling with rules; nested typed graphs; projections.	CG model	{Web based client access}	BCGCT; CGXML; XML	Browsing of ontology.	Yes	No	No	No	No	Open Source server, client and underlying C++ library; also Java API.
Coherence	2.0.1; 2.1	1/1/02 ; 4Q'02	Unicorn Solutions	Roundtrip transformation of ontologies from XML Schema and RDB schemas. Class and property hierarchies; business rules.	XML	{Internet client for sharing ontologies}	XML Schema; RDB schema; XML; RDF(S); DAML+OIL (in 2.1 release)	No, but planned.	Schema synchronization and dependency (referential integrity) to show impact of changes.	(Yes, in 2.1 release)	(Planned)	Explicit mapping between lexicons is possible.	No, except as explicit mappings from RDB.	Ontology functions are part of an enterprise data integration product. Additional input/output: entity-relation diagrams, COBOL Copybooks, HTML.
Contextia	2.1	8/1/02	Modulant	Basic concepts and relations with datatypes are represented in schemas.	Express	Referenced ontologies (URLs); URIs	Entity relation diagrams; XML Schema	For editing single ontology (using FirstStep XG).	Express model (ISO 10303) validation; cross-ontology consistencies	No	Schema mapping including aggregation/generalization; "context" mapping.	Synonym mappings; term matching	No, except as explicit mappings from structured and semi-structured sources.	Ontology functions are part of an enterprise data integration product. Ontology editing supported by FirstStep XG included with Contextia.
COPORUM OntoBuilder	1.5	8/1/02	CognIT AS	Basic concepts and relations are represented with single inheritance. Representation of concepts and relations extracted from content may be extended with WordNet information.	RDFS	{Web based repository; Web services in development}	DAML+OIL; RDF(S)	Browsing of ontology.	RDF consistency via repository.	(Under development)	Flat merging via Sesame.	Yes, based on WordNet and RDF Query Language; also in Sesame	Yes, based on meaning and distribution.	Tool embedded in On-To-Knowledge project tool set and requires Sesame RDF repository. Focus on generating editable ontologies automatically from natural language documents.
DAG-Edit	1.3.1.2	3/20/02	Berkeley Drosophila Genome Project (BDGP)	Mixed part-of and isa concept hierarchies are represented along with synonym and search facilities. No properties.	Directed (cyclic or acyclic) graph notation	{Read input via URLs}	Gene Ontology: RDF; Gene Ontology Postgres Database Schema (experimental); (DAML+OIL in GOET)	No, but tree view of flattened graph.	No	No	Yes, especially at the term level; also change history tracking.	Yes, for synonyms.	No, but allows regular expression search.	While intended for gene expression ontologies, it can be used for any taxonomy. Generic version - GOET - is under development (alpha).
DAMLImp (API)	0.7.0	7/15/02	AT&T Government Solutions	DAML+OIL constructs. Basic Java library for analysis and manipulation of DAML+OIL ontologies.	DAML+OIL	URIs	DAML+OIL; RDF	No	No	Possible	No, but Ontology Manager aids mapping.	No	No	DARPA DAML project
Differential Ontology Editor (DOE)	1.1	9/1/02	National Audiovisual Institute - INA (France)	Creates lattice of concepts and lattice of relationships between concepts, plus a set of instances. Concepts cannot be defined intentionally with constraints. Only types of the domains of relationships can be specified. No axiom editor is provided.	XML & CGXML	Load ontology by URL	DAML+OIL; RDFS	No, but tree view.	Arity and type inheritance on relation domains; detects cycles in hierarchies.	No	No	Term definitions, synonyms and preference; methodology for differential definitions.	No	Supports methodology of Bruno Bachimont; to be used with other editors.
Disciple Learning Agent Shell	2.6	4/1/02	George Mason University , Learning Agents Laboratory	Semantic network representation with functions, extended to allow partially learned entities. A hierarchy of objects and a hierarchy of features, with their descriptions, are represented as frames. Also, general problem solving rules can be expressed with terms from the	OKBC-like	{Ontology summaries output in HTML}	Import: CYC ontologies	Browse classes, properties and individuals.	Syntactic consistency is always maintained; can commit multiple changes to persistent ontology in single operation.	No	Yes, two ontologies.	Search for terms	No	The shell is used by subject matter experts to rapidly form knowledge and reason about a specific domain. Users, via a set of task reduction rules, create Disciple-RKF agents that can be combined into a single knowledge base.

