

Modeling Life Science Knowledge with OWL 1.1

Michel Dumontier^{1,2,3}, Natalia Villanueva-Rosales¹,

¹ School of Computer Science, ²Department of Biology, ³Institute of Biochemistry,
Carleton University, 1125 Colonel By Drive,

K1S 5B6, Ottawa, Canada

michel_dumontier@carleton.ca, nvillanu@scs.carleton.ca

Abstract. The OWL 1.1 specification has created new opportunities for the design of increasingly expressive and useful ontologies in the modeling of life science knowledge. Here, we describe the application of expressive features in the design of an ontology of basic relations and how, in combination with an upper level ontology, they can be used to guide the formulation of life science knowledge. We report on our experiences to enhance existing ontologies so as to facilitate knowledge representation and question answering. Finally, we identify some outstanding challenges towards building an ontology-based semantic web.

Keywords: semantic web, knowledge representation, knowledge engineering, ontology, life sciences, question answering, OWL-DL, OWL 1.1.

1 Introduction

The life sciences aim to increase our understanding of living things by investigating the structure, function, growth and evolution of living things. The subject matter is complex, rife with exceptions and compounded by the use of domain-specific terminology. In this respect, ontologies can help provide clarity of terminology by describing the semantic relationship between terms, such as describing those terms that are broader or narrower in their meaning. Furthermore, ontologies can be used to place our knowledge in a consistent and structured environment, opening new possibilities for machine interpretation. A key feature of logic-based ontologies is that fairly sophisticated inferences may be derived through automated reasoners, thus helping people discover knowledge that was not immediately apparent. Moreover, non-experts can make use of the granularity of terms to make meaningful queries suited to their level of interest and expertise.

OWL, the Web Ontology Language [1], is a core knowledge representation language for designing expressive ontologies for the Semantic Web [2]. OWL-DL, a variant that is based on a family of description logics (DL), facilitates the description of complex concepts from simpler ones with an emphasis on decidability of reasoning tasks [3]. We have successfully designed OWL-DL ontologies for a range of applications including the discovery of chemical functional groups from molecular structure [4], the design of an online yeast knowledge base [5, 6], the representation and querying of statistical graphs [7] and most recently, in question answering on the pharmacogenomics of depression [8].

The recently proposed OWL 1.1 specification [9] provides new expressive features such as qualified cardinality restrictions and role chains that not only simplify the representation of complex scientific knowledge in a more natural manner, but also facilitates the discovery of knowledge obtained from complex reasoning. Here, we describe how OWL 1.1 features have been useful towards these goals.

2 Basic Relation Ontology (BRO)

Judicious use of a minimal set of re-usable relations is essential in building consistent, well-formed and sophisticated knowledge bases that span multiple domains. Building on efforts devoted to building a common relation ontology [10], we focus on compatibility with OWL-DL, including unary/binary predicates restrictions. We have designed an ontology of relations, termed BRO, that provides 50 object properties and 3 datatype relations towards the assertion of hierarchical, mereological, participatory, spatial and temporal relationships. The ontology itself is composed of three separate documents, i) bro-primitive¹ where the relations are labeled (rdfs:label), defined (rdfs:comment), hierarchically organized with asserted inverse relations, ii) bro² which applies OWL 1.0 property characteristics (i.e. symmetric, transitive) and iii) bro-owl11³ which adds applicable OWL 1.1 property characteristics (reflexive, irreflexive, asymmetric, disjoint roles, role chains). **Table 1** illustrates a slice of the BRO where OWL 1.0 and OWL 1.1 property characteristics have been applied.

Table 1. Part of the Basic Relation Ontology and use of OWL 1.0 and OWL 1.1 property characteristic axioms

BRO Relation	Property Characteristic						
	S ¹	T ¹	R ²	IRR ²	AS ²	Anti ³	Top ³
isRelatedTo	x						[x]
hasPart		x					
hasProperPart		x		(x)	(x)	[x]	
hasIntegralPart		x	x			[x]	
hasImproperPart		x	x				
hasParticipant							

Table key: T = Transitive; R = Reflexive; IRR = Irreflexive; S = Symmetric; AS = Asymmetric; Anti = Antisymmetric; Top = Top role; x – in use; (x) problems, [x] feature unavailable.

