
O M V

Ontology Metadata Vocabulary for the Semantic Web

Jens Hartmann, Raúl Palma and Elena Paslaru Bontas

?

| | |
|---------------------|------------------------------|
| Document Identifier | OMV Report |
| Project | Ontology Metadata Vocabulary |
| Version | v1.0 |
| Date | February, 2006 |
| State | Final |
| Distribution | public |

OMV Consortium

The Ontology Metadata Vocabulary is based on discussions and agreement among the following consortium.

University of Karlsruhe (UKARL) - Coordinator

Institut für Angewandte Informatik und Formale
Beschreibungsverfahren - AIFB
Universität Karlsruhe
D-76128 Karlsruhe
Germany
Fax: +49 721 6086580, Phone: +49 721 6086554
Contact person: Jens Hartmann
E-mail address: hartmann@aifb.uni-karlsruhe.de

Universidad Politécnica de Madrid (UPM)

Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Fax: +34-913524819, Phone: +34-913367439
Contact person: Raul Palma, Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

Freie Universität Berlin (FU Berlin)

Takustrasse 9
14195 Berlin
Germany
Fax: +49 30 83875220, Phone: +49 30 83875223
Contact person: Elena Paslaru Bontas
E-mail address: paslaru@inf.fu-berlin.de

Changes

Version 0.9.5

- Strict naming conventions applied for OC and OI. Hence several re-namings.
- Moved *Reviewer attributes into Evaluation extension
- Changed several cardinalities
- ReNaming: OI.language - naturalLangugage
- ReNaming: OI.usedTool - usedOntologyEngineeringTool
- ReNaming: OI.ontologyLanguage - representedByOntologyLanguage
- ReNaming: OI.ontologySyntax - representedByOntologySyntax
- ReNaming: OI.versionInfo - implementationVersion
- ReNaming: OI.formalityLevel - implementationFormalityLevel
- New Class *OntologyFormalityLevel*
- ReNaming: OI.ontologyURL - implementationURL
- ReNaming: OI.imports - implementationImports
- ReNaming: OI.priorVersion - implementationPriorVersion
- ReNaming: OI.backwardCompatibleWidth - imeplementationBackwardCompatibleWith
- ReNaming: OI.incompatibleWith - implementationIncompatibleWith
- ReNaming: OI.num* - implementationNum*
- ReNaming: Party.developedTool - developedEngineeringTool
- ReNaming: Party.specifiedLicense - specifiedLicenseModel
- ReNaming: Person.* - Person.person*
- ReNaming: Organisation.* - Organisation.organisation*
- ReNaming: Organisation.contactPerson - hasContactPerson

Version 0.9.1

- Rename of *ontology document* into *OntologyImplementation*
- Moved class `OntologyReview` into `Evaluation Extension`
- Several re-namings
- New extensions: `OntologyApplication`, `OntologyUsage`, `Directives`, ...
- Introduced property `formalityLevel` for `OntologyImplementation`
- New class `OntologyTask`

Version 0.9

- Rename of *ontology base* into *conceptualisation*
- Introduced class `OntologyReview`
- New objectproperty `contributorOfReview` for class `party`
- Several re-namings due introduction of *conceptualisation*
- Introduced class `Location`. Hence, removed property `adress` from class `party`.

Executive Summary

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. authorship information, such as provided by Dublin Core for text documents. This burden makes it difficult for academia and industry e.g. to identify, find and apply – basically meaning to reuse – ontologies effectively and efficiently.

Hence we propose a metadata vocabulary for ontologies, so called Ontology Metadata Vocabulary (OMV).

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Preliminary considerations | 2 |
| 2.1 | Terminology | 2 |
| 2.2 | Naming conventions | 3 |
| 2.3 | Notations | 3 |
| 3 | Ontology Metadata Requirements | 5 |
| 4 | OMV - Ontology Metadata Vocabulary | 7 |
| 4.1 | Core and Extensions | 7 |
| 4.2 | Ontological representation | 7 |
| 4.3 | Conceptualisation vs. Implementation | 8 |
| 4.4 | OMV metadata entities | 9 |
| 5 | OMV Core Ontology | 12 |
| 5.1 | OntologyConceptualisation | 12 |
| 5.2 | OntologyImplementation | 19 |
| 5.3 | OntologyType | 33 |
| 5.3.1 | Pre-defined ontology types | 35 |
| 5.4 | LicenseModel | 35 |
| 5.5 | OntologyEngineeringMethod | 39 |
| 5.6 | OntologyEngineeringTool | 42 |
| 5.7 | OntologySyntax | 44 |
| 5.8 | OntologyRepresentationLanguage | 47 |
| 5.9 | KnowledgeRepresentationParadigm | 50 |
| 5.9.1 | Pre-defined knowledge representation paradigms | 52 |
| 5.10 | FormalityLevel | 53 |
| 5.10.1 | Pre-defined formality levels | 54 |
| 5.11 | OntologyApplicationSystem | 55 |
| 5.11.1 | Pre-defined ontology types | 57 |
| 5.12 | OntologyTask | 59 |
| 5.12.1 | Pre-defined ontology types | 61 |

CONTENTS

5.13 Party 63

5.14 Person 69

5.15 Organisation 72

5.16 Location 74

6 Using Metadata 76

7 Conclusion 77

Chapter 1

Introduction

Ontologies are intended to be used as a shared means of communication between computers and between humans and computers. A core requirement for the achievement of this goal is the usage of open standards and technologies for the representation, description, access and exchange of the ontological sources. Consider, for example, the W3C standardized Web ontology language OWL [10]. Using this representation language instead of a proprietary format would clearly increase the usability of an ontology at Web scale. The same applies for the means employed to describe existing ontologies or for the technological infrastructure supporting their management and exchange.

In contrast to plain Web documents, the majority of implemented ontologies are currently put into widespread use on the Web without any additional metadata information. This deficiency seriously affects the reusability of Semantic Web ontologies: without any metadata information potential ontology users can not find and deploy them effectively and efficiently. In order to cope with this problem, it is necessary to agree on a *standard for ontology metadata*, a vocabulary of terms and definitions describing ontologies. Replicating the positive experiences in other information management areas e.g. Digital Libraries, implementing such a vocabulary in conjunction with a solid technological infrastructure for creating, maintaining and distributing metadata is expected to increase the real value of ontologies by facilitating their wide scale sharing and reuse.

In this report we describe our contribution to the alleviation of this situation: the ontology metadata standard OMV (**O**ntology **M**etadata **V**ocabulary), which specifies reusability-enhancing ontology features for human and machine processing purposes. The remaining of this report is organized as follows: after clarifying the applied terminology and naming conventions (Chapter 2) we perform an analysis of the requirements for the realization of the proposed ontology metadata scheme in Chapter 3. We introduce the main ideas behind the OMV vocabulary and give a detailed description of the metadata and its extensions in Chapters 4, 5 and ??, respectively. The usage of the metadata is illustrated in Chapter 6. Finally we summarize our work and sketch its current limitations and future research directions in Chapter 7.

Chapter 2

Preliminary considerations

2.1 Terminology

In this section we clarify our understanding of the concept of metadata for ontologies:

- **Metadata** - data about data
- **Ontology Metadata** - metadata which provides information about ontologies
- **Metadata Ontology** - an ontology representing metadata information
- **Metadata Entity** - an element of a metadata scheme
- **OMV** - **O**ntology **M**etadata **V**ocabulary - The acronym of the proposed ontology metadata scheme
- **Metadata Categories** - we differentiate among the following three occurrence constraints for metadata elements, according to their impact on the prospected reusability of the described ontological content:
 - **Required** - mandatory metadata elements. Any missing entry in this category leads to an incomplete description of the ontology.
 - **Optional** - important metadata facts, but not strongly required.
 - **Extensional** - specialized metadata entities, which are not considered to be part of the core metadata scheme.

Complementary to this classification we organize the metadata elements according to the type and purpose of the contained information as follows:

- **General** - elements providing general information about the ontology.

- **Availability** - information about the location of the ontology (e.g. its URI or URL where the ontology is published on the Web)
- **Applicability** - information about the intended usage or scope of the ontology.
- **Format** - information about the physical representation of the resource. In terms of ontologies these elements include information about the representation language(s) in which the ontology is formalized.
- **Provenance** - information about the organizations contributing to the creation of the ontology.
- **Relationship** - information about relationships to other resources. This category include versioning, as well as conceptual relationships such as extensions, generalization/specialization and imports.
- **Statistics** - various metrics on the underlying graph topology of an ontology (e.g. number of classes)
- **Other** - information not covered in the categories listed above.

Note that the introduced classification dimensions are not intended to be part of the metadata scheme itself, but will be taken into consideration by the implementation of several metadata support facilities. The first dimension is relevant for a metadata creation service in order to ensure a minimal set of useful metadata entries for each of the described resources. The second can be used in various settings mainly to reduce the user-perceived complexity of the metadata scheme whose elements can be structured according to the corresponding classes.

2.2 Naming conventions

2.3 Notations

In the following we give an overview of the notations used in this report for representing the OMV metadata entities. The metadata scheme is formalized as a Semantic Web ontology in OWL (the introduced examples conform to the OWL RDF/XML syntax).

