D1.2.3 Methods for ontology evaluation

Jens Hartmann (UKARL)
Peter Spyns (VUB)
Alain Giboin (INRIA)
Diana Maynard (USFD)
Roberta Cuel (UniTn)
Mari Carmen Suárez-Figueroa (UPM)
York Sure (UKARL)

Abstract.
The deliverable D1.2.3 analyses and evaluates existing methods for ontology content evaluation according requirements from an industrial point of view.

EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB
Deliverable D1.2.3 (WP 1.2)

Document Identifier:	KWEB/2004/D1.2.3/v1.3
Class Deliverable:	KWEB EU-IST-2004-507482
Version:	Revised V 1.3
Date:	January 31, 2005
State:	Review
Distribution:	Public

Knowledge Web Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2004-507482.

University of Innsbruck (UIBK) – Coordinator
Institute of Computer Science,
Technikerstrasse 13
A-6020 Innsbruck
Austria
Fax: +43(0)5125079872
Phone: +43(0)5125076485/88
Contact person: Dieter Fensel
E-mail address: dieter.fensel@uibk.ac.at

École Polythechnique Fédérale de Lausanne (EPFL)
Computer Science Department. Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne.
Switzerland
Fax: +41 21 6935225
Phone: +41 21 6932738
Contact person: Boi Faltings
E-mail address: boi.faltings@epfl.ch

France Telecom (FT)
4 Rue du Clos Courtel
35512 Cesson Sévigné
France. PO Box 91226
Fax: +33 2 99124098
Phone: +33 2 99124223
Contact person : Alain Leger
E-mail address: alain.leger@rd.francetelecom.com

Freie Universität Berlin (FU Berlin)
Takustrasse, 9
14195 Berlin
Germany
Fax: +49 30 83875220
Phone: +49 30 83875223
Contact person: Robert Tolksdorf
E-mail address: tolk@inf.fu-berlin.de

Free University of Bozen-Bolzano (FUB)
Piazza Domenicani 3
39100 Bolzano
Italy
Fax: +39 0471 315649
Phone: +39 0471 315642
Contact person: Enrico Franconi
E-mail address: franconi@inf.unibz.it

Institut National de Recherche en Informatique et en Automatique (INRIA)
ZIRST - 655 avenue de l'Europe - Montbonnot Saint Martin
38334 Saint-Ismier
France
Fax: +33 4 7661 5207
Phone: +33 4 7661 5366
Contact person: Jérôme Euzenat
E-mail address: Jerome.Euzenat@inrialpes.fr

Centre for Research and Technology Hellas / Informatics and Telematics Institute (ITI-CERTH)
1st km Thermi – Panorama road
57001 Thermi-Thessaloniki
Greece. Po Box 361
Fax: +30-2310-464164
Phone: +30-2310-464160
Contact person: Michael G. Strintzis
E-mail address: strintzi@iti.gr

Learning Lab Lower Saxony (L3S)
Expo Plaza 1
30539 Hannover
Germany
Fax: +49-511-7629779
Phone: +49-511-76219711
Contact person: Wolfgang Nejdl
E-mail address: nejdl@learninglab.de

National University of Ireland Galway (NUIG)
National University of Ireland. Science and Technology Building. University Road
Galway
Ireland
Fax: +353 91 526388
Phone: +353 87 6826940
Contact person: Christoph Bussler
E-mail address: chris.bussler@deri.ie

The Open University (OU)
Knowledge Media Institute. The Open University
Milton Keynes, MK7 6AA
United Kingdom.
Fax: +44 1908 653169
Phone: +44 1908 653506
Contact person: Enrico Motta
E-mail address: e.motta@open.ac.uk

Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Fax: +34-913524819
Phone: +34-913367439
Contact person: Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

University of Karlsruhe (UKARL)
Institut für Angewandte Informatik und Formale
Beschreibungsverfahren – AIFB. Universität
Karlsruhe
D-76128 Karlsruhe
Germany
Fax: +49 721 6086580
Phone: +49 721 6083923
Contact person: Rudi Studer
E-mail address: studer@aifb.uni-karlsruhe.de

University of Liverpool (UniLiv)
Chadwick Building, Peach Street
L697ZF Liverpool
United Kingdom
Fax: +44(151)7943715
Phone: +44(151)7943667
Contact person: Michael Wooldridge
E-mail address: M.J.Wooldridge@csc.liv.ac.uk

University of Manchester (UoM)
Room 2.32. Kilburn Building, Department of
Computer Science, University of Manchester,
Oxford Road
Manchester, M13 9PL
United Kingdom
Fax: +44 161 2756204
Phone: +44 161 2756248
Contact person: Carole Goble
E-mail address: carole@cs.man.ac.uk

University of Sheffield (USFD)
Regent Court, 211 Portobello street
S14DP Sheffield
United Kingdom
Fax: +44 114 2221810
Phone: +44 114 2221891
Contact person: Hamish Cunningham
E-mail address: hamish@dcs.shef.ac.uk

University of Trento (UniTn)
Via Sommarive 14
38050 Trento
Italy
Fax: +39 0461 882093
Phone: +39 0461 881533
Contact person: Fausto Giunchiglia
E-mail address: fausto@dit.unitn.it

Vrije Universiteit Amsterdam (VUA)
De Boelelaan 1081a
1081HV. Amsterdam
The Netherlands
Fax: +31842214294
Phone: +31204447731
Contact person: Frank van Harmelen
E-mail address: Frank.van.Harmelen@cs.vu.nl

Vrije Universiteit Brussel (VUB)
Pleinlaan 2, Building G10
1050 Brussels
Belgium
Fax: +32 2 6293819
Phone: +32 2 6293308
Contact person: Robert Meersman
E-mail address: robert.meersman@vub.ac.be

Changes

Version	Date	Author	Changes
0.1	08-01-2004	J. Hartmann (jha)	Initial Setup
0.2	08-03-2004	J. Hartmann (jha)	Update
0.3	09-10-2004	J. Hartmann (jha)	Updates from KWeb Meeting Trento
0.4	09-20-2004	J.Hartmann (jha)	General
0.5	19-10-2004	J. Hartmann (jha)	Update
0.6	21-10-2004	Diana Maynard	USFD
0.9	05.12.2004	Alain Giboin	Discussion

Executive Summary

A classification of methods and tools for the evaluation of ontologies for industrial practice is provided. It concerns methods and tools to:

- select existing ontologies (possibly from libraries)
- measure the correspondence between textual sources and the corresponding ontology
- evaluate the impact of an ontology on an information retrieval application
- check and improve the quality and consistency of ontologies
- monitor an ontology in use

The methods and tools intervene at different stages in the industrial life cycle of a software product. For each method and tool, its scientific basis and design purposes as well as its relevance and usefulness for industry are presented.

Contents

1 Introduction

In this deliverable, the evaluation of ontology content and its suitability for adoption by industry should be analysed. No industrial method currently exists for helping an ontology engineer to evaluate and select ontologies that best matches his/her needs. Even in academia, few methods have been proposed, which allows to qualify the topic of this deliverable as "emergent" and "cutting-edge". Several methods and supporting tools, as identified by the KnowledgeWeb partners active on this task, have been classified and included in this deliverable. For many of them, a use case demonstrates the usability of the evaluation method and its relevance for industrial applications.

1.1 The KnowledgeWeb WP2 Joint Research Activities

The main goal of the Knowledge Web Network of Excellence is the transfer of ontology-based technologies and methodologies (often originating from Semantic Web projects) from academic institutions to strategic industries and vice versa. Through business cases, use cases, and research projects, emerging problems, potential solutions and innovative perspectives should be discussed for the utility of both researchers and practitioners. In particular:

- Theoretical studies on formal ontologies are committed to provide sound bases for industrial applications and to allow formal representation of corporate knowledge;

- Business experiences on case studies single out concrete problems and possible solutions;

- Experience analyses should provide useful insights on social and strategic aspects that might be relevant in the creation and deployment of formal ontologies as well as useful criteria or methods to evaluate ontologies and their effectiveness in applications.

Before making recommendations to industry about how to incorporate semantic web technology into their IT systems, content evaluation and ontology-based tools and tool suites are needed taking into account the industrial needs profiles identified from the use cases and industrial scenarios. Significant efforts are required, falling into the following categories: utility of tools, interoperability of tools and services, content evaluation and usability.

1.2 The WP2 task T1.2.3

As an immediate goal, the Work Package 1.2 of the Knowledge Web project foresees the task named "Ontology content evaluation and usability, which includes the content evaluation and suitability of ontologies with the content needed in the industrial use cases" [81]. That task is aimed at evaluating content and usability of ontologies before using them in IT systems.

If we understand evaluation as an activity that includes setting up tests and studying the resulting outcomes, several scenarios are possible. The most important ones are addressed in the subsequent sections of this deliverable. A first important distinction to be made is between what is traditionally called "glass box" or "component" vs. "black box" or "task-based" testing.

In the latter type of evaluation procedure, an ontology typically is tightly integrated in an application – e.g. an information extraction application or a search engine – of which the overall performance is measured (and compared with previous evaluation data). Black box testing will be executed through the same interface that the end user uses, and will only be covered on an exemplary base by this deliverable as the evaluation criteria are mainly application(-type) specific. For instance from this kind of analysis it could happen that a complete, correct, accurate and precise ontology is avoided in favour of a system of categories or a simple taxonomy. Workers, very stressed by their daily activities, might prefer to use a very simple conceptualization system rather than complete and general knowledge representation systems. They might consider a complete and correct ontology as oppressive or irrelevant [85]. Summarizing, an ontology is always the result of a sense-making process (the conceptual modelling), and represents the point of view (the knowledge representation) of those who took part in that process [1][18]. Therefore ontology evaluation methods and tests should be developed and adopted to analyze the effectiveness of ontologies and ontology based tools in real practices, during their daily use within companies.

For glass box testing, we retain three important evaluation stages, which are covered by this deliverable: (i) an ontology in its pre-modelling stage, (ii) an ontology in its modelling stage, and (iii) an ontology after its release. The latter type involves existing ontology libraries, while the former two types are related with existing ontology platforms.

- Evaluating an ontology in its pre-modelling stage (type 1) is to be understood as evaluating the pre-processed material which a human ontology engineer has at his disposal for building the actual ontology. This material can be the result of re-engineering a DB-schema, of scraping Web pages, and of mining text corpora. Before an ontology engineer uses the material collected, (s)he wants to evaluate its quality.

- Evaluating an ontology in its modelling stage (type 2) happens when the ontology engineer, at regular intervals, decides to check the quality of the work done so far. If available, existing "equivalent" ontologies could be used as a reference point. These quality checks also involve consistency checks or other checks on logical errors or errors against the ontology language. Competency questions [84] can be used as a technique to evaluate the ontology content-wise (does the domain model represent the necessary concepts to cope with a type of question content-wise).

- Evaluating an ontology after its release (type 3) is mainly done by people other than the ontology engineer(s) who released the ontology. This can involve comparing different but equivalent ontologies. One can also include monitoring activities of an "ontology in action", i.e. the degree with which (zones of) concepts in an ontology are "used" can be an indication of how well the ontology represents the domain. But evaluation activities of this type are mainly meant to support a decision taking process in a selection procedure, and consist of matching characteristics of an ontology against a set of (qualitative) criteria. An important problem hereby is that, as ontologies are rarely evaluated, documentation is equally rarely available although a number of ontology libraries already exist (e.g., the DAML+OIL library). Even well-known and large ontologies (e.g. Cyc ontologies [82], SENSUS [83]) suffer from the evaluation and documentation problem.

1.3 Relevance for industry

The three evaluation stages we have specified above are highly relevant in the decision making policy of a (high tech) company. We shall discuss two of the most obvious ones: (i) the "make or buy" dilemma based on transaction costs and the (ii) "go or no go" decision.

- The *"make or buy" dilemma* can manifest itself in various forms ranging from a strategic managerial company-wide technology innovation decision to an operational IT-system upgrade decision. The make or buy decision is traditionally based on transaction cost economics evaluation as a widely tested explanation for boundary choices (cf. [86][87][88]), but can be supported by recent theoretical work on competing knowledge-based and measurement cost explanations [89]. These theories take into consideration the strategic impacts of the market and hierarchies as institutions of governance, and analyze strategic factors (such as power, few numbers, innovation factor, etc.) in negotiation processes [90]. In any case, both allow a company to ponder thoroughly the pros and cons of buying or making tools or services, e.g. buying an available external ontology vs. training in-house developers to build the ontology themselves. Clearly, an evaluation of type 3 will intervene at some point in the decision process, as this kind of decision typically involves matching the ontology to be acquired with a checklist of predefined characteristics.

- The *"go or no go"* decision can equally play on various levels of a company transgressing traditional department boundaries. For example, the decision to create an innovative ontology-based service can involve a high number of resources in a company (e.g., R&D and/or engineering divisions up to marketing to create in advance market awareness for the future service). Unlike the "make or buy" decision, which is mostly taken only once, a "go or no go" decision can be split into several lower level "gos or no gos". Depending on the level of technicality, evaluations of type 1 and type 2 help to support this kind of local decision taking. For an depth discussion see ([91], [92]).

Of course, situations can occur where all kinds of evaluation types can be mixed in a complex decision making process. For example, if the in-house engineers do not succeed in reaching a quality level for an ontology-based service in time, a "no go" decision can be taken and the "make or buy" dilemma resurfaces (or is transformed into an ultimate "no go" by the executive management board). These processes can be periodically repeated any time organizations and workers consider them important for their performances. In particular they can evaluate an ontology during its whole life cycle. In fact, during this time organizational behaviours can affect the concrete appropriation of technology, and use of personalized shared conceptualization (cf. [16][17]). Considering that there is no one correct way to model a domain, that different alternatives are always available, that the system of artefacts that workers use change in time, and that ontology based tools can be appropriated in different ways, ontologies can affect knowledge sharing and managing processes. Then periodical evaluation of ontology and “make or buy” or “go or no go” decisions should be taken.

1.4 Relation to other Workpackages

As defined in the Annex 1 the Deliverable and its corresponding Task 1.2.3 is dependent to the task 1.1.3 referring to Typology of ontology-based processing tasks and high level components needed to fulfil the prototypical application requirements. Considering the fact that the Task 1.1.3 is delivered on Month 12, we didn’t explicitly use its content, but we cooperate and shared knowledge with people continuously involved in that tasks. During the research activity we identified another related Work Packages: WP 1.3 of the Knowledge Web research track. In the WP 1.3 some relations refers to best practices guidelines. In particular, it analyzes organizational practices and procedures of ontology evaluation in terms of efficiency and effectiveness.

