OMV– Ontology Metadata Vocabulary

Jens Hartmann[1], Raúl Palma[2] and York Sure[1]

[1] Institute AIFB, University of Karlsruhe, Germany.
{hartmann,sure}@aifb.uni-karlsruhe.de
[2] Ontology Engineering Group, Laboratorio de Inteligencia Artificial,
Universidad Politécnica de Madrid, Spain.
rpalma@delicias.dia.fi.upm.es

Abstract. Ontologies have seen quite an enormous development and application in many domains within the last years, especially in the context of the next web generation, the Semantic Web. Besides the work of countless researchers across the world, industry starts developing ontologies to support their daily operative business. Currently, most ontologies exist in pure form without any additional information, e.g. missing domain specific or application related information, such as provided by Dublin Core for text documents, denoting the difficulty for academia and industry to identify, find and apply – basically meaning to reuse – ontologies effectively and efficiently. Our contribution consists of a proposal for a metadata standard, so called Ontology Metadata Vocabulary (OMV) which is based on discussions and agreement in the EU IST thematic network of excellence Knowledge Web[1]. OMV is available for download at `http://ontoware.org/projects/omv`. The current version 0.7 has been validated as OWL Lite by the WonderWeb OWL Ontology Validator (see `http://phoebus.cs.man.ac.uk:9999/OWL/Validator`).

1 Introduction

Ontologies are commonly used for a shared means of communication between computers and between humans and computers. To reach this aim, ontologies should be represented, described, exchanged, shared and accessed based on open standards. Consider, as an example, the W3C standardized web ontology language OWL [10]. At the moment, most ontologies exist in pure form without any additional information, e.g. domain of interest or authorship information, such as provided by Dublin Core for text documents. This burden hamper academia and industry to identify and apply – basically meaning to reuse – ontologies effectively and efficiently. As consequence, the reuse of ontologies as in the vision of the Semantic Web is nowadays a hard task if not impossible.

We argue that metadata in the sense of machine processable information for the Web[2] helps to improve accessibility and reuse of ontologies. Furthermore, it can provide other useful resource information to support maintenance. Thus, we claim that metadata not only help when applied (or, attached) to documents, but also to ontologies

[1] `http://knowledgeweb.semanticweb.org`
[2] `http://www.w3.org/Metadata/`

themselves. We contend that there is a demand for an ontology metadata standard which would enable amongst others the access and reuse of ontologies. To achieve this goal, it is necessary to agree on a standard for ontology metadata, that is a common set of terms and definitions describing ontologies, so called metadata vocabulary. Such a vocabulary is required to annotate ontologies according the vocabulary comparable to the well-known process of annotating documents [6]. Further, if ontologies are annotated using ontology metadata standards, an appropriate technology infrastructure is required as well. This includes tools and metadata repositories which comply to the ontology metadata standard. Such tools and repositories typically should support the engineering process, maintenance and distribution of ontologies.

Our contribution consists of a proposal for a metadata standard reflecting the most relevant properties of ontologies for supporting their reuse, so called Ontology Metadata Vocabulary (OMV), which is based on discussions and agreement in the EU IST thematic network of excellence Knowledge Web.

This paper is organised as follows: Section 1 provides the introduction. The developed metadata vocabulary is given in Section 3. Followed by two existing applications of OMV: The P2P system Oyster and the metadata portal ONTHOLOGY in Section 4. In Section 5 we provide related work and conclude in Section 6.

2 Requirements

As an initial step towards a standardized vocabulary, we analysed requirements for ontology metadata. Several aspects are similar to other metadata standards, like Dublin Core. However, important differences like the conceptual models (semantics) behind ontologies require a detailed analysis and require a different representation of metadata about ontologies. In a nutshell, an ontology normally reflects the (i) conceptualization from persons about a specific task or domain which then is (ii) realised by an ontology engineering process [12].

As a result, the main identified requirements are the following:

- **Accessibility:** Metadata[3], especially about ontologies, must be accessible and processable for humans and machines.
- **Usability:** A majority of users should be able to apply Metadata easily.
- **Reuse of Ontologies:** As ontologies are a core technology for the Semantic Web, its metadata should reflect key issues of the Semantic Web as well. In particular **reuse** and **sharing** of knowledge.
- **Conceptualisation vs. Realisation:** Metadata must reflect (and also distinguish between) a semantic *conceptualisation* and its particular *realisation* as a concrete ontology document.
- **Interoperability:** Metadata should be interoperable and conform to the major representation languages currently being used for Semantic Web applications. Indeed, this means that a metadata vocabulary should be representable e.g. in F-Logic and OWL as well.