Further on OMV uses the following namespaces:

```
owl = "http://www.w3.org/2002/07/owl#"
rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
rdfs = "http://www.w3.org/2000/01/rdf-schema#"
xsd = "http://www.w3.org/2001/XMLSchema#"
omv = "http://omv.ontoware.org/omv-core-1.0#"
```

A metadata entity can be a *class* or *property* (*DatatypeProperty*, *ObjectProperty*) of the OMV ontology. Every entity is described using the template illustrated by Table 2.1.

| Name of the OMV metadata entity | |
|--|--|
| Name | (Case sensitive) name of the metadata entity. |
| Type | The type of ontological primitive used to represent the entity in OWL: <code>Class</code> , <code>ObjectProperty</code> or <code>DatatypeProperty</code> . |
| Identifier | Unique identifier used for this entity. |
| Occurrence Constraint | One of the following: <code>required</code> , <code>optional</code> or <code>extensional</code> . |
| Category | The content/purpose category the entity belongs to, as introduced above. |
| Definition | A short definition of the purpose, which might be elaborated in the <code>comments</code> tag. |
| Domain | Domain of OMV entity (for OWL properties) |
| Range | Range of OMV entity (for OWL properties). |
| Cardinality | Cardinality of OMV entity (MIN:MAX). |
| OMV version | OMV version, in which the entity has been introduced. |
| Comments | Detailed description of the entity. |

Table 2.1: Template for a metadata entry

Chapter 3

Ontology Metadata Requirements

We elaborated an inventory of requirements for the metadata model as a result of a systematic survey of the state of the art in the area of ontology reuse. Besides analytical activities, we conducted extensive literature research, which focused on theoretical methods [8, 1, 4], but also on case studies on reusing existing ontologies [12, 9, 7], in order to identify the real-world needs of the community w.r.t. a descriptive metadata format for ontologies. Further on, the requirements analysis phase was complemented by a comparative study of existing (ontology-independent) metadata models and of tools such as ontology repositories and libraries (implicitly) making use of metadata-like information. Several aspects are definitely similar to other metadata standards such as Dublin Core. Differences arise however if we consider the semantic nature of ontologies, which are much more than plain Web information sources. In accordance to one of the major principles in Ontological Engineering an ontology comprises a conceptual model of a particular domain of interest, represented at knowledge level, and multiple implementations using knowledge representation languages. These two components are characterized by different properties, can be developed and maintained separately. The main requirements identified in this process step are the following:

Accessibility: Metadata should be accessible and processable for both humans and machines. While the human-driven aspects are ensured by the usage of natural language concept names, the machine-readability requirement can be implemented by the usage of Web-compatible representation languages (such as XML or Semantic Web languages, see below).

Usability: This requirement states for the necessity of building a metadata model which 1). reflects the needs of the majority of ontology users, as reported by current case studies in ontology reuse, but in the same time 2). allows proprietary extensions and refinements in particular application scenarios. From a content perspective, usability can be maximized by taking into account multiple metadata types, which correspond to specific viewpoints on the ontological resources and are applied in various application tasks. Despite the broad understanding of the metadata concept

and the use cases associated to each definition, several key aspects of metadata information have already established across computer science fields [6]:

- **Structural metadata** relates to statistical measures on the graph structure underlying an ontology. In particular we mention the number of specific ontological primitives (e.g. number of classes, instances). The availability of structural metadata influences the usability of an ontology in a concrete application scenario, as size and structure parameters constraint the type of tools and methods which are applied to aid the reuse process.
- **Descriptive metadata** relates to the domain modelled in the ontology in form of keywords, topic classifications, textual descriptions of the ontology contents etc. This type of metadata plays a crucial role in the selection of appropriate reuse candidates, a process which includes requirements w.r.t. the domain of the ontologies to be re-used.
- **Administrative metadata** provides information to help manage ontologies, such as when and how it was created, rights management, file format and other technical information.

Interoperability: Similarly to the ontology it describes, metadata information should be available in a form which facilitates metadata exchange among applications. While the syntactical aspects of interoperability are covered by the usage of standard representation languages (see “Accessibility”), the semantical interoperability among machines handling ontology metadata information can be ensured by means of an formal and explicit representation of the meaning of the metadata entities, i.e. by conceptualizing the metadata vocabulary itself as an ontology.

Separation between Knowledge and Implementation Levels : In accordance to the recommendations of current ontology engineering methodologies it should be clearly distinguished between the conceptual model of an ontology and particular implementations (in various languages, syntaxes, versions etc.).

Chapter 4

OMV - Ontology Metadata Vocabulary

This chapter gives an overview of the core design principles applied for the realization of the OMV metadata scheme, which is described in detail in the remaining of the report.

4.1 Core and Extensions

Following the usability constraints identified during the requirements analysis we decided to design the OMV scheme modularly; OMV distinguishes between the OMV Core and various OMV Extensions. The former captures information which is expected to be relevant to the majority of ontology reuse settings. However, in order to allow ontology developers and users to specify task- or application-specific ontology-related information we foresee the development of OMV extension modules, which are physically separated from the core scheme, while remaining compatible to its elements.

4.2 Ontological representation

Due to the high accessibility and interoperability requirements, as well as the nature of the metadata, which is intended to describe Semantic Web ontologies, the conceptual model designed in the previous step was implemented in the OWL language. An implementation as XML-Schema or DTD was estimated to restrict the functionality of the ontology management tools using the metadata information (mainly in terms of retrieval capabilities) and to impede metadata exchange at semantical level. Further on, a language such as RDFS does not provide a means to distinguish between required and optional metadata properties. The implementation was performed manually by means of a common ontology editor.

4.3 Conceptualisation vs. Implementation

OMV core distinguishes between an **ontology conceptualisation** and its **implementation(s)** in concrete representation languages. This is based on a core principle of knowledge engineering, which clearly recommends the separation between knowledge and implementation level. Any existing ontology on the Web is the implementation of an (abstract) *conceptual model* i.e. the ontology conceptualisation. For example, the intranet of a company might be conceptualized as a graph of workstations and servers and might be implemented using the OWL language. In this case the graph structure representing the intranet represents the *ontology conceptualization*, which can be formalized in various way, from an abstract image in the mind of an individual to graphics of vertices and edges or natural languages. The conceptual model is implemented in a particular representation language and stored in a file system, giving birth to a so called *ontology implementation*.

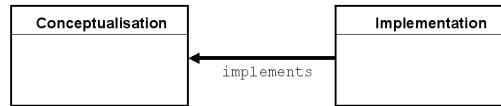
We two concepts are defined as follows:

- **[Ontology Conceptualisation]:** An *Ontology Conceptualisation (OC)* represents the (abstract) core model or idea behind an ontology. It describes the core properties of an ontology, independently from any implementation details. For a general illustration of the relationship between conceptualisations and implementations, we refer to Figure 4.1.
- **[Ontology Implementation]:** An *Ontology Implementation (OI)* represents a specific realization of a conceptualisation. Therefore, it describes properties of an ontology that are related to the realization or implementation.¹

The distinction between the two concepts provides an efficient mechanism for the realization of several ontology management utilities, such as tracking several versions and the evolution flow of an ontology or handling different representations of the same knowledge model. Technically a conceptualisation and an ontology implementation are modeled as two separate classes, with the relation `realizes` from the ontology document to the conceptualisation. This means that there may be many possible ontology implementations for one conceptualisation, but one ontology implementation can `implement` one conceptualisation.

From a theoretical point of view an instance of an OI can not exist without a corresponding conceptualisation. 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 the members of its engineering team, who are likely to have authored the ontology conceptualization. Once an ontology is implemented the extraction of the underlying conceptual structure is realizable only by means of non-trivial re-engineering activities. However, automatically extracting syntactical properties of an

¹Note that an ontology representation does not necessarily map to a physical document, as a Semantic Web ontology might be stored in multiple Web documents linked by `owl:import` statements.



1

Figure 4.1: Relationship between conceptualisation and implementation

existing ontology is quite simple. In this case a *minimal OI* description would exist without a concrete conceptualisation.

The main classes and properties of the OMV ontology are illustrated in Figure 4.2.²

Ontology conceptualisations and implementation entries share many features. However, they have different meanings and are used for different purposes in an ontology engineering scenario. For illustration, think of an ontology engineer A developing an ontology in RDF(S) syntax and annotating it with OMV. Then, the properties of the conceptualisation and OI individual are quite similar (e.g. the same `party` as `creator`, the same `domain`). A realization of the same conceptualisation in OWL instead of RDF(S) would however generate a new OI entry, which is comparatively different from the metadata information captured for the original OC.

4.4 OMV metadata entities

Additionally to the two main classes `Ontology Conceptualisation` and `Ontology Implementation` the metadata scheme contains further elements describing various aspects related to the creation, management and usage of an ontology. We will briefly discuss these in the following. In a typical ontology engineering process `Persons` or `Organisation(s)` are developing ontologies. We group these two classes under the generic class `Party` by a `subclass-of` relation. A `Party` can have several locations by referring to a `Location` individual and can *create*, *contribute* to and *review* ontological resources i.e. `Ontology Conceptualisation` or `Ontology Implementations`. Review details and further information can be captured in an

²Please notice, that not all classes and properties are included. The ontology is available for download in several ontology formats at <http://ontoware.org/projects/omv/>