1.5 Structure of Deliverable

A distinction has been made between methods (section 2) and tools (section 3). The former have clearly identifiable theoretical principles, while the latter are more practical (implemented) engineering tools.

The method section includes examples of each evaluaton type (see above), namely

- Evaluating an ontology in its pre-modelling stage (type 1): sections 2.4, 2.2.2.1, 2.2.2.2 and 2.2.2.3
- Evaluating an ontology in its modelling stage (type 2): sections 2.3 and 2.2.2.4]
- Evaluating an ontology after its release (type 3): section 2.1

The tools section contains examples of

- Evaluating an ontology in its modelling stage (type 2): section 3.1
- Evaluating an ontology after its release (type 3): section 3.2

For each method and tool, its scientific basis and design purposes as well as its relevance and usefulness for industry are presented. A discussion (section 4) ends this deliverable.

2 Methods for evaluating Ontologies

2.1 OntoMetric

2.1.1 Abstract

The development of the Semantic Web has encouraged the creation of ontologies in a great variety of domains. Although most of the methodologies for building ontologies [71] propose a phase of ontology reuse, there are no works that indicate to the users how to choose ontologies for a new project, and there are no methodologies that quantify the suitability of these ontologies for a new system. Knowledge engineers are currently looking for ontologies in different web servers in order to incorporate such ontologies into their systems, and they choose the ontologies just using their experience and intuition. This makes it difficult for them to justify their choices.

This election problem would be palliated if there existed a metric that quantified, for each of the candidate ontologies, how appropriate they are for a new system. To solve this deficiency, the *OntoMetric* method presents a set of processes that the user should carry out to obtain the measures of suitability of existing ontologies, regarding the requirements of a particular system.

2.1.2 Method description

a. *Principles*

The *OntoMetric* method ([73], [72]) helps knowledge engineers to choose the appropriate ontology for a new project; in order to do this, the engineer must compare the importance of the objectives, and study carefully the characteristics of ontologies. The method is based on a group of processes which help to choose the most appropriate ontologies to be reused in a particular project. This method supplies a measure about the suitability of a set of candidate ontologies to be incorporated in a new project. This method can be used to:

1) select the most appropriate ontology among various alternatives or,
2) decide the suitability of a particular ontology for the project.

Further, the *OntoMetric* method is based in the Analythic Hierarchy Process (AHP) [75], a multicriteria decision method, adapting some processes for the reuse of ontologies. The basic decision criteria of the method, so-called dimensions, are the fundamental aspects to be considered by the user before choosing an ontology. The dimensions specify the following features:

- The content represented in the ontology.
- The language in which the ontology is implemented.
- The methodology followed to develop the ontology.
- The software environments used for building the ontology.
- The costs of using the ontology in the system.

Each of these dimensions contains a set of factors which are used to determine the suitability of the ontology regarding the needs of the project. And each factor has a set of characteristics that fixes its value.

The final result of the method is a valuation of the suitability for each of the ontologies taken into account. This result helps the user to make a justified decision about which ontologies are the most appropriate for the application that the user is developing.

OntoMetric gets for every candidate ontology a quantitative measure of its suitability using:

- A multilevel framework of 160 characteristics that describe the ontology domain. This framework provides the outline to represent the information of existing ontologies, and to choose and to compare existing ontologies.

  The multilevel framework of characteristics can be represented like a hierarchical tree, and it has, in the superior level of the taxonomy, five basic aspects on the ontologies, the aforementioned dimensions. Thus, the engineers will be able to extend or to prune the criterion that they considers opportune, so that the new tree depends on the particularities of the project, the business and the organization that will reuse the ontology. This hierarchical tree is called multilevel tree of characteristics (MTC). It should be kept in mind that the framework is subject to the conceptual and technological novelties that will appear in the future in the ontology field. In this sense, the MTC constitutes a set of “living” criteria that should be actualized according to the produced changes.
- The conceptual model of the Reference Ontology [74] and its instances.
- An adaptation of the AHP.

b. *Steps /Process Model*

The *OntoMetric* method is an adaptation of the general steps of AHP to be used in the reuse of ontologies. The main steps in the *OntoMetric* method are the following:

STEP 1: specify the objectives of the project. The engineers should know the exact guidelines of their company and available resources in relation to the new business. They must decide on the importance of the terms of the ontology, the precision of the definitions, the suitability of relations between concepts, the reliability of the methodology used to build the ontology, etc.

STEP 2: build the decision tree from the multilevel tree of characteristics (MTC), so that the objective, "select the most appropriate ontology for a new software project", is placed at the rood node; the dimensions (content, language, methodology, tool and costs) are placed at the first level; the factors of each dimension at the second level; and underneath these factors, the sub-trees of specific characteristics of the particular evaluation project. The general characteristics of all types of ontologies should be specialised according to: the particular ontology, the specific target project and the organization that will develop the project.

STEP 3: for each set of brother nodes, make the pairwise comparison matrixes [75] with the criteria of the decision tree. These comparisons depend on the objectives and aims identified in step 1. The *eigenvectors* are calculated from these matrixes. These weights represent the relative importance between criteria.

STEP 4: for each alternative ontology, assess its characteristics. These values will (always multiplying by the weights calculated in step 3) ascend up to the superior nodes of the tree, until the node root is calculated. For each one of these characteristics, the engineer should establish a scale of appropriate ratings.

> *STEP 4.1*: this method assigns linguistic values (non-numbers) to the alternatives because the human beings, in their daily activities, usually make this type of judgement. It is more intuitive than a numeric scale between zero and ten. In this process, it is important that the groups of the linguistic values are precisely defined.
>
> However it is not possible to perform calculations with linguistic values. One possible representation of these linguistic values is fuzzy intervals. By assigning linguistic values with fuzzy intervals, we are able to perform basic mathematical operations for intervals.
>
> *STEP 4.2*: with these established linguistic scales for each one of the criteria, the engineer will proceed to study each of the ontologies that have been considered as alternatives, and to value them using these scales.

STEP 5: lastly, combine the vectors of weights obtained in step 3 with the values of the alternatives obtained in step 4.

In large projects, which require a team of analysts, each person can provide their own values, and it will be necessary to reach an agreement. In this case, all the steps up to step 4.1 should reach a common consensus among the members of the evaluation team. Later, each analyst can value each one of the candidate ontologies in an individual way. Finally, the suitable ontology is chosen based on the results obtained.

2.1.3 Usefulness / Relevance for practice

In recent years, the development of ontology-based applications has increased considerably, mainly related to the Semantic Web. In spite of the great increase that the use of ontologies has acquired, nowadays knowledge engineers need to look for ontologies disperse in web servers in order to incorporate them into their systems. When they find several ontologies that can be adapted, they must examine their characteristics attentively and decide which are the best are for reuse. This selection procedure usually depends on the knowledge engineers´ experience and intuition. When the system is being developed with commercial and industrial goals, it is very difficult for the engineers to justify the selection they have made.

Feedback from expert users have used the method reveals that specifying the characteristics of a certain ontology is complicated and takes time, and its assessment is quite subjective; however, they state that, once the framework has been defined and if it is applied to one particular type of ontology, the *OntoMetric* method helps to justify decisions taken, to “clarify ideas”, and to weigh up the advantages and the risks involved in choosing one ontology from other options.

Although the specialisation of the characteristics and the assessment of the criteria of a particular ontology require considerable effort, the *OntoMetric* method provides a useful schema to carry out complex multi-criteria decision making. This method helps the knowledge engineer to make a justified decision about which ontologies are the most appropriate for the application that the engineer is developing.

2.1.4 Conclusion

The *OntoMetric* method is an adaptation of the Analythic Hierarchy Process (AHP) to help knowledge engineers to choose the appropriate ontology for a new project. In order to do this, the engineer must compare the importance of the project objectives, and study carefully the characteristics of ontologies. This method helps to justify decisions taken, to “clarify ideas”, and to weigh up the advantages and the risks involved in choosing one ontology from other options.

2.2 Natural Language Application metrics

2.2.1 Abstract

In this section, we describe methodologies for evaluating the content of ontologies with respect to natural language applications. Natural Language (NL) methods can be used for both ontology population and semantic metadata creation. The first involves populating an ontology of concepts with instances drawn from textual data; the second involves associating the text with the correct concepts in the ontology by means of associating mentions in the text with instances in that ontology. The two tasks often involve very similar methods, however in this deliverable we shall restrict ourselves to discussion of the former, since it is the content of the ontologies that we are interested in here. Ontology population (i.e. adding instances to the concepts in an ontology) is a useful task because it can be very time consuming to carry out manually, and because ontologies for use in industry are often required to be specific to a domain or application of particular interest. This means that each ontology, and its contents, must be tailored to the needs of the user. It is important to be able to evaluate how well the ontology has been populated. This could be done from several points of view: for example, some people might be interested only in whether all the instances in the ontology are correctly positioned, but not if some things are missing, whereas others might be more interested in having as much information in the ontology as possible, even if some of it is not correctly positioned.

Natural language applications involving ontologies are a relatively new area of research, and while they are mainly derived from applications (and their respective evaluation methods) with a long history, methods for evaluation of such technologies are currently at the forefront of research and there are no well accepted standards as yet. We outline here some current proposals and examine to what extent existing methodologies can be reused in the context of ontology content evaluation.

2.2.2 Method description

In this section we present some explanation of the metrics required to evaluate the ontology population task. The task involves taking an ontology of concepts and a set of texts, and populating the ontology with instances from the texts. The gold standard in this case is an ontology with all the instances added. Here a metric is needed to compare the overlap between the automatically populated ontology and the gold standard one. The metric needs to measure how well the system has detected the presence of new instances in the text and added them to the ontology in the correct place (i.e. as instances of the right concept). Such a metric could be derived from one of the many algorithms for computing similarity between two ontologies (the automatically populated one and the gold standard one), for example the semantic comparison approach used by [15].

2.2.2.1 Precision and Recall Metrics

The simplest method of evaluation of the ontology population task described in the previous section is based on precision and recall. These are typically used in IE evaluations such as MUC (Message Understanding Conferences) [43] and CONLL[46][47]. Because much of the research in Information Extraction (IE) in the last decade has been connected with these competitions, the MUC evaluation metrics of precision, recall and F-measure [45] have been the most widely used in this field, albeit with slight variations from time to time. These metrics have a very long-standing tradition in the field of Information Retrieval.

Precision measures the number of correctly identified items as a percentage of the number of items identified. In other words, it measures how many of the items that the system identified were actually correct, regardless of whether it also failed to retrieve correct items. The higher the precision, the better the system is at ensuring that what has been identified is correct.

Error rate is the inverse of precision, and measures the number of incorrectly identified items as a percentage of the items identified. It is sometimes used as an alternative to precision.

Recall measures the number of correctly identified items as a percentage of the total number of correct items. In other words, it measures how many of the items that should have been identified actually were identified, regardless of how many spurious identifications were made. The higher the recall rate, the better the system is at not missing correct items.

Clearly, there must be a tradeoff between precision and recall, for a system can easily be made to achieve 100% precision by identifying nothing (and so making no mistakes in what it identifies), or 100% recall by identifying everything (and so not missing anything). The F-measure [56] is often used in conjunction with Precision and Recall, as a weighted average of the two. if the weight is set to 0.5, precision and recall are deemed equally important.

The precision, recall and F-measure metrics are defined as follows:

$$Precision = \frac{Correct + 1/2Partial}{Correct + Spurious + 1/2Partial}$$

$$Recall = \frac{Correct + 1/2Partial}{Correct + Missing + 1/2Partial}$$

$$F - measure = \frac{(\beta^2 + 1)P * R}{(\beta^2 R) + P}$$

where $\beta$is a value between 0 and 1 reflecting the weighting of P vs. R. If $\beta$is set to 0.5, the two are weighted equally.

$$FalsePositive = \frac{Spurious}{c}$$

where c is some constant independent from document richness, e.g. the number of tokens or sentences in the document.

Note that we consider annotations to be partially correct if the entity type is correct and the spans are overlapping but not identical. Partially correct responses are normally allocated a half weight.

False positives are also a useful metric when dealing with a wide variety of text types, because it is not dependent on relative document richness. By this we mean the relative number of entities of each type to be found in a set of documents.

When comparing different systems on the same document set, relative document richness is unimportant, because it is equal for all systems. When comparing a single system’s performance on different documents, however, it is much more crucial, because if a particular document type has a significantly different number of any type of entity, the results for that entity type can become skewed. Compare the impact on precision of one error where the total number of correct entities = 1, and one error where the total = 100. Assuming the document length is the same, then the false positive score for each text, on the other hand, should be identical.

2.2.2.2 Cost-Based Evaluation Metric

Evaluation mechanisms in Information Extraction (IE) and related tasks such as ontology population can also be affected by the notion of r*elative document richness*, i.e. the relative number of entities of each type to be found in a set of documents. For this reason, error rate (the number of wrongly identified answers divided by some fixed criteria such as document length) is sometimes preferred in the IE field, because, unlike precision, it is not dependent on relative document richness.

Using error rate instead of precision and recall means, however, that the F-measure can no longer be used. An alternative method of getting a single bottom-line number to measure performance is the cost-based evaluation (CBE) metric. This is a favorite with the DARPA competitions, such as TDT [49], and is the method used in the ACE (Automatic Content Evaluation) competitions [42]. The model stems from the field of economics, where the standard model ``Time Saved Times Salary" measures the use of the direct salary cost to an organisation as a measure of the value.

One of the main advantages of this method is that it enables the evaluation to be adapted depending on the user's requirements, and so is particularly suitable for use in industry. A CBE model characterises the performance in terms of the cost of the errors (or the value of the correct things, depending on whether you see the glass as half-empty or half-full). For any application, the relevant cost model is applied, and expected prior target statistics are defined.

For a cost-based error model, a cost would typically be associated with a miss and a false alarm (spurious answer), and with each category of result (e.g. recognising Person might be more important then recognising Date correctly). Expected costs of error would typically be based on probability (using a test corpus). This makes the assumption that a suitable test corpus is available, which has the same rate of entity occurrence (or is similar in content) to the evaluation corpus. If necessary, the final score can be normalised to produce a figure between 0 and 1, where 1 is a perfect score.

In the ACE evaluations, the systems are evaluated in terms of false alarms and misses, which are combined to form a (normalised) **value**. A value of 0 means the system achieved nothing; a negative value means the system did worse than not attempting the task at all; a score of 100 means a perfect system.

