Termontography and DOGMA for Knowledge Engineering within PROLIX

Peter De Baer[1], Robert Meersman[1], and Rita Temmerman[2]

[1] STARLab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2] Centre for Terminology and Communication, Erasmushogeschool Brussel, Pleinlaan 5, 1050 Brussels, Belgium
{pdebaer,meersman}@vub.ac.be, rita.temmerman@ehb.be

Abstract. In this article, we describe our ongoing research to combine two approaches, i.e. Termontography and DOGMA, for knowledge engineering. Both approaches have in common that they mainly rely on natural language to describe meaning. Termontography is a special form of terminography that results in an ontologically structured terminological resource. DOGMA is an abbreviation of Developing Ontology Guided Mediation for Agents. The DOGMA approach results in a scalable and modular ontology that can easily be (re)used for different domains and applications. Both Termontography and DOGMA have already been used separately during several research projects. In this article we explain how both approaches are being combined within the PROLIX project, and what the advantages of this combination are. The goal of PROLIX is to develop an open, integrated reference architecture for process-oriented learning and information exchange.

Keywords: Knowledge engineering, ontology engineering, terminography.

1 Introduction

In today's knowledge driven world, effective access to and use of information is a key enabler for progress. Modern technologies not only are themselves knowledge intensive technologies, but also produce large quantities of new information that must be processed and aggregated. These technologies require knowledge capture, which involves the extraction of useful knowledge from vast and diverse sources of information as well as its acquisition directly from users. Driven by the demands for knowledge-based applications, the constructed knowledge resources should best be structured in such a way that various applications may use the information. In order to build such resources, VUB STARLab combines two approaches for knowledge engineering, i.e. Termontography and DOGMA, as part of the European integrated project PROLIX[1].

In this article we will first introduce both approaches for knowledge acquisition and representation. Afterwards, we will describe the combination of both approaches within the PROLIX project and explain some of the advantages.

[1] http://www.prolixproject.org/

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 534–543, 2009.

2 Termontography

Termontography is based on the insights of sociocognitive terminology management [17], which in addition to linguistic information, takes cultural and cognitive perspectives into account during terminology description. Based on these insights an innovative methodology for sociocognitive terminography was developed. The goal of the methodology is to build comprehensive lexical resources that may help to overcome certain communication problems. The methodology therefore starts with a problem analysis of each communication problem, in order to find the requirements of a possible solution in the form of an ontologically structured terminological resource. An example of such communication problem may occur in specialized communication. To build a multilingual terminological resource, the methodology analyses terms as they appear and are used in multilingual (specialized) natural language, using techniques of corpus linguistics, terminography, and ontology engineering. Extracted terms from a specialized text corpus are described and linked to concepts, resulting in a terminological ontology. Techniques for ontology engineering are applied to structure the terminological information according to the different perspectives of interest for the end user. This formal knowledge representation, using an ontology, should also facilitate the application of the resulting resources in various software tools like an electronic dictionary, a translation memory system, or a knowledge management system.

Two examples of resources that were built using the methodology are: an ontologically structured lexical resource that helps to bridge communication gaps between welfare professionals in the European Union [4], and an ontologically-structured multilingual (EN, FR, NL) terminological resource of competences and occupations [3].

2.1 Termontography Methodology

The Termontography methodology consists of a cyclic process divided into six phases. During the first knowledge analysis phase (1), the communication problem is analyzed. The analysis results in a specification requirements document for the terminological resource and a categorization framework [5]. Such categorization framework may best be considered as a terminological domain ontology that contains the key concepts and relations for the domain. Using the key concepts a specialized text corpus is compiled during the information gathering phase (2). The gathered texts are then analyzed and relevant terminology is extracted and linked to the categorization framework. This we call the search phase (3). During the refinement phase (4) the terminological resource will be further developed. Verification (5) guarantees the correctness of the resource according to the accepted understanding of the domain. Validation (6) guarantees that the resource corresponds to the specification requirements document. Note: The methodology follows the typical ontology development life cycle [14]; however, the specification, conceptualization, formalization, and implementation phase all use the categorization framework [5] to capture the terminological and ontological information.