Tool	Version	Release Date	Source	Modeling Features/Limitations	Base Language	Web Support & {Use}	Import/Export Formats	Graph View	Consistency Checks	Multi-user Support	Merging	Lexical Support	Information Extraction	Comments
				ontology.										
Domain Ontology Management Environment (DOME)	2.0	8/1/02	Btexact Technologies	Concepts, relations and constraints are mapped to ER-like specifications.	CLASSIC & FaCT	{Web access}	OKBC; XML	ER diagrams	Yes	Yes	(Under development)	(Under development)	Semi-automatic and rule-based extraction from RDBs and web pages.	Available externally by individual agreements with limited support.
DUET	0.3.0	7/17/02	AT&T Government Solutions	Represents only UML static constructs available on class diagrams.	UML	URLs and namespaces are preserved in UML package naming	DAML	Editing using UML class diagrams (via Rose or Argo products)	Valid UML diagrams will produce valid DAML+OIL and conversely.	Supports multi-user capabilities of Rational Rose.	Multiple ontologies may be imported for comparison and merging.	No	No	DARPA DAML project. Additional output: HTML views of UML models. Also under development for GentileWare Poseidon UML.
Enterprise Semantic Platform (ESP) including Knowledge Toolkit	3.0	11/30/02 (expected)	Semagix, Inc	Description models composed of hierarchical categories and attributes with named relationships. Type system for heterogeneous media content. Instances supported by simple constraints on entities (cardinality, range) and entity properties, as well as inferencing. Automatic assertion and maintenance of instances is possible.	Graph & XML	URIs; {partial HTML client; HTTP API}	XML; (RDF(S) is planned)	Connected tree browsing via TouchGraph	Yes, includes automatic and user interactive checks; dynamic content management.	Limited, user privileges prevent concurrent update of the same ontology parts.	No	Synonym based term normalization.	Automatic ontology directed classification and semantic annotation of heterogeneous content.	ESP is an application platform for semantic integration of heterogeneous content including media and enterprise databases. It includes the Knowledge Toolkit for building ontologies.
EOR	1.01	7/10/01	Dublin Core Metadata Initiative	RDF models as sets of triples. Can be used to build, insert (infuse) and query instance knowledge bases for DAML+OIL, RDFS, etc. ontologies.	RDF	URIs	RDF	No	Validate RDF	No	Yes, by adding sets of RDF statements.	No	No	Developed by OCLC.
ExClaim & CommonKADS Workbench	release	12/1/01	National Institute for Research and Development in Informatics (Romania)	Description logic modeling plus primitive problem solving actions.	DL model	No	CML	Browsing of ontology.	Knowledge verification and model validation (for DL representation).	User roles	No	No	No	Uses the CommonKADS Workbench based on SWI-Prolog and the XPCE GUI.
GALEN Case Environment (GCE)	5	8/1/02	Kermanog	Description logic terminological modeling without support for individuals. Composite concepts are automatically classified according to their criteria (relationships with other concepts). New concepts can be created interactively and according to user-defined rules.	GRAIL	No	GRAIL	No, but filtered tree views allow editing.	Explicit grammatical and sensible sanctions are enforced when combining terms.	No	Compiles differences in concepts, hierarchies and criteria (properties) between two ontologies.	GALEN concept identifiers can be associated with synonyms and word forms.	No	Although, developed primarily as a medical terminology model builder, the tool can serve as a general purpose ontology editor. GCE is part of the Classification Workbench with support to manage domain classification schemes.
ICOM	1.1	4/25/01	Free University of Bozen-Bolzano, Italy	EER (extended entity relations) modeling plus inheritance hierarchies, multidimensional aggregations and multiple schema relations.	Description logic	No	XML; UML (future)	Native editing of ER diagrams (UML diagrams planned).	Verify the specification via DL classifier (FaCT).	No	Supports inter-ontology mappings with graphical interface.	No	No	Graphically editing of native UML class diagrams planned for next release.