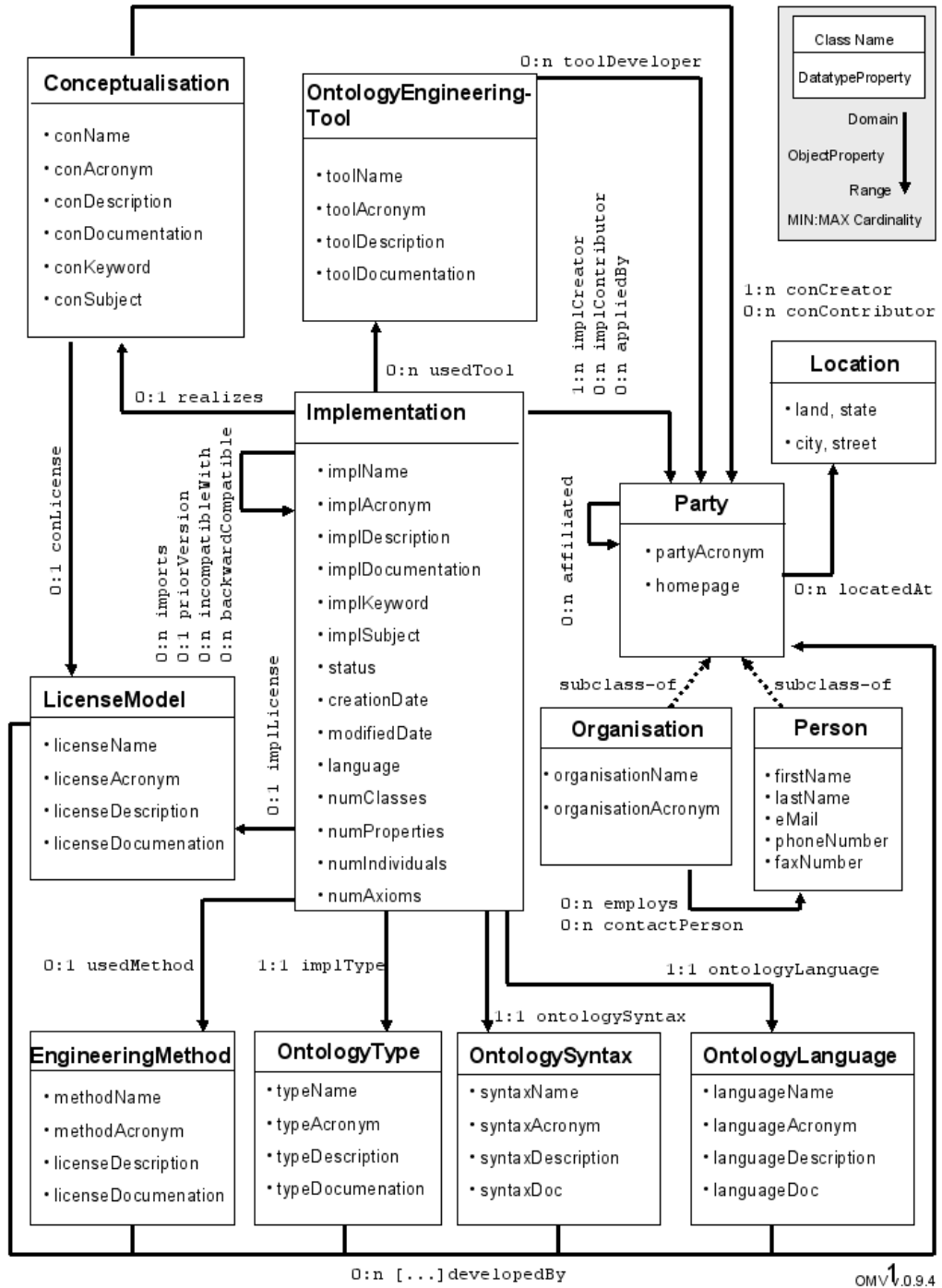


Figure 4.2: OMV overview

extensional OMV module (see Chapter ??). Further on we provide information about the engineering process the ontology originally resulted from in terms of the classes `OntologyEngineeringMethod`, `OntologyEngineeringTool` and the attributes `version`, `status`, `creationDate` and `modificationDate`. Again these can be elaborated as an extension of the core metadata scheme. The usage history of the ontology is modelled by the classes `OntologyTask`, `OntologyApplicationSystem` and `LicenceModel`. The scheme also contains a representation of the most significant intrinsic features of an ontology. Details on ontology languages are representable with the help of the classes `OntologySyntax`, `OntologyRepresentationLanguage` and `KnowledgeRepresentationParadigm`. Ontologies might be categorized along a multitude of dimensions. One of the most popular classification differentiates among application, domain, core, task and upper-level ontologies. A further classification relies on their level of formality and types of KR primitives supported, introducing catalogues, glossaries, thesauri, taxonomies, frames etc. as types of ontologies. These can be modeled as instances of the class `OntologyType`, while generic formality levels are introduced with the help of the class `FormalityLevel`. The domain the ontology describes is represented by the class `OntologyDomain` referencing a pre-defined topic hierarchy such as the DMOZ hierarchy. Further content information can be provided as values of the `DatatypeProperties` `description`, `keywords`, and `documentation`. Finally the metadata scheme gives an overview of the graph topology of an `OntologyImplementation` with the help of several graph-related metrics represented as integer values of the `DatatypeProperties` `numClasses`, `numProperties`, `numAxioms`, `numIndividuals`.

We now turn to a detailed description of the OMV model and its planned extensions.

Chapter 5

OMV Core Ontology

In the following we introduce the metadata elements of OMV , the first metadata standard for ontologies. As aforementioned, OMV is formalized as an OWL ontology. A metadata element is modelled either by means of classes and individuals or by means of valued properties. The decision for one of these two alternatives was justified by the complexity of the corresponding metadata element. If the value/content of a metadata element can be easily mapped to conventional data types (numerical, literal, list values) the metadata element is usually represented as a DatatypeProperty. Complex metadata elements which do not fall into the previous category are modelled by means of additional classes linked by ObjectProperties.

The description of the model is grouped along the core classes of the ontology. For each class we describe the meaning of its properties and additional usage and occurrence constraints.

5.1 OntologyConceptualisation

The class `OntologyConceptualisation` captures information about the knowledge model of an ontology. It is shared by multiple `OntologyImplementations` representing the model in particular representation languages.

| OntologyConceptualisation | |
|---------------------------|--|
| Name | OntologyConceptualisation |
| Type | class |
| Identifier | OC |
| Definition | The knowledge model of an ontology |
| OMV version | 0.9 |
| Comments | Renamed from <code>OntologyBase</code> to <code>OntologyConceptualisation</code> . |

Table 5.1: Class: `OntologyConceptualisation`

| hasImplementation | |
|--------------------------|--|
| Name | hasImplementation |
| Type | ObjectProperty, inverseOf{implements} |
| Identifier | |
| Occurrence constraint | optional |
| Category | General information |
| Definition | Relation to an ontology implementation (OI) of this OC |
| Domain | omv:OntologyConceptualisation |
| Range | omv:OntologyImplementation |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.2: Property: hasImplementation

| conceptualisationId | |
|----------------------------|--|
| Name | conceptualisationId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | The identifier of the metadata element |
| Domain | omv:OntologyConceptualisation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.3: Property: conceptualisationId

| conceptualisationName | |
|------------------------------|---|
| Name | conceptualisationName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | The name by which an OC is formally known |
| Domain | omv:OntologyConceptualisation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.4: Property: conceptualisationName

| conceptualisationAcronym | |
|---------------------------------|--|
| Name | conceptualisationAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | A short name by which the OC is formally known |
| Domain | omv:OntologyConceptualisation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.5: Property: conceptualisationAcronym

| conceptualisationDescription | |
|-------------------------------------|---------------------------------|
| Name | conceptualisationDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | Free text description of the OC |
| Domain | omv:OntologyConceptualisation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.6: Property: conceptualisationDescription

| conceptualisationDocumentation | |
|---------------------------------------|--------------------------------|
| Name | conceptualisationDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | URL for further documentation |
| Domain | omv:OntologyConceptualisation |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.2 |
| Comments | None |

Table 5.7: Property: conceptualisationDocumentation

| conceptualisationKeywords | |
|----------------------------------|------------------------------------|
| Name | conceptualisationKeywords |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | List of keywords related to the OC |
| Domain | omv:OntologyConceptualisation |
| Range | xsd:string |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | |

Table 5.8: Property: conceptualisationKeywords

| conceptualisationLicense | |
|---------------------------------|--|
| Name | conceptualisationLicense |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Availability information |
| Definition | Information about the underlying license model |
| Domain | omv:OntologyConceptualisation |
| Range | omv:LicenseModel |
| Cardinality | 0:1 |
| OMV version | 0.1 |
| Comments | Reference to a concrete LicenseModel. |

Table 5.9: Property: conceptualisationLicense

| conceptualisationDomain | |
|--------------------------------|---|
| Name | conceptualisationDomain |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Applicability information |
| Definition | Specifies the domain topic of an ontology |
| Domain | omv:OntologyConceptualisation |
| Range | omv:OntologyDomain |
| Cardinality | 0:n |
| OMV version | 0.8 |
| Comments | Typically, the subject can refer to established topic hierarchies such as the general purpose topic hierarchy DMOZ or the domain specific topic hierarchy ACM for the computer science domain |

Table 5.10: Property: conceptualisationDomain

| conceptualisationType | |
|-----------------------------------|--|
| Name Type Identifier | conceptualisationType ObjectProperty |
| Occurrence Constraint Category | required Applicability information |
| Definition | The nature of the content of the ontology |
| Domain Range Cardinality | omv:OntologyConceptualisation omv:OntologyType 1:1 |
| OMV version | 0.1 |
| Comments | See section 5.3 for details. |

Table 5.11: Property: conceptualisationType

| conceptualisationContributor | |
|-------------------------------------|---|
| Name Type Identifier | conceptualisationContributor ObjectProperty |
| Occurrence Constraint Category | optional Provenance information |
| Definition | Contributor to the OC |
| Domain Range Cardinality | omv:OntologyConceptualisation omv:Party 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.12: Property: conceptualisationContributor

| conceptualisationCreator | |
|-----------------------------------|---|
| Name Type Identifier | conceptualisationCreator ObjectProperty |
| Occurrence Constraint Category | required Provenance information |
| Definition | The main party responsible for the OC |
| Domain Range Cardinality | omv:OntologyConceptualisation omv:Party 1:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.13: Property: conceptualisationCreator

| conceptualExtension | |
|-----------------------------------|---|
| entity Name Type Identifier | conceptualExtension ObjectProperty |
| Occurrence Constraint Category | optional Relationship information |
| Definition | An OC which is a conceptual extension of another OC |
| Domain Range Cardinality | omv:OntologyConceptualisation omv:OntologyConceptualisation 0:1 |
| OMV version | 0.9.1 |
| Comments | Sub-property of conceptualRelation |

Table 5.14: Property: conceptualExtension

| conceptualIntegration | |
|-----------------------------------|---|
| entity Name Type Identifier | conceptualIntegration ObjectProperty |
| Occurrence Constraint Category | optional Relationship information |
| Definition | An OC which conceptually integrates other OCs |
| Domain Range Cardinality | omv:OntologyConceptualisation omv:OntologyConceptualisation 0:n |
| OMV version | 0.9.1 |
| Comments | Sub-property of conceptualRelation |