[3] In the requirements we mean ontology metadata.

– **Documentary:** Documentary aspects of metadata like information about *technical, statistical, accessibility, management information, etc.* should be provided.
– **Extensibility:** Reflecting special user needs, it is required that beyond such standard metadata facts can be added and extended easily.
– **Expressiveness:** Metadata must be expressive enough to represent all desired aspects, as presented above.

The main aspects are *Conceptualisation vs. Realisation* and *Reuse of Ontologies* which should be reflected by any ontology metadata. Already now, it is possible to capture several technical properties of ontologies, like *used syntax* or *number of classes*. Such properties, as shown e.g. by [3], can be derived automatically. Besides technical properties which are obviously relevant, there is a strong demand for representing conceptual metadata, like authorship information, categorizations or underlying methodologies. As consequence, representing these issues by a vocabulary requires an expressive language for the metadata itself which makes it impossible to reuse any existing metadata schema.

3 Ontology Metadata Vocabulary

3.1 Conceptualisation vs. Realisation

OMV distinguishes between an **ontology base** and an **ontology document**. This separation is based on following observation: any existing ontology document has some kind of *core idea* (conceptualisation) behind. From an ontology engineering perspective, initially a person develops such *core idea* of what should be modeled (and maybe how) in his mind. Further, this initial conceptualisation might be discussed with other persons and after all, an ontology will be *realised* using an ontology editor and stored in a specific format. Over time, there might be created several *realisations* of this initial *cconceptualisation* in many different formats, e.g. in RDF(S) [1] or OWL [10].

Therefore we distinguish between an *ontology base* and an *ontology document*:

– **[Ontology Base]:** An *Ontology Base (OB)* represents the abstract or core idea of an ontology, so called conceptualisation. It describes the core properties of an ontology, independent from any implementation details. For a general illustration of the relationship of an OB and OD, we refer to figure 1.
– **[Ontology Document]:** An *Ontology Document (OD)* represents a specific realization of an ontology base. Therefore, it describes properties of an ontology that are related to the realization or implementation.

The distinction between an OB and OD leads to an efficient mechanism, e.g. for tracking several versions and evolvements of ontologies as well as for different representations of one knowledge model (conceptualisation) in different languages. In particular, such an *ontology base* can be seen as representation of the conceptual model behind an ontology. Technically, an ontology base and an ontology document are modeled as two separate classes, with the relation `realizes` from the ontology document to the ontology base. This means that there may be many possible ontology documents for one ontology base, but one ontology document can only realize one ontology base.

Fig. 1. Relationship between OB and OD

Normally, an OD should not be able to exist without a corresponding OB. However, for practical reasons, we allow the existence of each class independently of each other. Hence, we cannot assume that every existing ontology will be annotated by its original author who might create an OB for his ontology. However, automatically extracting syntactical properties of an existing ontology is quite simple. Then, such *minimal OD* would exist without a concrete OB.

The main classes and properties of the OMV ontology are illustrated in figure 2. Please notice, that not all classes and properties are included. It is only to demonstrate the main idea behind OMV. The complete ontology is described in [7] and is available for download at `http://ontoware.org/projects/omv/`.

It should be noticed that there exist several properties within OB and OD which look similar at first. However, they have different meanings and semantics. Exemplary, think of an ontology engineer `A` developing an ontology in RDF(S) syntax and annotating it with OMV. Then, the properties of an OB and OD individual are quite similar. Exemplary, both would have the same `party` as `creator` and so on. However, over time, there might be another engineer `B` with similar needs according to the OB from `A`. Hence, `B` reuses the OB from `A` and only creates a new OD, e.g. realising the OB in OWL instead of RDF(S). As a result, a new OD would be created for this ontology and most of the properties would be different.

3.2 Basic classes of OMV

As mentioned above, the OMV models the two main classes OB and OD for representing core information about ontologies. However, additional classes are required to represent and support the reuse of ontologies by such metadata vocabulary, especially in the context of the Semantic Web. Hence, we modeled, as shown in figure 2, further classes and properties representing *environmental information* and *relations*. We will briefly discuss these classes in the following.