Integrated Ontology Development Environment	1.6.1	7/1/02	Ontology Works, Inc.	Distinguishes between properties and relations; allows contexts; default reasoning; temporal model relations; higher-arity relations; meta-properties and meta-relations.	OWL (based on KIF; not related to W3C WebOnt language of same name.)	{Web client - control panel}	KIF; UML; RDB; XML DTD	UML diagrams	Top-level ontology consistency per Guarino & Welty.	Yes	(slated for version 1.8 in Q2 2003)	Synonyms; English-language names	No	Supports OntoClean methodology (Guarino & Welty); supports relational and other databases
IsaViz	1.1	5/23/02	W3 Consortium	Supports RDFS level specifications. Can specify any model based on RDF such as DAML+OIL.	RDF model	URI namespaces	RDF; N-Triple; SVG	Native creation and editing of resources, literals and properties.	RDF model correctness.	No	Yes	No	No	None
JOE	Demo	7/21/99	University of South Carolina Center for IT	Basic concept and relations modeling ala ER.	KIF	No	ER (LDL++)	No	No	No	No	No	No	No current development. Available as an applet.
KAON (including OIModeller)	1.2	9/25/02	FZI Research Center & AIFB Institute, University of Karlsruhe	Extends RDFS with symmetric, transitive and inverse relations, relation cardinality, meta-modeling, etc. Similar to F-Logic using axiom patterns. Editor currently only supports concept hierarchy.	KAON (proprietary extension of RDFS)	{Browsing ontologies via KAON Portal; Web services API under development}	RDFS	No	Yes, for evolution of ontology.	Concurrent access control with transaction oriented locking and rollback.	(Under development)	Explicit lexical representation in model. Synonyms; stemming; multilingual.	(Under development)	OIModeller is part of KAON tool suite for business applications that uses RDB persistence layer for scalability. The ontology editor is under development.
KBE -- Knowledge Base Editor (for Zeus AgentBuilding Toolkit)	1.3	3/22/00	Institute for Software Integrated Systems, Vanderbilt University	Zeus ontology model of concepts, attributes and values; multiple inheritance; modularization within a closed world model. (Also defines agent interaction protocols.)	GME	No	Zeus ontology file (.edf)	UML-like diagrams for browsing only.	Yes	No	No	No	No	KBE is layered on top of the Zeus environment for building agents and extends the ontology editor functions. The underlying GBE model specification system could be

Tool	Version	Release Date	Source	Modeling Features/Limitations	Base Language	Web Support & {Use}	Import/Export Formats	Graph View	Consistency Checks	Multi-user Support	Merging	Lexical Support	Information Extraction	Comments
														used as the basis of other ontology builders.
LegendBurster Ontology Editor	1.1.2	10/5/02	GeoReference Online Ltd	Semantic network hierarchy of concepts, attributes, attribute values and explicitly represented truth-status flags. Inheritance within hierarchies with lateral links. Full reified relations; inverse relations (partial). Metadata for all entities (at node level). Separate tree list editor.	Proprietary (uses Prolog)	No	No, except across projects (proprietary).	No, except SVG export of instance and query graphs.	Partial, with strict attribute context checks but arities currently unchecked.	No	Yes, if from LegendBurster. (User must check semantic consistency.)	Term search and alphabetical sort.	Semi-automatically capture and import vocabulary present in attribute tables of maps of interest.	While LedgendBurster is principally a GIS application, the Ontology Editor is suitable for general purpose ontology development. Standalone editor with instance description and fuzzy query expected in early 2003.
LinKFactory Workbench	release	7/1/02	Language & Computing nv	Description logic T-box (terminological) and A-box (assertional) model. Multiple inheritance over concepts and relationships; identification of necessary and sufficient criteria for concept definition. Manage multiple conflicting ontologies in one T-box. Versioning metadata.	Extended description logic	{LinKFactory Server supports Internet clients; WebInfo spider component.}	XML; RDF(S); DAML+OIL/OWL	No	Checks cover role restrictions, formal disjoints, sanctioning over subsumers, etc.	Yes, with author privileges and auditing specific to concept hierarchies.	Compares and links ontologies via a core ontology; related concepts matched on formal relationships and lexical information.	Strict concept/term distinction; lexeme-description; part-of-speech. Search with wildcards.	Yes, via text analyses and automatic linkage to ontology. WebInfo spider gleans domain-specific concepts/terms on Web.	LinKFactory Workbench includes a database server, application server and clients. Originally designed for very large medical ontologies. It has a Java beans API and optional Application Generators for semantic indexing, automatic coding, and information extraction.
