

# A metaontology for applied ontology

Pawel Garbacz\* and Robert Trypuz

*The John Paul II Catholic University of Lublin, Lublin, Poland*

**Abstract.** The paper defines a schema in which to describe the cornucopia applied ontologies. In contradistinction to the main trend in engineering meta-ontology the main ideas that support this schema are inspired by philosophy. Namely, we look at the domain of applied ontologies from the point of view of a certain metaphilosophical tradition. In a nutshell, we identify an applied ontology by means of (i) a description of its language, (ii) the employed methodology (if any), (iii) a description of sources of ontological knowledge used in its development (if any), (iv) a description of its domain, (v) the list of its categories (if any), and (vi) a description of its use (if any). The final result of our considerations is an engineering artefact, i.e., an OWL ontology. For the sake of validation we employ it to describe several applied ontologies of diverse kinds.

Keywords: Meta-ontology, metadata, philosophy

## 1. Introduction

A symptom of maturity in any branch of knowledge is its methodological framework. A mature framework of this kind does not only list practical rules of thumb on how to perform research within this branch, but also describes the distinctive features of this knowledge: its goal(s), methods, open problems, etc. Although the field of applied ontology is relatively young, we have recently witnessed a growing awareness of its needs, goals, specific methods, types of approaches, etc. This methodological self-awareness has brought about a number of research results at the meta level, e.g., engineering methodologies, classifications of applied ontologies, etc.

The majority of the ideas that shape this meta-research is of the engineering or applicative nature. This predominance seems to be unbalanced if we compare it to the theoretical research on the ontological level. The domain of applied ontology is famous (or infamous) for being prone to the interdisciplinary influences, including those coming from philosophy. Our paper aims to extend the scope of this type of theoretical reflection into engineering meta-ontology. Namely, we present a meta-ontological schema for formal characterisation of applied ontologies, which schema is inspired by a certain metaphilosophical understanding of the notion of ontology.

The novelty of our approach consists in a new perspective from which we attempt to describe applied ontologies as if they were philosophical systems of thought. It turns out that this perspective is not that far from the engineering insight, however, it makes it possible to supplement this perspective with a few new components. In particular, our description of an applied ontology

- (1) focuses on the non-extensional aspects of its domain and categories:
  - (a) intensionality, due to which different entities may belong to this domain (resp. its categories) in different circumstances, e.g., at different moments in time;

---

\*Corresponding author: Pawel Garbacz, E-mail: garbacz@kul.pl.

- (b) intentionality, due to which that whether a given entity belongs to the domain (resp. its categories) depends on the intentional aspects of the process of ontology development, e.g., on the beliefs of its developers that shaped the ontology itself even if they did not make it to the ontology's axioms and definitions;
- (2) specifies the sources of knowledge used in ontology development;
- (3) provides a novel classification of ontological methodologies;
- (4) distinguishes between intended and actual application of the ontology.

In Section 2 we very briefly summarise the main results in engineering meta-ontology. This outline is not a survey of the state-of-the-art research, but a kind of conceptual set-up for our own contribution, which is presented in Section 3. The next section shows how our meta-ontological conception may be implemented as an engineering artefact in the form of an OWL ontology. (The DL formalisation of this ontology is given in the Appendix.) Section 5 compares our proposal to OMV, which is a recently defined meta-ontological schema.

## 2. Related work

The meta-ontological research in applied ontology does not match the scope and richness of existing ontological artefacts. The major part of the former endeavour revolves around (i) the problem of an adequate definition of the notion of applied ontology and (ii) subsequent classification(s) thereof.

As for the former, first let us note that the number of definitions of applied ontology matches the number of definitions of philosophical ontology if we factor both with the span of their lifetimes; for example, an early paper of N. Guarino (Guarino, 1997) lists eight cases of the former. In ontological engineering there seems to exist two different traditions or approaches of defining the notion of ontology. One tradition provides textual descriptions in ethnic language. Here the mainstream paradigm follows the (in)famous definition of Thomas Gruber (Gruber, 1993) by adding new components thereto (as in e.g., Borst, 1997 or Studer et al., 1998). None definition obtained within this tradition is formal in the sense of being rendered in a formal language. Their distinctive feature is semantic flexibility, which results in the fairly broad denotation of the term they define. In order to tame this open-texture the other tradition attempts to capture the precise meaning of the natural language phrases used in those textual definitions by means of a certain formalism, which usually happens to be the language of set theory. One of the most popular definition of this kind stems from Guarino and Giaretta (1995).

As far the classification of ontologies is concerned our survey of engineering meta-ontology showed that there are at least five orthogonal ways of classifying applied ontologies:

- (1) with respect to the “ontological type” of an applied ontology;
- (2) with respect to the domain of an applied ontology;
- (3) with respect to the formal type of an applied ontology's language;
- (4) with respect to the complexity of the structure of an applied ontology;
- (5) with respect to the intended usage of an applied ontology.

We believe that all of these types are self-explanatory except perhaps for the first.<sup>1</sup> An example of the first type of classification is presented in Guarino and Giaretta (1995). N. Guarino and P. Giaretta describe two meanings of the term “ontology”: (i) ontology as a logical theory (of a certain type) and (ii) ontology as a conceptualisation. An applied ontology construed as a logical theory refers to an

---

<sup>1</sup>In Garbacz and Trypuz (2011) we describe these types in greater detail.

“engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words” (Guarino, 1998, p. 4). Its content should be rich enough so that the ontology could provide the meanings of its non-logical terms. If the ontology construed as a logical theory is created in the context of model theory, it is usually accompanied by the model-theoretic structures it represents, which structures eventually turn out to be sets. Guarino and Giaretta claim that these set-theoretical structures themselves may also be considered as ontologies although only in the derivative sense, namely, ontologies construed as conceptualisations.

The requirement that an applied ontology needs to be a logical theory seems to be too restrictive as far as the common use of this term is concerned. Its usage is rather relaxed, so the above distinction may be generalised to the distinction between ontologies construed as representing artefacts and ontologies construed as represented structures, which need not necessarily be built out of sets. The former are bound to be stored in a medium of some sort, which usually takes the form of a formal language.

Although the systematic research on the proper meta-ontological level is not predominant in engineering ontology, there has been several initiatives that go beyond simple definitions or classifications of ontologies. For example, a number of schemas or standards for representing meta-information are already available, e.g., the Dublin Core standard (<http://dublincore.org>), the FOAF vocabulary ([www.foaf-project.org/](http://www.foaf-project.org/)), the metadata for the Swoogle search engine (Ding et al., 2004), etc. Ontology Metadata Vocabulary (OMV) is one of the most recent proposals in this respect (see Hartmann et al., 2005; Palma et al., 2008). OMV is a formal meta-ontological framework in which to describe applied ontologies from the point of view of knowledge representation. Currently it consists of two modules: OMV Core and OMV Extensions, which deal, respectively, with the general aspects of engineering ontologies and with the task- and application-specific ontology-related information. We will focus here on the former module.

The main conceptual distinction drawn within OMV Core separates ontology bases from ontology documents.

An *Ontology Base* (OB) represents the abstract or core idea of an ontology, so called conceptualisation. It describes the core properties of an ontology, independent from any implementation details. [...]

An *Ontology Document* (OD) represents a specific realization of an ontology base. Therefore, it describes properties of an ontology that are related to the realization or implementation (Hartmann et al., 2005, p. 908).<sup>2</sup>

Needless to say, the properties of an ontology base may be different from the properties of those ontology documents that implement it. Both are characterised by the property that represents ontology’s authorship and the property that represents license models utilised in ontology’s development. An ontology document is further described by:

- (1) the ontology’s domain;
- (2) the engineering methodology and software tools that were used by the ontology authors of the ontology documents;
- (3) the language in which the ontology document is expressed;
- (4) the type of the ontology document (e.g., upper-level ontology, domain ontology, etc.);
- (5) the task specifications for the ontology.

---

<sup>2</sup>The later version of OMV from Palma et al. (2008) uses a different terminology. An ontology base is called there a conceptualisation and ontology document is called simply an ontology.

OMV distinguishes between a formal language and its syntax, however, the precise import of this distinction is not further elaborated.

Figure 1 presents all main categories of this approach and the relationships therebetween. The whole OMV schema is rendered in OWL with a number of “predefined” instances and datatype properties’ values.

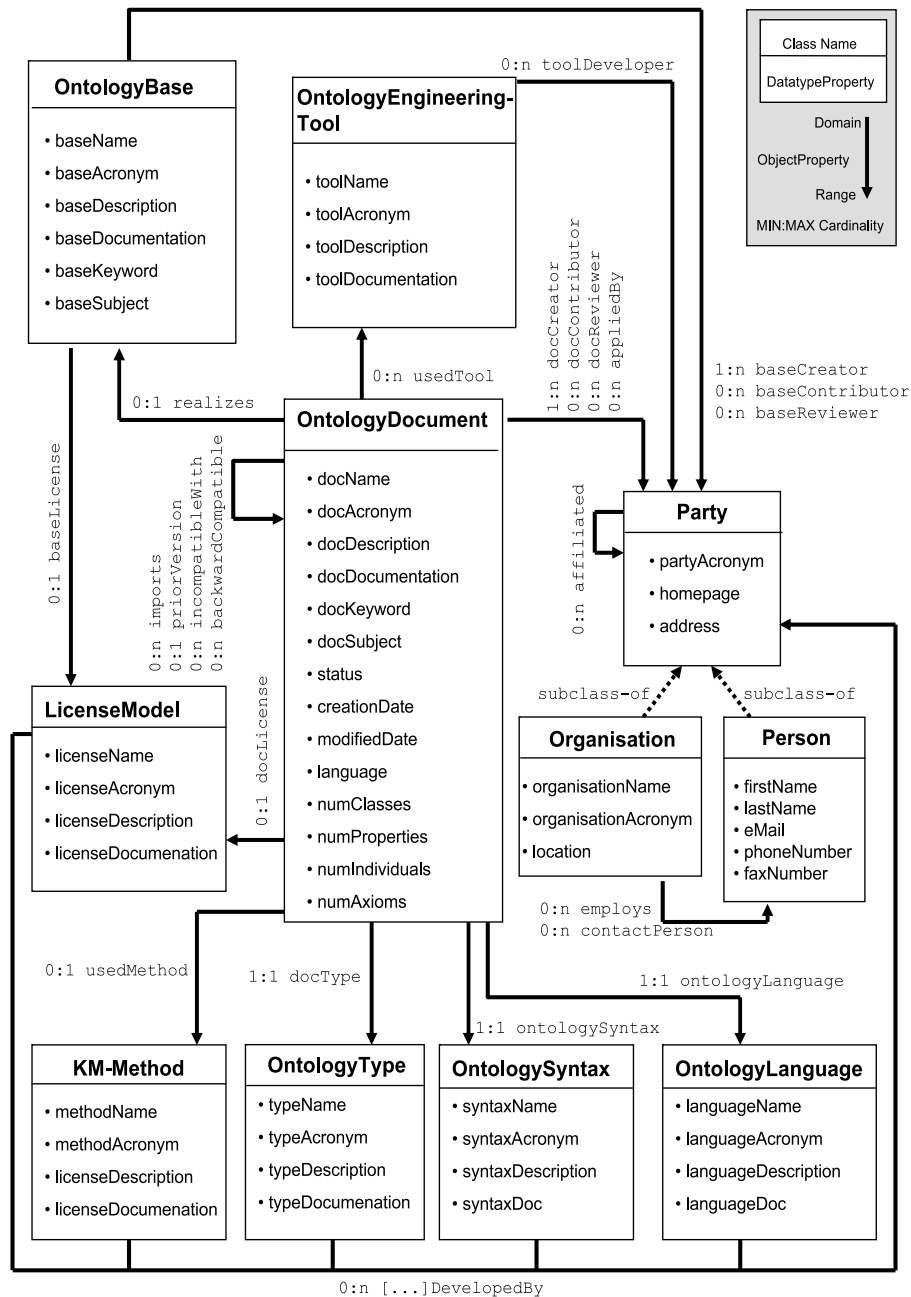


Fig. 1. UML model of OMV (from Hartmann et al., 2005).