Table 5.15: Property: conceptualIntegration

| conceptualRelation | |
|-----------------------------------|---|
| entity Name Type Identifier | conceptualRelation ObjectProperty |
| Occurrence Constraint Category | optional Relationship Information |
| Definition | An OC which is conceptually similar to other OCs. |
| Domain Range Cardinality | omv:OntologyConceptualisation omv:OntologyConceptualisation 0:n |
| OMV version | 0.9.1 |
| Comments | Super-property of the remaining conceptual relationships |

Table 5.16: Property: conceptualRelation

| conceptualSpecialisation | |
|---------------------------------|---|
| entity Name | conceptualSpecialisation |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Relationship information |
| Definition | An OC which conceptually specialises another OC |
| Domain | omv:OntologyConceptualisation |
| Range | omv:OntologyConceptualisation |
| Cardinality | 0:1 |
| OMV version | 0.9.1 |
| Comments | Sub-property of conceptualRelation |

Table 5.17: Property: conceptualSpecialisation

| conceptualGeneralisation | |
|---------------------------------|---|
| entity Name | conceptualGeneralisation |
| Type | ObjectProperty |
| Identifier | Used Identifier for this entity. |
| Occurrence Constraint | optional |
| Category | Relationship information |
| Definition | An OC which conceptually generalises another OC |
| Domain | omv:OntologyConceptualisation |
| Range | omv:OntologyConceptualisation |
| Cardinality | 0:1 |
| OMV version | 0.9.1 |
| Comments | Sub-property of conceptualRelation |

Table 5.18: Property: conceptualGeneralisation

5.2 OntologyImplementation

Aspects of specific realizations are covered modular (and extendable) by the class `OntologyImplementation`.

| OntologyImplementation | |
|-------------------------------|---|
| Name | OntologyImplementation |
| Type | class |
| Identifier | OI |
| Definition | An implementation of a conceptual model |
| OMV version | 0.1 |
| Comments | None |

Table 5.19: Class: OntologyImplementation

| implements | |
|-----------------------|---|
| Name | implements |
| Type | ObjectProperty, inverseOf hasImplementation |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | Relation to the OC of which this OI is an implementation |
| Domain | omv:OntologyImplementation |
| Range | omv:OntologyConceptualisation |
| Cardinality | 0:1 |
| OMV version | 0.1 |
| Comments | Every OI should have one conceptualisation. Only for practical reasons we allow a cardinality of null |

Table 5.20: Property: implements

| implementationId | |
|-------------------------|--|
| Name | implementationId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | The identifier of the metadata element |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.21: Property: implementationId

| implementationName | |
|---------------------------|---|
| Name | implementationName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | The name by which an OI is formally known |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.22: Property: implementationName

| implementationAcronym | |
|------------------------------|---|
| Name | implementationAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | A short name by which an OI is formally known |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.23: Property: implementationAcronym

| implementationDescription | |
|----------------------------------|--------------------------------|
| Name | implementationDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | Free text description of an OI |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.24: Property: implementationDescription

| implementationDocumentation | |
|------------------------------------|-------------------------------|
| Name | implementationDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | URL for further documentation |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.2 |
| Comments | None |

Table 5.25: implementationDocumentation

| implementationKeywords | |
|-------------------------------|--|
| Name | implementationKeywords |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | List of keywords related to an OI |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | Typically this set includes names of the classes, properties, etc. |

Table 5.26: Property: implementationKeywords

| status | |
|-----------------------|--|
| Name | status |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | It specifies the tracking information for the contents of the ontology |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.1 |
| Comments | Pre-defined values to be specified |

Table 5.27: Property: implementationStatus

| creationDate | |
|-----------------------|--|
| Name | creationDate |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | Date when the OI was initially created |
| Domain | omv:OntologyImplementation |
| Range | xsd:date |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.28: Property: creationDate

| modificationDate | |
|-------------------------|--|
| Name | modifiedDate |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | Date of the last modification made to the OI |
| Domain | omv:OntologyImplementation |
| Range | xsd:date |
| Cardinality | 0:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.29: Property: modificationDate

| implementationContributor | |
|----------------------------------|----------------------------|
| Name | implementationContributor |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | Contributor to the OI |
| Domain | omv:OntologyImplementation |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.30: Property: implementationContributor

| implementationCreator | |
|------------------------------|-----------------------------|
| Name | implementationCreator |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Provenance information |
| Definition | Main responsible for the OI |
| Domain | omv:OntologyImplementation |
| Range | omv:Party |
| Cardinality | 1:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.31: Property: implementationCreator

| implementationDomain | |
|-----------------------------------|---|
| Name Type Identifier | implementationDomain ObjectProperty |
| Occurrence Constraint Category | optional Applicability information |
| Definition | Specifies the domain of an ontology |
| Domain Range Cardinality | omv:OntologyImplementation omv:OntologyDomain 0:n |
| OMV version | 0.8 |
| Comments | Usage of pre-defined topic hierarchies recommended |

Table 5.32: Property: implementationDomain

| implementationType | |
|-----------------------------------|---|
| Name Type Identifier | implementationType ObjectProperty |
| Occurrence Constraint Category | required Applicability information |
| Definition | The nature of the content of the ontology |
| Domain Range Cardinality | omv:OntologyImplementation omv:ontologyType 1:1 |
| OMV version | 0.1 |
| Comments | See section 5.3 for details |

Table 5.33: Property: implementationType

| naturalLanguage | |
|-----------------------------------|--|
| Name Type Identifier | language DatatypeProperty |
| Occurrence Constraint Category | optional Applicability information |
| Definition | The language of the content of the OI, i.e. English, German, etc. |
| Domain Range Cardinality | omv:OntologyImplementation xsd:string 0:n |
| OMV version | 0.1 |
| Comments | Pre-defined vocabulary to be specified |

Table 5.34: Property: naturalLanguage

| usedInApplicationSystem | |
|--------------------------------|--|
| Name | usedInApplicationSystem |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Applicability information |
| Definition | Declares where the ontology has been applied |
| Domain | omv:OntologyImplementation |
| Range | omv:OntologyApplicationSystem |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.35: Property: usedInApplicationSystem

| designedForOntologyTask | |
|--------------------------------|---|
| Name | designedForOntologyTask |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Applicability information |
| Definition | Declares for which purpose the ontology was originally designed |
| Domain | omv:OntologyImplementation |
| Range | omv:OntologyTask |
| Cardinality | 0:n |
| OMV version | 0.9 |
| Comments | See Section ?? |

Table 5.36: Property: designedForOntologyTask

| usedOntologyEngineeringTool | |
|------------------------------------|--|
| Name | usedTool |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Other |
| Definition | Information about tool used to create the ontology |
| Domain | omv:OntologyImplementation |
| Range | omv:OntologyEngineeringTool |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.37: Property: usedOntologyEngineeringTool

| usedOntologyEngineeringMethod | |
|--------------------------------------|--|
| Name | usedOntologyEngineeringMethod |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Other |
| Definition | Information about the method model used to create the ontology |
| Domain | omv:OntologyImplementation |
| Range | omv:OntologyEngineeringMethod |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.38: Property: usedOntologyEngineeringMethod

| ontologyLanguage | |
|-------------------------|------------------------------------|
| Name | ontologyLanguage |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | Specifies the ontology language |
| Domain | omv:OntologyImplementation |
| Range | omv:ontologyRepresentationLanguage |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.39: Property: ontologyLanguage

| ontologySyntax | |
|-----------------------|--|
| Name | ontologySyntax |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | It specifies the presentation syntax for the ontology language |
| Domain | omv:OntologyImplementation |
| Range | omv:OntologyRepresentationLanguageSyntax |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.40: Property: ontologySyntax

| formalityLevel | |
|-----------------------|------------------------------------|
| Name | formalityLevel |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Applicability information |
| Definition | Level of formality of the ontology |
| Domain | omv:OntologyImplementation |
| Range | omv:formalityLevel |
| Cardinality | 0:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.41: Property: formalityLevel

| ontologyURL | |
|-----------------------|--------------------------------|
| Name | ontologyURL |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Availability information |
| Definition | URL where the OI can be found. |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.42: Property: ontologyURL

| version | |
|-----------------------|---|
| Name | version |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Availability information |
| Definition | Specifies the version information of the OI |
| Domain | omv:OntologyImplementation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | Versioning could be useful for tracking, comparing and merging ontologies The number could be incremented by 1, or a smaller or larger value, depending on the personal preference of the author. Pre-defined value to be specified |

Table 5.43: Property: version

| implementationLicense | |
|------------------------------|--------------------------------------|
| Name | implementationLicense |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Availability information |
| Definition | Underlying license model |
| Domain | omv:OntologyImplementation |
| Range | omv:LicenseModel |
| Cardinality | 0:1 |
| OMV version | 0.1 |
| Comments | Reference to a concrete LicenseModel |

Table 5.44: Property: implementationLicense

| imports | |
|-----------------------------------|---|
| Name Type Identifier | imports ObjectProperty |
| Occurrence Constraint Category | optional Relationship information |
| Definition | References another OI containing definitions, whose meaning is considered to be part of the meaning of the importing ontology |
| Domain Range Cardinality | omv:OntologyImplementation omv:OntologyImplementation 0:n |
| OMV version | 0.1 |
| Comments | Each reference consists of a URI specifying from where the OI is to be imported |

Table 5.45: Property: imports

| priorVersion | |
|-----------------------------------|---|
| Name Type Identifier | priorVersion ObjectProperty |
| Occurrence Constraint Category | optional Relationship information |
| Definition | Contains a reference to another OI |
| Domain Range Cardinality | omv:OntologyImplementation omv:OntologyImplementation 0:1 |
| OMV version | 0.1 |
| Comments | Identifies the specified OI as a prior version of the containing OI. It may be used to organize ontology implementations by versions and is NULL for initial ontology |

Table 5.46: Property: priorVersion

| backwardCompatibleWith | |
|-----------------------------------|--|
| Name Type Identifier | backwardCompatibleWith ObjectProperty |
| Occurrence Constraint Category | optional Relationship information |
| Definition | This property identifies the specified OI as a compatible prior version of the containing OI |
| Domain Range Cardinality | omv:OntologyImplementation omv:OntologyImplementation 0:n |
| OMV version | 0.1 |
| Comments | This also indicates that all identifiers from the previous version have the same intended interpretations in the new version |