Medius Visual Ontology Modeler	0.18 beta	7/1/02	Sandpiper Software, Inc	UML modeling of ontologies with frame systems support.	UML with extensions for OKBC model	URI support in DAML generator; {read-only browser support from Rose}	XML Schema; RDF; DAML+OIL	Yes, as UML diagrams via Rose.	Limited	Yes	Native Rose model merging support	Search for terms and relations.	No	Operates as a Rational Rose plug-in.
NeoClassic	release	12/15/00	Bell Labs (Lucent Technologies)	Framework representation of descriptions, concepts, roles, individuals and rules. Concepts can be derived from necessary and sufficient conditions for individual membership. Subsumption and classification are inherent inference. (Command line editor only.)	DL model	No	No	No	Yes	No	No	No	No	This C++ implementation of the original CLASSIC system is the only currently supported version.
OilEd	3.4	4/12/02	University of Manchester Information Management Group	DAML constraint axioms; same-class-as; limited XML Schema datatypes; creation metadata; allows arbitrary expressions as fillers and in constraint axioms; explicit use of quantifiers; one-of lists of individuals; no hierarchical property view.	DAML+OIL	RDF URIs; limited namespaces; very limited XML Schema	RDFS; SHIQ	Browsing Graphviz files of class subsumption only.	Subsumption and satisfiability (FaCT)	No	No	Limited synonyms	No	None
OLR3 Schema Editor	1	4/1/02	Institute for Information Systems, University of Hannover	Instantiation and editing of external or custom schemas conforming to RDFS. Concept-specific filtering to present choice of legal properties.	RDFS	RDF URIs; {browser based}	RDF	No	Yes, for property constraints, etc.	No	No	No	No	Part of the Open Learning Repository Version 3 (OLR3) system for course specification.
OntoBuilder	1.0	6/13/02	Institute for Medical Information, Statistics and Epidemiology University of Leipzig	Manages compilation of domain terms, their description, and contexts using natural language.	Natural language; (logical representation language planned)	{Web access}	No	No	Not automatically	Yes, with editor, moderator and administrator user group types.	No	Representation of synonyms; search on terms and descriptions; lexical rules for term input	No	Semantic analysis using a formal model based on a top level ontology and a logic-based representation language are planned. Domain focus is on medicine.
Onto-Builder	3.0	5/1/02	University of Savoy ; Ontologos	Distinguishes "what contributes to the essence of things and what describes them", defining concepts by their "specific difference". Thus, logical and set-oriented semantics are derived a posteriori.	LOK (Language for Ontological Knowledge) written in Smalltalk	{Web access}	Input: DAML-OIL; XML, LOK Output: DAML+OIL; XML; KIF; Conceptual Graph	Yes, for browsing.	Yes, based on logic and on the specific-difference theory.	User groups.	Yes, for ontologies based on the OK Model.	Lexicon management including synonyms	Extraction of lexicons from texts with OK lexical tools (based on Brill's tagger).	Part of Ontological Knowledge Station. Building OK ontologies is based on a dedicated methodology.
OntoEdit	2.5.2	8/6/02	Ontoprise GmbH	F-Logic axioms on classes and relations; algebraic properties of relations; creation of metadata; limited DAML property constraints and datatypes; no class combinations, equivalent instances.	F-Logic	Resource URIs	RDFS; F-Logic; DAML+OIL (limited); RDB	Yes, via plug-in	Yes, via OntoBroker	Transaction locking at the object and whole subtree levels.	Yes	Multiple lexicons via plug-in	No	Free and commercial (Professional) versions are available, with continuing development of the commercial version.