Further details and formulae for the ACE cost-based metrics used for entity detection and relation detection can be found at ftp://jaguar.ncsl.nist.gov/ace/doc/ACE-EvalPlan-2002-v06.pdf, and an improved version at ftp://jaguar.ncsl.nist.gov/ace/doc/ace_evalplan-2003.v1.pdf.

2.2.2.3 Ontology fit – the tennis measure

This method, proposed by [44], uses a vector-space model of instances (terms) in a corpus and an ontology to give a measure of the "fit" between the ontology and the corpus (domain of knowledge). The standard method of evaluating an ontology, by comparing it with a gold standard ontology (e.g. [50],[15]), suffers from the problem that when the ontologies differ, it is not clear whether this is because the generated ontology is wrong, simply different (but still correct), if the methodology is flawed, or if the corpus is inappropriate. For semantic web technologies, Brewster suggests that it is better to choose out of a set of x possible ontologies the most appropriate one for the domain and/or application.

There are 2 possible approaches to comparing an ontology with a corpus. The first is to perform IE on the corpus and compare the overlap between the extracted instances and those in the populated ontology (as described in Section 2.7.5 above). The second is the tennis measure [55], which evaluates the extent to which items in the same cluster are closer together in the ontology than those in different clusters. The "tennis problem" was noted by [48] as a phenomenon occurring in WordNet where related words could occur in two completely different parts of the ontology with no apparent link between then, e.g. "ball boy" could occur as a descendant of "male child" and tennis ball as a descendant of "game equipment", despite an obvious semantic relation. The idea behind the tennis measure is that it would deal with two ontologies with identical concept sets, but which have the concepts differently organised, i.e. at a different distance from each other.

2.2.2.4 Lexical Comparison Level measure

This measure, proposed by [15], is used to compare the contents of two ontologies without considering their conceptual structure, in a direct attempt to combat the problem of precision and recall and their restrictive binary nature. The measure is based on the edit distance of [51], a long established and well known method for comparing two strings by measuring the minimum number of insertions, deletions and substitutions needed to transform one string into the other. Maedche's string matching measure compares two lexical entries using the edit distance, and returns a similarity score between 0 and 1. These scores are summarised for all the instances of a concept in the hierarchy, and averaged over the whole ontology, using an aymmetric measure which determines the extent to which the lexical level of the gold standard ontology is covered by the lexical level of the system generated ontology. It is useful

because it diminishes the importance of trivial factors such as hyphenation, capitalisation, alternative spellings etc. Maedche et al. claim that although this means that similar strings with no relation (e.g. power and tower) can give deceptive results (since they are accorded a falsely high similarity), in practice this happens rarely and can be largely ignored.
Cimiano et al. extend this idea further by also considering lexical similarity as well as taxonomic similarity, in the same vein. This is measured by considering the recall of one lexicon compared to that of the gold standard.

2.2.3 Usefulness / Relevance for practice

In terms of suitable evaluation models for use in industry, all the methods described are potential candidates. To a certain extent, the choice of model depends on the resources available and the kind of evaluation required (as mentioned earlier in Section 2.2.1). In this section we discuss the relevance and usefulness of the models described.

The precision/recall model is the most well known and widely used evaluation metric in the IE community, and is the easiest to use, understand and to develop tools for (for example the GATE Corpus Benchmark Tool [101]). For example, a Precision of 90% is intuitively easy to understand, but an ACE value of 20 is rather harder to interpret. However, this does not mean that it is necessarily the most appropriate way of evaluating the ontology content, because of a number of problems. As discussed earlier, there is no one perfect solution for the best method of evaluation. In some circumstances, the problem of misclassification, as detailed below, could be an important factor; in other cases, the user might not care about such problems. We therefore discuss below some of the factors which could influence a user’s choice of method.

The Precision/Recall method suffers from the problem of double penalization of misclassification. For example, if the system wrongly classifies a Person as a Location, it suffers both Precision and Recall penalties as there is both a missing Person and a spurious Location. Taking this a stage further, we can extend the problem to all metrics which use a binary kind of evaluation (where something is either right or wrong). The cost-based evaluation, on the other hand, is designed to prevent this in a flat structure, by assigning different weights to different errors, i.e. the score is scalar rather than binary.

There are two particular cases in which we might wish to have a more scalar approach to evaluation in order to deal with misclassification. The first is connected with the fact that a mistake at the top of the ontology is intuitively worse than one nearer the bottom. For example, misclassifying an orange as a vegetable rather than a fruit is less wrong than misclassifying an orange as a animate rather than a inanimate object. The second case is connected with the fact that the extent of the error should be proportional to the distance between the correct and submitted response. This “semantic distance”, as it is commonly known, is taken into account by many traditional similarity measures [42,55]. While these measures typically measure the similarity between two concepts or instances in the same ontology, they could equally be used to compare concepts or instances in two separate ontologies, as in our evaluation scenario. To deal with the misclassification problem, we can use a metric which assigns the weight of the error according to the similarity of the given and target responses, for example using both position and commonality measures, as above. Note that this also covers the problem of assigning something to the correct type but the wrong level of generality, e.g. assigning a banana as a kind of food but not as a fruit. This would get a higher score than assigning it as e.g. a vehicle, because the distance between two is less.

Other methods of dealing with misclassification include the CBE approach as used in ACE. Here there is no full ontology, but there are two levels of classification: a class and subclass. The subclass is annotated as a feature, and the system can then be scored taking into account features or not (and assigning different features different values), and also just scoring features independently. Note that this is just one example of how the CBE approach was used. The approach could equally be extended to a full ontology, whereby the level of classification could be scored separately (or included/excluded as part of the overall score).

Another solution is to extend the definition of a partial match to cover annotations with subsuming classes. Then depending on the distance in the ontology between the target class and the given class, a progressively diminishing weight could be assigned. However, this does not account for the problem that different parts of the ontology tend to be more or less heavily populated with concepts, so depth between concepts is not really uniform across the whole ontology. Again, this might or might not be an issue, depending on the nature of the ontology.

One problem with the CBE metric is that it contains many different weights, which are assigned by the end-user, and it is not easy to decide on appropriate weights or to find a way to calculate these automatically, again since there are few precedents. However, this could be rectified by making all weights the same initially, and including a distance-based metric for ensuring that partially correct items which are assigned a tag at the wrong level of the ontology are penalised appropriately according to the distance.

The CBE metric is designed specifically for different applications or different users, who might have different requirements of a system. For example, one user might be more concerned with precision than recall, or one user might be more concerned about getting particular types of entities right, and not so concerned about other types, or one user might be more concerned with the fact that getting something partially right is important. So a cost-based model is useful because it enables the parameters to be modified according to the particular evaluation or task. evaluation or task. If this model were to be adopted as a standard for ontology content evaluation, we would have to devise:

- some simple and heuristic method of weight assignment;
- some scoring tool, with the ability to be adapted easily by the user to reflect changes to the weights,
- some fairly generic set of weights that could be used as a default.

This is not a problem, just a requirement if the model is to be adopted.

Cimiano et al [100] apply the lexical comparison measure to evaluate the performance of a method to acquire taxonomies automatically from text, comparing a machine-built taxonomy with a hand-crafted one. They use this measure because it is the only available method they know of for comparing two ontologies. However, the task for which they make use of the measure is slightly different from the task of ontology population we describe in this section, in that Cimiano et al. wanted to measure how well the ontology is constructed, rather than how well it is populated, given an existing ontology structure. It is possible that the measure could be adapted to the needs of this task, but it is not clear at the moment how suitable this would be.

Finally, we should consider also the use of a multi-dimensional evaluation, where a single score is not generated, but instead the evaluation is carried out simultaneously along several axes. This is pursued in the ACE evaluations, where although a single score is generated, many other aspects are also evaluated separately. Olsson et al. [52] evaluate the performance of protein name taggers in this way to overcome the limitations of Precision and Recall being too inflexible. They propose measures such as “Sloppy” where if an answer is partially correct (i.e. there is any overlap between the system and key response), it is classified as Correct, “Left Boundary” where if the left boundaries of the system and key responses match, it is classified as correct, “Right Boundary” and so on.

2.2.4 Case study

The h-TechSight Knowledge Management Portal (KMP) [99] enables support for knowledge-intensive industries in monitoring information resources on the Web, as an important factor in business competitiveness. The portal contains tools for identification of concepts and terms from an ontology relevant to the user's interests, and enables the user to monitor them over time. It also contains tools for ontology management and modification, based on the results of targeted knowledge extraction from the web. By monitoring the instances of concepts from the ontology in which they are interested, businesses can keep track of trends and topics of interest in their field, and be alerted to changes.

One particular application of this KMP is the employment portal. This contains an ontology containing information about the chemical engineering domain and employment, for example job types, qualifications, skills required, etc. The employment portal was tested by IChemE (Institution of Chemical Engineers) to see how it could help gain more insight about employment activities in their field. IChemE is a leading international body, providing services for and representing the interests of those involved in chemical, biochemical and process engineering world-wide. The portal provided them with information about instances found in chemical engineering job advertisements of concepts of interest, such as skills required, remuneration, benefits etc. The instances were used to populate an ontology so that they could be monitored over time and reused in the future.

In many cases it was clear to the user when the system had returned results that were incorrectly identified, for example if the system returned “£2000” as belonging to the concept Skill instead of to the concept Salary. But it was not clear whether the system had missed instances which it should have found, because there was no way to tell this from looking at the populated ontology. There were also cases where mistakes could have been made by the system without the user realizing, for example “2000” could have been returned as a Salary when it actually referred to the year. So a means of evaluation was necessary for the users to know how good the system was and whether it was really helping them with their search. For this evaluation, the Precision and Recall metrics were used, because the main things that the users were interested in were whether things were missing or not, and whether things had been wrongly identified. If something had been identified correctly but wrongly categorized, to these users it was just as bad as returning something totally incorrect, so the CBE method of evaluation was not really necessary. In a sanple evaluation, the system achieved an average of 97% Precision and 91.5% Recall. This enabled the user to be able to tell very easily how well the system was doing, because the figures are quite understandable. The results could also be broken down further into scores for each concept in the hierarchy.

2.2.5 Conclusion

The evaluation methods described above are particularly useful for industry, because they are aimed at comparing different systems with each other, rather than simply comparing a single system with a gold standard solution. This makes it easier to tell which system is better at which aspects. Typically each system will be good in some aspects and poor in others, e.g.

they may be fast and easily configurable, but less accurate. Ultimately a graphical representation of evaluation results would be useful here, in order to compare different features of the systems more easily; however, discussion of this is beyond the scope of this report. Which notion of "correctness" to use when describing the overall performance of a system depends to a large extent on the environment in which it is to be used. For example, in a query-based system, more inexact responses may be completely acceptable, because the user still has some input, and use of wildcards may be a possibility. For an information extraction system where there is no user interaction, a more exact match may be necessary. The CBE metric and multi-dimensional evaluation approaches are particularly useful in industry, where not only will systems be used in different environments, but also by users with differing levels of expertise, different expectations, and for different applications and domains.

2.3 OntoClean

2.3.1 Abstract

Formal ontology evaluations such as those proposed by the OntoClean methodology are presented within this section. The OntoClean methodology is based on philosophical notions for a formal evaluation of taxonomical structures. It focuses on the cleaning of taxonomies and is currently being applied for cleaning the upper level of the WordNet taxonomy (cf. [21]). Core to the methodology are the four fundamental ontological notions of *rigidity*, *unity*, *identity*, and *dependence*. By attaching them as meta-relations to concepts in a taxonomy they are used to represent the behaviour of the concepts.

2.3.2 Method description

The following building blocks constitute the basic infrastructure for implementing OntoClean: (i) a set of axioms that formalize definitions, constraints and guidelines given in OntoClean and (ii) a "meta-ontology", *viz*. the so-called "taxonomy of properties", that provides a frame of reference for evaluations. An ontology can be compared with a predefined ideal taxonomical structure to detect inconsistencies. Thus, the integration of the OntoClean methodology into OEEs enables an integrated quality control for ontologies.

We briefly sketch the methodology in a simplified way and mention two of the introduced philosophical notions, viz. *rigidity* and *unity*:

- **Rigidity** is defined based on the idea of essence. A *property* is essential to an individual if and only if it necessarily holds for that individual. Thus, a *property* is rigid (+R) if and only if it is necessarily essential to all its instances. A *property* is non-rigid (-R) if and only if it is not essential to some of its instances, and anti-rigid if and only if it is not essential to all its instances.

  Example: Consider for example the property of *being hard*. We may say that it is an essential property of hammers, but not of sponges. Some sponges (dry ones) are hard, and some particular sponge may be hard for its entire existence, however this does not make being hard an essential property of that sponge. The fact is that it *could have* been soft at some time, it just happened that it never was.

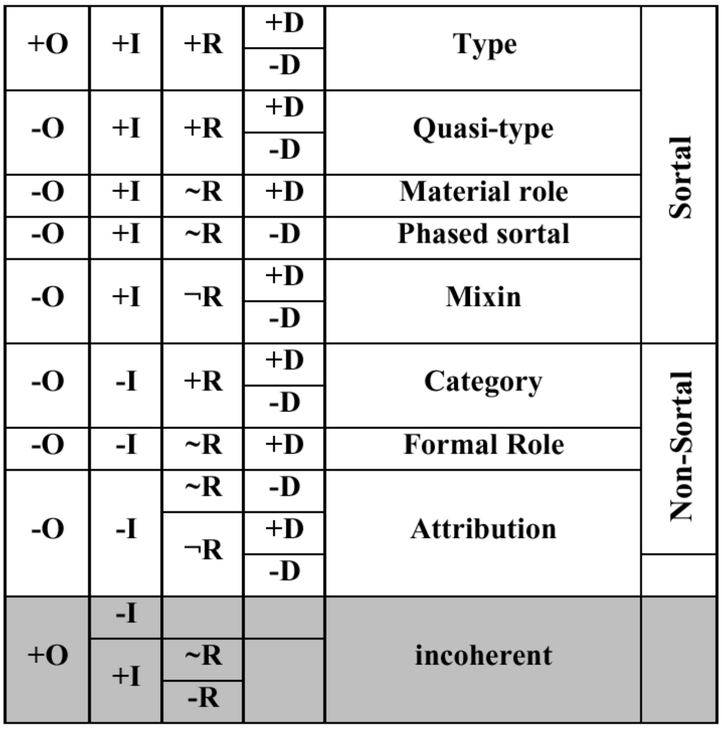
  Furthermore, *being a person* is usually conceptualized as rigid, while, as shown above, *being hard* is not. Rigidity is a subtle notion: every entity that *can exhibit* the property *must* exhibit it. So, every entity that is a person must be a person, and there are no entities that can be a person but aren't.