2.2 Termontography Software

Currently, the Termontography methodology is supported by the following software tools: the didactic CatTerm wizard teaches student-translators to use the methodology to acquire bilingual domain knowledge, the Termontography Workbench (see Fig. 1) is used to manage a specialized corpus and the ontologically structured terminology base, and the Categorization Framework Editor [3] that is used to manage a terminological ontology in the form of a categorization framework. All these software applications make use of the Categorization Framework API [2] to manage the underlying information store.

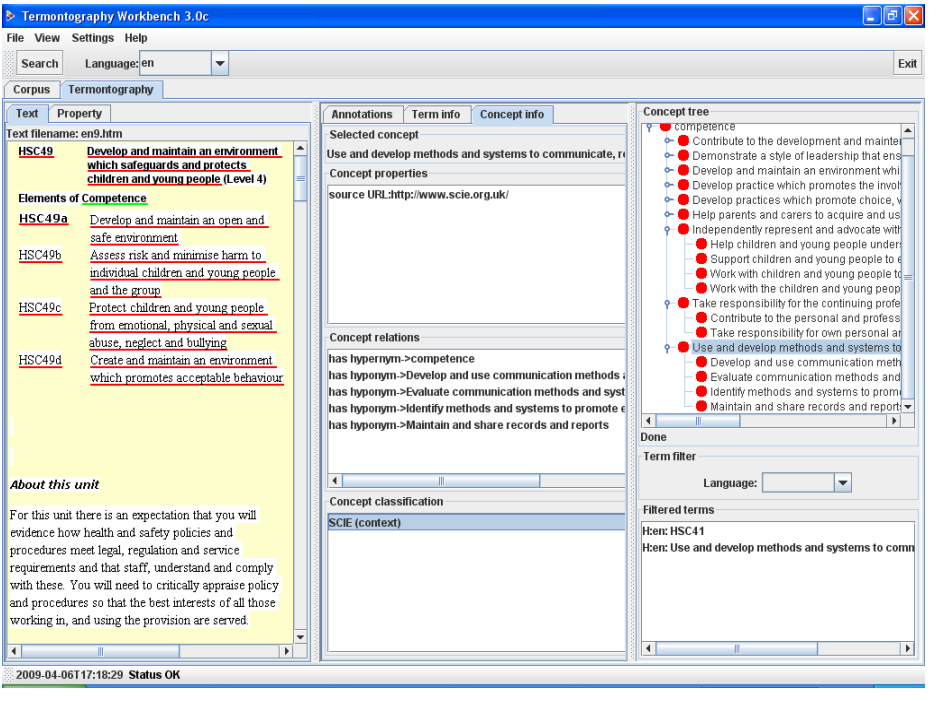

Fig. 1. The Termontography Workbench is used to manage a text corpus, extract terminology, and build a terminological ontology

3 DOGMA

DOGMA is a research initiative of VUB STARLab where various theories, methods, and tools for ontology engineering and ontology driven application design are studied and developed. A DOGMA inspired ontology is based on the classical model-theoretic perspective [15] and decomposes an ontology into a lexon base and into a layer of ontological commitments [11, 12]. This is called the principle of double articulation [16].

A lexon base holds simple, commonly agreed upon, context-specific facts that are part of the conceptualization of a particular domain. These facts are called lexons. A

lexon is formally defined as a 5-tuple <context-id, head concept term, role, co-role, tail concept term>. A lexon may be read as: within the context identified by the context-id, the implied head concept may play this role for the tail concept, and conversely the tail concept may play this co-role for the head concept. The modality "may" indicates plausibility since lexons are assumed to be defined independent of subsequent applications (and therefore also may come in many shapes and formulations for the same "fact"). For example the lexon <HRM, programming skills, part of qualification, providing competence, Bachelor in Computer Science> states that within the context of human resource management, programming skills (may be) part of the qualification for a Bachelor in Computer Science, and conversely that this degree (may be) providing the competence 'programming skills'.