Ontolingua	1.0.649;	11/214/01;	Stanford	OKBC model with	Ontolingua	{Web access	Import & Export:	No	Elaborate with	Write-only	Semi-automated via	Search for terms	No	Online service

Tool	Version	Release Date	Source	Modeling Features/Limitations	Base Language	Web Support & {Use}	Import/Export Formats	Graph View	Consistency Checks	Multi-user Support	Merging	Lexical Support	Information Extraction	Comments
with Chimaera	0.1.42	7/24/02	Knowledge Systems Lab	full KIF axioms.		to service.}	DAML+OIL; KIF; OKBC; Loom; Prolog; Ontolingua; CLIPS. Import only: Classic; Ocelot; Protégé.		Chimaera; Theorem proving (via JTP)	locking; user access levels.	Chimaera	in all loaded ontologies.		only (at http://www-ksl-svc.stanford.edu); Chimaera is being enhanced under DARPA funding in 2002.
Ontology Builder & Server	1.1	9/1/01	Verticalnet, Inc.	Classes with slots, datatypes and cardinality constraints; node documentation; inclusion. No axioms.	Partial OKBC model	Fully qualified names; {HTTP browser and server}	RDFS & DAML+OIL (future)	No	Limited to term validity and graph cycles.	User roles and security; global locking.	Simple difference and merge process.	Yes	No	Currently available only as part of their enterprise solution product.
Ontology Directed Extraction (ODE) Tools	alpha	4/8/02	XSB, Inc.	Multiple inheritance subsumption class hierarchies. Support for typed attributes of classes and relations between classes. Supports schema and object information.	Tabled Prolog	No	No	No	Yes	No	Yes, with limitations.	Yes	Yes	Tool supports construction of domain ontologies used to guide lexical classification and information extraction.
Ontopia Knowledge Suite	1.3.4	8/28/02	Ontopia AS	Constraint modeling specifically and solely for Topic Map representations.	Ontopia Schema Language (OSL)	{Web access and Web API}	OSL; XTM; LTM (import only); HyTM	No, but tree view.	Validation against the OSL schema.	Full concurrency and transaction support when running with RDBMS.	For ontologies and instance data, but not (currently) for constraints.	Full-text search	No, but application framework allows this.	Although primarily an IDE for Topic Map applications, the framework supports ontologies.
Ontosaurus	1.9	3/28/02	USC Information Sciences Institute	Rich KB browser with simple editing; contexts; same-class-as; metaclasses.	Loom	{HTTP browser}	KIF; Loom; OKBC	Browsing class hierarchy	Yes	Global locking	No	No	No	Online access to KBs hosted on CL HTTP server.
OntoTerm	0.9.98	6/1/99	University of Malaga	Concept and property hierarchies with concept instances; properties distinguished as attributes or relations. Metadata (natural language definitions).	n/a	{HTML output}	n/a	No, but cross-linked tree views indicate legal element associations or types, and allow editing.	No	No	No	Word lists	No	Although intended to be a terminology management system, OntoTerm can be used for general ontology development. Ongoing development and support of the software is unknown.
OpenCyc Knowledge Server	0.6.0b	4/3/02	Cyc Corp.	FOPC extended with contexts, equality, default reasoning, skolemization, quantification over predicates. (Basic ontology editing via KB Browser Create Term tool.)	CycL (& SubL)	{HTTP server}	DAML+OIL (native KB only)	No	Directed inferencing and queries; truth maintenance	Yes	No	Yes, via Cyc-NL with KB-linked lexicon for syntactic and semantic disambiguation.	English parsing possible with Cyc-NL.	Knowledge base subset and browser only. Future release of ontology building tools: Template-based knowledge entry, Index Overlap, Similarity Tool and Salient Descriptor.