### 3. From metaphilosophy to metaontological engineering

Our goal is to provide a meta-ontological schema in which to detail the philosophically outstanding aspects of engineering ontologies. We recognise our approach as complementary to the current engineering meta-ontology. Therefore, for instance, our schema does not include any information on the engineering aspects of applied ontologies, e.g., the specification of the software used in ontology development. The methodology we followed to define such a schema starts from the search for a metaphilosophical description of philosophical ontology. The most adequate description found is then adapted to the specific features and needs of applied ontologies.

Our starting point is the meta-ontological schema developed by Antoni B. Stepień for the sake of (metaphilosophical) analysis of philosophical ontologies (cf. Stepień, 1993). He argues that the adequate description of a given philosophical ontology requires four components:

- (1) characterisation of the sources of ontological knowledge and the methods used in the development of the ontology, e.g.,
  - rational intuition or insight (cf. BonJour, 1998);
  - eidetic variation (cf. Smith, 2007);
  - abstraction (cf. Klima, 2008);
  - linguistic analysis (cf. Ziff, 1960);
  - formal deduction;
  - etc.;
- (2) explanation of the concept of being that is presupposed or implied by the ontology, e.g.,
  - the being as something definite, e.g., as having certain properties;
  - the being as something possible;
  - the being as an object of thought, i.e., as something that someone can think of;
  - etc.;
- (3) list of the main ontological categories together with the relations therebetween, e.g., endurants, properties, parthood, etc.;
- (4) description of the role or function of the ontology in the broad philosophical framework, e.g., whether the ontology in question provides a foundation for a theory in ethics.

We consider an *applied ontology* as an informational artefact whose main role is to represent a certain domain in such a way that a properly configured computer system is capable of storing and processing this representation. This definition is relatively broad since it covers all kinds of meaningful and processable documents, e.g., even an MS Word document is an ontology in this sense provided that it does represent something in a computer-readable way. Our understanding of the term “informational artefact” is such that an applied ontology is hybrid with respect to the first type of classification of ontologies in the list on page 2. Namely, any applied ontology is construed as if it was composed both of a representational carrier of some kind (e.g., a text) and a related conceptualisation (of a certain domain). In other words, it is a fusion of (i) a thing that represents something and (ii) a thing that is represented thereby. We do not interpret this description as a definition of applied ontology since we want to approach ontological engineering from the descriptive and not from the prescriptive perspective, i.e., we do not want to exclude certain software artefacts just because they do not fall under some definition of applied ontology. Instead, we pursue for a (meta-ontological) schema to describe applied ontologies of any type.

In order to adapt Stepień's conception for the purposes of knowledge engineering we transformed it into the following schema. An applied ontology is described there by means of:

- (1) description of its medium, usually the formal language in which the ontology is rendered;
- (2) methodology used in its development;
- (3) description of the sources of ontological knowledge utilised; this usually will include various characterisations of the cognitive, psychological, philosophical ideas that shape the main traits of the ontology;
- (4) description of the ontology's domain;
- (5) list of the ontology's categories (if any);
- (6) description of actual and/or possible (intended) applications of the ontology in an information system.

### 3.1. Language

Firstly, let us emphasize that we do not distinguish between ontological conceptualisations and ontological realisations/implementations (as OMV does) because usually even minor changes in the way an "abstract" idea is formulated lead to different views on the domain in question. In particular, when an ontology is rendered in two formal languages, say, the full first-order logic and some weak description logic language like OWL DL, the two formalisations are in fact two different theories due to the difference in the expressivity of their languages. Consequently, each will represent its domain in a slightly (or, as it may happen, radically) different way. There are well-known examples of such cases within the class of the so-called upper level ontologies, where the OWL version of a given ontology is logically much leaner than its full-fledged first-order formalisation – *nomina sunt odiosa*. The resulting differences are of particular importance when an ontology in question is to disambiguate the natural language discourse for the sake of, say, semantic negotiations within a network of agents. Then even minor changes in the formalisation may result in significant semantic discrepancies and consequently in negotiation failure. In sum, the hybrid or "holistic" conception of ontology we presuppose implies that each ontology is expressed exactly in one language. In the practical terms, it gives an engineer a more precise tool relieving him or her from the effort to pin down the elusive notion of conceptualisation.

We adopted the classic understanding of language by means of which a language is construed as a triad constituted by the syntactic, semantic and pragmatic aspects (cf. Peirce, 1931, Section 227, Vol. 2; Morris, 1938). This implies that two different syntactic formats of one knowledge representation language are conceptualised here as two different languages. Therefore, we cannot incorporate the OMV distinction between a language and its syntax as our notion of language includes its syntactical features. For instance, OWL RDF/XML and OWL Manchester Syntax are understood here as two different formal languages. Brushing the philosophical motivation aside we believe that there is a practical rationale for this understanding. If one of the aims of ontological engineering is knowledge exchange, then using an applied ontology for this purpose presupposes that this very ontology is sharable between two computer systems. If the ontology in question comes covered in two different syntactic disguises, then it requires an extra functionality from these systems – each of them should be able to read and write this ontology in both formats. Thus, "from the point of view of the computer system" that employs this ontology there is no one but two different information artefacts that it needs to handle. And the reason is that there is no one but two languages in which the ontology is encoded. Bluntly speaking, for each language your application needs an extra piece of code to handle it. In our meta-ontological schema an applied ontol-

ogy does not amount to a theory in the sense of formal logic, but is an engineering artefact that may “implement” such a theory down to the nitty-gritty details.

We believe that one needs to draw an important distinction in the domain of ontological languages that differentiates between the languages that support formal inference and the languages that do not. By an *inference* we understand any cognitive process whereby a new piece of knowledge is obtained on the basis of some previously acquainted knowledge. An inference is *formal* if its validity depends on the structures of its premises and conclusion and not on their content. A language *supports* formal inference if its semantics is sufficiently developed so that one can define therein the standard model-theoretic notion of validity. For instance, if an ontology is a simple list of objects rendered as an CSV (*comma-separated values*) file, no formal inference within this ontology is supported by the CSV format.

### 3.2. Methodology

As for the description of the methodology used, we restrict this aspect to a limited number of generic engineering methodologies, e.g., OntoClean (Guarino and Welty, 2009), METHONTOLOGY (Fernández et al., 1997), DILIGENT Pinto et al., 2009, etc.

We organised these generic methodologies into the following classification:

- (1) postontological methodologies, i.e., the methodologies that presuppose the existence of some (applied) ontologies:
  - (a) evaluative methodologies, e.g., OntoClean (cf. Guarino and Welty, 2009);
  - (b) ontology mapping methodologies, e.g., ONIONS (cf. Gangemi et al., 1996), (cf. FCA-Merge Stumme and Maedche, 2001);
  - (c) ontology modification methodologies, e.g., scenarios 3–6 of the NeOn framework (cf. Gómez-Pérez et al., 2011);
- (2) ontological methodologies, i.e., the methodologies that do not presuppose the existence of any (applied) ontologies:
  - (a) methodologies for developing ontologies “from the scratch”, e.g., On-To-Knowledge (cf. Staab et al., 2001), METHONTOLOGY (cf. Fernández-López et al., 1997);
  - (b) methodologies for developing ontologies from non-ontological resources
    - (i) methodologies for developing ontologies from structured resources, e.g., scenario 2 of the NeOn framework (cf. Gómez-Pérez et al., 2011);
    - (ii) methodologies for developing ontologies from non-structured resources, e.g., various methodologies of ontology population from text (see Cimiano, 2006).

We do not require that each applied ontology should be developed by means of an engineering methodology, but if it was, we record this fact.

### 3.3. Sources of ontological knowledge

In contradistinction to Stepień’s schema we separate the methodological aspect from the epistemological aspect of an applied ontology. The latter is represented here by means of the description of a source of ontological knowledge. The role of this description is to document the ontological choices made when the ontology was designed and developed.

We divide the category of knowledge sources into two subcategories: descriptive knowledge and normative knowledge. A source of knowledge is *descriptive* if it provides factual information on how certain

things are. A source of knowledge may be descriptive without being truthful, reliable, authoritative, etc. Thus, we include here not only scientific conceptions and theories, but also philosophical or common-sense beliefs. For the sake of clarity, we organise the descriptive sources of knowledge according to the top-most categories of the Dewey Decimal Classification (Dewey, 1876), however, this assumption is not crucial to our approach and may be dropped in favour of some other classification. On the other hand, a source of knowledge is *normative* if it provides non-factual information on how certain things are expected (by someone or someones) to be. This category includes all kinds of standards, requirement specifications, technical designs for the legacy data systems, etc. This distinction is parallel to the distinction between two kinds of the direction of fit among speech acts: “word-to-world” and “world-to-word” – see Searle and Vanderveken (1985):

For example, if I make the prediction that you will leave the room, then it is a consequence of the illocutionary point of my prediction that the propositional content is supposed to match an independently existing reality. If you leave the room my prediction is said to be true, if you fail to leave the room it is said to be false [word-to-world fit]. If on the other hand I order you to leave room, then it is part of the illocutionary point of my order to you to act in a such a way as to match your behavior to propositional content of the order. If you do leave the room by way of carrying out the order, the order is not said to be true, rather to have been obeyed, and if you fail to leave the room the order is not said to be false, but rather to have been disobeyed [world-to-word fit] (Searle & Vanderveken, 1985, p. 92–93).

Orthogonally to the previous distinction, we distinguish two types of sources of ontological knowledge: explicit and implicit, or, more precisely speaking, we distinguish two types of relations that a source of information may bear to a given applied ontology. A source of knowledge is called *explicit* with respect to an ontology if it is mentioned as such (i.e., as being used in its development) in the technical documentation of this ontology. On the other hand, a source of knowledge is qualified as *implicit* if its propositional content follows from the axioms of this ontology (if any) or from this documentation. For example, when an ontology contains a category whose all instances are endurants, we assume that this ontology is based on the so-called endurantism<sup>3</sup> even if this philosophical position is not mentioned in its documentation.