Table 5.47: Property: backwardCompatibleWith

| incompatibleWith | |
|-----------------------------------|---|
| Name Type Identifier | incompatibleWith ObjectProperty |
| Occurrence Constraint Category | optional Relationship information |
| Definition | This property indicates that the containing ontology is a later version of the referenced ontology, but is not backward compatible with it It can be used to explicitly state that OI s cannot upgrade to use the new version without checking whether changes are required. |
| Domain Range Cardinality | omv:OntologyImplementation omv:OntologyImplementation 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.48: Property: incompatibleWith

| numClasses | |
|-----------------------|-----------------------------|
| Name | numClasses |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Statistic information |
| Definition | Number of classes in the OI |
| Domain | omv:OntologyImplementation |
| Range | xsd:unsignedLong |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | Language specific value |

Table 5.49: Property: numClasses

| numProperties | |
|-----------------------|----------------------------------|
| Name | numProperties |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Statistic information |
| Definition | Number of properties in the OI . |
| Domain | omv:OntologyImplementation |
| Range | xsd:unsignedLong |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | Language specific value |

Table 5.50: Property: numProperties

| numIndividuals | |
|-----------------------|-----------------------------------|
| Name | numIndividuals |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Statistic information |
| Definition | Number of individuals in the OI . |
| Domain | omv:OntologyImplementation |
| Range | xsd:unsignedLong |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | Language specific value |

Table 5.51: Property: numIndividuals

| numAxioms | |
|-----------------------|----------------------------|
| Name | numAxioms |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Statistic information |
| Definition | Number of axioms in the OI |
| Domain | omv:OntologyImplementation |
| Range | xsd:unsignedLong |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | Language specific value |

Table 5.52: Property: numAxioms

5.3 OntologyType

This class subsumes types of ontologies according to well-known classifications in the Ontology Engineering literature [2].

| OntologyType | |
|---------------------|------------------------|
| Name | OntologyType |
| Type | class |
| Identifier | |
| Definition | Categorizes ontologies |
| OMV version | 0.3 |
| Comments | None |

Table 5.53: Class: OntologyType

| typeId | |
|-----------------------|----------------------------|
| Name | typeId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | An identifier for the type |
| Domain | omv:OntologyType |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.54: Property: typeId

| typeName | |
|-----------------------|--|
| Name | typeName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | The name by which an ontology type is formally known |
| Domain | omv:OntologyType |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.55: Property: typeName

| typeAcronym | |
|-----------------------|--|
| Name | typeAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | A short name by which an ontology type is formally known |
| Domain | omv:OntologyType |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.56: Property: typeAcronym

| typeDescription | |
|------------------------|---|
| Name | typeDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | General information |
| Definition | Free text description of an ontology type |
| Domain | omv:OntologyType |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.57: Property: typeDescription

| typeDocumentation | |
|--------------------------|-------------------------------|
| Name | typeDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | URL for further documentation |
| Domain | omv:OntologyType |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.2 |
| Comments | None |

Table 5.58: typeDocumentation

| definedBy | |
|-----------------------|--|
| Name | definedBy |
| Type | ObjectProperty, inverseOf(definesOntologyType) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | General information |
| Definition | Reference a party that defined the ontology type |
| Domain | omv:OntologyType |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.59: typeDefinedBy

5.3.1 Pre-defined ontology types

Individuals of the class `OntologyType` refer to well-known classifications for ontologies in the literature. Currently the OMV model resorts to a classification on the generality levels of the ontology conceptualisation [3, 13]:

- upper level ontologies describing general, domain-independent concepts e.g. space, time.
- core ontologies describing the most important concepts in a specific domain
- domain ontology describing some domain of the world
- task ontology describing generic types of tasks or activities e.g. selling, selecting.
- application ontology describing some domain in an application-dependent manner

The class can be extended to support additional classifications (e.g. the one in [5]).

5.4 LicenseModel

| LicenseModel | |
|---------------------|---|
| Name | LicenseModel |
| Type | class |
| Identifier | LM |
| Definition | A license model describing the usage conditions for an ontology |
| OMV version | 0.3 |
| Comments | None |

Table 5.60: Class: LicenseModel

| licenseId | |
|-----------------------|----------------------------|
| Name | licenseId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Availability information |
| Definition | Identifier of the resource |
| Domain | omv:OntologyLicense |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.61: Property: licenseId

| licenseName | |
|-----------------------|---|
| Name | licenseName |
| Type | DatatypeProperty |
| Identifier | Used Identifier for this entity. |
| Occurrence Constraint | required |
| Category | Availability information |
| Definition | The name by which a license model is formally known |
| Domain | omv:OntologyLicense |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.62: Property: licenseName

| licenseAcronym | |
|-----------------------------------|---|
| Name Type Identifier | licenseAcronym DatatypeProperty |
| Occurrence Constraint Category | required Availability information |
| Definition | A short name by which a license model is formally known |
| Domain Range Cardinality | omv:OntologyLicense xsd:string 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.63: Property: licenseAcronym

| licenseDescription | |
|-----------------------------------|---|
| Name Type Identifier | licenseDescription DatatypeProperty |
| Occurrence Constraint Category | required Availability information |
| Definition | Descriptive free text about a license model |
| Domain Range Cardinality | omv:OntologyLicense xsd:string 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.64: Property: licenseDescription

| licenseDocumentation | |
|-----------------------------------|--|
| Name Type Identifier | licenseDocumentation DatatypeProperty |
| Occurrence Constraint Category | optional Availability information |
| Definition | URL for further documentation |
| Domain Range Cardinality | omv:OntologyLicense xsd:string 0:1 |
| OMV version | 0.2 |
| Comments | None |

Table 5.65: licenseDocumentation

| licenseSpecifiedBy | |
|---------------------------|--|
| Name | licenseSpecifiedBy |
| Type | ObjectProperty, inverseOf(specifiesLicense) |
| Identifier | Used Identifier for this element. |
| Occurrence Constraint | optional |
| Category | Availability information |
| Definition | Reference a party that specified the license model |
| Domain | omv:LicenseModel |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.66: licenseSpecifiedBy

5.5 OntologyEngineeringMethod

| OntologyEngineeringMethod | |
|----------------------------------|--|
| Name | OntologyEngineeringMethod |
| Type | class |
| Identifier | |
| Definition | Information about the ontology engineering methodology |
| OMV version | 0.3 |
| Comments | None |

Table 5.67: Class: OntologyEngineeringMethod

| methodId | |
|-----------------------|------------------------------------|
| Name | methodId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | Identifier of the metadata element |
| Domain | omv:OntologyEngineeringMethod |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.68: Property: methodId

| methodName | |
|-----------------------|---|
| Name | methodName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | The name by which a ontology engineering method is formally known |
| Domain | omv:OntologyEngineeringMethod |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.69: Property: methodName

| methodAcronym | |
|-----------------------|--|
| Name | methodAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | A short name by which a ontology engineering method is known |
| Domain | omv:OntologyEngineeringMethod |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.70: Property: methodAcronym

| methodDescription | |
|--------------------------|---|
| Name | methodDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | Free text description of an ontology engineering method |
| Domain | omv:OntologyEngineeringMethod |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.6 |
| Comments | None |

Table 5.71: Property: methodDescription

| methodDocumentation | |
|----------------------------|-------------------------------|
| Name | methodDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Other |
| Definition | URL for further documentation |
| Domain | omv:OntologyEngineeringMethod |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.6 |
| Comments | None |

Table 5.72: methodDocumentation

| methodDeveloper | |
|------------------------|---|
| Name | methodDeveloper |
| Type | ObjectProperty |
| Identifier | inverseOf(developesOntologyEngineeringMethod) |
| Occurrence Constraint | optional |
| Category | Other |
| Definition | A party that developed the ontology engineering methodology |
| Domain | omv:OntologyEngineeringMethod |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.73: methodDeveloper

5.6 OntologyEngineeringTool

| OntologyEngineeringTool | |
|--------------------------------|------------------------------------|
| Name | OntologyEngineeringTool |
| Type | class |
| Identifier | |
| Definition | A tool used to create the ontology |
| OMV version | 0.3 |
| Comments | None |

Table 5.74: Class: OntologyEngineeringTool

| toolId | |
|-----------------------|--|
| Name | toolId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | The identifier of the metadata element |
| Domain | omv:OntologyEngineeringTool |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.75: Property: toolId

| toolName | |
|-----------------------|--|
| Name | toolName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | The name by which a tool is formally known |
| Domain | omv:OntologyEngineeringTool |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.76: Property: toolName

| toolAcronym | |
|-----------------------|---------------------------------------|
| Name | toolAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | A short name by which a tool is known |
| Domain | omv:OntologyEngineeringTool |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.77: Property: toolAcronym

| toolDescription | |
|------------------------|-----------------------------------|
| Name | toolDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Other |
| Definition | Free text description of the tool |
| Domain | omv:OntologyEngineeringTool |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.78: Property: toolDescription

| toolDocumentation | |
|--------------------------|--------------------------------|
| Name | toolDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Other |
| Definition | URL for further documentation. |
| Domain | omv:OntologyEngineeringTool |
| Range | xsd:string |
| Cardinality | 0:n |
| OMV version | 0.2 |
| Comments | None |

Table 5.79: toolDocumentation

| toolDeveloper | |
|-----------------------|---|
| Name | toolDeveloper |
| Type | ObjectProperty |
| Identifier | inverseOf(developesOntologyEngineeringTool) |
| Occurrence Constraint | optional |
| Category | Other |
| Definition | References the tool developer party |
| Domain | omv:OntologyEngineeringTool |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.4 |
| Comments | None |

Table 5.80: toolDeveloper

5.7 OntologySyntax

| OntologySyntax | |
|-----------------------|--|
| Name | OntologySyntax |
| Type | class |
| Identifier | |
| Definition | Information about the syntax used in an OI |
| OMV version | 0.3 |
| Comments | None |