OpenKnoMe	5.4c	9/27/02	University of Manchester Medical Informatics	Description logic terminological modeling without support for individuals or type system. Arbitrarily complex structures may be composed from primitive concepts and relations. Role hierarchy with inverses, and reasoning over relationships such as part-of. No formal negation, disjunction or conjunction. Limited support for cardinality. No reasoning over numbers or ranges. Toolset for managing intermediate representations.	GRAIL	Not as configured.	CLIPS; XML	No	Logical coherence ala DL and a meta-model system for declaring inherited semantic constraints and permissions. Also, declarative query language (GQL) can be used to author checks of modeling consistency.	User roles and read/write privileges; version control. Users see each other's changes only when they check modules back in.	Via explicit mappings (reifications) to GALEN Common Reference Model. Focus is on linking rather than mapping to reference model.	Can use GALEN language module that links its concept identifiers with synonyms and word forms, and provides segment grammar for semantic links.	No	Although, developed primarily as a medical terminology model builder, the tool can serve as a general purpose ontology editor. Currently requires OpenGALEN terminology server and CINCOM VisualWorks runtime environment.
PC Pack 4	1.0 beta (unreleased)	4/27/2001 (predecessor)	Epistemics Ltd	Knowledge acquisition and modeling. Multiple inheritance; n-ary relations; rules and methods. User definable templates for modeling formalisms like CommonKADS and Moka.	XML	{HTML output via XSLT}	XML	ER diagrams; class hierarchies; OO views	Only logically consistent models can be created.	Yes	No	No	No	Suite of many integrated KADS inspired tools.
Protégé-2000	1.7; 1.8 beta	4/10/02 ; 10/22/02	Stanford Medical Informatics	Multiple inheritance concept and relation hierarchies (but single class for instance); meta-classes; instances specification support; constraint axioms ala Prolog, F-Logic, OIL and general axiom language (PAL) via plug-ins.	OKBC model	Limited namespaces; {can run as applet; access through servlets}	RDF(S); XML Schema; RDB schema via Data Genie plug-in; (DAML+OIL backend due 4Q'02 from SRI)	Browsing classes & global properties via GraphViz plug-in; nested graph views with editing via Jambalaya plug-in.	Plug-ins for adding & checking constraint axioms: PAL; FaCT.	No, but features under development.	Semi-automated via Anchor-PROMPT.	WordNet plug-in; wildcard string matching (API only).	No	Support for CommonKADS methodology.
RDFAuthor	alpha	5/9/02	Damian Steer	Create RDF instance data against RDFS schemas.	RDF	URIs; {Web links; remote RDF query}	XML; RDF	Creating and editing instances as graphs.	RDF errors	No	No	No	No	Currently available for Mac OS X; also as a Java Swing application. Additional output:

Tool	Version	Release Date	Source	Modeling Features/Limitations	Base Language	Web Support & {Use}	Import/Export Formats	Graph View	Consistency Checks	Multi-user Support	Merging	Lexical Support	Information Extraction	Comments
														SVG, PNG, TIFF, PDF.
RDFedt	1.02	2/25/01	Jan Winkler	Textual language editor only.	RDF model	RSS	RDFS; DAML; OIL; Shoe	No, but tree view.	Writing mistakes only.	No	No	No	No	None
SemTalk	1.1.3	8/18/02	Semtation GmbH	Subset of RDFS and DAML extended with inverse relations and process modeling.	Visual Basic	URI namespaces; {distributed development}	XML; F-Logic; ARIS models; Bonapart models	Yes, for design and browsing.	Subsumption and name usage across multiple models; meta-model specific checks.	No	Yes, with simple filtering.	Synonyms; homonyms; stop words; some POS; glossaries via Babylon .	No, but interfaces to appropriate Ontoprise and TextTech products.	Microsoft Visio extension and SmartTags. Additional output include: Rational Rose UML class diagrams, RDF annotated HTML, MS Excel, MS Project, SAP IPC, HTML/VML.
Specware	3.1	6/1/01	Kestrel Technology	Logical and functional axioms. (Text based language editor only.)	Metaslang	No	None	No	Proofs via Gandalf and SNARK.	No	Yes, via composition operations (e.g., co-limits).	No	No	While primarily a tool for the formal, compositional specification of software, Specware can be used to define domain theories.
SymOntos	2.2	4/1/02	Institute for the Analysis of Information Systems - CNR (Italy)	XML Schema modeling constructs with subsumption of classes and relations; specified relation types of isa, part-of, similarity and predicate. Business-oriented predefined classes such as: actor, process, event, and message.	XML	{Web access}	XML; RDF(S)	(Planned release 4Q'02)	Concept hierarchy validity, range restrictions and graph cycles.	Simple user groups	Possible via XML encoding.	Word lists of synonyms; term query support.	No	Online service; academic level support; can support collaborative ontology building. SymOntoX version in progress with language for process, actor, event and goal.