The distinction between explicit and implicit knowledge source is of crucial importance in two contexts: ontology evaluation and ontology alignment. Both context favour explicit knowledge as it increases conceptual transparency and verifiability and decreases the need for human interpretation, which is prone to error. As for the former context, it seems to be a truism now that an ontology that is better documented is more functional, *ceteris paribus*, than the one that is less or worse documented (cf. Uschold and Grüninger, 1996, p. 110, Farquar et al., 1997, p. 720, Fernandez et al., 1997, p. 33–34, Masolo et al., 2003, p. 4). This implies then the explicit sources of knowledge may be a feature with respect to which we evaluate an applied ontology. As for the latter context, an explicit source of ontological knowledge

<sup>3</sup>One of the popular characterisation of this philosophical position, together with a competing view, opens McKinnon (2002):

It is now usual to say that something persists iff it is located at more than one time. This neutral term gives us a means of framing the question, how does a thing persists? One answer is to say that a thing’s persistence involves its perduring. What is it for a thing to perdure? Generally, it has been held that perdurance involves persisting in virtue of having temporal, as well as spatial parts. And what is it for a thing to endure? Often, this is put in terms of thing’s being wholly present at all times at which it exists (McKinnon, 2002, p. 288).

The notion of temporal part can be defined as follows:  $x$  is a temporal part of  $y$  at time  $t$  if  $x$  is part of  $y$ ,  $x$  exists only at  $t$ , and  $x$  overlaps every part of  $y$  that exists at  $t$ .



(e.g., endurantism) makes it possible to arrange various ontologies in clusters and make certain distinctions that are otherwise inaccessible. For instance, the ontology ISO 15926, which is explicitly based on the perdurantism, contains the category Event. And the ontology PROTON also contains a category by the same name. Still, since we know that the latter is based on the endurantism<sup>4</sup> and the former is based on perdurantism, we will not be tempted to align both categories.

Our schema allows for one ontology being a source of ontological knowledge for another ontology. Moreover, it makes it possible to differentiate between that a source of knowledge informs a whole ontology and that it informs one (or many) of its categories. As in the case of methodological aspect, we do not require that an applied ontology should be informed by any source of knowledge.

### 3.4. Domain

The description of the domain of an applied ontology corresponds, *mutatis mutandis*, to philosophical characterisation of the concept of being. In philosophy, the latter concerns the formal properties and/or content of the most general notion whose range is intended to cover all genres of existence. In knowledge representation the range of entities which a given applied ontology commits to depends on a particular engineering problem that this ontology is to solve. Thus, its domain may be compared to the concept of being *modulo* this problem, i.e., it will contain those entities that in the view of the developer(s) of this ontology we need to posit in order to solve this problem. For instance, if the problem at stake concerns chemical engineering, one can expect that the description of the domain will mention events and processes and not sets or proper classes (in the sense of Neumann–Bernays–Gödel set theory).<sup>5</sup>

Even a cursory survey of existing applied ontologies clearly shows that in most cases one cannot simply identify the domain of an ontology with the set of entities that this ontology represents. The reason is that the former exhibits two non-extensional aspects:

- (1) might be intensional, i.e., there are ontologies such that the extension of their domains depends on a particular circumstance, e.g., they change with respect to a moment in time;
- (2) might be intentional, i.e., for the majority of applied ontologies the extension of their domains eventually depends on the beliefs (and perhaps other intentional attitudes) of the ontology's developer(s) and not all of those beliefs become part of the ontology itself.

Although the latter aspect is subjective, from the epistemological point of view it is usually objectified in the technical documentation of this ontology, which may include the relevant research publications of their developers. Consequently, although we assume that all intentional factors that define the domain of the ontology are expressed in this documentation, for the practical purposes our schema identifies the author(s) of this ontology as well.

The non-extensional character of the ontological domains is captured in our approach by means of the notion of flag. Namely, the domain of an applied ontology may be characterised with respect to the following features:

- (1) modality flag:
  - (a) actual – if the domain contains *only* actual entities, i.e., those that either existed, exist, or will exist;

<sup>4</sup>As a matter of fact, PROTON ontology specifies the DOLCE ontology as one of its knowledge sources and it is the latter ontology that explicitly mentions endurantism as its source of ontological knowledge Terziev et al. (2005).

<sup>5</sup>This idea of parallelism between philosophy and knowledge representation can be traced back to the seminal paper McCarthy (1980) and his use of the term “ontology”.

- (b) possible – if besides actual entities the domain contains also possible ones, e.g., unicorns.  
In order to establish the value of this flag one needs to consult the technical documentation of the ontology in question and review the instances of its categories (if any are represented in the ontology or described in the documentation);

(2) objectivity flag:

- (a) mind-dependent – if the domain contains only mind dependent entities, i.e., those whose existence rigidly (either constantly or historically) depends on the existence of (beliefs, desires, or intentions, etc. of) some particular agent;<sup>6</sup>
- (b) mind-independent – if the domain contains only mind independent entities;
- (c) mixed – if the domain contains entities of both kinds.  
In order to establish the value of this flag one needs to review the list of categories of the ontology in question together with its technical documentation in the search of the mind-dependent categories;

(3) scope flag:

- (a) global – if, according to the developers of the ontology, the domain covers the whole of reality, i.e., if according to their beliefs
  - there exists no ontological category that is more general than the most general category of the ontology in question and
  - there exists no individual entity that does not fall under one from the categories of this ontology;
- (b) local – if, according to the developers of the ontology, the domain does not cover the whole of reality.  
In order to establish the value of this flag one needs to consult the technical documentation of the ontology in question in the search of any declarations of the above kind. The well-known qualifications: “upper-level ontology”, “top-level ontology”, or “general ontology” vs “domain ontology” or “middle-level ontology” may be instructive in this respect; however, they are not decisive as their semantics is not sufficiently clear and stable.<sup>7</sup>

The reader is advised to bear in mind that all flags are to account for the non-extensional character of the domain of a given applied ontology. In particular, they are to absorb at least some of the beliefs of its developers including those that have not been formally expressed in the ontology’s axioms or definitions. For instance, if an ontology is built by a developer under the realist assumption to the effect that all entities in the domain of this ontology are mind-independent, then the objectivity flag for this ontology should be set accordingly. Unless the developer’s beliefs turn out to be inconsistent with the propositional content of the ontology we interpret them as infallible. Thus, for instance, if describing a given ontology (i) you know that it is an intention of its developer that its domain contain only mind-independent entities and (ii) you know that a certain entity belongs to this domain but (iii) you are wholeheartedly convinced that this entity is after all mind-dependent, then you still should set the objectivity flag equal to “mind-independent”.

---

<sup>6</sup>We refer here to the definition of the dependence relation and its kinds defined in (Thomasson, 1999, p. 29–33), which is used, among other ontologies, in the DOLCE ontology (Masolo et al., 2003).

<sup>7</sup>Needless to say, an ontology and its documentation may not be specific enough to determine the value of any of these three flags.

We found the idea of domain flags useful, for instance, as a means to characterise those aspects of applied ontologies that are the subject of the realism/anti-realism debate (see the recent exchange between Merrill (2010) and Smith and Ceusters (2010)).

In contradistinction to the previous two aspects, we do require that each applied ontology should represent a unique domain. As in the case of its language this is a consequence of the “holistic” construal of the notion of applied ontology we assumed.

### 3.5. Categories

The description of the domain leads us to the list an ontology’s categories, including those expressible by  $n$ -ary ( $n \geq 2$ ) predicates. Thus, we shall look at the categories as sets of entities which are either individual objects, in the case of unary categories, or  $n$ -tuples of individual objects, in the case of  $n$ -ary categories. From the philosophical point of view, the latter may be viewed as states of affairs (e.g., as *Sachverhalten* of A. Reinach – see Smith, 1987).

Although one can characterise the notion of ontological category in a similar way to the above description of the notion of ontological domain (i.e., by defining flags for categories), we will focus on the extensional aspect of the former. Perhaps we should only add that the intensional aspect of an ontological category has an additional twist to it. One object may fall under two categories from two ontologies although both categories represent it in a different way. This difference usually implies different types of properties that these categories ascribe to this object. For example, English is an instance both of the category Language from the ISO 15926 ontology and Language from the SUMO ontology, however both ontologies represent it in a different way. ISO 15926 represents all languages as sets of sets of individual objects and SUMO represents them as individual physical objects that represent something. Focusing on the extensional aspect of ontological categories we neglect also this type of intensionality.

We capture the non-extensional nature of ontological categories in the standard way, i.e., by means of the notion of possible world. We introduce this notion for two reasons. First, the documentation of some ontologies envisage the possibility to the effect that some (or even all) of their categories may have extensions different from the ones that they actually have. Such cases are either stated explicitly, when, for instance, those extensions are claimed to be dependent on moments in time (as in the case of BFO – see Grenon, 2003, p. 2–3), or implicitly, when (i) the language of a given ontology contains modal connectives (as in the case of DOLCE) or (ii) when the ontology’s documentation mentions some non-existent situations or entities, e.g., unicorns. Secondly, following Guarino and Giarretta (1995) we intend to account for possible changes to the extensions of categories. Similarly to Guarino and Giarretta we believe that the meanings or interpretations of categories are supposed to remain the same even if the extensions of the categories change due to different courses of events in the world (see also Guarino et al., 1994; Guarino, 1998; Guarino et al., 2009).

In other words, the notion of possible world is to do justice to the fact that (at least) some ontologies “take into account” alternatives to the actual world. In order to capture this fact, we use the notion of possible world because of its known versatility, but we do not impose any particular interpretation thereof since this interpretation is established in each case by the intentions of the ontology’s developers when they mention what kinds of alternatives are envisaged by their artefact. In principle, it may happen that each ontology defines its own set of possible worlds, in which case any “trans-ontological” comparison of categories would be impossible, except for the basic case of the actual world.

Then, the *intended extension* of an applied ontology’s category  $c$  in possible world  $w$  – formally:  $\text{ext}(c, w)$  – is understood here a set of objects that are believed by the authors of this ontology to fall

under this category in this world. If an ontology does not take into account any alternatives to the actual course of events, its categories are relative just to one possible world: the actual world.

Since we provide neither a theory of ontological categories nor even their criteria of identity, we are not in a position to say that category  $c_1$  from ontology  $O_1$  is identical to category  $c_2$  from ontology  $O_2$ . However, in order to be able to draw at least some comparisons between different ontological categories one may use the following relations:

- (1) category  $c_1$  from ontology  $o_1$  is *extensionally included relative to* possible world  $w$  in category  $c_2$  from ontology  $o_2$  if  $\text{ext}(c_1, w) \subseteq \text{ext}(c_2, w)$ ;
- (2) category  $c_1$  from ontology  $o_1$  is *extensionally included* in category  $c_2$  from ontology  $o_2$  (formally:  $c_1 \preceq c_2$ ) if  $\forall w [\text{ext}(c_1, w) \subseteq \text{ext}(c_2, w)]$ ;
- (3) category  $c_1$  from ontology  $o_1$  is *extensionally identical* to category  $c_2$  from ontology  $o_2$  (formally:  $c_1 \equiv c_2$ ) if  $c_1 \preceq c_2$  and  $c_2 \preceq c_1$ .

Including both the description of the domain of an applied ontology and the list of its categories we are able identify not only which portion of reality is represented by the ontology, but also on what aspects or features of this portion the ontology focuses. Since neither the notion of domain nor the notion of ontological category is reduced to the notion of set, our schema allows for a number of intensional interpretations (as, for example, described in Guarino, 1998).