Both (context, head) and (context, tail)-pairs are axiomatically assumed to identify a unique concept. Any specific (application-dependent) interpretation rules are moved to a separate layer of ontological commitments. The commitment layer mediates between the ontology base and its applications. Each ontological commitment defines a partial semantic account of an intended conceptualization [9]. It consists of a finite set of axioms that specify which lexons of the ontology base are used and how they should be annotated in and used by the committing application. Experience shows that it is much harder to reach an agreement on such application specific constraints than on the generic plausible domain facts [10]. E.g. a rule stating that each employee has a single function may hold in one organization, but could be too strong for another application.

As DOGMA intends to be a generic approach to ontology engineering it is not restricted to a specific ontology language (e.g. RDF [13] or OWL [20]). Once the elicitation process is finished, and the ontology formalized, the DOGMA tools can output the information to the target language. Conversely, existing ontologies may be converted from specific representation languages into DOGMA, so that they may be maintained and updated using the DOGMA Studio tools [1, 18].

The DOGMA principles derive in part from an early database design methodology called Natural Language Information Analysis Method [7] and uses natural language based techniques and terminology in order to elicit concepts from domain experts. The latter therefore do not have to tackle formal language issues, or learn to think in a new paradigm.

Due to the strict separation between lexon base (i.e. lexical representation of concepts and their relationships) and commitments (i.e. adding application-specific interpretation rules) a DOGMA ontology achieves a more convenient balance between use and reuse. For example, a well accepted fact that an employee holds a position within an organization, could be reused in many applications. An application specific commitment that an employee holds just one position could, however, be too strict for most organizations. Usage complexity concerns also become separated similar to the way databases implement data independence [11].

3.1 DOGMA Software

Currently, the DOGMA approach is supported by the following software tools[2]: the DOGMA Studio suite and the DOGMA-MESS web platform.

[2] We limit our discussion to the research prototypes; commercial versions of the Studio suite are available from the STARLab spin-off company Collibra (www.collibra.com).

The DOGMA Studio suite contains both a Server and a Workbench. The DOGMA Server is an ontology library system that features context-driven disambiguation of lexical labels. It provides a framework for managing multiple context dependency types and operators. The DOGMA Workbench is based on the plug-in architecture of the Eclipse IDE. Plug-ins exist for ontology viewing and querying, and editing modules support the different ontology engineering activities.

DOGMA-MESS (Meaning Evolution Support System) is a web-based platform that supports interorganizational ontology engineering [6]. The main focus in DOGMA-MESS is how to capture relevant interorganizational commonalities and differences in meaning. It provides a community grounded methodology to address the issues of relevance and efficiency. In DOGMA-MESS, there are three user roles: (1) Knowledge Engineer, (2) Core Domain Expert, and (3) Domain Expert. The task of the Knowledge Engineer is to assist the (Core) Domain Experts in their tasks. The major chunk of knowledge is captured by the Domain Experts themselves. The Core Domain Expert builds high-level templates in the so-called Upper Common Ontology. The Domain Experts specialize these templates to reflect the perspective of their organization in their Organizational Ontologies. The Domain Experts are shielded from complexity issues by assigning specific tasks in the elicitation process (e.g. to specialize a 'Subtask' template for 'Baking'). In every version of the process, common semantics are captured in the Lower Common Ontology whilst organizational differences are kept in the Organizational Ontologies. Information in the Lower Common Ontology is distilled from both the Upper Common Ontology and the Organizational Ontologies using meaning negotiation between (Core) Domain Experts. The Lower Common Ontology is then used as input for future versions in the process.

4 Termontography and DOGMA

DOGMA and Termontography have already been applied separately during the research projects FFPoirot (http://www.ffpoirot.org/) and PoCeHRMOM (http://starlab.vub.ac.be/website/PoCehrMOM). Currently, we use a combination of both approaches for the PROLIX project.

The objective of PROLIX is to align learning with business processes in order to enable organizations to faster improve the competencies of their employees according to continuous changes of business requirements. To reach this goal, PROLIX develops an open, integrated reference architecture for process-oriented learning and information exchange. Third party vendors may integrate specific solutions into the overall approach by introducing or replacing modular components and/or services. One of the tasks of STARLab in the project was to develop ontologies based on the existing competence frameworks used by the test bed partners, e.g. British Telecom (BT), the Social Care Institute for Excellence (SCIE), Klett Lernen und Wissen GmbH, and the Baden-Württembergische Genossenschaftsverband.