The property *being a student* is typically anti-rigid -- *every* instance of student is not essentially a student (i.e. may also be a non-student).

- **Unity** is defined by saying that an individual is a whole if and only if it is made by a set of parts unified by a relation *R*. A *property P* is said to carry unity (+U) if there is a common unifying relation R such that all the instances of *P* are wholes under *R*. A *property* carries anti-unity if all its instances can possibly be non-wholes.

  Example: The enterprise British Airways is a whole unified by the relation *has president*. To generalize, an *enterprise with president* carries unity since the relation *has president* is the relation that unifies every instance.

  Based on these meta-relations OntoClean classifies concepts into categories as shown in `Figure 11` (the figure is taken from [22]). E.g., a concept that is tagged with "+O +I +R" is called a "Type".



+O	**+I**	**+R**	**+D**	**Type**	**Sortal**
			-D		
-O	**+I**	**+R**	**+D**	**Quasi-type**	
			-D		
-O	**+I**	**~R**	**+D**	**Material role**	
-O	**+I**	**~R**	**-D**	**Phased sortal**	
-O	**+I**	**¬R**	**+D**	**Mixin**	
			-D		
-O	**-I**	**+R**	**+D**	**Category**	**Non-Sortal**
			-D		
-O	**-I**	**~R**	**+D**	**Formal Role**	
-O	**-I**	**~R**	**-D**	**Attribution**	
		¬R	**+D**		
			-D		
+O	**-I**			**incoherent**	
	+I	**~R**			
		-R			

Figure 11: Combinations of OntoClean meta-relations

The aim of the methodology is to produce a "clean" taxonomy as shown in the ideal structure in Figure `Figure 22` (figure is taken from [22]).

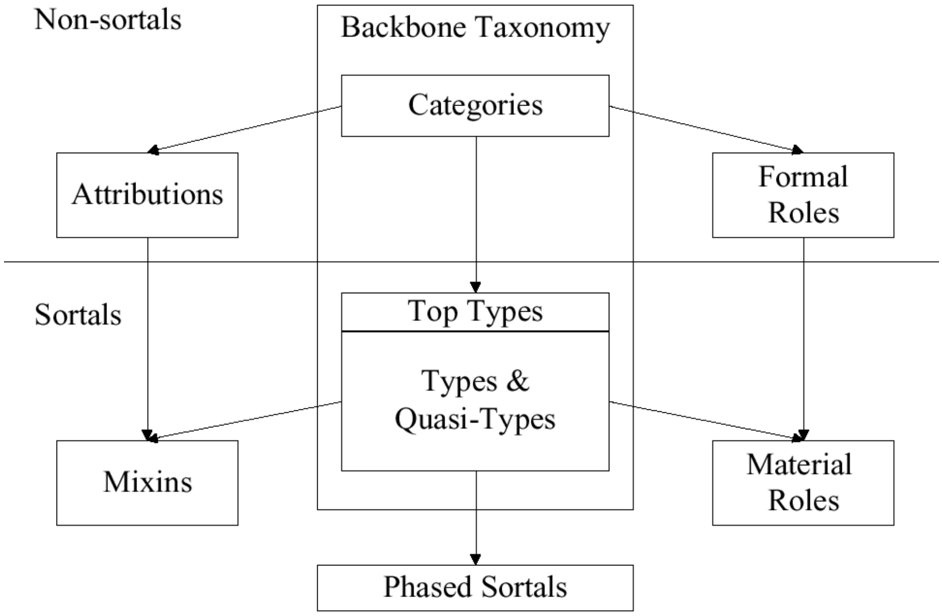


Figure 22: Ideal taxonomy structure

Beside these meta-relations OntoClean contains axioms that can be applied to evaluate the correctness of a given taxonomy. For instance, an axiom suggested in OntoClean is “a property carrying anti-unity has to be disjoint of a property carrying unity”. As a consequence, “a property carrying unity cannot be a subclass of a property carrying anti-unity” and “a rigid property and an anti-rigid property are ever disjoint”, to name but a few. As an example we present the formalization of the disjointness in
F-Logic:

```
FORALL C check("A property cannot carry +R and -R",C)
    <- C[carryR->>"true"]
    AND C[carryNotR->>"true"].
```

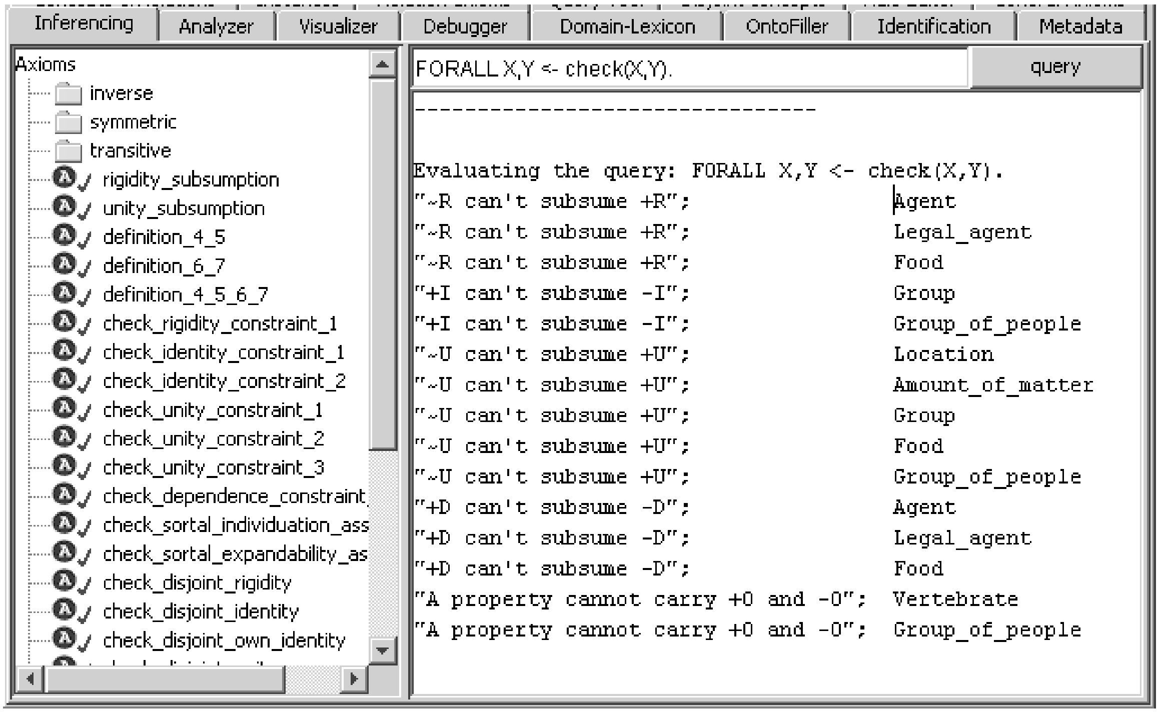
Another example is, that a property that is defined as “anti-rigid” cannot subsume a property that is “rigid” (the heck message is abbreviated for means of simplicity):

```
FORALL B ( check("~R can't subsume +R",B) )
    <- EXISTS C
    C::B
    AND B[antiR->>"true"]
    AND C[carryR->>"true"].
```

We modelled the “meta ontology” and an example ontology (taken from [22]) that has to be evaluated, in OntoEdit. Each concept of the example ontology, i.e. all subconcepts of the root concept of the example ontology, viz. “Entity”, is then specified as being also an instance of the top-level concept “Property” of the meta ontology through an axiom:

```
FORALL C C:Property
    <- C::Entity.
```

Figure 33 shows the inconsistencies derived from applying the OntoClean axioms to the example ontology by using the Inferencing plugin. On the left side the list of implemented axioms is shown, for testing purposes they can be switched on and off. On the right side the result from an evaluation is shown, e.g. the concept *Agent* is defined as "anti-rigid" and subsumes the concept *Animal* which is defined as "rigid". According to the OntoClean methodology this is a violation of a given constraint. To enhance the quality of the taxonomical structure an ontology engineer can now reconsider the modelled hierarchy.



```
FORALL X,Y <- check(X,Y).
```

```
-------------------------------


Evaluating the query: FORALL X,Y <- check(X,Y).
"~R can't subsume +R";                Agent
"~R can't subsume +R";                Legal_agent
"~R can't subsume +R";                Food
"+I can't subsume -I";                Group
"+I can't subsume -I";                Group_of_people
"~U can't subsume +U";                Location
"~U can't subsume +U";                Amount_of_matter
"~U can't subsume +U";                Group
"~U can't subsume +U";                Food
"~U can't subsume +U";                Group_of_people
"+D can't subsume -D";                Agent
"+D can't subsume -D";                Legal_agent
"+D can't subsume -D";                Food
"A property cannot carry +O and -O";  Vertebrate
"A property cannot carry +O and -O";  Group_of_people
```

Figure 33: Deriving inconsistencies with the OntoClean Plugin

2.3.3 Usefulness / Relevance for practice

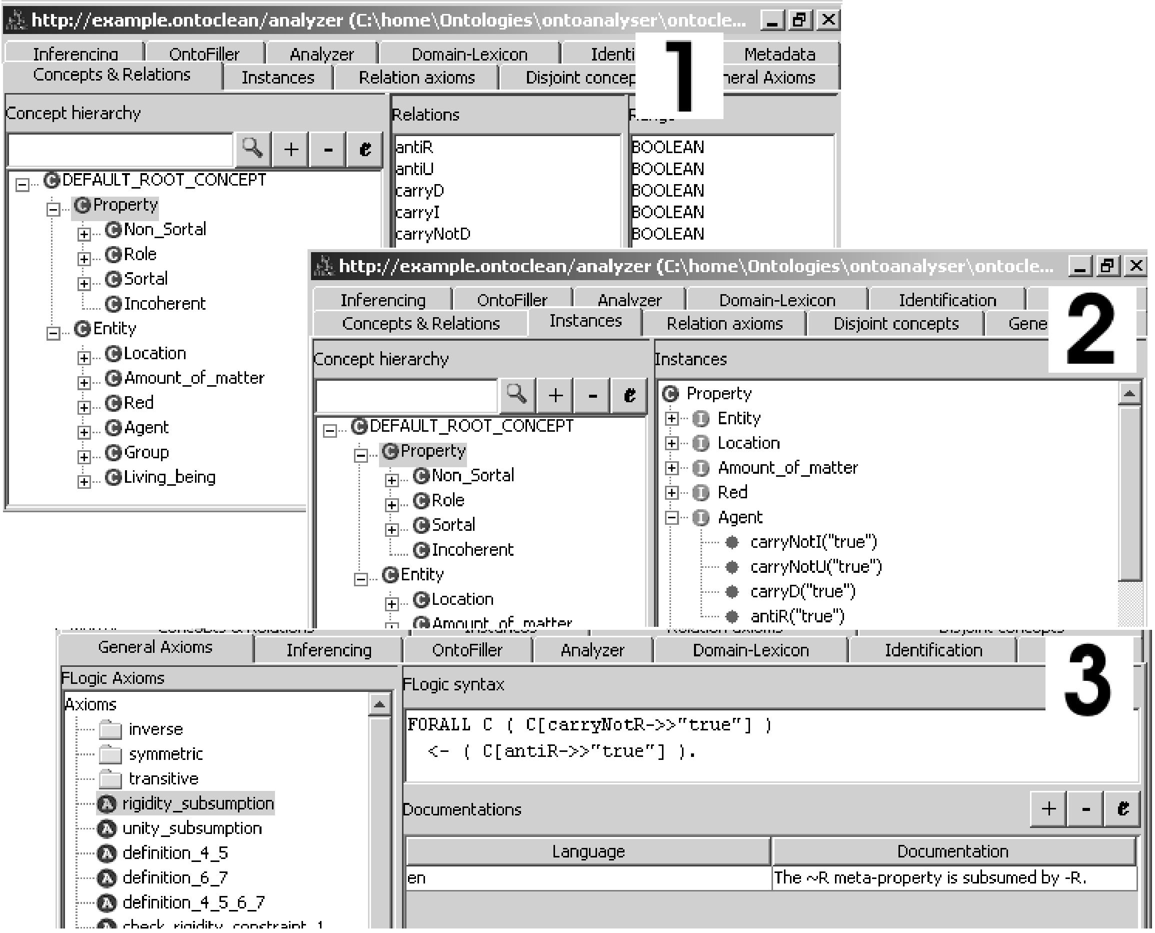
Generally, formal ontology evaluation provides useful insights into semantic models. However, these insights are more structural and formally driven and do not allow to infer anything about the usability of an analysed ontology.

The usage of the presented OntoClean method for industry is limited. At least, it might be used as a supporting method to ensure formal correctness where needed but this would assume that the method can be applied by "end-users".

2.3.4 Case Study

To analyse practical relevance and possible applications in industry we describe a technical showcase of how to apply the OntoClean methodology within industrial applications, indeed the engineering tool OntoEdit.

OntoEdit[1] is an ontology engineering environment which allows for inspecting, browsing, codifying and modifying ontologies. Modeling ontologies using OntoEdit means modelling at a conceptual level, viz. (i) as much as possible independent of a concrete representation language, (ii) using graphical user interfaces (GUI) to represent views on conceptual structures conceptual structures, i.e. concepts ordered in a concept hierarchy, relations with domain and range, instances and axioms, rather than codifying conceptual structures in ASCII. To implement the OntoClean methodology (as presented in section 2.3) in OntoEdit, we (i) formalized the constraints and definitions as axioms, and (ii) formalized the meta-relations and classifications as a "meta ontology" that can be used to classify concepts of an ontology. Figure 44 shows the subsequent steps during implementation and employment of OntoClean as a plugin in OntoEdit (the numbers in the figure correspond to the following enumeration):



Relations	Range
antiR	BOOLEAN
antiU	BOOLEAN
carryD	BOOLEAN
carryI	BOOLEAN
carryNotD	BOOLEAN

```
FORALL C ( C[carryNotR->>"true"] )
  <- ( C[antiR->>"true"] ).
```

Language	Documentation
en	The ~R meta-property is subsumed by -R.

Figure 44: Implementation of OntoClean in OntoEdit

1. model both ontologies, the taxonomy of properties and the example ontology,
2. fill the meta-relations with values (i.e. tag the concepts of the example ontology with "carryR" (+R) etc.), and
3. specify the definitions and constraints from OntoClean as axioms (here by using the General Axiom Editor). One can now ask queries to find inconsistencies in an ontology according to the OntoClean methodology.