Support for characteristic in ¹OWL 1.0; ²OWL 1.1; ³OWL Candidate?

We advocate the addition of a **syntactic Top Role to facilitate ontology mapping and knowledge discovery**. In the BRO ontology, *isRelatedTo* is a symmetric top level property that allows all imported object properties to be mapped as sub-properties and doing so further enables trivial queries over all relations: i) all objects related to a given entity can be discovered (what is related to X?), ii) between entities

¹ <http://ontology.dumontierlab.com/bro-primitive>

² <http://ontology.dumontierlab.com/bro>

³ <http://ontology.dumontierlab.com/bro-owl11>

(i.e. *is X directly related to Y?*) and iii) indirect relations by asserting the top level role as transitive. An equivalent Top Role offered in future OWL specifications could provide new opportunities for knowledge discovery and simplify query formulation over generalized relations (as we do with classes using owl:Thing).

New expressive OWL 1.1 property characteristics are valuable in **constraining mereological relations**, but we found that they are neither sufficient nor could be applied as expected. The BRO mereological relations (*hasPart*, *hasProperPart*, *hasIntegralPart*, *hasImproperPart*) and their inverses with expected attributes are detailed in **Table 1**. Briefly, *hasPart* is generally considered to be transitive, reflexive (a relation *R* on set *X* is **reflexive** when for all *a* in *X*, *a* is *R*-related to itself) and antisymmetric (a relation *R* on a set *X* is **antisymmetric** if, for all *a* and *b* in *X*, if *a* is related to *b* and *b* is related to *a*, then *a* = *b*). Antisymmetric characteristic is also required for *hasPart* sub-properties *hasProperPart* and *hasIntegralPart*. Unfortunately, OWL 1.1 does not specify antisymmetry. It does specify asymmetry (a relation *R* on a set *X* is **asymmetric** if, for all *a* and *b* in *X*, if *a* is related to *b*, then *b* is not related to *a*). Effectively, asymmetry implies that a relation is antisymmetric and irreflexive (a relation *R* on set *X* is **irreflexive** if, for all *a* in *X*, *a* is never *R*-related to itself). Hence, we can specify *hasProperPart* as a transitive and asymmetric. However, by applying these characteristics, we obtained a reasoner error “Nonstructural Restrictions on Axioms” using Protégé 4 (build 60) and the FaCT++ (Version 1.1.10) reasoner. The lack of either explanations (reasoner or otherwise⁴) makes debugging this situation overly challenging.

We have found OWL 1.1 **chain inclusion axioms** extremely useful in both knowledge representation and in question answering [7, 8]. In particular, we have specified the following role chain: *hasPart* o *hasParticipant* → *hasParticipant*. This allows us to discover that participants of process part are also the participants of the process whole. This role chain allows us to break apart processes into several parts, each with their participants, but be able to query the whole process to obtain all participants, and, by inverse, all processes that involve a participant [8].

3 Upper Level Entity Management

Upper level ontologies promise increased semantic coherency by helping to identify the basic type of domain entities and imposing restrictions on the relationships that these entities may hold. We use the Basic Formal Ontology (BFO) as a simplified framework to distinguish qualities, roles, objects and processes from purely spatial or temporal entities [11]. To guide proper use of relations, we designed the NULO ontology⁵ by semantically constraining the relations by i) assigning the domain and range to BFO entities and ii) adding additional restrictions in the form of necessary conditions. For instance, the domain and range values of *hasParticipant* are respectively set to the disjoint bfo:Occurrent and bfo:Continuant. Hence, if it is later determined that the domain or range is not a member of occurrent or continuant, an inconsistency will emerge. Second, while *hasPart* can be held by any entity, it is

⁴ <http://www.w3.org/2007/OWL/wiki/Syntax>

⁵ <http://ontology.dumontierlab.com/nulo>

strictly required that occurrents have occurrent parts and continuants have continuant parts, defined as universal restrictions in NULO-constraints⁶. Hence, not only does this ensure correct usage, but also creates new opportunities for knowledge discovery (e.g. that the part of a spatial region is also a spatial region). As we found universal restrictions to be computationally expensive, placing them in a separate ontology allows the import of these restrictions for consistency checking during the construction of a knowledge base, but may be omitted for query answering purposes after consistency checking.