Sometimes the authors of an applied ontology explicitly intend to *exclude* certain entities from its domain. Therefore, we posit that the list of the ontology's categories may be accompanied by the list of the rejected (i.e., believed by the authors to be empty) categories. For instance, Batres et al. (2007) specifies the four-dimensional perspective for the ISO 15926 ontology, which makes any ontological category that contains endurants empty. We interpret this fact saying either that the ISO 15926 rejects any category of endurants or that it rejects the category of all endurants.

### 3.6. Domain and categories

Suppose that one is able to fix the intended extensions of all categories of an applied ontology  $O$ . The *intended extension of the domain* of ontology  $O$  (in possible world  $w$ ) is the union of all intended extensions of those categories in  $w$ .<sup>8</sup> As in the case of ontological categories one can explicitly define the relations of extensional inclusion and identity between ontological domains. Then we posit that domain  $d_1$  of ontology  $O_1$  is identical to domain  $d_2$  of ontology  $O_2$  if (i)  $O_1$  and  $O_2$  set the same flags for their domains and (ii)  $d_1$  is extensionally identical to  $d_2$ .

### 3.7. Use

Finally, our description of the ontology's applications is composed of two parts. The first part specifies the *intended type* of the ontology's usage. To this end, we currently employ the concepts and terminology defined in Mizoguchi (2003). The second component describes the *actual* use of the ontology in a *particular instance* of computer system.

As for the former let us simply report that Mizoguchi distinguishes (and describes) 8 types of roles that an engineering ontology may perform:

---

<sup>8</sup>Our schema also deals with a borderline case where an applied ontology does not contain any categories at all but is just a list of URIs or GUIDs that identify certain individual entities. Then we construe the intended extension of this ontology (in every possible world) as the set of these individuals.

- (1) being a common vocabulary;
- (2) defining a data structure;
- (3) explicating implicit presuppositions;
- (4) enhancing semantic interoperability;
- (5) explicating a design rationale (for an engineering artifact);
- (6) systematising a body of knowledge;
- (7) being a meta-model;
- (8) being a theory of content.

An ontology may be developed without being intended for any general usage situation and without being actually applied to any instance of information system.

\*\*\*

Our “holistic” interpretation of the notion of ontology implies a necessary criterion of identity for ontologies. One ontology is *identical* to another ontology only if (i) both ontologies are expressed in the same language, (ii) they represent the same domain, and (iii) if one of them contains any (general) categories, then their categories are extensionally identical, i.e., if  $C_1$  and  $C_2$  are the sets of categories from the ontologies in question, then relation  $\equiv$  is a bijection from  $C_1$  onto  $C_2$ . Although this criterion does not provide the sufficient condition for the identity of ontologies, it can provide negative results, which state that two ontologies are not identical.

This criterion also leads to the following adequacy condition for meta-ontological descriptions of applied ontologies. Among the six aspects of applied ontologies listed on page 6 we find the descriptions of the ontology’s language and domain (mentioned as items 1 and 4 in the list) mandatory for any adequate characterisation of this ontology. The list of categories (item 5 in the list) is also mandatory if the ontology in question has any categories at all. This means that any meta-ontological description of an applied ontology that lacks any of these components does not sufficiently characterise this ontology as an informational (representational) artefact. Other aspects are optional, i.e., a meta-ontological description of an applied ontology is not claimed to be defective if it lacks any of them.

#### 4. Implementation

Our meta-ontological schema is not restricted to any specific language although we reckon ethnic language is the most flexible tool fit for this purpose. Nonetheless, because we engaged ourselves in a discipline of applied research, we built a system of OWL ontologies that incorporates the main components of this schema so that the philosophical perspective we provide could become computationally operable. We consider this system as a corner stone of a prospective repository of applied ontologies to be built.

The aforementioned system of ontologies is depicted in Fig. 2. It contains four components:

- (1) METAONT – an OWL ontology that contains all (meta-)categories of our schema;
- (2) METAONT\_POPULATED – an OWL ontology that besides importing the METAONT (meta-)categories also contains these instances thereof that we reckon to be shared by more than one of our OWL descriptions of applied ontologies;
- (3) a set of OWL ontologies describing applied ontologies;

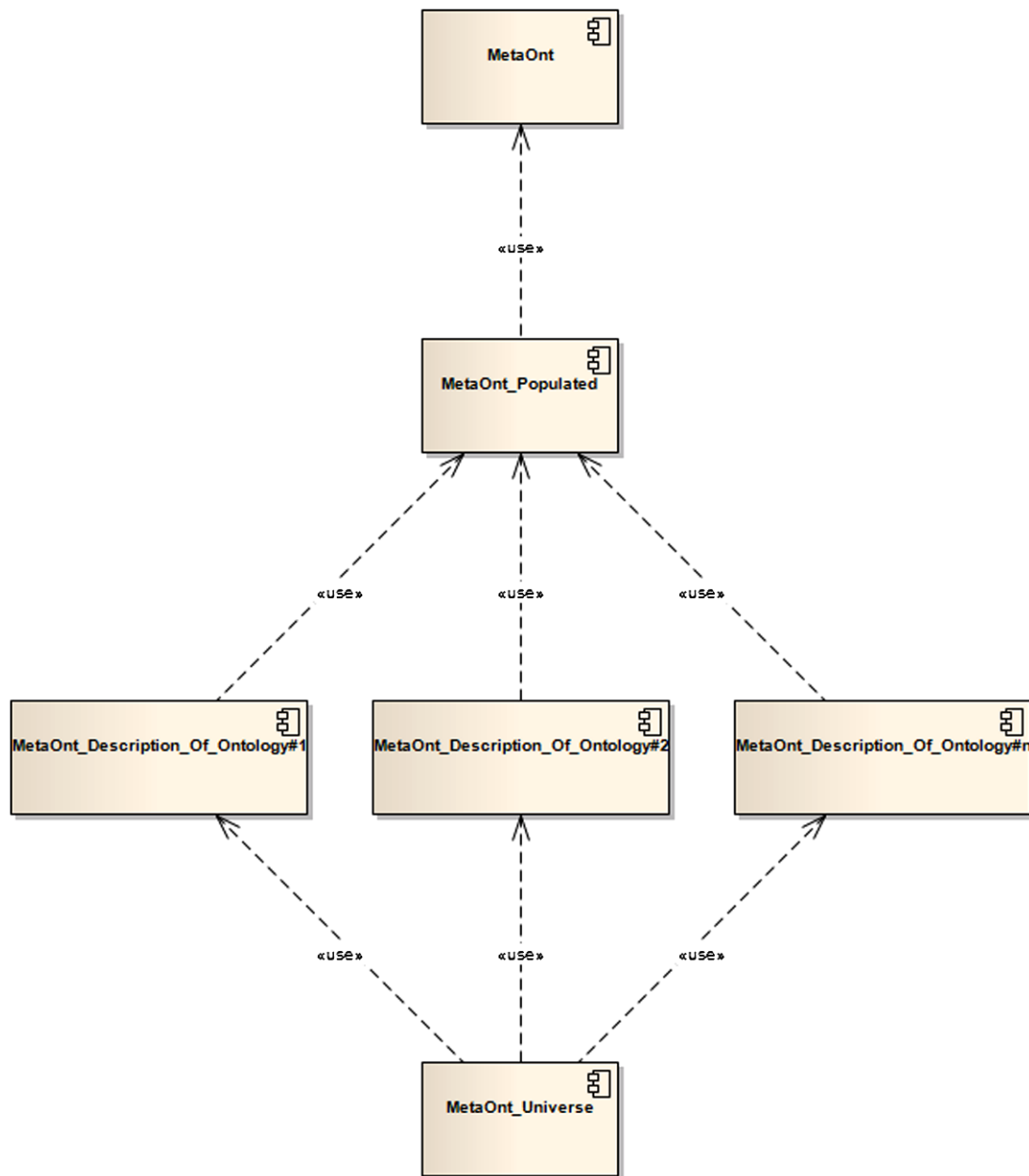


Fig. 2. METAONT architecture. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-130120>.)

- (4) METAONT\_UNIVERSE – an OWL ontology that imports all these descriptions and provides a platform for a comparison thereof.

Due to the lack of space in what follows we present here only the first component of this system.

Incorporating the main ideas of our conceptual schema METAONT is an OWL 2 DL ontology with the DL expressivity equal to SRIQ(D). It contains 36 OWL classes (besides owl:Thing and owl:Nothing) with eight top-level categories:

- (1) Individual,
- (2) Category,
- (3) Domain,
- (4) Language,
- (5) Methodology,
- (6) Ontology,
- (7) SourceOfKnowledge,
- (8) Usage.

METAONT's classes correspond to the items in the list from page 6. The full subsumption hierarchy for OWL classes is presented in Fig. 3.

Moreover, the OWL (meta-)ontology contains 18 OWL object properties, among which we should mention the following ones:

- (1) acknowledges,
- (2) rejects,
- (3) represents,
- (4) isDevelopedBy,
- (5) isDevelopedWithin,
- (6) isExpressedIn,
- (7) isInformedBy,
- (8) isUsedAs,
- (9) isExtensionallyIncludedIn and its subtype
  - (a) isExtensionallyEquivalentTo,
- (10) belongsTo,
- (11) fallsUnder.

The role of the first eight properties is to relate the Ontology class to other top-level classes in METAONT – see Fig. 4. That is to say, those properties (together with their values) describe all aspects of our meta-ontological schema we discussed in the previous section. We should add that the *acknowledges* property has a subtype that identifies the root category within a given ontology. (We allow that one and the same category may be a root in one ontology and be subsumed by some other category in another ontology.) Property 9 and its subtype are to relate the categories of one ontology (or its domain) to the relevant categories (resp. domain) of another ontology as described in page 11 above. The last two connect the three classes: Category, Domain and Individual.