[1] Can be obtained from http://ww.ontoprise.de

OntoClean in general or as implemented in OntoEdit requires well-trained users and assumes expert knowledge about ontological engineering which makes it difficult for use by industrial end-users.

2.3.5 Conclusion

The implementation shown is a first proof of concept. The next version of the plugin encapsules the meta ontology by using a dynamically built GUI to handle the tagging of concepts with meta-relations more intuitively. The results should automatically guide users through a set of possible actions that can be performed to fix the detected inconsistencies.

However, the application of the OntoClean methodology requires significant training, as only few people are currently able to apply it properly. In collaboration with the group from the Artificial Intelligence Laboratory of the Technical University of Madrid (UPM) and the inventors of the methodology, we are therefore planning to implement a more user-friendly and intuitive solution.

The relevance for industry appears low. In domains in which formal correctness is required, OntoClean might be used to support the process of ontology evaluation additionally.

2.4 EvaLexon

2.4.1 Abstract goals / purpose

This evaluation method is meant to be applied on the results of automatic ontology mining techniques. The aim of the mining process is to create ontologies, and not to populate ontologies with instances. The method stays at the linguistic level, as the mining results are words rather than concepts. The method is applied in the context of the VUB DOGMA ontology framework [65] and the UA-CNTS unsupervised miner [58], which results in the miner producing triples (or lexons in DOGMA terminology) [64]. The triples are easily convertible into RDF/OWL-triples, so that the evaluation procedure can be applied to existing ontologies as well. The method is still in the process of being refined, although first results have already been reported [63][66]. Other text miners (see [60] for an overview of miners of ontologies from text) can be used as well. Some other methods to evaluate text-based ontology learning and population methods are presented in [57].

The purpose of this method is to be simple to understand by laymen, objective, automatable, and easily applicable to any text that describes an application domain and that serves as input for the ontology mining/creation process. The reference point for the evaluation is the text itself, as otherwise one has to assume the existence of a gold standard ontology. This is generally not the case. It is not the aim to provide very sophisticated evaluation results, but rather a rough but good enough reference to determine whether or not the results of ontology mining capture most of the notions of the input text.

2.4.2 Method description

2.4.2.1 Principles

The main principle is to compare the vocabulary of the triples mined with the input text as such and with a set of words considered to be relevant for that text. It is clear that this is only an approximation (as words do not equal concepts), but this constitutes the simplicity of the procedure.

Two basic scientific insights are used:

- A corollary of Zipf's law [69] (roughly summarised by "the high frequency words are meaningless")
- A statistical formula to compare two proportions as applied in corpus linguistics (e.g. [59]).

Four measures have been defined that should express how well the resulting triples lexically represent the important notions of the application domain:

- recall and precision (the classical IE measures) using a set of relevant words derived from the text
- coverage and accuracy (derived measures) using the text itself

2.4.2.2 Steps

The (technical) input text is processed in order to derive a frequency list and related text statistics about relative frequencies of frequency classes. Similar frequency list and related statistics have been produced for a general corpus, in this case the Wall Street Journal. The two texts are compared (or rather the distribution of their vocabulary) and a list of statistically significant words that characterise the technical text is collected. This list (also called "relevant words") is used for almost all the subsequent computations.

- *Coverage* is computed as the average of the overlap between the vocabulary of the triples and the input text for each frequency class.
- *Accuracy* is computed as the average of the coverage of those frequency classes that include at least 60% of relevant words.
- *Precision* is computed as the intersection of the words of the triples and the related words divided by the number of words of the triples.
- *Recall* is computed as the intersection of the words of the triples and the related words divided by the number of relevant words

2.4.2.3 How to apply

Several scripts and Dos/Linux commands are applied in sequence. The first series are Perl scripts that produce the frequency list and related text statistics. The second series are Tawk scripts (a commercial version of Awk [68]) that create the set of relevant words. The third series are also Tawk scripts that compute the actual values for the four metrics. All the intermediate results are tab or space delimited ASCII files, which facilitates processing, e.g., some intermediate results are re-ordered using the classical "sort" command. The Tawk scripts are compiled for a Windows machine, but using the appropriate compiler they can run on a Unix/Linux machine as well.

2.4.3 Usefulness/Relevance for practice

Only few evaluation methods for ontology mining and learning exist [57] and these require an important human intervention. The fundamental problem is the point of reference, or stated otherwise, determining a gold standard. As ontology mining is becoming more popular (to circumvent the knowledge acquisition bottleneck), largely automated and highly objective methods to evaluate mining outcomes or modelling results are particularly useful, especially in an industrial environment.

In principle, this method can also be applied to an existing ontology, i.e. an ontology in a more or less final state, to compare it with its source text(s). To the extent that an existing ontology, encoded in RDF or OWL triples, uses language terms to denote concepts, this method can be applied for type 3 evaluations (as presented in the overall introduction of this deliverable). However, the main application as presented is the modelling phase (type 1). Some of the principles of our method resemble a lexical comparison method for ontologies presented in [61], but the second element of the comparison here is a textual source instead of another ontology. Unfortunately, our method has currently not yet been applied in this sense.

2.4.3.1 Why is it useful for industry

Industry, almost by definition and in contrast to academia, must have performance measures for the products they are building. In order to stay ahead of the competition, a company constantly tries to find ways to improve its products. Benchmarking their products with competing ones is but one manner. Therefore, companies that are incorporating (or willing to incorporate) ontologies (be it as a result of text mining or of human modelling) need metrics to measure and compare results. The measures themselves are maybe even less important than the agreement on the common use of a measure in order to create benchmark tests.

Another important concern for industry is quality assurance. A classical way of checking the (improving) quality of a product is by performing regression tests. Several experiments, varying on a specific point one from one another, are run and the results of the runs are compared (in the most ideal case using statistics). Companies that deliver ontologies to customers definitely want to test internally various versions of an ontology before delivery. A Professional Services group (i.e. delivering customised end-solutions) of an IT-company embracing semantic web technology can use this evaluation procedure to tune ontologies for a specific customer by performing regressions tests.

Even for customers, evaluation measures are important. It is not unusual for companies to deliver (and customers to request) a certificate stating that a product, in this case an ontology, complies with certain criteria. As one of the most interesting criteria (at least from the point of view of the customer) is the degree to which the ontology covers the knowledge of an application domain of a customer, the measures proposed above can be very helpful. One can easily imagine a scenario where a company delivers, together with the ontology produced, two sets of metrics: one that shows how well a “sample ontology” built as the result of a certain methodology (human modelling or automated mining) “represents” a commonly accepted reference corpus, and one that details how the ontology actually ordered “corresponds” with the input text(s) of the customer’s application domain. A similar scenario (for speech recognition performance) exists for quite some time in the speech processing industry.

Having this kind of documentation on the ontology modelling methodology and the associated results can be crucial if a company wants obtain the ISO9001 certification or formalise its internal quality assurance and product delivery and/or acquisition procedures (see the overall introduction on the evaluation scenarios).

2.4.3.2 Case study

2.4.3.2.1 Background

The use case we are currently working on involves a Belgian company that specialises in VAT legislation (EU directive 77/388/EEC of 27 January 2001). The company “VAT@” provides consultancy and tools to its customers, who are international trading companies, to fill in the appropriate set of VAT documents in the required way. The problem for any company that imports or exports goods is that the EU directive is implemented at a national level in (more or less) different legislation characterised by idiosyncratic vocabulary and slightly different notions. Therefore, the company wants to integrate a VAT ontology in their applications to bridge these conceptual differences, to facilitate maintainability (when the directive is updated) and enhance interoperability with third-party applications (e.g. national e-government VAT applications). Research on these activities forms part of the Flemish IWT OntoBasis project (IWT GBOU 2001 #10069).

2.4.3.2.2 Results

The English EU directive has been mined by the CNTS unsupervised miner and triples have been produced. The evaluation methodology is still in an early stage, although intermediate results have already been published [62]. A human knowledge engineer has worked in parallel and evaluated the results of the miner as well as the outcomes of the automated evaluation. He considered around 50% of the triples produced as noise. Recent results give a coverage ratio of 39.68%, an accuracy of 52.1%, a precision of 75.81% and a recall of 9.84% [67]. A detailed study revealed that the Perl scripts have to be replaced by a proper concordancy program (e.g. WordSmith) to eliminate the high number of noise words from the corpus that enter in the computation and distort the results severely. Also, some more work on detecting compounds is needed. The evaluation procedure showed that the miner discarded too many low frequency words that are deemed relevant by the statistical formula. Although the evaluation procedure might not yet be interesting for VAT@ directly, indirectly it proved already to be useful as it revealed a weak spot of the text miner. A human evaluator would have needed considerably more time to detect this.

2.4.4 Conclusion

The aim is to develop a simple, objective and automated evaluation procedure for ontology miners from text. The four metrics presented above allow to set up benchmarking and regression tests. Even if the evaluation stays on the linguistic level (triples of words), the fact that the point of reference is the text itself constitutes a main asset of this procedure. This method can also be applied to existing ontologies. Another use case on privacy legislation is scheduled.

3 Applications and Tools

In this section we present existing tools realising evaluation methods which were introduced in section 2.

3.1 ODEval

3.1.1 Abstract

Ontology evaluation is a crucial activity, which needs to be carried out during the whole ontology lifecycle. The goal of this evaluation is to determine what the ontology correctly defines, does not define at all, or even incorrectly defines.

Ontologies implemented in Semantic Web languages (such as RDF(S), DAML+OIL, and OWL) should be evaluated from the point of view of knowledge representation before using them in Semantic Web applications. In ([79], [78],[76]) it has been discovered that current ontology tools (language-dependent ontology validation tools and ontology platforms) do not detect taxonomic problems in RDF(S), DAML+OIL, and OWL concept taxonomies from the point of view of knowledge representation. Indeed, such tools do not focus on detecting the inconsistencies and redundancies in concept taxonomies identified in [77]. However, ODEval[2] (a complement to such ontology tools) performs syntactic evaluation of RDF(S), DAML+OIL, and OWL ontologies, and evaluates their concept taxonomies from the point of view of knowledge representation using the ideas proposed in [77]. ODEval detects inconsistencies and redundancies in ontology concept taxonomies.

3.1.2 Method description

ODEval uses a set of algorithms, based on graph theory [80], to detect possible problems in ontology concept taxonomies (inconsistencies and redundancies). An ontology concept taxonomy is considered by ODEval as a directed graph *G(V,A)*, where *V* is a set of nodes or vertex and *A* is a set of directed arcs. The elements included in the sets *V* and *A* are different depending on each language and on each type of problem that we want to detect.

ODEval detects possible taxonomic problems (inconsistencies and redundancies) in each considered language in the following way:

- *RDF(S)*: In this language disjoint and exhaustive knowledge cannot be defined with any of its primitives. Consequently, the only problems that can be resolved in RDF(S) ontologies are circularity and redundancy.
  - Circularity Problems. ODEval looks for cycles in the graph *G(V,A)*.
  - Redundancy Problems. For each class *class_A* in the set *V*, and for each arc r_i in the set *A* whose origin is *class_A*, we take r_i out of the set *A* and check whether this change affects the set of elements that are reachable from the *class_A*. If there is no change, this means at least one of the r_i is dispensable. And therefore, ODEval has found at least one grammatical redundancy problem.
- *DAML+OIL*:
  - Circularity Problems. ODEval looks for cycles in the graph *G(V,A)*.
  - Partition Errors. In this type of error, ODEval performs out two distinctions:
    - Disjoint groups. An error occurs in a disjoint decomposition or a partition, formed by the classes {$Class\_P_1$, $Class\_P_2$,…, $Class\_P_n$}, if there are common elements in two or more branches of the partition.

[2] http://minsky.dia.fi.upm.es/odeval

        - Exhaustive groups. If an element is only reachable from the base class (or its equivalents) and it is not reachable from the classes of the decomposition, then there is an error in the exhaustive group.

    - Redundancy Problems. For each class *class_A* in the set *V*, and for each arc r_i in the set *A* whose origin is *class_A*, we take r_i out of the set *A* and check whether this change affects the set of elements that are reachable from the *class_A*. If there is no change, this means at least one of the r_i is dispensable. And therefore, ODEval has found at least one grammatical redundancy problem.

- *OWL*:

    - Circularity Problems. ODEval looks for cycles in the graph *G(V,A)*.

    - Partition Errors. ODEval detects errors in disjoint groups. In this case, an error occur in a disjoint decomposition or a partition, formed by the classes {$Class\_P_1$, $Class\_P_2$,…, $Class\_P_n$}, if there are common elements in two or more branches of the partition.

    - Redundancy Problems. For each class *class_A* in the set *V*, and for each arc r_i in the set *A* whose origin is *class_A*, we take r_i out of the set *A* and check whether this change affects the set of elements that are reachable from the *class_A*. If there is no change, this means at least one of the r_i is dispensable and therefore that ODEval has found at least one grammatical redundancy problem.

3.1.3 Usefulness / Relevance for practice

Ontologies play an important role for the Semantic Web as a source of formally defined terms for communication. Like any other resource used in software applications, ontology content needs to be evaluated before being reused it in other ontologies or applications. Ontology content evaluation is a critical process to be undertaken before ontologies can be integrated in final applications. It is unwise to publish an ontology or to implement software that relies on ontologies without first evaluating its content (concept definitions, taxonomy and formal axioms).

As ontologies move from academic institutions into commercial environments they have to fulfil stronger requirements (correctness, consistency, completeness, conciseness, etc.). For this reason, ontology tools (like ODEval) are needed to prevent possible anomalies in ontologies, both in the research area and in the industrial area, in order to provide reliable ontology-based systems.

3.1.4 Conclusion

Current ontology tools (language-dependent ontology validation tools and ontology platforms) do not focus on detecting inconsistencies and redundancies in ontology concept taxonomies. The (re)use of ontologies without anomalies is a critical point in the industrial area in order to produce successful projects. For this reason, it is necessary to work on the ontology evaluation area and to create evaluators (like ODEval) in order to complement current ontology tools.

ODEval is a tool that evaluates RDF(S), DAML+OIL, and OWL concept taxonomies from a knowledge representation point of view. This tool is meant to help ontology developers in designing ontologies, without anomalies, in such ontology languages, and to help ontology engineers in reusing ontologies without problems in their concept taxonomies.