Taxonomy Builder	2.0	8/1/02	Semansys Technologies	General taxonomy of elements assigned data types and substitution groups. Predefined XBRL relation types via links.	XML Schema	XML namespaces; {taxonomy browser; Internet client}	XML; XML Schema	No	Yes, relative to XBRL core schema.	No	Yes	No	No	Available separately or as part of the Semansys XBRL Composer Professional. Additional outputs include CSV, TXT and SQL.
TOPKAT	prototype	6/17/95	AIAI, University of Edinburgh	Supports representation of the various models of CommonKADS (circa 1995). Underlying these models are dictionaries of concepts, properties, property values, inferences, and tasks. Production rules can be represented using a combination of these primitives.	HARDY and CLIPS	No	CML	Native graph view for editing.	Limited	No	No, except for models within a single ontology.	Term equivalence through the data dictionary.	Simple natural language parser can identify possible concepts and property values in a protocol transcript.	The Open Practical Knowledge Acquisition Toolkit (TOPKAT) supports CommonKADS knowledge acquisition techniques including: laddered grid, card sort, repertory grid, protocol analysis. Final diagrams also output in HTML. No support.
Visio for Enterprise Architects	2002 SR-1	4/2/02	Microsoft Corp.	Most object-role modeling (ORM) constructs, but imposes relational logical constraints on specification.	ORM	No	XML (via add-on); DDL	ORM class diagrams	Yes	Yes	Yes	No	No	ORM modeler may be effective for specifying domain ontologies; part of Visual Studio.NET Enterprise Architect
WebKB	2.0	9/10/02	Distributed Systems Technology Centre (DSTC), Australia	Basic conceptual graph modeling and manipulation that includes contexts, constraint checking and querying. Can derive new statements (e.g., relations) from necessary and sufficient conditions.	FS (extended CGs)	URIs for element references; Web access of KBs; {CGI and HTML interfaces; Web script language}	Export (partial only): DAML/RDF; CGIF; KIF	Hyperbolic-like browsing (of taxonomies only) via KVO's OntoRama.	Syntactic and logical including transitive cycles, disjoint relations, relation signatures. Also lexical checking.	KB sharing restricts editing so each element has an associated author.	No, but separate ontologies can share the same KB "framework" including a WordNet based upper ontology.	WordNet nouns and adjectives; aliases; element searching by author or name.	No	On-line service (www.webkb.org); also source and binary code available.
WebODE	2.0.6	7/10/02	Technical University of Madrid UPM	Concepts (class and instance), attributes and relations of taxonomies; disjoint and exhaustive class partitions; part-of and ad-hoc binary relations; properties of relations; constants; axioms; and multiple inheritance. Inference engine for subset of OKBC primitives and axioms.	Prolog translation of FOL and frames per OKBC model.	URIs as imported terms; {browser client}	DAML+OIL; RDFS; X-CARIN; FLogic; Prolog; XML	Native graph view with editing of classes, relations, partitions, meta-properties, etc.	Type and cardinality constraints; disjoint classes and loops, taxonomy style (OntoClean), etc.	Yes, with synchronization; authentication and access restrictions per user groups.	Unsupervised (ODEMerge methodology) using synonym and hyperonym tables; custom dictionaries and merging rules.	Synonyms and abbreviations; (EuroWordNet support under development)	Using WebPicker (UNSPSC, RosettaNet)	Supports Methontology methodology (Fern‡ndez-L—pez et al, 1999); offered as online service; successor to ODE; ontology storage in RDB.
WebOnto	2.3	5/1/02	Knowledge Media Institute of Open University (UK)	Multiple inheritance and exact coverings; meta-classes; class level support for prolog-like inference.	OCML	{Web service deployment site}	Import: RDF; Export: RDFS, GXL, Ontolingua, OIL	Native graph view of class relationships.	For OCML code	Global write-only locking with change notification.	No	No	(Available from OCML based tool MnM.)	Online service only.