In typical ontology engineering a `Person` (or multiple) or an `Organisation` as a whole are developing ontologies. We group these two classes under the generic class `Party` by a `subclass-of` relation. A `Party` can *create*, *contribute* and *review* an `OntologyBase` resp. an `OntologyDocument`. We here distinguish between the development of an OB and OD.

Further, tools such as ontology editors can be referred to by the class `OntologyEngineeringTool` which itself can be `developedBy` a `Party`. The different existing syntactical representations and ontology languages are representable by `OntologySyntax` and `OntologyLanguage`. OMV further consists of the class

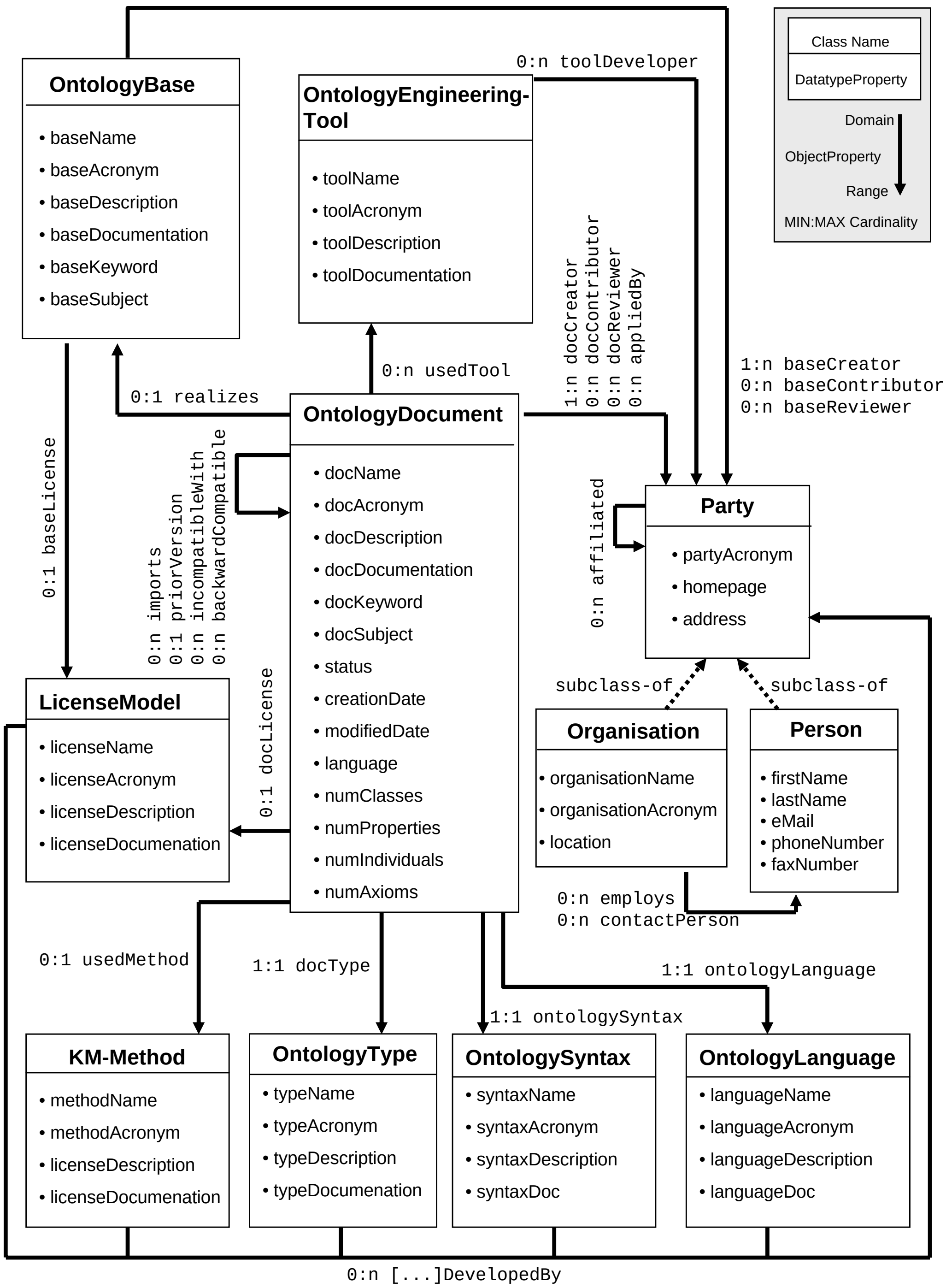

Fig. 2. General OMV overview

KM-Method to make explicit the methodology (or methodologies) used during engineering.