Table 5.81: Class: OntologySyntax

| syntaxId | |
|-----------------------|--|
| Name | syntaxId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | The identifier of the metadata element |
| Domain | omv:OntologySyntax |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.82: Property: syntaxId

| syntaxName | |
|-----------------------|--|
| Name | syntaxName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | The name by which an ontology syntax is formally known |
| Domain | omv:OntologySyntax |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.83: Property: syntaxName

| syntaxAcronym | |
|-----------------------|---|
| Name | syntaxAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | A short name by which an ontology syntax is known |
| Domain | omv:OntologySyntax |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.84: Property: syntaxAcronym

| syntaxDescription | |
|--------------------------|--|
| Name | syntaxDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | Free text description of the used syntax |
| Domain | omv:OntologySyntax |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.6 |
| Comments | None |

Table 5.85: Property: syntaxDescription

| syntaxDocumentation | |
|----------------------------|--------------------------------|
| Name | syntaxDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | URL for further documentation. |
| Domain | omv:OntologySyntax |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.6 |
| Comments | None |

Table 5.86: syntaxDocumentation

| syntaxDeveloper | |
|------------------------|---|
| Name | syntaxDeveloper |
| Type | ObjectProperty |
| Identifier | inverseOf(developedOntologySyntax) |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | The party who developed the used syntax |
| Domain | omv:OntologySyntax |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.87: syntaxDeveloper

5.8 OntologyRepresentationLanguage

| OntologyRepresentationLanguage | |
|---------------------------------------|---|
| Name | OntologyRepresentationLanguage |
| Type | class |
| Identifier | |
| Definition | Information about the language in which the ontology is implemented |
| OMV version | 0.3 |
| Comments | None |

Table 5.88: Class: OntologyRepresentationLanguage

| languageId | |
|-----------------------|--|
| Name | languageId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | The identifier of the metadata element |
| Domain | omv:OntologyRepresentationLanguage |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.89: Property: languageId

| languageName | |
|-----------------------|--|
| Name | languageName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | The name by which an ontology language is formally known |
| Domain | omv:OntologyRepresentationLanguage |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.90: Property: languageName

| languageAcronym | |
|------------------------|---|
| Name | languageAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | A short name by which an ontology language is known |
| Domain | omv:OntologyRepresentationLanguage |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.91: Property: languageAcronym

| languageDescription | |
|----------------------------|--|
| Name | languageDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | Free text description of an ontology language. |
| Domain | omv:OntologyRepresentationLanguage |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.6 |
| Comments | None |

Table 5.92: Property: languageDescription

| languageDocumentation | |
|------------------------------|------------------------------------|
| Name | languageDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | URL for further documentation |
| Domain | omv:OntologyRepresentationLanguage |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.6 |
| Comments | None |

Table 5.93: Property: languageDocumentation

| languageDeveloper | |
|--------------------------|--|
| Name | languageDeveloper |
| Type | ObjectProperty, inverseOf(developesOntologyLanguage) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | References the party who developed the language |
| Domain | omv:OntologyRepresentationLanguage |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.94: Property: languageDeveloper

| supportsRepresentationParadigm | |
|---------------------------------------|---|
| Name | supportsRepresentationParadigm |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | References the party who developed the language |
| Domain | omv:OntologyRepresentationLanguage |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.95: Property: supportsRepresentationParadigm

| hasSyntax | |
|-----------------------|---|
| Name | hasSyntax |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | References the syntactical alternatives of the language |
| Domain | omv:OntologyRepresentationLanguage |
| Range | omv:OntologyRepresentationLanguageSyntax |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.96: Property: hasSyntax

5.9 KnowledgeRepresentationParadigm

| KnowledgeRepresentationParadigm | |
|--|--|
| Name | KnowledgeRepresentationParadigm |
| Type | class |
| Identifier | |
| Definition | Information about a knowledge representation paradigm a particular language adheres to |
| OMV version | 0.9.1 |
| Comments | E. g. Description Logics, Frames |

Table 5.97: Class: KnowledgeRepresentationParadigm

| knowledgeRepresentationParadigmId | |
|--|---|
| Name | knowledgeRepresentationParadigmId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | The identifier of this metadata element |
| Domain | omv:KnowledgeRepresentationParadigm |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.98: Property: knowledgeRepresentationParadigmId

| knowledgeRepresentationParadigmName | |
|--|---|
| Name | knowledgeRepresentationParadigmName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | The name by which a KR paradigm is formally known |
| Domain | omv:KnowledgeRepresentationParadigm |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.99: Property: knowledgeRepresentationParadigmName

| knowledgeRepresentationParadigmAcronym | |
|---|--|
| Name Type Identifier | knowledgeRepresentationParadigmAcronym DatatypeProperty |
| Occurrence Constraint Category | required Format information |
| Definition | A short name by which a kR paradigm is known |
| Domain Range Cardinality | omv:KnowledgeRepresentationParadigm xsd:string 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.100: Property: knowledgeRepresentationParadigmAcronym

| knowledgeRepresentationParadigmDescription | |
|---|--|
| Name Type Identifier | knowledgeRepresentationParadigmDescription DatatypeProperty |
| Occurrence Constraint Category | required Format information |
| Definition | Free text description of the knowledge representation paradigm |
| Domain Range Cardinality | omv:KnowledgeRepresentationParadigm xsd:string 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.101: Property: knowledgeRepresentationParadigmDescription

| knowledgeRepresentationParadigmDocumentation | |
|---|--|
| Name Type Identifier | knowledgeRepresentationParadigmDocumentation DatatypeProperty |
| Occurrence Constraint Category | optional Format information |
| Definition | URL for further documentation |
| Domain Range Cardinality | omv:KnowledgeRepresentationParadigm xsd:string 0:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.102: knowledgeRepresentationParadigmDocumentation

| knowledgeRepresentationParadigmSpecifiedBy | |
|---|--|
| Name | knowledgeRepresentationParadigmSpecifiedBy |
| Type Identifier | inverseOf specifiesKnowledgeRepresentationLanguageParadigm ObjectProperty |
| Occurrence Constraint | optional |
| Category | Format information |
| Definition | Author of the KR paradigm |
| Domain | omv:KnowledgeRepresentationParadigm |
| Range | omv:Party |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.103: Property: implementationContributor

5.9.1 Pre-defined knowledge representation paradigms

In this version we foresee two main classes of KnowledgeRepresentationParadigms:

- DescriptionLogics
- Frames

5.10 FormalityLevel

| FormalityLevel | |
|-----------------------|---|
| Name | FormalityLevel |
| Type | class |
| Identifier | |
| Definition | The level of formality of an ontology |
| OMV version | 0.9.1 |
| Comments | According to classifications in the OE literature |

Table 5.104: Class: FormalityLevel

| formalityLevelId | |
|-------------------------|---|
| Name | formalityLevelId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | The identifier of this metadata element |
| Domain | omv:FormalityLevel |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.105: Property: formalityLevelId

| formalityLevelName | |
|---------------------------|--|
| Name | formalityLevelName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | The name by which this element is formally known |
| Domain | omv:FormalityLevel |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.106: Property: formalityLevelName

| formalityLevelDescription | |
|----------------------------------|--|
| Name | formalityLevelDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Format information |
| Definition | Free text description of the formality level |
| Domain | omv:FormalityLevel |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.107: Property: formalityLevelDescription

5.10.1 Pre-defined formality levels

We use the classification in [11], which differentiates among four formality levels: highly informal, semi-informal, semi-formal, rigorously formal. The four are modelled as individuals of the class `FormalityLevel`.

5.11 OntologyApplicationSystem

| OntologyApplicationSystem | |
|----------------------------------|---|
| Name | OntologyApplicationSystem |
| Type | class |
| Identifier | |
| Definition | Information about the task the ontology was intended to be used for |
| OMV version | 0.9.1 |
| Comments | Super-class of classes modelling typical ontology-related tasks |

Table 5.108: Class: OntologyApplicationSystem

| applicationSystemId | |
|----------------------------|---|
| Name | applicationSystemId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | The identifier of this metadata element |
| Domain | omv:OntologyApplicationSystem |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.109: Property: applicationSystemId

| taskName | |
|-----------------------|--|
| Name | taskName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | The name by which an ontology task is formally known |
| Domain | omv:OntologyApplicationSystem |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.110: Property: taskName

| taskAcronym | |
|-----------------------|---|
| Name | taskAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | A short name by which an ontology task is known |
| Domain | omv:OntologyApplicationSystem |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.111: Property: taskAcronym

| taskDescription | |
|------------------------|--|
| Name | taskDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | Free text description of the ontology task |
| Domain | omv:OntologyApplicationSystem |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.112: Property: taskDescription

| taskDocumentation | |
|--------------------------|-------------------------------|
| Name | taskDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Applicability information |
| Definition | URL for further documentation |
| Domain | omv:OntologyApplicationSystem |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.113: taskDocumentation

5.11.1 Pre-defined ontology types

Individuals of the class `OntologyApplicationSystem` refer to particular application scenarios for ontologies, in which the benefits of using ontologies are widely acknowledged. We differentiate among the following tasks:

AnnotationTask : the ontology is used as a controlled vocabulary to annotate Semantic Web resources. This task includes the usage of a semantically rich ontology for representing arbitrarily complex annotation statements on these resources. The task can be performed manually or (semi-)automatically.

ConfigurationTask : the ontology is designed to provide a controlled and unambiguous means to represent valid configuration profiles in application systems. As the aim of the ontology is to support the operationalization of particular system-related processes; this task is performed automatically in that the ontology is processed in an automatic manner by means of reasoners or APIs.

FilteringTask : the task describes at a very general level how ontologies are applied to refine the solution space of a certain problem, such as information retrieval or personalization. The task is targeted at being performed semi-automatically or automatically.

IndexingTask : in this scenario, the goal of the ontology is to provide a clearly defined classification and browsing structure for the information items in a repository. Again, the task can be performed manually by domain experts or as part of an application in an automatic or semi-automatic way.

IntegrationTask : the task characterizes how ontologies provide an integrating environment, an inter-lingua, for information repositories or software tools. In this scenario the ontology is applied (semi-)automatically to merge between heterogeneous data pools in the same or in adjacent domains.