To develop the ontologies we made use of the existing competence materials of the test bed partners. For example, SCIE provided us with documents describing the competence standards, e.g. 'Health and Social Care' (HSC) and 'Leadership and Management for Care Services' (LMC), they use within their organization. To build the SCIE competence ontology based on these standards we made use of the Termontography Workbench (see Fig. 1) to build the text corpus, extract the terminology, and

build the terminological ontology. A first advantage of this approach is that, during the development of the ontology, the link with the source texts is preserved. Since the source texts help to document the derived ontology, the collaborating knowledge engineers, who may not be domain experts themselves, are better able to understand the concepts and concept relations. Another advantage is that the Termontography Workbench allows publishing the ontology in HTML[3]. The HTML format makes it easy to discuss the ontology with domain experts and users. Each competence also has a unique URL which may be used to specify the competence. For example, a job candidate might specify his competences (e.g. in a portfolio) by referring to the corresponding competence URLs. This enables HRM managers to better understand the competences by analyzing the HTML competence information. A third advantage is that (multilingual) terminological information can easily be added during the conceptualization process. Adding (multilingual) terminological information helps, both knowledge engineers and end users, to better understand the ontology. Terminological documentation is considered important in many ontology engineering methodologies, e.g. ENTERPRISE [19] and METHONTOLOGY [8].

Another task of STARLab within PROLIX was the development of a data matching service and the evaluation of different ontology-based data matching algorithms. Using the DOGMA Studio tools, knowledge engineers could, in collaboration with domain experts from the test beds, develop a domain ontology for competency matching. This domain ontology consists of the concepts: competence, competence level, competency, context, function, person, qualification, qualification level, and task.

- A *competence* is a generic competence that may be reused within different organizations.
- A *competence level* indicates the proficiency level of a person for competences. Examples are the reference levels in the Common European Framework of Reference for Languages[4]. Each competence level has a minimum and maximum value between 0 and 1 to enable inter-organizational comparison. If an organization uses four competence levels A, B, C, and D, competence level A might, for example, have the minimum value 0 and the maximum value 0.25, i.e. 1 / 4.
- A *competency* is a specific competence that includes a certain competence level and optionally a context.
- A *context* is a category that may help to contextualize competences. For example the context BT would indicate that the competence should be interpreted within the context of that organization.
- A *function* is an organization specific occupational title.
- A *person* represents an employee within an organization or a job candidate.
- A *qualification* is a training certificate issued by an organization that certifies a set of competencies for a person. The qualification may correspond to a set of training materials, a set of begin competencies, and a set of end competencies.

[3] SCIE competence ontology: http://starlab.vub.ac.be/prolix/SCIE/competences/index.htm
[4] Common European Framework of Reference for Languages: http://tinyurl.com/2k7zko

- A *qualification level* is a level that indicates the difficulty level of a qualification. Examples are the European Qualification Levels[5]. Each qualification level has a minimum and maximum value between 0 and 1 to enable inter-organizational comparison.
- A *task* is an organization specific task that requires certain competencies.

To test different matching strategies, application ontologies with instance information linked to terminological resources and the domain ontology were needed. To manage these ontological and terminological resources the Context Ontology Editor (COE) (see Fig. 2) was developed. The COE uses the Categorization Framework API (CF API) [2] and the DOGMA ontology format to bridge between both ontology formats. Although the CF API makes a distinction between abstract contexts and concrete objects, both types share the same super type 'meaning'. Therefore it is possible to handle both contexts - for the domain ontology - and objects - for the application ontology - in a similar way. Since the 'meaning' type may be linked to (multilingual) terminology, the data matching algorithms have access to both the ontological and the terminological information via the CF API. This makes it easier to implement data matching algorithms compared to using different types of resources. The Categorization Framework does not make use of description logics. However, the matching algorithms may commit to specific contexts, objects, object relations, and object properties in the resource. Therefore, semantically rich reasoning is possible. For