Finally, we use eight OWL datatype properties:

- (1) hasModalityFlag,
- (2) hasObjectivityFlag,
- (3) hasScopeFlag,
- (4) hasArity,
- (5) hasGuid,
- (6) hasVersionID,
- (7) hasPrefix,
- (8) isDescribedIn and its two sub-properties:

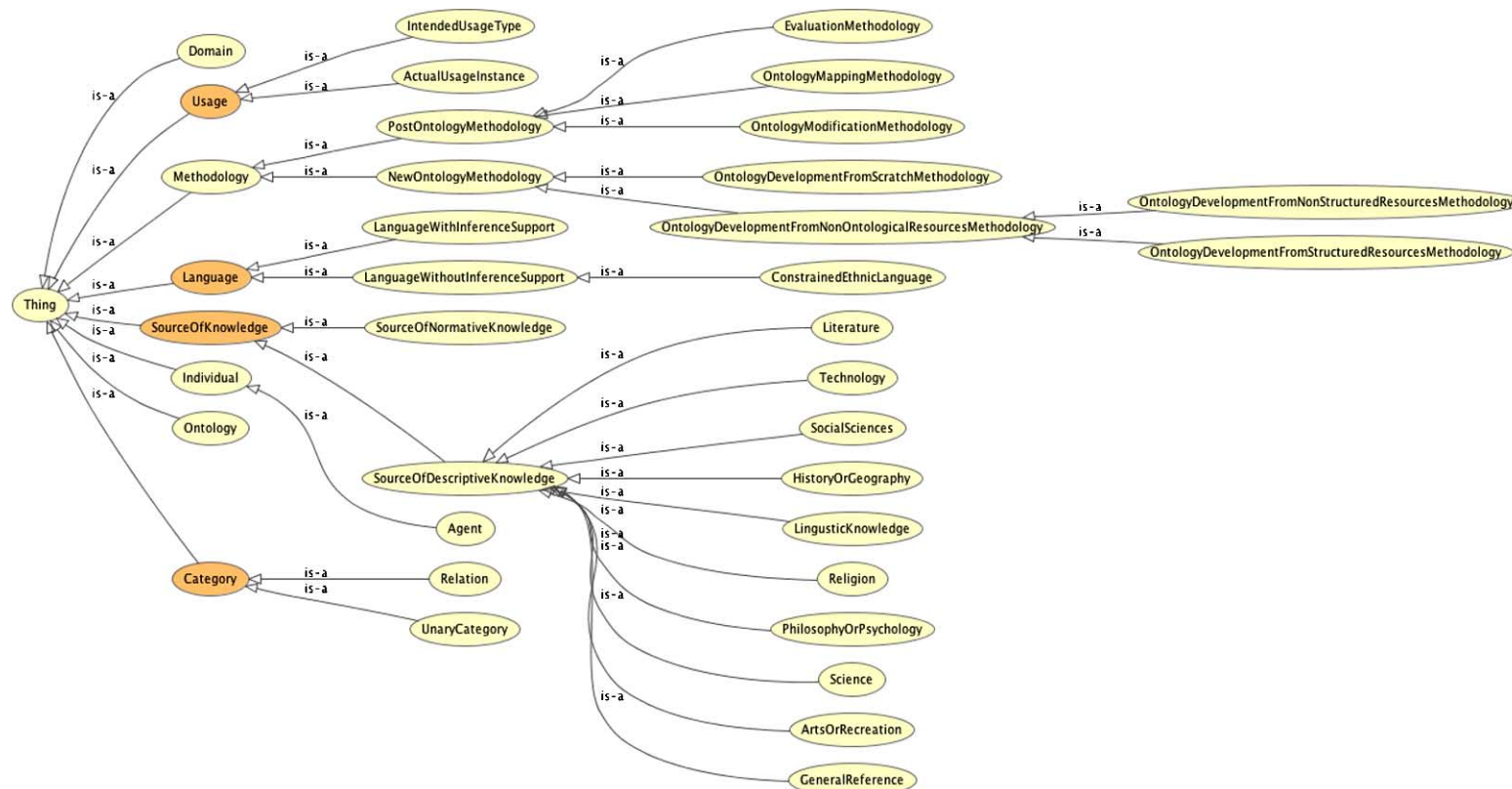


Fig. 3. Subsumption in METAONT. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-130120>.)



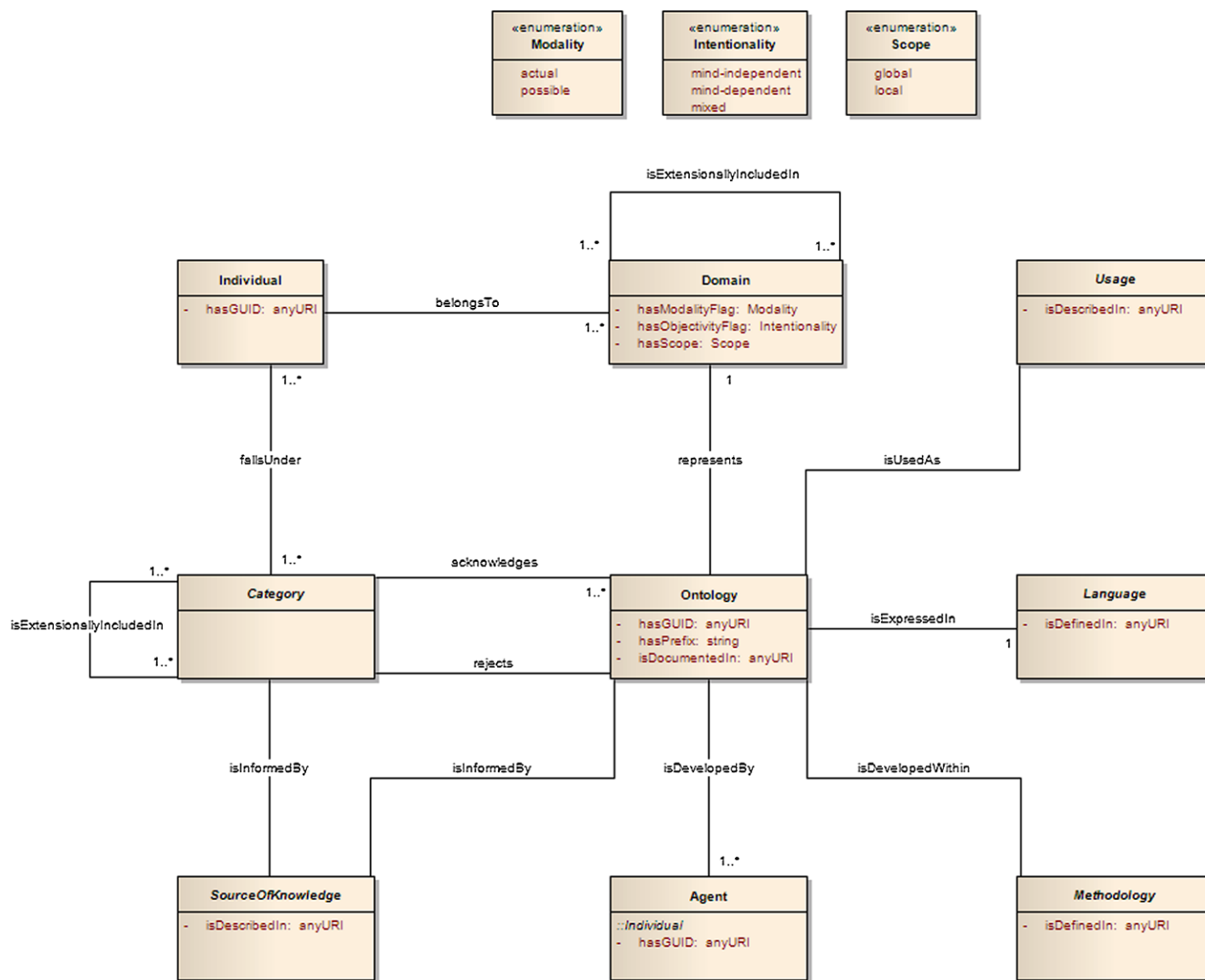


Fig. 4. UML model of METAONT. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-130120>.)

- (a) `isDefinedIn`,
- (b) `isDocumentedIn`.

The first three properties represent the three domain flags. Property 4 describes the number of arguments for an ontological relation. The next properties identify various classes of METAONT. Property 5 uniquely identifies instances of `Individual`, including instances of `Agent`. Properties 6 and 7 refer to an applied ontology itself. The last three properties are responsible for storing the URIs of other information resources represented by a meta-ontological description. We maintain the distinction between the relation denoted by “`isDefinedBy`” and the relation denoted by “`isDocumentedIn`”. The former relates certain elements of the meta-ontological description, namely instances of the `IntendedUsageType`, `Language`, `Methodology`, and `SourceOfGeneralKnowledge` categories, to those informational resources that *describe* (define) those instances. The latter relates other elements that need to be documented, namely instances of the `ActualUsageInstance`, `Ontology`, `SourceOfSpecificKnowledge` categories, to those informational resources that actually *store* them.

Appendix contains the formal outline of METAONT in a DL dialect that was obtained as a slightly modified Latex export from the Protege 4.1 editor.

We should indicate that our schema forces its user to pay attention to the language in which a given ontology is represented (because `isExpressedIn` is functional). Thus, if the developers of this ontology attached the same label (and perhaps even the same version identification) to two or more formalisations, each of which is rendered in a different language, we treat these formalisations as different ontologies.

Using our schema we described ten examples of applied ontologies of different types:

- (1) BFO – as stored in <http://www.ifomis.org/bfo/1.1>
- (2) CIDOC CRM – as stored in [http://www.cidoc-crm.org/official\\_release\\_cidoc.html](http://www.cidoc-crm.org/official_release_cidoc.html)
- (3) DOLCE (Lite) – as stored in <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl>
- (4) Dublin Core Metadata Element Set – as stored in <http://purl.org/dc/elements/1.1/>
- (5) FAO Geopolitical Ontology – as stored in <http://aims.fao.org/aos/geopolitical.owl>
- (6) GFO – as stored in <http://www.onto-med.de/ontologies/gfo-basic.owl>
- (7) ISO 15926 – as stored in [http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2\\_2003](http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003)
- (8) PROTON\_KM – as stored in <http://proton.semanticweb.org/2005/04/protonkm>
- (9) WordNet – as stored in <http://www.w3.org/2006/03/wn/wn20/schema/>
- (10) YAMATO – as documented in [http://www.ei.sanken.osaka-u.ac.jp/hozo/onto\\_library/YAMATO101216.pdf](http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/YAMATO101216.pdf)

We regard them as a proof of concept for our meta-ontological investigation, however, due to the lack of space we could not specify them here in detail, so the interested reader should consult the webpage <http://www.l3g.pl/metaont.html>, where they are all stored as OWL ontologies. We will, however, describe two possible applications thereof.

The first application has to do with comparison of ontological categories across different ontologies – see Table 1. For the sake of this comparison we fix the extension of each category with respect to the actual world at a certain point in time.

The second application concerns the possibility of finding similarities between ontologies that go far beyond the standard alignment mechanisms. Given an appropriate reasoner one can query METAONT\_UNIVERSE with respect to the features of ontologies represented by our meta-ontology. We list a few examples of such queries performed by the DL Query plugin to Protege 4.1 and their results below:

**(query)** `isExpressedIn` value `OWL_DL_RDF/XML`

**(answer)** BFO

**(answer)** ISO\_15926

Table 1  
Comparison of ontological categories

	BFO : Continuant	BFO : Occurent	CIDOC CRM : E2 Temporal Entity	CIDOC CRM : E77 Persistent Item	CIDOC CRM : E55 Type	DOLCE : Abstract	DOLCE : Endurant	DOLCE : Perdurant	DOLCE : Particular	GFO : Occurent	GFO : Individual	ISO 15926 : Possible_Individual	ISO 15926 : Class	PROTON : Abstract	PROTON : Happening	PROTON : Object
BFO : Continuant	///			⌞												
BFO : Occurent		///	⌞					///		⌞		⌞			///	
CIDOC CRM : E2 Temporal Entity		⌞	///					⌞		⌞		⌞			⌞	
CIDOC CRM : E77 Persistent Item	⌞			///			⌞									⌞
CIDOC CRM : E55 Type					///											
DOLCE : Abstract						///								///		
DOLCE : Endurant				⌞			///									///
DOLCE : Perdurant		///	⌞					///		⌞		⌞				
DOLCE : Particular									///		///					
GFO : Occurent		⌞	⌞					⌞		///		⌞				
GFO : Individual									///		///					
ISO 15926 : Possible_Individual		⌞	⌞					⌞		⌞		///			⌞	
ISO 15926 : Class													///			
PROTON : Abstract						///								///		
PROTON : Happening		///	⌞									⌞			///	
PROTON : Object				⌞			///									///

(answer) PROTON\_KM

(answer) FAO\_Geopolitical\_Ontology

(query) isInformedBy value endurantism

(answer) DOLCE

(answer) CIDOC\_CRM

(query) isDevelopedWithin some NewOntologyMethodology

(answer) BFO

(answer) PROTON\_KM

(query) isUsedAs value as\_common\_vocabulary

(answer) CIDOC\_CRM

(answer) Dublin\_Core\_Metadata\_Element\_Set

(answer) FAO\_Geopolitical\_Ontology

(query) hasModalityFlag value “possible”

(answer) DOLCE : domain

(answer) ISO 15926 : domain  
 (answer) PROTON\_KM : domain  
 (answer) WordNet : domain

The last query indirectly shows which ontologies have the domains that include possible entities, i.e., which take into account not only actual, but also possible entities. The direct query would require the OWL 2.0 notion of chain property, which is not available in the DL Query plugin.