MatchingTask : the goal of matching is to establish links between semantically similar data items in information repositories. In contrast to the previous task, matching does not include the production of a shared final schema/ontology as a result of aggregating the matched source elements to common elements. W.r.t. the automation level the range varies from manual to fully-automatical execution.

MediationTask : the ontology is built to reduce the ambiguities between communicating human or machine agents. It can act as a normative model which formally and clearly defines the meaning of the terms employed in agent interactions. In the context of programmed agents, the task is envisioned to be performed automatically.

QueryFormulationTask : the ontology is used in information retrieval settings as a controlled vocabulary for representing user queries. Usually the task is performed au-

tomatically in that the concepts of the ontology is are listed in a query formulation front-end in order to allow users to specifies their queries.

QueryRewritingTask : complementary to the query formulation dimension, this task applies ontologies to semantically optimize query expressions by means of the domain knowledge (constraints, subsumption relations etc.) The task can be interpreted as a particular art of filtering information. The task is performed automatically; however, it assumes the availability of patterns describing the transformations at query level.

PersonalizationTask : the ontology is used mainly for providing personalized access to information resources. Individual user preferences w.r.t. particular application settings are formally specified by means of an ontology, which, in conjunction with appropriate reasoning services, can be directly integrated to a personalization component for filtering purposes. The usage of ontologies in personalization tasks might be carried out in various forms, from a direct involvement of the user who manually specifies ontological concepts which optimally describe his preferences, to the ontological modelling of user profiles.

SearchTask : the task characterizes how ontologies are used to refine common keyword-based search algorithms using domain knowledge in form of subsumption relations. Ontology-driven search is usually performed automatically by means of reasoning services handling particular aspects of an ontology representation language.

5.12 OntologyTask

| OntologyTask | |
|---------------------|---|
| Name | OntologyTask |
| Type | class |
| Identifier | |
| Definition | Information about the task the ontology was intended to be used for |
| OMV version | 0.9.1 |
| Comments | Super-class of classes modelling typical ontology-related tasks |

Table 5.114: Class: OntologyTask

| taskId | |
|-----------------------|---|
| Name | taskId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | The identifier of this metadata element |
| Domain | omv:OntologyTask |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.115: Property: taskId

| taskName | |
|-----------------------|--|
| Name | taskName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | The name by which an ontology task is formally known |
| Domain | omv:OntologyTask |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.116: Property: taskName

| taskAcronym | |
|-----------------------|---|
| Name | taskAcronym |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | A short name by which an ontology task is known |
| Domain | omv:OntologyTask |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.117: Property: taskAcronym

| taskDescription | |
|------------------------|--|
| Name | taskDescription |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Applicability information |
| Definition | Free text description of the ontology task |
| Domain | omv:OntologyTask |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.118: Property: taskDescription

| taskDocumentation | |
|--------------------------|-------------------------------|
| Name | taskDocumentation |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Applicability information |
| Definition | URL for further documentation |
| Domain | omv:OntologyTask |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9.1 |
| Comments | None |

Table 5.119: taskDocumentation

5.12.1 Pre-defined ontology types

Individuals of the class `OntologyTask` refer to particular application scenarios for ontologies, in which the benefits of using ontologies are widely acknowledged. We differentiate among the following tasks:

AnnotationTask : the ontology is used as a controlled vocabulary to annotate Semantic Web resources. This task includes the usage of a semantically rich ontology for representing arbitrarily complex annotation statements on these resources. The task can be performed manually or (semi-)automatically.

ConfigurationTask : the ontology is designed to provide a controlled and unambiguous means to represent valid configuration profiles in application systems. As the aim of the ontology is to support the operationalization of particular system-related processes; this task is performed automatically in that the ontology is processed in an automatic manner by means of reasoners or APIs.

FilteringTask : the task describes at a very general level how ontologies are applied to refine the solution space of a certain problem, such as information retrieval or personalization. The task is targeted at being performed semi-automatically or automatically.

IndexingTask : in this scenario, the goal of the ontology is to provide a clearly defined classification and browsing structure for the information items in a repository. Again, the task can be performed manually by domain experts or as part of an application in an automatic or semi-automatic way.

IntegrationTask : the task characterizes how ontologies provide an integrating environment, an inter-lingua, for information repositories or software tools. In this scenario the ontology is applied (semi-)automatically to merge between heterogeneous data pools in the same or in adjacent domains.

MatchingTask : the goal of matching is to establish links between semantically similar data items in information repositories. In contrast to the previous task, matching does not include the production of a shared final schema/ontology as a result of aggregating the matched source elements to common elements. W.r.t. the automation level the range varies from manual to fully-automatical execution.

MediationTask : the ontology is built to reduce the ambiguities between communicating human or machine agents. It can act as a normative model which formally and clearly defines the meaning of the terms employed in agent interactions. In the context of programmed agents, the task is envisioned to be performed automatically.

QueryFormulationTask : the ontology is used in information retrieval settings as a controlled vocabulary for representing user queries. Usually the task is performed au-

tomatically in that the concepts of the ontology is are listed in a query formulation front-end in order to allow users to specifies their queries.

QueryRewritingTask : complementary to the query formulation dimension, this task applies ontologies to semantically optimize query expressions by means of the domain knowledge (constraints, subsumption relations etc.) The task can be interpreted as a particular art of filtering information. The task is performed automatically; however, it assumes the availability of patterns describing the transformations at query level.

PersonalizationTask : the ontology is used mainly for providing personalized access to information resources. Individual user preferences w.r.t. particular application settings are formally specified by means of an ontology, which, in conjunction with appropriate reasoning services, can be directly integrated to a personalization component for filtering purposes. The usage of ontologies in personalization tasks might be carried out in various forms, from a direct involvement of the user who manually specifies ontological concepts which optimally describe his preferences, to the ontological modelling of user profiles.

SearchTask : the task characterizes how ontologies are used to refine common keyword-based search algorithms using domain knowledge in form of subsumption relations. Ontology-driven search is usually performed automatically by means of reasoning services handling particular aspects of an ontology representation language.

5.13 Party

| Party | |
|--------------|--|
| Name | Party |
| Type | class |
| Identifier | |
| Definition | A party is a person or an organisation |
| OMV version | 0.4 |
| Comments | None |

Table 5.120: Class: Party

| partyId | |
|-----------------------|--|
| Name | partyId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | The identifier for this metadata element |
| Domain | omv:Location |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9 |
| Comments | Inherited by Person and Organisation |

Table 5.121: Property: partyId

| locatedAt | |
|-----------------------|--------------------------------------|
| Name | locatedAt |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Availability Information |
| Definition | The geographical location of a party |
| Domain | omv:Party |
| Range | omv:Location |
| Cardinality | 0:n |
| OMV version | 0.9 |
| Comments | None |

Table 5.122: Property: locatedAt

| developesOntologyEngineeringTool | |
|---|--|
| Name | developesOntologyEngineeringTool |
| Type | ObjectProperty, inverseOf(toolDeveloper) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References a tool developed by a party |
| Domain | omv:Party |
| Range | omv:OntologyEngineeringTool |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.123: Property: developesOntologyEngineeringTool

| developesOntologyLanguage | |
|----------------------------------|--|
| Name | developesOntologyLanguage |
| Type | ObjectProperty, inverseOf(languageDeveloper) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References an ontology language developed by a party |
| Domain | omv:Party |
| Range | omv:OntologyLanguage |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.124: Property: developeaOntologyLanguage

| developesOntologyRepresentationLanguageSyntax | |
|--|--|
| Name | developesOntologyRepresentationLanguageSyntax |
| Type | ObjectProperty, inverseOf(syntaxDeveloper) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References an ontology syntax developed by a party |
| Domain | omv:Party |
| Range | omv:OntologySyntax |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.125: Property: developesOntologyRepresentationLanguageSyntax

| specifiesKnowledgeRepresentationLanguageParadigm | |
|---|---|
| Name | specifiesKnowledgeRepresentationLanguageParadigm |
| Type | ObjectProperty |
| Identifier | inverseOf(knowledgeRepresentationParadigmSpecifiedBy) |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References an KR paradigm specified or defined by a party |
| Domain | omv:Party |
| Range | omv:KnowledgeRepresentationParadigm |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.126: Property: specifiesKnowledgeRepresentationLanguageParadigm

| definesOntologyType | |
|----------------------------|---|
| Name | definesOntologyType |
| Type | ObjectProperty, inverseOf(definedBy) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | Reference an ontology type defined by a party |
| Domain | omv:Party |
| Range | omv:OntologyType |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.127: Property: definesOntologyType

| developesOntologyEngineeringMethod | |
|---|---|
| Name | developesOntologyEngineeringMethod |
| Type | ObjectProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References a ontology engineering method developed by a party |
| Domain | omv:Party |
| Range | omv:OntologyEngineeringMethod |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.128: Property: developesOntologyEngineeringMethod

| specifiesLicense | |
|-----------------------------------|---|
| Name Type Identifier | specifiesLicense ObjectProperty, inverseOf(licenseSpecifiedBy) |
| Occurrence Constraint Category | optional Provenance information |
| Definition | References a license specified by a party |
| Domain Range Cardinality | omv:Party omv:LicenseModel 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.129: Property: specifiesLicense

| affiliatedParty | |
|-----------------------------------|---|
| Name Type Identifier | affiliatedParty ObjectProperty |
| Occurrence Constraint Category | optional Provenance information |
| Definition | References another party that is affiliated with the instance |
| Domain Range Cardinality | omv:Party omv:Party 0:n |
| OMV version | 0.2 |
| Comments | None |

Table 5.130: Property: affiliatedParty

| createsConceptualisation | |
|-----------------------------------|---|
| Name Type Identifier | createsConceptualisation ObjectProperty, inverseOf(conceptualisationCreator) |
| Occurrence Constraint Category | optional Provenance information |
| Definition | References an ontology conceptualisation created by a party |
| Domain Range Cardinality | omv:Party omv:Conceptualisation 0:n |
| OMV version | 0.7 |
| Comments | None |