3.2 OntoManager

3.2.1 Abstract

In an ontology-based information portal, ontologies often support the process of “indexing” the content of an information resource – so called semantic annotation – and the navigation through the knowledge repository – so called conceptual navigation. However, ontologies, as a conceptual model for the given business domain, should react to all changes in the business environment. This includes accounting for modifications in the application domain or in the business strategy; incorporating additional functionality according to changes in the users’ needs; organizing information in a better way, etc. If the underlying ontology is not up-to-date or the annotation of knowledge resources is inconsistent, redundant or incomplete, then the reliability, accuracy and effectiveness of the system decrease significantly [23]. In order to avoid these real problems, ontology-based applications have to be supported by a mechanism for discovering these changes, analyzing and resolving them in a consistent way [24].

3.2.2 Method description

We have developed such an approach for ontology management and implemented it in the ***OntoManager*** tool (see http://ontoware.org/projects/ontomanager). It concerns the truthfulness of an ontology with respect to its problem domain - does the ontology represent a piece of reality and the users' requirements (user are end-user of ontology-based portals or applications) correctly? Indeed, it helps to find the “weak places” in the ontology regarding the users’ needs, ensures that generated recommendations for the ontology improvement reflect the users' needs, and promotes the accountability of portal managers. In this way, the ***OntoManager*** provides an easy-to-use management system for administrators, domain experts, and business analysts, since they are able to use it productively, with a minimum of the training. As known to the authors, none of the existing other ontology management systems offer support for (semi-) automatic ontology improvement in response to the users’ needs analysis.

The conceptual architecture – the MAPE model

Our management system is realised according to the **MAPE** (**M**onitor **A**nalyse **P**lan **E**xecute) model [25], which abstracts a management architecture into four common functions: collect the data, analyse the data, create a plan of action, and execute the plan. Indeed, our architecture decomposes the control loop into four parts:

- **Monitor** – mechanism that collects, organises and filters the data about users’ interactions with the ontology-based application;
- **Analyse** – mechanism that aggregates, transforms, correlates, visualises the collected data, and makes proposals for changes in the ontology;
- **Plan** – mechanism to structure actions needed to apply the discovered changes by keeping the consistency of the ontology. The planning mechanism uses evolution strategies [23] to guide its work;
- **Execute** – mechanism to update the underlying ontology-based application according to the changes applied in the ontology.

By monitoring (**M**) the behaviour of users and analysing (**A**) this data, planning (**P**) which actions should be taken and executing (**E**) them, a kind of a “usage loop” is created.

Figure 1 depicts this “usage loop” in an information portal scenario. A user is searching for information by querying and/or navigating through the portal (cf. 1 in Figure 1). All activities the user has performed are acquired in the Semantic Log (cf. 2), which is structured according to the Log Ontology, and contains meta-information about the content of visited pages [26].

This log data is aggregated and visualised in the ***OntoManager*** (cf. 3). Moreover, the ***OntoManager*** helps ontology managers discover changes in the ontology, which are mostly important for enhancing the usability of the application. The architecture of the ***OntoManager*** is described in the next section in more details. Since the application of a single ontology change can cause inconsistency in another part of this ontology and all the artefacts that depend on it [27], we applied the ontology evolution process (cf. 4) that guarantees the transfer of the ontology and dependent artefacts into another consistent state. Moreover, in the case of creating a new concept, **METIS**[3] – an ontology-based web crawler - can be started to complete the concept with the most promising instances that can be found in an intranet (or on the Internet, in general). Finally, since the underlying application is ontology-based, all changes in the ontology are reflected in the structure of the portal (cf. 1), by tailoring the portal to the users' needs, which implicitly arose. For example, if none of users were interested in a topic, then the ***OntoManager*** can recommend to the ontology manager to remove the corresponding concept from the topic hierarchy. Consequently, new users will be not be "bored" by browsing topics which are useless for the domain shown in the portal. In that way, our management system aims to be a user-friendly platform that integrates the results from the analysis of the usage data with the tools that guide the process of modifying the ontology.

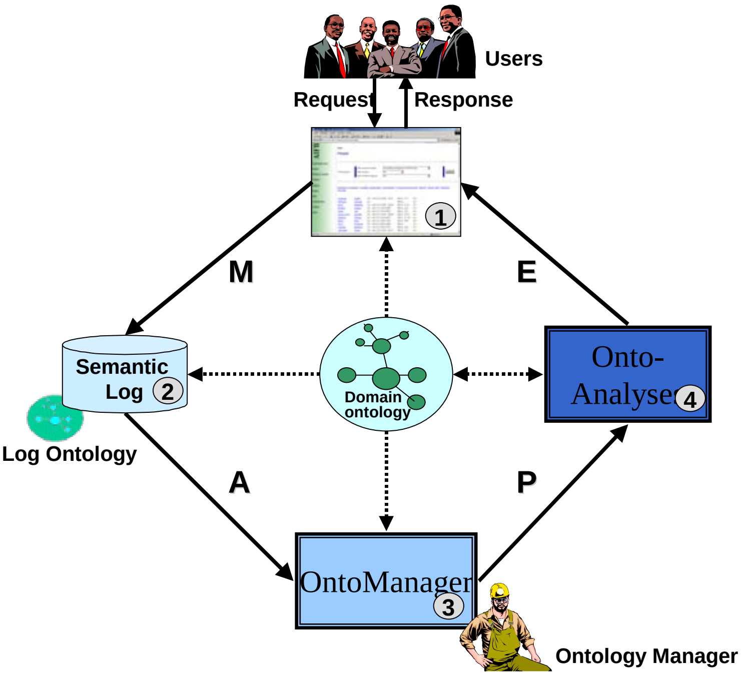


Figure 1 The conceptual architecture of the ontology management system according to the MAPE model

The ***OntoManager*** has been designed to provide the methods and tools that support the ontology managers in managing and optimising the ontology according to the users' needs. This system incorporates mechanisms that assess how the ontology (and by extension the application) is performing based on different criteria, and then enables the user to take action to optimise it.

One of the key tasks is to check how the ontology fulfils the perceived needs of the users. In that way, we obtain an in-depth view of the users' perspective on the ontology and the ontology-based application, since it is on the top of this ontology the application is going to be conducted. The technique that can be used to evaluate/estimate the users' needs depends on

[3] http://ontoware.org/projects/metis

the information source. By tracking users’ interactions with the application in a log file, it is possible to collect useful information that can be used to assess what the main interests of the users are. In this way, we avoid asking the users explicitly, since they tend to be reluctant to provide feedback via filling in questionnaires or forms.

3.2.3 Relevance for practice

The application OntoManager represents a pragmatic approch based on the usage statistics of ontologies to identify the relevance of concepts in a specific application or domain.

OntoManager is easy to apply and to use also by end-users. The evaluation quality however is not so high and does not allow one to discover deep insights.

The main application field of OntoManager is domains in which usage information of ontologies is present and further evaluation relevant information is missing or difficult to extract. For example, OntoManager is used within a SEmantic PorAL (SEAL) [70] to evolve the underlying portal ontology towards the users interest in the EU project OntoWeb[4] portal.

3.2.4 Case study

OntoManager has been developed within the project SemIPort. The intention of the project **Semantic Methods and Tools for Information Portals (SemIPort)** is to evolve a set of methods and tools for representing and accessing information within a semantically structured information portal, while offering the possibility to integrate one´s own information.

For testing purposes, the developed approaches will be evaluated on data from the German bibliography portal DBLP, and the tools are planed to be integrated into the competency and service network portal of the German Informatics Society (GI) which is currently under construction[5].

At the moment OntoManager is used within a test-bed scenario for the bibliography portal to ensure high content quality with respect to the users interest.

3.2.5 Conclusion

To summarize, OntoManager is best applied in domains in which usage information of ontologies is available to identify relevant concepts of an ontology. This occurs mainly in the area of web portals or any other ontology-based application producing so called semantic log files. However, the limitation on usage information does not allow to evaluate an ontology in general. Therefore OntoManager might be used as an additional analysis of an ontology within an existing evaluation process.

[4] http://www.OntoWeb.org

[5] http://www.io-port.net

4 Discussion

In this deliverable, we have reported methods and tools for evaluating ontologies, and discussed the relevance and usefulness of these methods and tools for industrial practice. We will now summarize the main technical characteristics of these methods and tools, and their main usefulness characteristics. To conclude, we will envision further work necessary to make the methods and tools more useful and usable to industrial practitioners, and to develop other methods and tools.

4.1 The ontology-evaluation methods and tools considered in this deliverable

4.1.1 Methods

The methods and their main technical characteristics are summarized in Table 1. For each method are mentioned: its goal, a short description, the ontology or system lifecycle stage where the method can be used, and the tools that may support the method.

4.1.2 Tools

The tools and their main technical characteristics are summarized in Table 2. For each tool are mentioned: its goal, a short description, the ontology or system lifecycle stage where the tool can be used, and the method it supports.

TABLE 1.- The methods considered.

Method	Goal	Description	Lifecycle Stage	Tools Supporting the Method
OntoMetric	Helps to choose the appropriate ontology for a new project	□ Compares the importance of the project objectives and study the characteristics of the ontologies □ Gets for every candidate ontology a quantitative measure of its suitability	After ontology release (type 3):	
Natural Language Application metrics	Helps evaluate the content of ontologies with respect to various metrics:		□ Pre-modelling stage (type 1), except Maedche's string matching measure for lexical comparison. □ Modelling stage (type 2)	
	▸ Precision and Recall Metrics	Measures for each ontology (a) how many identified items are correct and (b) how many items that would have been identified are effectively identified		
	▸ Cost-based evaluation metric	Characterizes the performance in terms of the cost of errors, or the value of correct things (e.g., the importance of recognizing Person vs. Date correctly)		
	▸ Tennis measure	Gives a measure of the "fit" between an ontology and a corpus (domain knowledge) by using a vector space model of instances (terms)		
	▸ Lexical comparison level measure	Compares the contents of two ontologies without considering their conceptual structure		
OntoClean	Helps evaluate a formal ontology.	□ Cleans the taxonomical structure of ontologies (e.g., upper level ontologies). □ Compares the ontology *vs.* a predefined ideal taxonomical structure to detect inconsistencies	□ Pre-modelling stage (type 1). □ Modelling stage (type 2)	OntoEdit
EvaLexon	Helps evaluate ontologies created (but not populated) by ontology miners from text.	□ The method stays at the linguistic level (compares words rather than concepts). □ *Reference point for the evaluation:* the text itself (and not a gold standard ontology). □ *Principle:* compares the vocabulary of the triples mined with the input text as such and with a set of words considered to be relevant for that text. □ *What is measured* (through regressions tests): recall and precision using a set of relevant words derived from the text; coverage and accuracy (derived measures) using the text itself.	Pre-modelling stage (type 1)	Perl scripts and Tawk scripts

TABLE 2.- The tools considered.

Tool	Goal	Description	Lifecycle stage	Supported Method
ODEval	Helps evaluate ontologies from the point of view of knowledge representation.	□ Evaluates the concept taxonomy of the ontologies. □ Detects inconsistencies and redundancies in taxonomies. □ Furthermore, performs syntactic evaluation of RDF(S), DAML+OIL, and OWL ontologies.	Type 2, when the ontology is finished.	Method from [77]
OntoManager	Helps determine the truthfulness of an ontology with respect to its problem domain.	□ Finds the "weak places" in the ontology, and modifying it, regarding the end-users' needs/requirements (of a portal or application). □ Relies on the analysis of usage data: By tracking users' interactions with the application in a log file, it is possible to collect useful information that can be used to assess what the main interests of the users are.	Type 3, (Evaluating an ontology after its release; evaluating ontology in usage)	

4.2 Relevance and usefulness of the methods and tools for industrial practice

A main goal of the current KnowledgeWeb WP 1.2 study on the "Methods for ontology evaluation" was to determine the relevance and usefulness of some methods and tools for industrial practice. What can be concluded from this study? What can be said about the relevance and usefulness of the methods OntoMetric, Natural Language Application metrics (hereafter, NLAM), OntoClean and EvaLexon, and what can be said about the tools ODEval and OntoManager. Summarizing, tables 3 and 4 give us some answers. In these tables are mentioned for each method/tool: (a) the users of the method/tool, (b) the relevance and usefulness of the method/tool, (c) its usability, (d) its applications and related use cases.

4.3 Are the methods/tools useful or relevant to enterprises?

Except for the OntoClean method – for which relevance (usage) for industry is said to be very low (limited), all the methods are considered as useful or relevant to enterprises (see column "Usefulness/Relevance for practice" of Table 3). Concerning the tools, OntoManager seems to be the more relevant for evaluating ontologies.

4.4 To which extent are the methods/tools useful or relevant to enterprises?

The extent to which a method or a tool is usable or relevant can be considered from three points of view: *(a)* from the intended usage viewpoint, *(b)* from the application (or use case) viewpoint, and *(c)* from the user viewpoint.

4.4.1 Which is or can be the usage of the methods/tools?

Usefulness or relevance can be first seen in the intended industrial usage of the methods/tools (see the column "Usefulness/Relevance for practice" of Table 3 and 4). It can be noticed that the methods and tools are intended to be used, and are sometimes used to:

- select existing ontologies for some industrial purpose;
- (as a selection process) measure the correspondence between corporate textual sources and some candidate ontology (to see if the ontologies reflect the content of the sources, and so can be used for annotating corporate documents);
- monitor an ontology in use (i.e., adapting the ontology once installed in some corporate system).[6]

If we could situate the methods on some industrial relevance scale, we could see that the NLAM, OntoMetric and EvalLexon are the most relevant, and OntoClean the least relevant. The weak relevance of OntoClean can be explained by its academic purpose: OntoClean is a method which focuses on the evaluation of formal aspects of ontologies, and their creators seem to be more[7] interested ~~on~~ by the evaluation of ontology upper-levels (i.e., more abstract levels, or the levels the most distant from the applications.)

4.4.2 Which are or can be the applications (use cases) of the methods/tools?

The usefulness or relevance of the methods/tools can be seen also in the applications and use cases mentioned, i.e., in actual usages of the methods/tools. It can be seen from this viewpoint that methods like NLAM and EvaLexon have applications and/or use cases: KM Portal

[6] Note that applying an evaluation method is sometimes considered as applying some ISO verification procedure to improve the quality of industrial products. This is a sound argument for industrials.

[7] "More" doesn't mean "exclusively."

applications (esp., an Employment Portal) for NLAM; and legislation (in particular VAT Legislation) use cases for EvaLexon.. The other methods or tools do not have such specific applications or use cases, suggesting a lower relevance to industry.