## 5. METAONT and OMV – Comparison

Except for the differences we already mentioned above, at first sight the degree of similarity between METAONT and OMV seems to be substantial. Both meta-ontologies represent the domain, language, methodology, and usage for an applied ontology. Nonetheless, these categories are conceptualised differently there:<sup>9</sup>

### (1) domain

- (a) *omv* : *OntologyDomain* is characterised by means of its name, acronym, description and documentation. These attributes are not specific for this category as a number of other categories are characterised thereby, e.g., *omv* : *OntologyTask*. OMV allows for multiple domains for a single ontology;
- (b) *meta* : *Domain* is characterised by means of its name and flags. The flags' attributes are specific for this category. METAONT specifies a unique domain for each ontology, but, at the same time, makes it possible to extensionally compare different domains;

### (2) language

- (a) OMV has two categories that represent ontological languages:  
*omv* : *OntologyLanguage* and *omv* : *OntologySyntax*. Both are characterised by means of their name, acronym, description, and documentation. Additionally, OMV provides also information on the developers of a given language. Distinguishing between ontologies and ontological documents OMV does allow for multiple languages for a single ontology;
- (b) *meta* : *Language* is characterised by means of its name and definition, where the latter may be stored in a number of different documents. Incorporating Morris's view on language METAONT does not distinguish between a language and its syntax. Moreover, it maintains a one-to-one correspondence between an ontology and its language;

### (3) methodology

- (a) From the formal point of view the category *omv* : *OntologyEngineeringMethodology* is identical (i.e., isomorphic) to both *omv* : *OntologyLanguage* and *omv* : *OntologySyntax*, i.e. both have the same relationships to the same elements of OMV, with the exception that one ontology may be developed by means of multiple methodologies;
- (b) Besides providing information on the name and documentation for a methodology, METAONT also classifies nine main types of ontological methodologies;

---

<sup>9</sup>In order to avoid ambiguity we will use two prefixes: "omv" and "meta" to distinguish between the two.

## (4) usage

- (a) From the formal point of view the category `omv : OntologyEngineeringTask` is identical (i.e., isomorphic) to `omv : OntologyEngineeringMethodology`;
- (b) METAONT distinguishes between intended usage types for an ontology and actual usage instances of this ontology.

Secondly, there is a number of OMV elements that do not have any counterparts in METAONT:

## (1) classes:

- (a) `omv : LicenseModel`
- (b) `omv : KnowledgeRepresentationParadigm`
- (c) `omv : FormalityLevel`
- (d) `omv : Location`
- (e) `omv : OntologyEngineeringTool`
- (f) `omv : OntologyType`

(2) all relations between those classes and `omv : Ontology`

## (3) attributes:

(a) for `omv : Ontology`:

- (i) `omv : resourceLocator`
- (ii) `omv : keywords`
- (iii) `omv : creationDate`
- (iv) `omv : modificationDate`
- (v) `omv : naturalLanguage`
- (vi) `omv : numberOfAxioms`

(b) for all other classes that do have their counterparts in METAONT: the acronym `omv : attribute`.

Moreover, METAONT does not make the distinction between individual and collective ontology developers and the distinction between the description and documentation of an attribute.

On the other hand:

- (1) METAONT stores all categories of an ontology instead of simply counting them. This facilitates making (extensional) comparisons between categories from different ontologies.
- (2) METAONT stores also all relations from an ontology while OMV seems to be confined to counting the number of binary relations.
- (3) METAONT represents and classifies sources of knowledge used in ontology development as instances of its (meta-)categories. This “reification” facilitates making queries with respect to the design rationales that drove this development.

## 6. Conclusions and further work

It appears to us that our paper supports the claim that metaphilosophy may provide a number of non-trivial insights for engineering meta-ontology. Compared to the existing meta-ontological standards (e.g., to OMV) our schema exposes (i) the non-extensional aspects of ontological domains and categories,

(ii) the need for a proper specification of the sources of ontological knowledge, and (iii) the difference between the intended and actual application of a given ontology.

The main outstanding problem with our proposal concerns the non-extensionality of ontological domains and categories. Although one can capture this aspect by means of the notion of possible world, this kind of solution is hardly operational. It seems that it is also one of the main obstacles to an effective and adequate method of ontology comparison.

Nevertheless, we believe that further development of this initial proposal may lead to a repository of applied ontologies with an extended list of functionalities as compared to such repositories as BioPortal (<http://bioportal.bioontology.org/>). In particular we have in mind (i) queries related to the conceptual foundations of applied ontologies, (ii) queries on the non-extensional aspects of ontological domains, and finally (iii) comparisons between ontological categories from different ontologies, whose examples we provided in Section 4.

## Acknowledgements

Both authors were supported by the grant N N101 150037 from the Ministry of Science and Higher Education.

## Appendix. Formalisation of METAONT

### *Classes*

#### *ActualUsageInstance*

$\text{ActualUsageInstance} \sqsubseteq \text{Usage}$

$\text{ActualUsageInstance} \sqsubseteq \exists \text{ isDocumentedIn } \text{Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$

#### *Agent*

$\text{Agent} \sqsubseteq \text{Individual}$

$\text{Agent} \sqsubseteq \exists \text{ hasGuid } \text{Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$

#### *ArtsOrRecreation*

$\text{ArtsOrRecreation} \sqsubseteq \text{SourceOfDescriptiveKnowledge}$

### *Category*

$\text{Category} \equiv \text{Relation} \sqcup \text{UnaryCategory}$

$\text{Category} \sqsubseteq \exists \text{ inverse\_of\_ } \text{acknowledges } \text{Ontology}$

$\text{Category} \sqsubseteq \neg \text{SourceOfKnowledge}$

$\text{Category} \sqsubseteq \neg \text{Language}$

$\text{Category} \sqsubseteq \neg \text{Usage}$

$\text{Category} \sqsubseteq \neg \text{Methodology}$

$\text{Category} \sqsubseteq \neg \text{Domain}$

$\text{Category} \sqsubseteq \neg \text{Ontology}$

$\text{DisjointUnion } \text{Relation } \text{UnaryCategory}$

#### *ConstrainedEthnicLanguage*

$\text{ConstrainedEthnicLanguage} \sqsubseteq \text{LanguageWithoutInferenceSupport}$

*Domain*

$\text{Domain} \sqsubseteq \neg \text{Language}$   
 $\text{Domain} \sqsubseteq \neg \text{Methodology}$   
 $\text{Domain} \sqsubseteq \neg \text{SourceOfKnowledge}$   
 $\text{Domain} \sqsubseteq \neg \text{Ontology}$   
 $\text{Domain} \sqsubseteq \neg \text{Category}$   
 $\text{Domain} \sqsubseteq \neg \text{Usage}$

*EvaluationMethodology*

$\text{EvaluationMethodology} \sqsubseteq \text{PostOntologyMethodology}$

*GeneralReference*

$\text{GeneralReference} \sqsubseteq \text{SourceOfDescriptiveKnowledge}$

*HistoryOrGeography*

$\text{HistoryOrGeography} \sqsubseteq \text{SourceOfDescriptiveKnowledge}$

*Individual*

$\text{Individual} \sqsubseteq \exists \text{ fallsUnder } \text{UnaryCategory}$   
 $\text{Individual} \sqsubseteq \text{Thing}$   
 $\text{Individual} \sqsubseteq \exists \text{ belongsTo } \text{Domain}$

*IntendedUsageType*

$\text{IntendedUsageType} \sqsubseteq \text{Usage}$   
 $\text{IntendedUsageType} \sqsubseteq \exists \text{ isDefinedIn } \text{Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$

*Language*

$\text{Language} \equiv \text{LanguageWithInferenceSupport} \sqcup \text{LanguageWithoutInferenceSupport}$   
 $\text{Language} \sqsubseteq \exists \text{ isDefinedIn } \text{Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$   
 $\text{Language} \sqsubseteq \neg \text{Domain}$   
 $\text{Language} \sqsubseteq \neg \text{Category}$   
 $\text{Language} \sqsubseteq \neg \text{Methodology}$   
 $\text{Language} \sqsubseteq \neg \text{Ontology}$   
 $\text{Language} \sqsubseteq \neg \text{SourceOfKnowledge}$   
 $\text{Language} \sqsubseteq \neg \text{Usage}$   
 $\text{DisjointUnion } \text{LanguageWithInferenceSupport } \text{LanguageWithoutInferenceSupport}$

*LanguageWithInferenceSupport*

$\text{LanguageWithInferenceSupport} \sqsubseteq \text{Language}$

*LanguageWithoutInferenceSupport*

$\text{LanguageWithoutInferenceSupport} \sqsubseteq \text{Language}$

*LingusticKnowledge*

$\text{LingusticKnowledge} \sqsubseteq \text{SourceOfDescriptiveKnowledge}$

*Literature*

$\text{Literature} \sqsubseteq \text{SourceOfDescriptiveKnowledge}$

*Methodology*

Methodology  $\sqsubseteq$  NewOntologyMethodology  $\sqcup$  PostOntologyMethodology  
 Methodology  $\sqsubseteq \exists$  isDefinedIn Datatypehttp://www.w3.org/2001/XMLSchema#anyURI  
 Methodology  $\sqsubseteq \neg$  Domain  
 Methodology  $\sqsubseteq \neg$  Language  
 Methodology  $\sqsubseteq \neg$  SourceOfKnowledge  
 Methodology  $\sqsubseteq \neg$  Category  
 Methodology  $\sqsubseteq \neg$  Usage

*NewOntologyMethodology*

NewOntologyMethodology  $\sqsubseteq$  OntologyDevelopmentFromNonOntologicalResourcesMethodology  $\sqcup$   
 $\sqcup$  OntologyDevelopmentFromScratchMethodology  
 NewOntologyMethodology  $\sqsubseteq$  Methodology

*Ontology*

Ontology  $\sqsubseteq$  Thing  
 Ontology  $\sqsubseteq =$  isExpressedIn Language  
 Ontology  $\sqsubseteq \exists$  isDocumentedIn Datatypehttp://www.w3.org/2001/XMLSchema#anyURI  
 Ontology  $\sqsubseteq \exists$  represents Domain  
 Ontology  $\sqsubseteq \exists$  isDevelopedBy Agent  
 Ontology  $\sqsubseteq \neg$  Usage  
 Ontology  $\sqsubseteq \neg$  SourceOfKnowledge  
 Ontology  $\sqsubseteq \neg$  Domain  
 Ontology  $\sqsubseteq \neg$  Language  
 Ontology  $\sqsubseteq \neg$  Category