Ontologies might be categorized by different types of ontologies, exemplary think of *domain, lingusitic* or *upper-level* ontologies. Those types can be modeled by the class OntologyType. For industry it might be relevant to propose usage licenses which can be realized by the class LicenseModel. So, that each OntologyBase or OntologyDocument is related to a pre-defined LicenseModel.

The presented OMV tries to model as much information about ontologies and the important aspects for ontology reuse as possible and at the same time intends to stay as simple as possible.

4 Existing Applications of OMV

We now present running applications based on the proposed OMV. In detail, we present two complementary applications, namely the *decentralised* P2P system Oyster and the *centralised* metadata portal ONTHOLOGY. In general, the two tools differ in their usage perspective and are appropriate for different tasks. However, as we will see, only the combined application of both tools will offer users the full potential of ontology metadata management.

4.1 Oyster – A Peer-to-Peer System for Sharing Ontologies

Oyster[4] is a java-based system that exploits semantic web techniques in order to provide an innovative and useful solution for exchanging and reusing ontologies. For this purpose, Oyster provides facilities for managing, searching and sharing ontology metadata in a P2P network, thereby implementing the OMV proposal for the standard set of ontology metadata.

Oyster offers a user driven approach where each peer has its own local repository of ontology metadata and also has access to the information of others repositories, thus creating a virtual decentralized Ontology repository. The Oyster client on its own (e.g. disconnected from the P2P network) will already provide added value to its users as it will give developers an overview and search facilities of his/her own ontology metadata stored in its local repository. The goal is a decentralized knowledge sharing environment using Semantic Web technologies that allows developers to easily share ontology documents.

The Oyster system has been implemented as an instance of the Swapster system architecture [5]. It uses ontologies extensively in order to provide some of its main functions: creating and importing data, formulating queries, routing queries and processing answers.

[4] http://oyster.ontoware.org
[5] http://swap.semanticweb.org/

4.2 ONTHOLOGY – A Central Ontology Metadata Portal

As the importance of metadata increases with the number of existing ontologies, the demand for supporting technologies like storage and access techniques becomes important as well. We present the conceptual design of a centralised ontology metadata portal and its implementation, so-called ONTHOLOGY[6] standing for an "anthology of ontologies".

Centralised systems allow to reflect long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. Such well accepted ontologies and in particular their metadata need to be stored in a central metadata portal which can be accessed easily by a large number of users whereby the management procedures are well defined. Hence, a main goal of a centralised metadata portal is to act as large evidence storage of metadata resp. their related ontologies to facilitate access, reuse and sharing as required for the Semantic Web.

A metadata portal mainly consists of *a large data repository* in which metadata can be stored. Exemplary, Sesame[7] or KAON[8] can be used as back-end metadata repository. Furthermore, *access* and in particular the *managament* of the repository must be guaranteed, too. Therefore, ONTHOLOGY is based on SEAL, the AIFB conceptual architecture for building SEmantic portALs. Further information can be found at [8].

4.3 Discussion

Both presented applications are covering a variety of different tasks. Indeed, users who wants to store metadata individually similar to managing his personal favorite song list, a repository is required to which a user has full access and can perfom any operation (e.g. create, edit or delete metadata) without any consequences to other users. Exemplary, users from academia or industry might use a personal repository for a task-dependant investigation or ontology engineers, might use it during their ontology development process to capture information about different ontology versions. We argue, that a decentralised system is the technique of choice, since it allows the maximum of individuality while it still ensures exchange with other users.

Centralised systems allow to reflect long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. Such well accepted ontologies and in particular their metadata need to be stored in a central metadata portal which can be accessed easily by a large number of users.

The benefit of connecting both systems lies mainly in the simple use of existing ontology metadata information within Oyster. So, while users are applying or even developing their own ontologies they can manage their own metadata along with other existing metadata in one application (in Oyster). If some metadata entries from Oyster have reached a certain confidence, an import into ONTHOLOGY can be performed

[6] http://www.onthology.org/
[7] http://www.openrdf.org/
[8] http://kaon.semanticweb.org/

easily. In combination, both systems ensure efficient and effective ontology metadata management for various use cases.