Table 5.131: Property: createsConceptualisation

| contributesToConceptualisation | |
|---------------------------------------|---|
| Name | contributesToConceptualisation |
| Type | ObjectProperty |
| Identifier | inverseOf(conceptualisationContributor) |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References an ontology conceptualisation to which a party made contributions to |
| Domain | omv:Party |
| Range | omv:Conceptualisation |
| Cardinality | 0:n |
| OMV version | 0.7 |
| Comments | None |

Table 5.132: Property: contributesToConceptualisation

| createsImplementation | |
|------------------------------|--|
| Name | createsImplementation |
| Type | ObjectProperty, inverseOf(implementationCreator) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References an ontology implementation created by a party |
| Domain | omv:Party |
| Range | omv:OntologyImplementation |
| Cardinality | 0:n |
| OMV version | 0.7 |
| Comments | None |

Table 5.133: Property: createsImplementation

| contributesToImplementation | |
|------------------------------------|---|
| Name | contributesToImplementation |
| Type | ObjectProperty |
| Identifier | inverseOf(implementationContributor) |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | References an ontology implementation a party made contributions to |
| Domain | omv:Party |
| Range | omv:OntologyImplementation |
| Cardinality | 0:n |
| OMV version | 0.7 |
| Comments | None |

Table 5.134: Property: contributesToImplementation

5.14 Person

| Person | |
|---------------|---|
| Name | Person |
| Type | class |
| Identifier | |
| Definition | A named individual |
| OMV version | 0.1 |
| Comments | Represents an individual responsible for the creation, or contribution to an ontology |

Table 5.135: Class: Person

| lastName | |
|-----------------------|-------------------------|
| Name | lastName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Provenance information |
| Definition | The surname of a person |
| Domain | omv:Person |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.2 |
| Comments | None |

Table 5.136: lastName

| firstName | |
|-----------------------|----------------------------|
| Name | firstName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Provenance information |
| Definition | The first name of a person |
| Domain | omv:Person |
| Range | xsd:string |
| Cardinality | 1:n |
| OMV version | 0.2 |
| Comments | None |

Table 5.137: firstname

| eMail | |
|-----------------------|-------------------------------|
| Name | email |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Provenance information |
| Definition | The email address of a person |
| Domain | omv:Person |
| Range | xsd:string |
| Cardinality | 1:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.138: eMail

| phoneNumber | |
|-----------------------|------------------------------|
| Name | phoneNumber |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | The phone number of a person |
| Domain | omv:Person |
| Range | xsd:string |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.139: Property: phoneNumber

| faxNumber | |
|-----------------------|----------------------------|
| Name | faxNumber |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | The fax number of a person |
| Domain | omv:Person |
| Range | xsd:string |
| Cardinality | 0:n |
| OMV version | 0.1 |
| Comments | None |

Table 5.140: faxNumber

| isContactPerson | |
|------------------------|---|
| Name | isContactPerson |
| Type | ObjectProperty, inverseOf(contactPerson) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | Instance is contact person of an organisation |
| Domain | omv:Person |
| Range | omv:Organisation |
| Cardinality | 0:n |
| OMV version | 0.7 |
| Comments | None |

Table 5.141: isContactPerson

5.15 Organisation

| Organisation | |
|---------------------|--|
| Name | Organisation |
| Type | class, subclassOf(Party) |
| Identifier | |
| Definition | An organisation of some kind |
| OMV version | 0.6 |
| Comments | Represents social institutions such as universities, companies, societies etc. |

Table 5.142: Class: Organisation

| organisationName | |
|-------------------------|---|
| Name | organisationName |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | required |
| Category | Provenance information |
| Definition | The name by which an organisation is formally known |
| Domain | omv:Organisation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.143: Property: organisationName

| organisationAcronym | |
|----------------------------|--|
| Name | organisationAcronym |
| Type | DatatypeProperty |
| Identifier | Used Identifier for this entity. |
| Occurrence Constraint | required |
| Category | Provenance information |
| Definition | A short name by which an organisation is known |
| Domain | omv:Organisation |
| Range | xsd:string |
| Cardinality | 1:1 |
| OMV version | 0.1 |
| Comments | None |

Table 5.144: Property: organisationAcronym

| contactPerson | |
|-----------------------|--|
| Name | contactPerson |
| Type | ObjectProperty, inverseOf(isContactPerson) |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | A contact person in the organisation |
| Domain | omv:Organisation |
| Range | omv:Person |
| Cardinality | 0:n |
| OMV version | 0.6 |
| Comments | None |

Table 5.145: contactPerson

5.16 Location

| Location | |
|-----------------|---|
| Name | Location |
| Type | class |
| Identifier | |
| Definition | A location. |
| OMV version | 0.9 |
| Comments | The geographical location of a party. To keep things simple we use only DatatypeProperties instead of introducing classes for country, street, etc. |

Table 5.146: Class: Location

| locationId | |
|-----------------------|--|
| Name | locationId |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | The identifier for this metadata element |
| Domain | omv:Location |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9 |
| Comments | None |

Table 5.147: Property: locationId

| state | |
|-----------------------|-------------------------|
| Name | state |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | The state of a country. |
| Domain | omv:Location |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9 |
| Comments | None |

Table 5.148: Property: state

| country | |
|-----------------------|---------------------------------------|
| Name | country |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | The name of the country |
| Domain | omv:Location |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9 |
| Comments | Changed the name from land to country |

Table 5.149: Property: country

| city | |
|-----------------------|----------------------------------|
| Name | city |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | Name of the city (and zip code). |
| Domain | omv:Location |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9 |
| Comments | None |

Table 5.150: Property: city

| street | |
|-----------------------|--|
| Name | street |
| Type | DatatypeProperty |
| Identifier | |
| Occurrence Constraint | optional |
| Category | Provenance information |
| Definition | Name of the street and number (address). |
| Domain | omv:Location |
| Range | xsd:string |
| Cardinality | 0:1 |
| OMV version | 0.9 |
| Comments | None |

Table 5.151: Property: street

Chapter 6

Using Metadata

Metadata information can be embedded in a variety of ways in the majority of ontology management services proposed so far. In this document we focus on the main usage scenarios.

We identified the following roles w.r.t. developing and deploying ontology metadata and ontologies:

- **Ontology developer** - The party primarily responsible for developing an ontology. They are expected to provide the majority of the metadata information and might use ontology metadata during ontology reuse processes.
- **Ontology contributor** - A party involved in an ontology development process. They are expected to create and use metadata in the same manner as the previous category.
- **Ontology reviewer** - A party responsible for evaluating an ontology. Ontology metadata information provides them with a useful means to ease the evaluation process. Further on, reviewers are expected to report on the reviewing process and its results in form of an entry in the metadata extension module for evaluation.
- **Ontology user** - A party applying an ontology for a specific purpose. They are expected to provide information about the application scenario in the appropriate extension module.

Chapter 7

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 (Oyster¹) and centralized (ONTHOLOGY²) sharing of ontology metadata based on OMV .

Next steps include the standardization of OMV on a wider scope by particularly including non-KnowledgeWeb 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.

Acknowledgements

This proposal is based on a huge number of discussions and many helpful arguments by persons from academia and industry. Especially we would like to thank our colleagues York Sure (AIFB), M. Carmen Suárez-Figueroa (UPM), Peter Haase (AIFB), Asunción Gómez-Pérez (UPM), Denny Vrandečić (AIFB) and Rudi Studer (AIFB). Furthermore, we thank our partners from the EU project Knowledge Web for their supporting discussions.

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2004-507482. The partners in this project are: University of Innsbruck (UIBK, Coordinator), École Polytechnique Fédérale de Lausanne (EPFL), France Telecom (FT), Freie Universität Berlin (FU Berlin), Free University of Bozen-Bolzano (FUB), Institut National de Recherche en Informatique et en Automatique (INRIA), Centre for Research and Technology Hellas /

¹<http://ontoware.org/projects/oyster>

²<http://www.onthology.org>

Informatics and Telematics Institute (ITI-CERTH), Learning Lab Lower Saxony (L3S), National University of Ireland Galway (NUIG), The Open University (OU), Universidad Politécnica de Madrid (UPM), University of Karlsruhe (UKARL), University of Liverpool (UniLiv), University of Manchester (UoM), University of Sheffield (USFD), University of Trento (UniTn), Vrije Universiteit Amsterdam (VUA), Vrije Universiteit Brussel (VUB).

Bibliography

- [1] A. Gangemi, D. M. Pisanelli, and G. Steve. An overview of the ONIONS project: Applying ontologies to the integration of medical terminologies. *Data Knowledge Engineering*, 31(2):183–220, 1999.
- [2] A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. Springer, 2003.
- [3] N. Guarino. Formal Ontology and Information Systems. In *Proceedings of the FOIS'98*, pages 3–15, 1998.
- [4] A. Lozano-Tello and A. Gomez-Perez. ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management*, 15(2), 2004.
- [5] D. L. McGuinness. Ontologies Come of Age. In *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.
- [6] National Information Standards Organization. Understanding metadata. NISO Press, 2004.
- [7] E. Paslaru Bontas, M. Mochol, and R. Tolksdorf. Case Studies on Ontology Reuse. In *Proceedings of the IKNOW05 International Conference on Knowledge Management*, 2005.
- [8] H. S. Pinto and J. P. Martins. A methodology for ontology integration. In *Proc. of the International Conf. on Knowledge Capture K-CAP01*, 2001.
- [9] T. Russ, A. Valente, R. MacGregor, and W. Swartout. Practical Experiences in Trading Off Ontology Usability and Reusability. In *Proc. of the Knowledge Acquisition Workshop (KAW99)*, 1999.
- [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] M. Uschold and M. Grninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2), 1996.

- [12] M. Uschold, M. Healy, K. Williamson, P. Clark, and S. Woods. Ontology Reuse and Application. In *Proc. of the Int. Conf. on Formal Ontology and Information Systems FOIS98*, 1998.
- [13] Y. Wand and R. Weber. Information Systems and Conceptual Modelling: A Research Agenda. *Information Systems Research*, 13(4), 2002.