*OntologyDevelopmentFromNonOntologicalResourcesMethodology*

OntologyDevelopmentFromNonOntologicalResourcesMethodology  $\sqsubseteq$  NewOntologyMethodology  
 DisjointUnion OntologyDevelopmentFromNonStructuredResourcesMethodology  
 OntologyDevelopmentFromStructuredResourcesMethodology

*OntologyDevelopmentFromNonStructuredResourcesMethodology*

OntologyDevelopmentFromNonStructuredResourcesMethodology  $\sqsubseteq$   
 $\sqsubseteq$  OntologyDevelopmentFromNonOntologicalResourcesMethodology

*OntologyDevelopmentFromScratchMethodology*

OntologyDevelopmentFromScratchMethodology  $\sqsubseteq$  NewOntologyMethodology

*OntologyDevelopmentFromStructuredResourcesMethodology*

OntologyDevelopmentFromStructuredResourcesMethodology  $\sqsubseteq$   
 $\sqsubseteq$  OntologyDevelopmentFromNonOntologicalResourcesMethodology

*OntologyMappingMethodology*

OntologyMappingMethodology  $\sqsubseteq$  PostOntologyMethodology

*OntologyModificationMethodology*

OntologyModificationMethodology  $\sqsubseteq$  PostOntologyMethodology

*PhilosophyOrPsychology*

PhilosophyOrPsychology  $\sqsubseteq$  SourceOfDescriptiveKnowledge



*PostOntologyMethodology*

PostOntologyMethodology  $\sqsubseteq$  EvaluationMethodology  $\sqcup$  OntologyMappingMethodology  $\sqcup$   
 $\sqcup$  OntologyModificationMethodology  
 PostOntologyMethodology  $\sqsubseteq$  Methodology

*Relation*

Relation  $\sqsubseteq \exists$  hasArity DatatypeRestrictionDatatypehttp://www.w3.org/2001/XMLSchema#int  
 FacetminInclusive"2"^^http://www.w3.org/2001/XMLSchema#integer  
 Relation  $\sqsubseteq$  Category

*Religion*

Religion  $\sqsubseteq$  SourceOfDescriptiveKnowledge

*Science*

Science  $\sqsubseteq$  SourceOfDescriptiveKnowledge

*SocialSciences*

SocialSciences  $\sqsubseteq$  SourceOfDescriptiveKnowledge

*SourceOfDescriptiveKnowledge*

SourceOfDescriptiveKnowledge  $\sqsubseteq$  SourceOfKnowledge  
 SourceOfDescriptiveKnowledge  $\sqsubseteq \exists$  isDefinedIn Datatypehttp://www.w3.org/2001/XMLSchema#  
 anyURI  
 DisjointUnion ArtsOrRecreation GeneralReference HistoryOrGeography LinguisticKnowledge  
 Literature PhilosophyOrPsychology Religion Science SocialSciences Technology

*SourceOfKnowledge*

SourceOfKnowledge  $\equiv$  SourceOfDescriptiveKnowledge  $\sqcup$  SourceOfNormativeKnowledge  
 SourceOfKnowledge  $\sqsubseteq \exists$  isDescribedIn Datatypehttp://www.w3.org/2001/XMLSchema#anyURI  
 SourceOfKnowledge  $\sqsubseteq \neg$  Category  
 SourceOfKnowledge  $\sqsubseteq \neg$  Ontology  
 SourceOfKnowledge  $\sqsubseteq \neg$  Domain  
 SourceOfKnowledge  $\sqsubseteq \neg$  Methodology  
 SourceOfKnowledge  $\sqsubseteq \neg$  Language  
 SourceOfKnowledge  $\sqsubseteq \neg$  Usage  
 DisjointUnion SourceOfDescriptiveKnowledge SourceOfNormativeKnowledge

*SourceOfNormativeKnowledge*

SourceOfNormativeKnowledge  $\sqsubseteq$  SourceOfKnowledge  
 SourceOfNormativeKnowledge  $\sqsubseteq \exists$  isDocumentedIn Datatypehttp://www.w3.org/2001/  
 XMLSchema#anyURI

*Technology*

Technology  $\sqsubseteq$  SourceOfDescriptiveKnowledge

*Thing**UnaryCategory*

UnaryCategory  $\sqsubseteq \exists$  inverse\_of\_fallsUnder Individual  
 UnaryCategory  $\sqsubseteq$  Category

*Usage*

$\text{Usage} \equiv \text{ActualUsageInstance} \sqcup \text{IntendedUsageType}$   
 $\text{Usage} \sqsubseteq \exists \text{ isDescribedIn Datatype} \text{http://www.w3.org/2001/XMLSchema\#anyURI}$   
 $\text{Usage} \sqsubseteq \neg \text{Ontology}$   
 $\text{Usage} \sqsubseteq \neg \text{Category}$   
 $\text{Usage} \sqsubseteq \neg \text{Domain}$   
 $\text{Usage} \sqsubseteq \neg \text{SourceOfKnowledge}$   
 $\text{Usage} \sqsubseteq \neg \text{Methodology}$   
 $\text{Usage} \sqsubseteq \neg \text{Language}$   
 $\text{DisjointUnion ActualUsageInstance IntendedUsageType}$

*Object properties**acknowledges*

$\sqsubseteq \text{topObjectProperty}$   
 $\langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#acknowledges} \rangle \equiv$   
 $\equiv \langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#inverse\_of\_acknowledges} \rangle^-$   
 $\exists \text{ acknowledges Thing} \sqsubseteq \text{Ontology}$   
 $\top \sqsubseteq \forall \text{ acknowledges Category}$   
 $\text{DisjointObjectProperties acknowledges rejects}$

*belongsTo*

$\sqsubseteq \text{topObjectProperty}$   
 $\exists \text{ belongsTo Thing} \sqsubseteq \text{Individual}$   
 $\top \sqsubseteq \forall \text{ belongsTo Domain}$

*fallsUnder*

$\langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#fallsUnder} \rangle \equiv$   
 $\equiv \langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#inverse\_of\_fallsUnder} \rangle^-$   
 $\exists \text{ fallsUnder Thing} \sqsubseteq \text{Individual}$   
 $\top \sqsubseteq \forall \text{ fallsUnder UnaryCategory}$

*hasRoot*

$\sqsubseteq \text{acknowledges}$   
 $\top \sqsubseteq \leq 1 \text{ hasRoot Thing}$   
 $\exists \text{ hasRoot Thing} \sqsubseteq \text{Ontology}$   
 $\top \sqsubseteq \forall \text{ hasRoot UnaryCategory}$

*inverse\_of\_acknowledges*

$\langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#acknowledges} \rangle \equiv$   
 $\equiv \langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#inverse\_of\_acknowledges} \rangle^-$   
 $\exists \text{ inverse\_of\_acknowledges Thing} \sqsubseteq \text{Category}$   
 $\top \sqsubseteq \forall \text{ inverse\_of\_acknowledges Ontology}$

*inverse\_of\_fallsUnder*

$\langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#fallsUnder} \rangle \equiv$   
 $\equiv \langle \text{http://l3g.pl/ontologies/Metaontology/Version\_2.0/Metaontology.owl\#inverse\_of\_fallsUnder} \rangle^-$   
 $\exists \text{ inverse\_of\_fallsUnder Thing} \sqsubseteq \text{UnaryCategory}$   
 $\top \sqsubseteq \forall \text{ inverse\_of\_fallsUnder Individual}$

*isDevelopedBy* $\exists \text{ isDevelopedBy Thing} \sqsubseteq \text{Ontology}$  $\top \sqsubseteq \forall \text{ isDevelopedBy Agent}$ *isDevelopedWithin* $\exists \text{ isDevelopedWithin Thing} \sqsubseteq \text{Ontology}$  $\top \sqsubseteq \forall \text{ isDevelopedWithin Methodology}$ *isExplicitlyInformedBy* $\sqsubseteq \text{isInformedBy}$  $\exists \text{ isExplicitlyInformedBy Thing} \sqsubseteq \text{Category} \sqcup \text{Ontology}$  $\top \sqsubseteq \forall \text{ isExplicitlyInformedBy (Ontology} \sqcup \text{SourceOfKnowledge)}$ 

DisjointObjectProperties isExplicitlyInformedBy isImplicitlyInformedBy

*isExpressedIn* $\top \sqsubseteq \leq 1 \text{ isExpressedIn Thing}$  $\exists \text{ isExpressedIn Thing} \sqsubseteq \text{Ontology}$  $\top \sqsubseteq \forall \text{ isExpressedIn Language}$ *isExtensionallyEquivalentTo* $\sqsubseteq \text{isExtensionallyIncludedIn}$  $\text{isExtensionallyEquivalentTo} \equiv \text{isExtensionallyEquivalentTo}^-$ 

TransitivePropertyisExtensionallyEquivalentTo

 $\exists \text{ isExtensionallyEquivalentTo Thing} \sqsubseteq \text{Category} \sqcup \text{Domain}$  $\top \sqsubseteq \forall \text{ isExtensionallyEquivalentTo (Category} \sqcup \text{Domain)}$ *isExtensionallyIncludedIn*

TransitivePropertyisExtensionallyIncludedIn

 $\exists \text{ isExtensionallyIncludedIn Thing} \sqsubseteq \text{Category} \sqcup \text{Domain}$  $\top \sqsubseteq \forall \text{ isExtensionallyIncludedIn (Category} \sqcup \text{Domain)}$ *isImplicitlyInformedBy* $\sqsubseteq \text{isInformedBy}$  $\exists \text{ isImplicitlyInformedBy Thing} \sqsubseteq \text{Category} \sqcup \text{Ontology}$  $\top \sqsubseteq \forall \text{ isImplicitlyInformedBy (Ontology} \sqcup \text{SourceOfKnowledge)}$ 

DisjointObjectProperties isExplicitlyInformedBy isImplicitlyInformedBy

*isInformedBy* $\exists \text{ isInformedBy Thing} \sqsubseteq \text{Category} \sqcup \text{Ontology}$  $\top \sqsubseteq \forall \text{ isInformedBy (Ontology} \sqcup \text{SourceOfKnowledge)}$ *isSyntacticVariantOf* $\text{isSyntacticVariantOf} \equiv \text{isSyntacticVariantOf}^-$ 

TransitivePropertyisSyntacticVariantOf

 $\exists \text{ isSyntacticVariantOf Thing} \sqsubseteq \text{Language}$  $\top \sqsubseteq \forall \text{ isSyntacticVariantOf Language}$ *isUsedAs* $\exists \text{ isUsedAs Thing} \sqsubseteq \text{Ontology}$  $\top \sqsubseteq \forall \text{ isUsedAs Usage}$

*rejects*

$\exists \text{ rejects Thing} \sqsubseteq \text{Ontology}$   
 $\top \sqsubseteq \forall \text{ rejects Category}$   
 $\text{DisjointObjectProperties acknowledges rejects}$

*represents*

$\top \sqsubseteq \leq 1 \text{ represents Thing}$   
 $\exists \text{ represents Thing} \sqsubseteq \text{Ontology}$   
 $\top \sqsubseteq \forall \text{ represents Domain}$