4.4.3 Who are or can be the users of the methods/tools?

Another way of seeing the usefulness or relevance of the methods/tools can be to look at the persons who are designated to use them. According to whether one mentions, as users, specialists (ontologists) vs. non specialists (end-users), researchers (who could be referred to as “basic-research ontologists”) vs. practitioners (who could be referred to as “industrial ontologists”), we can determine if the methods/tools are relevant to industrial practice, or still remain in the academic realm: for example, NLAM, OntoMetric, and EvaLexon are rather practitioner-oriented methods; and OntoClean is rather a method oriented towards basic research ontologists.

4.5 Are the methods/tools usable, and to which extent?

We can know if a method or a tool is (potentially) usable when enterprise members are mentioned as users of this method or tool. For example, OntoMetric mentioned project managers. If NLAM does not explicitly mention enterprise members, they implicitly supposed what can be referred to as Application users. EvaLexon mentions a Professional Services group of an IT-Company as users. On the other side, if OntoClean mentions ontology developers as users, these users in fact are basic-research ontologists.

We can know if a method or a tool is (potentially) usable when non-specialists or practitioners do not encounter usability problems, that is, when they can easily understand the method or tool, learn them without huge efforts, and operate them easily. EvaLexon is said to be simple to understand by laymen. NLAM are partially understandable, because they rely on Natural Language rather than on formal languages. With the NLAM “Cost-based evaluation metric”, it is difficult to assign a weight. To use OntoClean, it is necessary to be trained., and to have a deep knowledge in ontology engineering.

We can know if a method or a tool can be usable in a short- or mid-term when usability improvements are proposed. For example, it has been proposed to simplify the NLAM.

In brief, the most usable methods for practitioners seem to be EvalExon and NLAM, and the less usable, OntoClean Concerning the tools, OntoManager seems to be the more usable for practitioners.

TABLE 3.- Usefulness and relevance for practice of the methods.

Method	Users	Usefulness/Relevance for practice	Usability	Application/Use Case
OntoMetric	Knowledge engineers needing to look for ontologies disperse in Web servers to incorporate them into their systems Project managers	Helps justify decisions and weigh up the advantages and the risks of choosing an ontology	□ Specifying the characteristics of an ontology is complicated and takes time □ Assessing the characteristics is quite subjective (from the project managers' viewpoint)	
Natural Language Application metrics	Application users needing to know how good the system is and whether it is really helping them with their search	Useful to compare different systems (ontologies), rather than comparing a system (ontology) to a gold standard one		□ *Applications :* Natural Language applications involving ontologies; support for knowledge-intensive industries in monitoring information resources on the Web.
▸ Precision and Recall Metrics		□ Number of problems, esp., misclassification (due to binary scoring)	□ Most well known and widely used evaluation method	□ *Most relevant use case :* h-TechSight Knowledge Management Portal (KMP); particular application: KMP employment portal, tested by IChemE (Institution of Chemical Engineers) to see how it could help gain more insight about employment activities in their field.
▸ Cost-based evaluation metric		□ Overcomes the misclassification problem (thanks to a scalar scoring)	□ Many different weights : difficult for a user to assign a weight, or to find a way to calculate □ Needs to be adapted to be adopted as a standard for ontology content evaluation	
▸ Tennis measure				
▸ Lexical comparison level measure				

OntoClean	Ontology developers Currently not applicable by end-users	□ Evaluates formal ontologies. □ Focuses on the cleaning of taxonomies (e.g., the upper level). □ Provides useful structural and formally driven insights into semantic models. □ Does not allow to infer something about ontology usability. □ Relevance for industry is very low. □ Usage for industry is very limited	□ Requires significant training □ Only few people are currently able to apply it properly. □ *Planned (with UPM):* implementing a more user-friendly and intuitive solution OntoClean in general or as implemented in OntoEdit requires well-trained users and assumes expert knowledge about ontological engineering.	□ *Most relevant use case :* Cleaning the upper level of the WordNet taxonomy
EvaLexon	A Professional Services group (i.e. delivering customised end-solutions) of an IT-company embracing semantic web technology	Tuning ontologies for a specific customer. Setting up benchmarking and regression tests.	□ Simple to understand by laymen (comparison of words, not concepts). □ Automated evaluation procedure.	□ *Application domains:* legislation. □ *Most relevant use case :* A Belgian company specialised in VAT legislation; the company wants to integrate a VAT ontology in their applications to bridge conceptual differences, to facilitate maintainability (when the directive is updated) and enhance interoperability with third-party applications (e.g. national e-government VAT applications) □ *Scheduled use case :* Privacy legislation.
		Given an existing ontology, i.e. an ontology in a more or less final state, comparing it to its source text(s).		Currently not yet applied in this sense.

TABLE 3.- Usefulness and relevance for practice of the tools.

Tool	Users	Usefulness/Relevance for practice	Usability	Application/Use Case
ODEval	Ontology developers (creation) Ontology engineers (reuse)	Evaluates the content of some ontology before reusing it in other ontologies or applications		□ Semantic Web applications. □ New ontologies
OntoManager	Ontology managers Ontologists Domain experts Business analysts	□ Ensures that generated recommendations for the ontology improvement reflect the users' needs.□ Supports ontology management and optimising according to the users' needs, relying on the analysis of the usage data; in other words, allows managing ontologies from the users' perspective on the ontology and the ontology-based application. □ Promotes the accountability of managers. □ By tracking users' interactions with the application in a log file, it is possible to collect useful information that can be used to assess what the main interests of the users are met. □ Example: If none of users were interested in a topic, then the OntoManager can recommend the ontology manager to remove the corresponding concept from the topic hierarchy. Consequently, new users will be not "bored" by browsing topics, which are useless for the domain shown in the portal. □ The evaluation quality however is not so high and does not allow one to discovery deep insights.	□ An easy-to-use management system: Users are able to use the tool productively, with a minimum of the training. □ Helps ontology managers discover changes in the ontology, which are mostly important for enhancing the usability of the application.	□ *Application domains:* Domains in which usage information of ontologies are present and further evaluation relevant information is missing or difficult to extract. □ *Use case 1:* SEmantic PortAL (SEAL). □ *Use case 2:* Test-bed scenario for the bibliography portal to ensure high content quality with respect to the users' interests.

4.6 Further Work

To the question, “Are the methods and tools considered in this deliverable useful for industrial practice?”, we can answer: “Yes, but to differing degrees”. The most relevant methods seem to be OntoMetric, NLAM and EvalExon. The less useful is OntoClean. The most relevant tool seems to be OntoManager. Strictly speaking, we can say that industrial ontology evaluation methods and tools do not exist. All the methods here considered come from the academic realm. In other words, there remains some work to develop this transfer. There remains also some work to survey existing evaluation methods and tools to better learn from previous experiences.

4.6.1 Further Work for transferring the methods and tools to industry

Several transfer issues could be discussed, but we will limit the discussion to two of them: (1) adapting the methods and tools to industrial needs; (2) transferring and/or calling for ontology specialists. Behind these two issues is a practical question for industrials: Do we need to adapt the methods/tools to practitioners? Or will it be necessary to call for specialists, at least in some cases? This practical question is related to the “make or buy” and "go or no go" decisions discussed in section 1.3.

4.6.1.1 Adapting the methods and tools to industrial needs

To meet industrial needs, several kinds of adaptation can be performed, some of them having been suggested in the presentation of the methods/tools. An example is *simplifying the method/tool,* e.g., reducing the formalizing constraints. Another example is *making sense of the methods/tools:* industrials may not see very well the interest of some method or tool; to make this sense clearer to them, one can link the method/tool not only to the ontology lifecycle, but also to the ontology-based application; another way is to clearly determine which enterprise members can use the method because adapting a method/tool is, first of all, adapting them to intended types of users.

If the (financial or human) cost of adapting a method/tool is acceptable, the adaptation strategy can be adopted. If the cost is not acceptable, it can be necessary to ensivion another strategy, for example: transferring and/or calling for ontology specialists (who will be able to apply the method or the tool).

4.6.1.2 Transferring and/or calling for ontology specialists

If such a strategy could be chosen for the methods that are too costly to adapt, it can also be employed for methods or tools which are a priori not very relevant to industrial pratice, like OntoClean. In some cases, in effect, it could be necessary to go back to more formal methods to improve the usefulness of some ontology-based application. For example, it could be necessary to modify the upper levels of the ontology underlying some application to make a search engine’s inferences more relevant to the application’s users.

4.6.1.3 Further work on surveying evaluation methods and tools:

In this work, we have surveyed well-known methods and tools already used to evaluate ontologies and ontology-based tools. To benefit more from previous evaluation experiences, it would be necessary to have a follow-up to the current survey. Several follow-up directions can be taken, e.g.:

- Undertaking a larger state-of-the-art on the evaluation methods and tools reported in the literature, not forgetting to consider the connection between these methods and tools and design methods and tools (see, e.g., the method used in the Usable Ontology project by [98], to make the ontologies more usable); evaluation should not be separated from design; or design and evaluation should be considered as two indissociable phases of a series of iterative design-evaluation cycles.
- Identifying evaluation/design methods and tools used in communities other than the Ontology Engineering Community, that could be transferred to the ontology engineering community. See, for example, the attempt by [95] to transfer scenario-based methods from the Human-Computer interaction (HCI), Computer-supported Cooperative Work (CSCW), and Requirements Engineering (RE) communities to the Ontology Engineering community, in order to systemize the *motivating scenario method* introduced in the Ontology Engineering Community by ([96],[94]) through the TOVE ontological engineering method. Originally applied to the IST project CoMMA, the resulting scenario method proposed by [95] has been recently employed and deployed within the RNRT[8] Project KMP (Knowledge Management Platform), leading to the design of a Semantic Web Server[9] to map the competencies of enterprises and research labs situated in a given geographical area (the Telecom Valley in Sophia Antipolis, France), and belonging to a given community of interest (Telecoms, Microelectronics, and Informatics), to help the enterprises and labs cooperate, and the area economically develop. The method was adapted to this inter-firm perspective. In particular, the scenario structure used in the CoMMA project was adapted to capture crucial organizational features identified in the Giddens'(1983) structuration theory (see, e.g., [97]).

Lastly, we can suggest to complement this surveying activity with a validating activity of the most relevant evaluation methods and tools surveyed. If not assessed in the literature, the industrial validity of these methods and tools should be assessed to convince industrials to adopt them, or to call for specialists that can apply them to meet industrial needs.

[8] RNRT stands for Réseau National de Recherche en Télécommunications, a French organism which supports collaborative research projects in telecommunications between public research laboratories, large industrial groups and SME, around clearly defined priorities.

[9] Based on the Corese semantic search engine (Corby, Dieng, & Faron Zucker, 2004).

Bibliography

[1] Benerecetti, M., Bouquet, P. & Ghidini, C. (2000), Contextual Reasoning Distilled, *Journal of Theoretical and Experimental Artificial Intelligence* 12 no. 3, 279– 305.

[2] Bonifacio, M., Bouquet, P., & Cuel, R. (2002). Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management. *Journal for Universal Computer Science,* Vol. 8/6, pp 652-661. Springer Pub. & Co.

[3] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., & Stuckenschmidt, H. (2003). “C-OWL: Contextualizing ontologies” In *Proceedings of the 2nd International Semantic Web Conference (ISWC2003),* 20-23 October, Sanibel Island, Florida, USA.

[4] Cristani M. & Cuel R., (2004a) "A comprehensive guideline for building a domain ontology from scratch". In proceeding of *"International Conference on Knowledge Management (I-KNOW '04)"*, Graz, Austria

[5] Cristani M. & Cuel R., (2004b) “Methodologies for the Semantic Web: state-of-the-art of ontology methodology”. *Column of SIGSEMIS Bulletin*. Theme “SW Challenges for KM” V. 1 I. 2

[6] Davenport, TH., & Prusak, L. (1997). Working Knowledge: How Organizations Manage What They Know. *Harvard Business School Pr*, Boston.

[7] Dougherty, D. (1992) Interpretative barriers to successful product innovation in large firms, *Organization Science* 3, no. 2.

[8] Fauconnier, G., (1985) *Mental spaces: aspects of meaning construction in natural language*, MIT Press, 1985.

[9] Fernández, M., Gòmez-Pérez, A., & Juristo, N., (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering, *In Working Notes of the AAAI Spring Symposium on Ontological Engineering. Stanford University,* AAAI Press. Stanford, CA.

[10] Ghidini, C. & Giunchiglia, F., (2001) Local Models Semantics, or Contextual Reasoning = Locality + Compatibility, *Artificial Intelligence* 127 no. 2, 221–259.

[11] Giddens, A., 1984, *The Constitution of Society.* Berkeley. University of California Press

[12] Goffman, I., (1974) *Frame analysis,* Harper & Row, New York.

[13] Gruber, T. R., (1998) A translation approach to portable ontology specifications, *Knowledge Acquisition* 5, 199–220.

[14] Kuhn, T., (1979*) The structure of scientific revolutions*, University of Chicago Press.

[15] Maedche, A., & Staab, S. (2002) Measuring Similarity between Ontologies. *In: Proc. Of the European Conference on Knowledge Acquisition and Management - EKAW-2002*. Madrid, Spain, October 1-4, 2002. LNCS/LNAI 2473, Springer, pp. 251-263.

[16] Orlikowski, W.J. (1992), Learning from Notes: Organizational Issues in Groupware Implementation, *Proceedings of CSCW'92 Conference*, Turner, J. and Kraut, R. (Eds), The Association for Computing Machinery, New York, NY, pp. 362-369

[17] Orlikowski, W.J., & Gash, D. (1994). Technological frames: Making sense of information technology in organizations. *ACM transactions on information systems*, 12, 174-207.

[18] Sowa, J. F., (2000) *Knowledge Representation. Logical, Philosophical and Computational Foundations,* Brooks/Cole.

[19] Ushold, M., (2000) Creating, integrating and maintaining local and global ontologies. *Proceedings of the First Workshop on Ontology Learning (OL-2000) in conjunction with the 14th European Conference on Artificial Intelligence (ECAI 2000).*

[20] Weick, E.K., (1979) *The social psychology of organizing,* McGraw-Hill, Inc.

[21] A. Gangemi and N. Guarino and A. Oltramari and S. Borgo, (2002) Cleaning-up WordNet's Top-Level. of the 1st International WordNet Conference. Mysore, India.

[22] C. A. Welty and N. Guarino, (2001) Supporting ontological analysis of taxonomic relationships. In Data & Knowledge Engineering. vol 39, pages 51-74.

[23] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz, *Ontologies for Enterprise Knowledge Management*, IEEE Intelligent System, pp. 26-34, March/April 2003.