5 Related Work

We will briefly mention related metadata standards, including in particular those ones relevant for the Semantic Web. The **Dublin Core (DC)** metadata standard [2] is a simple yet effective element set for describing a wide range of networked resources. It includes two levels: Simple and Qualified. Simple DC comprises fifteen elements; Qualified DC includes an additional element as well as a group of element refinements (or qualifiers) that refine the semantics of the elements in ways that may be useful in resource discovery. **FOAF** [5], or "Friend Of A Friend", provides a way to create machine-readable Web homepages for people (their interests, relationships and activities), groups, companies and other kinds of things. To achieve this, FOAF project use the "FOAF vocabulary" to provide a collection of basic terms that can be used in these Web pages. The initial focus of FOAF has been on the description of people. The Semantic Web search engine **SWOOGLE** [3] makes use of particularly those metadata which can be extracted automatically. Our approach includes and extends this metadata vocabulary. Ideally, future versions of SWOOGLE would also take into account the additional vocabulary defined in OMV. There exist some similar approaches to our proposed solution to share ontologies, but in general they are limited in scope. E.g. the **DAML ontology library** [9] provides a catalog of DAML ontologies that can be browsed by different properties. The **FIPA ontology service** [11] defines an agent wrapper of open knowledge base connectivity. Finally we mention the **SchemaWeb Directory** [4] that is a repository for RDF schemas expressed in RDFS, OWL and DAML+OIL.

6 Conclusion

To conclude, reusing existing ontologies is a key issue for sharing knowledge on the Semantic Web. Our contribution aims at facilitating reuse of ontologies which are previously unknown for ontology developers by providing an Ontology Metadata Vocabulary (OMV) and two prototypical applications for decentralized (Oyster) and centralized (ONTHOLOGY) sharing of ontology metadata based on OMV.

OMV has been discussed and agreed upon in the industry area of the EU thematic network of excellence Knowledge Web (KWeb). Next steps include the standardization of OMV on a wider scope by particularly including non-KWeb parties in this process, followed by a close cooperation with tool providers for ontology engineering environments and applications providers for e.g. ontology based search engines to enhance their tools with support for OMV. The agreement and application of a standard on a global level will greatly facilitate the reuse of ontologies for all participating parties.

OMV is available for download at `http://ontoware.org/projects/omv`. The current version 0.6 has been validated as OWL Lite by the WonderWeb OWL Ontology Validator (see `http://phoebus.cs.man.ac.uk:9999/OWL/Validator`).

References

1. D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Rec. 10 February 2004, 2004. available at http://www.w3.org/TR/rdf-schema/.
2. Dublin Core. http://dublincore.org/.
3. Li Ding et al. Swoogle: A search and metadata engine for the semantic web. In *In Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 58–61, November 2004.
4. SchemaWeb directory. http://www.schemaweb.info/.
5. FOAF. http://www.foaf-project.org/.
6. S. Handschuh and S. Staab. Authoring and annotation of web pages in CREAM. In *The Eleventh Int. World Wide Web Conf. (WWW2002), Honolulu, Hawaii, USA 7-11 May*, pages 277–293, 2002.
7. J. Hartmann and R. Palma. OMV - Ontology Metadata Vocabulary for the Semantic Web, 2005. v. 1.0, available at http://ontoware.org/projects/omv/.
8. J. Hartmann and Y. Sure. An infrastructure for scalable, reliable semantic portals. *IEEE Intelligent Systems*, 19(3):58–65, May/June 2004.
9. DAML Ontology Repository. http://www.daml.org/ontologies/.
10. M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Rec. 10 February 2004, available at http://www.w3.org/TR/owl-guide/.
11. H. Suguri et al. Implementation of fipa ontology service. In *Proc. of the Workshop on Ontologies in Agent Systems, 5th Int. Conf. on Autonomous Agents Montreal, Canada*, 2001.
12. C. Tempich, H. S. Pinto, Y. Sure, and S. Staab. An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies (diligent). In A. Gmez-Prez and J. Euzenat, editors, *2nd European Semantic Web Conference, ESWC 2005*, volume 3532 of *LNCS*, pages 241–256, Heraklion, Crete, Greece, MAY 2005. Springer.