*topObjectProperty**Data properties**hasArity*

$\top \sqsubseteq \leq 1 \text{ hasArity}$   
 $\exists \text{ hasArity Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Relation}$   
 $\top \sqsubseteq \forall \text{ hasArity Datatypehttp://www.w3.org/2001/XMLSchema\#int}$

*hasGuid*

$\top \sqsubseteq \leq 1 \text{ hasGuid}$   
 $\exists \text{ hasGuid Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Individual}$   
 $\top \sqsubseteq \forall \text{ hasGuid Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$

*hasModalityFlag*

$\top \sqsubseteq \leq 1 \text{ hasModalityFlag}$   
 $\exists \text{ hasModalityFlag Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Domain}$   
 $\top \sqsubseteq \forall \text{ hasModalityFlag } \{ \text{"actual"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \} \sqcup$   
 $\sqcup \{ \text{"possible"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \}$

*hasObjectivityFlag*

$\top \sqsubseteq \leq 1 \text{ hasObjectivityFlag}$   
 $\exists \text{ hasObjectivityFlag Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Domain}$   
 $\top \sqsubseteq \forall \text{ hasObjectivityFlag } \{ \text{"mind-dependent"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \} \sqcup$   
 $\sqcup \{ \text{"mind-independent"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \} \sqcup$   
 $\sqcup \{ \text{"mixed"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \}$

*hasPrefix*

$\top \sqsubseteq \leq 1 \text{ hasPrefix}$   
 $\exists \text{ hasPrefix Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Ontology}$   
 $\top \sqsubseteq \forall \text{ hasPrefix Datatypehttp://www.w3.org/2001/XMLSchema\#string}$

*hasScopeFlag*

$\sqsubseteq \text{topDataProperty}$   
 $\top \sqsubseteq \leq 1 \text{ hasScopeFlag}$   
 $\exists \text{ hasScopeFlag Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Domain}$   
 $\top \sqsubseteq \forall \text{ hasScopeFlag } \{ \text{"global"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \} \sqcup$   
 $\sqcup \{ \text{"local"}^{\sim} \text{http://www.w3.org/2001/XMLSchema\#string} \}$

*hasVersionID*
 $\top \sqsubseteq \leq 1 \text{ hasVersionID}$ 
 $\exists \text{ hasVersionID Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Ontology}$ 
 $\top \sqsubseteq \forall \text{ hasVersionID Datatypehttp://www.w3.org/2001/XMLSchema\#string}$ 
*isDefinedIn*
 $\sqsubseteq \text{isDescribedIn}$ 
 $\exists \text{ isDefinedIn Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{IntendedUsageType} \sqcup$ 
 $\sqcup \text{Language} \sqcup \text{Methodology} \sqcup \text{SourceOfDescriptiveKnowledge}$ 
 $\top \sqsubseteq \forall \text{ isDefinedIn Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$ 
 $\text{isDefinedIn} \not\equiv \text{isDocumentedIn}$ 
*isDescribedIn*
 $\exists \text{ isDescribedIn Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq \text{Language} \sqcup \text{Ontology} \sqcup$ 
 $\sqcup \text{SourceOfKnowledge} \sqcup \text{Usage}$ 
 $\top \sqsubseteq \forall \text{ isDescribedIn Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$ 
*isDocumentedIn*
 $\sqsubseteq \text{isDescribedIn}$ 
 $\exists \text{ isDocumentedIn Datatypehttp://www.w3.org/2000/01/rdf-schema\#Literal} \sqsubseteq$ 
 $\sqsubseteq \text{ActualUsageInstance} \sqcup \text{Ontology} \sqcup \text{SourceOfNormativeKnowledge}$ 
 $\top \sqsubseteq \forall \text{ isDocumentedIn Datatypehttp://www.w3.org/2001/XMLSchema\#anyURI}$ 
 $\text{isDefinedIn} \not\equiv \text{isDocumentedIn}$ 
**References**

- Batres, R., West, M., Price, D., Shimada, K.M., Tetsuo, F.Y.F. & Naka, Y. (2007). An upper ontology based on ISO 15926. *Computers and Chemical Engineering*, 31, 519–534.
- BonJour, L. (1998). *In Defense of Pure Reason*. Cambridge: Cambridge University Press.
- Borst, W. (1997). Construction of engineering ontologies for knowledge sharing and reuse. PhD thesis. University of Twente.
- Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. New York: Springer.
- Dewey, M. (1876). *A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library*. Amherst: Amherst.
- Ding, L. et al. (2004). Swoogle: A search and metadata engine for the semantic web. In: *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management* (pp. 58–61). ACM.
- Farquar, A., Fikes, R. & Rice, J. (1997). The Ontolingua Server: a tool for collaborative ontology construction. *International Journal of Human–Computer Studies*, 46, 707–727.
- Fernandez, M., Gomez-Perez, A. & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering* (pp. 33–40).
- Fernández-López, M., Gómez-Pérez, A. & Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. In *Ontological Engineering on Spring Symposium Series*. Stanford.
- Gangemi, A., Steve, G. & Giacomelli, F. (1996). ONIONS: An ontological methodology for taxonomic knowledge integration. In *ECAI-96 Workshop on Ontological Engineering*.
- Garbacz, P. & Trypuz, R. (2011). A metaontology for ontological engineering: A philosophers' perspective. In V. Dignum and P. Vermaas (Eds.), *Formal Ontologies Meet Industry*. Frontiers in Artificial Intelligence and Applications (Vol. 229, pp. 16–27).
- Gómez-Pérez, A., Motta, E., Suárez-Figueroa, M.C., d'Aquin, M., Gangemi, A., Haase, P., Fernández-López, M., Lewen, H., Villazón-Terrazas, B., Presutti, V., Montiel-Ponsoda, E., Espinoza, M., Sabou, M., Zablith, F., Euzenat, J., Duc, C.L., Erdmann, M., Waterfeld, W., Harth, A., Lewen, H., Caracciolo, C., Sini, M., Herrero, G. & Pariente-Lobo, T. (2011). *NeOn Book, NeOn Methodology in a Nutshell*. [http://www.neon-project.org/nw/NeOn\\_Book](http://www.neon-project.org/nw/NeOn_Book).

- Grenon, P. (2003). BFO in a Nutshell: A Bi-categorical axiomatization of BFO and comparison with DOLCE. Tech. Rep. 6, Institut für Formale Ontologie und Medizinische Informationswissenschaft.
- Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220.
- Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human and Computer Studies*, 46, 293–310.
- Guarino, N. (1998). Formal ontology in information systems. In: I.P. Amsterdam (Ed.), *Proceedings of FOIS'98*, Trento, Italy, 6–8 June (pp. 3–15).
- Guarino, N. & Giarretta, P. (1995). *Ontologies and Knowledge Bases: Towards a Terminological Clarification* (pp. 25–32). IOS Press.
- Guarino, N. & Welty, C.A. (2009). An overview of OntoClean. In S. Staab and R. Studer (Eds.), *Handbook on Ontologies* (pp. 201–220). Dordrecht: Springer-Verlag.
- Guarino, N., Carrara, M. & Giarretta, P. (1994). Formalizing ontological commitment. In M. Kaufmann (Ed.), *National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA (pp. 560–567).
- Guarino, N., Oberle, D. & Staab, S. (2009). What is an ontology? In S. Staab and R. Studer (Eds.), *Handbook on Ontologies* (pp. 1–17). New York: Springer.
- Hartmann, J., Palma, R., Sure, Y., Suarez-Figueroa, M.C., Haase, P., Gomez-Perez, A. & Studer, R. (2005). Ontology metadata vocabulary and applications. In R. Meersman et al. (Ed.), *OTM Workshops 2005*, LNCS (Vol. 3762, pp. 906–915). Dordrecht: Springer-Verlag.
- Klima, G. (2008). The medieval problem of universals. In E.N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*, Winter 2008 edn.
- Masolo, C., Borgo, S., Gangemini, A., Guarino, N. & Oltramari, A. (2003). The WonderWeb Library of Foundational Ontologies and the DOLCE ontology. WonderWeb Deliverable D18, Final Report. Tech. Rep. vr. 1.0. 31-12-2003, Laboratory For Applied Ontology – ISTC-CNR.
- McCarthy, J. (1980). Circumscription – a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1,2), 27–39.
- McKinnon, N. (2002). The endurance/perdurance distinction, *Australasian Journal of Philosophy*, 80(3), 288–306.
- Merrill, G.H. (2010). Ontological realism: Methodology or misdirection? *Applied Ontology*, 5, 79–108.
- Mizoguchi, R. (2003). Tutorial on ontological engineering. Part 1: Introduction to ontological engineering. *New Generation Computing*, 21, 365–384.
- Morris, C.W. (1938). *Foundations of the Theory of Signs*. Chicago: Chicago University Press.
- Palma, R., Hartmann, J. & Haase, P. (2008). OMV Ontology Metadata Vocabulary for the SemanticWeb. Final 2.4, OMV Consortium, <http://sourceforge.net/projects/omv2/>.
- Peirce, C.S. (1931). *Collected Papers of Charles Sanders Peirce*. Harvard: Harvard University Press.
- Pinto, H.S., Tempich, C. & Staab, S. (2009). Ontology Engineering and Evolution in a Distributed World Using DILIGENT. In S. Staab and R. Studer (Eds.), *Handbook on Ontologies* (pp. 153–176). Dordrecht: Springer-Verlag.
- Searle, J.R. & Vanderveken, D. (1985). *Foundations of Illocutionary Logic*. Cambridge: Cambridge University Press.
- Smith, B. (1987). On the cognition of states of affairs. In *Speech Act and Sachverhalt: Reinach and the Foundations of Realist Phenomenology* (pp. 189–225). Dordrecht/Boston/Lancaster: Nijhoff.
- Smith, B. & Ceusters, W. (2010). Ontological realism: A methodology for coordinated evolution of scientific ontologies. *Applied Ontology*, 5, 139–188.
- Smith, D.W. (2007). *Husserl*. London and New York: Routledge.
- Staab, S., Studer, R., Schnurr, H.-P. & Sure, Y. (2001). Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16, 26–34.
- Stepien, A.B. (1993). Ontologia: typy i statut poznawczy. Ontologia a metafizyka. In A.B. Stepien and T. Szubka (Eds.), *Studia metafizyczne* (Vol. 1, pp. 9–37). TN KUL, Lublin.
- Studer, R., Benjamins, V. & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1,2), 161–197.
- Stumme, G. & Maedche, A. (2001). FCA-Merge: Bottom-up merging of ontologies. In *IJCAI-2001 – Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Seattle, WA, USA, August, 1–6, 2001 (pp. 225–234). San Francisco: Morgan Kaufmann.
- Terziev, I., Kiryakov, A. & Manov, D. (2005). D1.8.1 Base upper-level ontology (BULO) Guidance. Eu-ist project ist-2003-506826 sekt. SEKT: Semantically Enabled Knowledge Technologies.
- Thomasson, A.L. (1999). *Fiction and Metaphysics*. Cambridge: Cambridge University Press.
- Uschold, M. & Grüninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2), 93–155.
- Ziff, P. (1960). *Semantic Analysis*. Ithaca, NY: Cornell University Press.