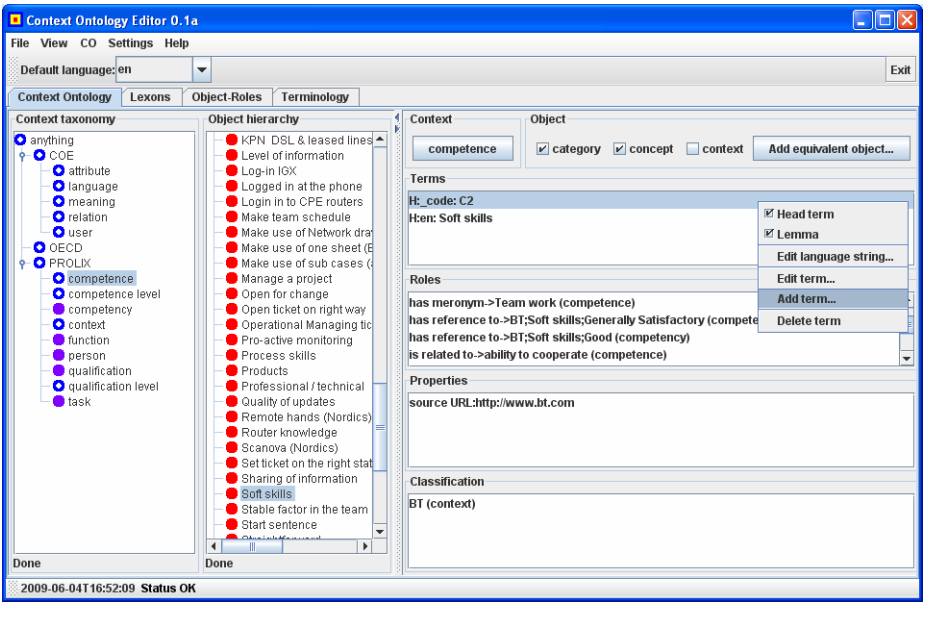

Fig. 2. The Context Ontology Editor is used to manage both the domain (*Context taxonomy*) and application ontology (*Object hierarchy*), and to link contexts and objects to terminology

[5] European Qualification Levels: http://tinyurl.com/lfrq6u

example, the relationship 'has required competency' may be used to list the required competencies for a certain task.

The reason why we prefer not to make use of description logics is that we want ontologies that are easy to build, reuse, and understand. Therefore, we prefer to use natural language to describe the intended meaning. We also believe that the consistency of the ontologies may be guaranteed by the commitments in the ontology software tools (or the CF API) without imposing formal restrictions on the ontology itself. Ultimately, the responsibility for the correct use and implementation of an ontology will always be in the hands of the application developers and the ontology engineers who use and construct the ontology.

Not all the competency information, provided by the test beds, was available in text format. For example, BT provided samples on competences, competence levels, functions, tasks, and qualifications in MS Excel format. To import such existing structured information, the Categorization Framework XML (CF XML) [2] format could be used. CF XML is a simple XML format to import and export a Categorization Framework via the CF API. The advantage of the format is that objects are identified by the combination of both the context head term and the object head term. This allows the automatic merging of two Categorization Frameworks and the modular management of the resources. Simple scripts may be used to convert structured information into CF XML.

5 Conclusion

Our experience in combining Termontography and DOGMA for knowledge engineering shows that both approaches complement each other well. Termontography is useful to develop terminological ontologies starting from structured and unstructured textual material. This allows ontology engineers to make use of the wisdom contained in existing resources, for example standards, while developing ontologies. The approach is also useful to document existing ontologies with (multilingual) terminological information.

DOGMA is especially suited to develop more formal ontologies by allowing domain experts to create and update the ontology using a combination of top-down and bottom-up techniques.

Both approaches allow for the modular development of the information resources. CF XML may be used to import / export parts of the terminology base, and lexons and ontological commitments may be imported / exported within DOGMA Studio.

Acknowledgments. The research described in this paper was sponsored in part by the EU IP 027905 Prolix project.