[24] I. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999.

[25] N. Stojanovic, A. Maedche, S. Staab, R. Studer, Y. Sure, *SEAL — A Framework for Developing SEmantic PortALs*, ACM K-CAP 2001. October, Vancouver, 2001.

[26] J. Kephart, D. Chess, *The Vision of Autonomic Computing*, IEEE Computer, January 2003., pp. 41-50.

[27] L. Stojanovic, A. Maedche, B. Motik, N. Stojanovic, *User-driven Ontology Evolution Management*, Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW'02, Madrid, 2002.

[28] N. Stojanovic, L. Stojanovic, J. Gonzalez, *On Enhancing Searching for Information in an Information Portal by Tracking Users' Activities*, First International Workshop on Mining for Enhanced Web Search (MEWS 2002), held in conjunction with WISE 2002, Singapore, 2002.

[29] N. Stojanovic, *On the Query Refinement in the Ontology-based Searching for Information*, the 15th Conference On Advanced Information Systems Engineering, CAiSE'03, Austria, 2003.

[30] R. Kimball, R. Merz, *The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse*, John Wiley & Sons, 2000.

[31] S. Card, J. Mackinlay, B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann, 1999.

[32] M. Uschold and M. Gruninger, *Ontologies: Principles, methods, and applications*, Knowledge Engineering Review, vol. 11, no. 2, pp. 93--155, 1996.

[33] R. Botafogo, E. Rivlin, and B. Shneiderman. *Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics*, ACM Transactions on Office Information Systems*, 10(2):142-180, 1992.*

[34] J. I. Kiger, *The Depth/Breadth Trade-Off in the Design of Menu-Driven User Interfaces*, Int J of ManMachine Studies, 20(2), pp. 201-213., 1984.

[35] K. Norman, *The Psychology of Menu Selection: Designing Cognitive Control of the Human/Computer Interface*. Ablex Publishing Corporation., 1991.

[36] V. Ramana, *The Importance of Hierarchy Building in Managing Unstructured Data*, Special Supplement to KM World, March 2002

[37] G. Salton, C. Buckley, *Improving retrieval performance by relevance feedback,*. Journal of the American Society for Information Science. 41(4): 288-297, 1990.

[38] L. Stojanovic, B. Motik, *Ontology Evolution within Ontology Editors*, EKAW'02/EON Workshop, Madrid, 2002.

[39] C. Hardless, R. Lindgren, U. Nulden, K., Pessi, *The Evolution of knowledge management system need to be managed*, http://www.viktoria.informatik.gu.se/groups/ KnowledgeManagement/Documents/kmman.pdf, 2000.

[40] N. Stojanovic, L. Stojanovic, *Usage-oriented Evolution of Ontology-based Knowledge Management Systems*, Proceedings of the 1st Int'l Conf. on Ontologies, Databases and Application of Semantics (ODBASE-2002), Irvine, CA, 2002.

[41] Rigau, G., Atserias, J. and Agirre, E. (1997) Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. In *Proc. of ACL/EACL*, Madrid, Spain, 48-55.

[42] ACE (2004). Annotation Guidelines for Entity Detection and Tracking (EDT), Available at http://www.ldc.upenn.edu/Projects/ACE/.

[43] Advanced Research Projects Agency (1993). *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, Morgan Kaufmann, California}.

[44] Brewster, C., Alani, H., Dasmahapatra, S. and Wilks, Y. (2004). Data Driven Ontology Evaluation. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2004)*, Lisbon, Portugal.

[45] Chinchor, N. MUC-4 Evaluation Metrics (1992). In *Proceedings of the Fourth Message Understanding Conferene*, 22-29.

[46] Introduction to the CONLL'02 shared task: Language Independent NE Recognition. http://cnts.uia.ac.be/conll2002/ner/

[47] Introduction to the CONLL'03 shared task. Language-independent NE Recognition. http://cnts.uia.ac.be/conll2003/pdf/14247tjo.pdf

[48] Fellbaum, C. (1998) *WordNet - An Electronic Lexical Database*, MIT Press.

[49] Fiscus, J. G., Doddington, G., Garofolo, J.S. and Martin, A. (1998). 1998 Topic Detection and Tracking Evaluation (TDT2), *Proc. of the DARPA Broadcast News Worksho*, Virginia, US.

[50] Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*, Kluwer Academic Publishers.

[51] Levenshtein, V.I. (1966).Binary Codes capable of correcting deletions, insertions and reversals,

In *Soviet Phys. Dokl.*, 10, 707-710.

[52] Olsson, F., Eriksson, G., Franzén, K., Asker, L. and Lidén, P. (2002). Notions of Correctness when Evaluating Protein Name Taggers. In *Proceedings of COLING 2002*, Taipei, Taiwan.

[53] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff and M. Goranov, 2004. KIM -- Semantic Annotation Platform, *Journal of Natural Language Engineering.*

[54] Smeaton, A. and Quigley, I. (1996). Experiments on using semantic distances between words in image caption retrieval. In *Proc. of 19th International Conference on Research and Development in Information Retrieval*, Zurich, Switzerland.

[55] Stevenson, M. (2002). Combining Disambiguation Techniques to Enrich an Ontology. In *Proceedings of ECAI-02 workshop "Machine Learning and Natural Language Processing for Ontology Engineering,* Lyon, France.

[56] van Rijsbergen, C. Information Retrieval. Butterworths, London, 1979.

[57] Buitelaar P., Handschuh S. & Magnini B., (2004), ECAI 2004 Workshop on Ontology Learning and Population: towards evaluation of text-based methods in the Semantic Web and knowledge discovery life cycle [in preparation as IOS Press volume]

[58] Daelemans W., Buchholz S. & Veenstra J., (1999), Memory-Based Shallow Parsing. In: Proceedings of CoNLL-99, Bergen, Norway, June 12, pp.53-60

[59] De Kock J., (1984), Elementos para una estilística computacional – tomo I, Editorial Colloquio, Madrid

[60] Gómez-Pérez A. & Manzano-Macho D. (eds.), (2003), A survey of ontology learning methods and techniques, OntoWeb Deliverable #D1.5, Universidad Politécnica de Madrid

[61] Maedche A. & Staab S., (2002), Measuring similarities between ontologies, Proceedings of the European Conference on Knowledge Acquisition and Management, Springer

[62] Reinberger M.-L., Spyns P., Daelemans W. & Meersman R., (2003), Mining for lexons: applying unsupervised learning methods to create ontology bases. In, Meersman R., Zahir T., Schmidt D. et al.,(eds.), *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, LNCS 2888, pp. 803 - 819, 2003. Springer Verlag.

[63] Reinberger M.-L., Spyns P., Pretorius A.J. & Daelemans W., (2004), Automatic initiation of an ontology. In, Meersman R., Tari Z. et al.,(eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE (part I), LNCS 3290, pp. 600 -617, 2004. Springer Verlag.

[64] Reinberger M.-L. & Spyns Peter, (2004), Discovering Knowledge in Texts for the learning of DOGMA-inspired ontologies. In, Buitelaar P., Handschuh S. & Magnini B.,(eds.), *Proceedings of the ECAI 2004 Workshop on Ontology Learning and Population*, pp. 19-24,. [extended version in preparation for an IOS Press volume on Ontology Learning and Population]

[65] Spyns P., Meersman R. & Jarrar M., (2002), Data modelling versus Ontology engineering, in Sheth A. & Meersman R. (ed.), SIGMOD Record Special Issue 31 (4), pp. 12-17

[66] Spyns P, Pretorius A.J. & Reinberger M.-L., (2004), Evaluating DOGMA-lexons generated automatically from a text corpus. In, Cimiano Ph., Ciravegna F., Motta E., & Uren V.,(eds.), *Proceedings of the EKAW 2004 Workshop on Language and Semantic Technologies to support Knowledge Management Processes*, pp. 38 – 44

[67] Spyns P & Reinberger M.-L., (2005), Evaluating ontology triples generated automatically from texts, [in preparation]

[68] Tawk Compiler v5, Thompson Automation Software, Jefferson OR, US

[69] Zipf G., (1949), Human behaviour and the principle of least effort, Addison-Wesley, Cambridge MA

[70] Jens Hartmann, York Sure. An Infrastructure for Scalable, Reliable Semantic Portals. IEEE Intelligent Systems 19 (3): 58-65. May 2004.

[71] Fernández-López, M (1999) “Overview of Methodologies for Building Ontologies”. Proceedings of the IJCAI’99. Workshop on Ontologies and PSMs. Stockholm.

[72] López-Pérez, A (2004). “Integración de la aplicación OntoMetric en la plataforma WebODE”. Proyecto fin de carrera. Facultad de Informática (Universidad Politécnica de Madrid).

[73] Lozano-Tello, A (2002). “Métrica de idoneidad de ontologías”. Tesis doctoral. Universidad de Extremadura.

[74] Lozano-Tello, A; Gómez-Pérez, A (2004). “ONTOMETRIC: A Method to Choose the Appropriate Ontology”. Journal of Database Management. Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods. Volumen: 15(2). Abril-June 2004.

[75] Saaty, T (1977) “A Scaling Method for Priorities in Hierarchical Structures”. Journal of Mathematical Psychology, Vol.15, (1977) 234–281.

[76] Corcho O, Gómez-Pérez A, González-Cabero R, Suárez-Figueroa MC (2004) *ODEval: a Tool for Evaluating RDF(S), DAML+OIL, and OWL Concept Taxonomies*. 1st IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI 2004). Toulouse, France. PP: 369-382.

[77] Gómez-Pérez A (2001) *Evaluating ontologies: Cases of Study*. IEEE Intelligent Systems and their Applications. Special Issue on Verification and Validation of ontologies. March 2001, Vol 16, Nº 3. PP: 391-409.

[78] Gómez-Pérez A, Suárez-Figueroa MC (2004) *Evaluation of RDF(S) and DAML+OIL Import/Export Services within Ontology Platforms*. 3rd Mexican International Conference on Artificial Intelligence (MICAI 2004) Mexico City, Mexico. PP: 109-118.

[79] Gómez-Pérez A, Suárez-Figueroa MC (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*. Evaluation of Ontology-based Tools (EON2003) 2nd International Workshop located at the 2nd International Semantic Web Conference (ISWC 2003) Sundial Resort, Sanibel Island, Florida, USA. PP: 13-26.

[80] Goodaire E, Parmenter M (1998) *Discrete Mathematics with Graph Theory*. Ed. Prentice Hall. 1998.

[81] Knowledge Web Annex I Description of Work, p. 29

[82] Lenat D. & Guha V., (1990), Building large knowledge-based systems: representation and inference in the CYC project. Addison-Wesley, Reading MA

[83] Swartout B., Ramesh P., Knight K., & Russ T., (1997), Toward distributed use of large scale ontologies, in Symposium on Ontological Engineering Engineering of AAAI, Stanford

[84] Ushold M. & Grüninger M., (1996), Ontologies: principles, methods and applications, Knowledge Sharing and Review, 11 (2)

[85] Geoffrey C. Bowker and Susan Leigh Star.*Sorting Things Out:Classification and Its Consequences*.Cambridge, MA:MIT Press, 1999

[86] Williamson, Oliver E. 1975. Markets and Hierarchies: Analysis and Antitrust Implications. New York: Free Press.

[87] Strategy research: governance and competence perspectives Oliver E. Williamson, on *Strategic Management Journal* Volume 20, Issue 12, 1999. Pages 1087-1108

[88] The Theory of the Firm as Governance Structure: From Choice to Contract, Williamson O.E. *The Journal of Economic Perspectives*, 1 August 2002, vol. 16, no. 3, pp. 171-195(25), American Economic Association

[89] Laura Poppo, Todd Zenger, Testing alternative theories of the firm: transaction cost, knowledge-based, and measurement explanations for make-or-buy decisions in information services, on *Strategic Management Journal* Volume 19, Issue 9, 1998. Pages 853-877

[90] Do make or buy decisions matter? The influence of organizational governance on technological performance, Michael J. Leiblein, Jeffrey J. Reuer, Frédéric Dalsace, *Strategic Management Journal*, Volume 23, Issue 9, 2002. Pages 817-833

[91] Seung H. Han and James E. Diekmann, Approaches for Making Risk-Based Go/No-Go Decision for International Projects, Journal of Construction Engineering and Management, Volume 127, Issue 4, pp. 300-308 (July/August 2001)

[92] Bove, T., & Andersen, H. B. (1999, June). *The effect of an advisory system on pilots' go/no-go decision during take-off.* Presented at *HESSD-99, Workshop on Human Error, Safety and System Development* , Liège.]

[93] Corby O., Dieng R. et Faron-Zucker C. (2004), Querying the semantic web with the Corese search engine, *Proceedings of the European Conference on Artificial Intelligence (ECAI'2004), subconference PAIS*, pp 705-709, 2004.

[94] Fox, M.S., Grüninger, M..: Enterprise Modelling, *AI Magazine* (1998) 109-121.

[95] Giboin, A., Gandon, F., Corby, O., & Dieng, R. (2002). User Assessment of Ontology-based Tools: A Step Towards Systemizing the Scenario Approach, *Proceedings of EON'2002: Evaluation of Ontology-based Tools, OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*, Sigüenza (Spain), September 30, 2002, pp. 63-73.

[96] Grüninger, M., and Fox, M.S.: Methodology for the design and evaluation of ontologies, *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, AAAI Press, Menlo Park CA, (1995). Available at: http://www.ie.utoronto.ca/EIL/public/org.ps

[97] Lazaric N., & Thomas C. (2003), "Coordination and codification of knowledge inside a network or the building of an "epistemic community" : The Telecom Valley Case study" *19th EGOS COLLOQUIUM*, July 3th - 5th, Copenhagen, Denmark.

[98] Missikoff, M., Navigli, R., & Velardi, P. (2002). The Usable Ontology: An Environment for Building and Assessing a Domain Ontology. *Proceedings of the First International Semantic Web Conference on The Semantic Web*, June 09 - 12, 2002, Lecture Notes In Computer Science, pp. 39 - 53.

[99] Maynard, D., Yankova, M., Aswani, N. and Cunningham, H. (2004). Automatic Creation and Monitoring of Semantic Metadata in a Dynamic Knowledge Portal. *Proceedings of the 11th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA)*, Varna, Bulgaria, 2004.

[100] Cimiano, P., Staab, S. and Tane, J. (2003). Automatic Acquisition of Taxonomies from Text: FCA meets NLP In Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia, pp. 10-17.

[101] Cunningham, H., Maynard, D., Bontcheva, B. and Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).* Philadelphia, July.