References

1. Coessens, B., Christiaens, S., Verlinden, R.: Ontology guided data integration for computational prioritization of disease genes. In: Meersman, R., Zahir, T., Herrero, P., et al. (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops (KSinBIT 2006). Springer, Heidelberg (2006)

2. De Baer, P., Kerremans, K., Temmerman, R.: Constructing Ontology-underpinned Terminological Resources. A Categorisation Framework API. In: Proceedings of the 8th International Conference on Terminology and Knowledge Engineering, Copenhagen (2008)
3. De Baer, P., Kerremans, K., Temmerman, R.: A Categorisation Framework Editor for Constructing Ontologically underpinned Terminological Resources. In: Proceedings of the 6th international conference on Language Resources and Evaluation, Marrakech (2008)
4. De Baer, P., Kerremans, K., Temmerman, R.: Bridging Communication Gaps between Legal Experts in Multilingual Europe: Discussion of a Tool for Exploring Terminological and Legal Knowledge Resources. In: Corino, E., Marello, C., Onesti, C. (eds.) Proceedings of the XII Euralex International Congress, Turin, Italy, September 6-9, pp. 813–818 (2006)
5. De Baer, P., Kerremans, K., Temmerman, R.: Facilitating Ontology (Re)use by Means of a Categorization Framework. In: Meersman, R., et al. (eds.) On the Move to Meaningful Internet Systems 2006. Proceedings of the AWeSOMe workshop, Montpellier, France, October 2006, pp. 126–135 (2006)
6. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 189–202. Springer, Heidelberg (2006)
7. De Troyer, O., Meersman, R., Verlinden, P.: RIDL* on the CRIS Case: a Workbench for NIAM. In: Olle, T.W., Verrijn-Stuart, A.A., Bhabuta, L. (eds.) Computerized Assistance during the Information Systems Life Cyle, pp. 375–459. Elsevier Science Publishers B.V., Amsterdam (1988)
8. Fernández, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: from ontological art towards ontological engineering. In: Proceedings of AAAI 1997 spring symposium series, workshop on ontological engineering, Stanford, CA, pp. 33–40 (1997)
9. Guarino, N., Giaretta, P.: Ontologies and knowledge bases: Towards a terminological clarification. In: Mars, N. (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)
10. Meersman, R.: Semantic web and ontologies: Playtime or business at the last frontier in computing? In: NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises, pp. 61–67 (2002)
11. Meersman, R.: Ontologies and databases: More than a fleeting resemblance. In: d'Atri, A., Missikoff, M. (eds.) OES/SEO 2001 Rome Workshop. Luiss Publications (2001)
12. Meersman, R.: The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In: Zhang, Y., Rusinkiewicz, M., Kambayashi, Y. (eds.) Proceedings of the Conference on Cooperative Database Systems (CODAS 1999), pp. 1–14. Springer, Heidelberg (1999)
13. Miller, E., Manola, F.: RDF primer. W3C recommendation, W3C (2004), http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
14. Pinto, H.S., Martins, J.P.: Ontologies: How can They be Built? Knowledge and Information Systems (6), 441–464 (2004)
15. Reiter, R.: Towards a logical reconstruction of relational database theory. In: Brodie, M., Mylopoulos, J., Schmidt, J. (eds.) On Conceptual Modelling, pp. 191–233 (1984)
16. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Record 31(4), 12–17 (2002)
17. Temmerman, R.: Towards New Ways of Terminology Description. The sociocognitive approach. John Benjamins, Amsterdam (2000)

18. Trog, D., Vereecken, J., Christiaens, S., De Leenheer, P., Meersman, R.: T-lex: A role-based ontology engineering tool. In: Meersman, R., Tari, Z., Herrero, P., et al. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1191–1200. Springer, Heidelberg (2006)
19. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Proceedings of IJCAI 1995's workshop on basic ontological issues in knowledge sharing, Montreal, Canada (1995)
20. van Harmelen, F., McGuinness, D.L.: OWL web ontology language overview, W3C recommendation, W3C (2004), http://www.w3.org/TR/2004/REC-owl-features-20040210/