OWL 1.1 disjoint properties (a relation *R* on a set *X* and *S* on a set *Y* are **disjoint** if, for all *a* and *b* in *X*, and for all *c* and *d* in *Y*, then *a* != *c* and *b* != *d*), are useful in ensuring that fundamentally different properties cannot be used over the same entities. Since *hasParticipant* is a role between an occurrent and continuant, and *hasPart* can only be used between occurrents or continuants, we should assert that *hasPart* and *hasParticipant* are disjoint. However, when adding this restrictions in our testing environment (Protégé 4.0 build 60 and FACT ++), the transitive characteristic of *hasPart* causes an inconsistency to arise from “nonstructural restrictions”.

4 Experience and Design of OWL 1.1 Ontologies

In this section, we describe our experiences in the design of ontologies that make use some of the new OWL 1.1 features.

4.1 Atom Ontology

The Atom Ontology⁷ is composed of 118 atom types and provides the basic building blocks for describing molecular composition and chemical reaction in chemistry and biochemistry. Atom types are determined by atomic number which is a count of the number of protons in the nucleus. Hence, we may describe the necessary and sufficient conditions of class membership by specifying the atomic number⁸ or number of protons⁹. Specifying the number of protons requires **qualified cardinality restrictions** i.e. CarbonAtom *hasPart* exactly 6 Proton. We can represent atomic number in one of two ways. The first is to use place a value restriction on a specific datatype property (i.e. *hasAtomicNumber*), with the caveat that this may lead to a proliferation of domain specific datatype properties. A second, more general approach that we favor, is to represent the atomic number as a type of measurement, represented as a class, whose value is captured using a generic *hasValue* datatype property. In this way, every quality, whether a count or descriptor, is an ontological entity that can be described and reasoned about. We may specify that every atom has exactly 1 atomic number, and for each specific atom type, we can assert its value e.g.

⁶ <http://ontology.dumontierlab.com/nulo-constraints>

⁷ <http://ontology.dumontierlab.com/atom-primitive>

⁸ <http://ontology.dumontierlab.com/atom-complex-atomic-number>

⁹ <http://ontology.dumontierlab.com/atom-complex-proton>

the definition of CarbonAtom in Manchester Syntax is: Atom that *hasQuality* exactly 1 (AtomicNumber that *hasValue* value 6).

An essential aspect of atoms is that each one is different from all other types. The assertion of the pairwise disjoints in OWL 1.0 leads to a very large RDF/XML representation, but OWL 1.1's **disjoint union** significantly reduces the size of XML/RDF representation¹⁰. The disjunction is essential for defining more complex structures that depend on specific atom types. For instance, certain chemical functional groups [4] must be specified by both the presence and absence of particular atom types (i.e. a tertiary amine is a nitrogen group that does not have any hydrogen atoms).

4.2 Modeling the Pharmacogenomics of Depression

Pharmacogenomics aims to better understand the pharmacological response of a drug with respect to genetic variation. We have designed a Pharmacogenomics Ontology¹¹ and applied it in capturing and querying knowledge on the pharmacogenomics of depression [8]. A key aspect of the knowledge representation involves the *hasPart* o *hasParticipant* → *hasParticipant* **chain inclusion axiom**. Our representation breaks apart a complicated process (DrugTreatment) into several sub-processes (DrugGeneInteraction, DrugInducedSideEffect) which are directly linked to their participants (e.g. drug, gene, side effect). Through the role chain we can then query all drug treatments that involve certain genes or side effects as their participants will be inferred (see Fig. 1).

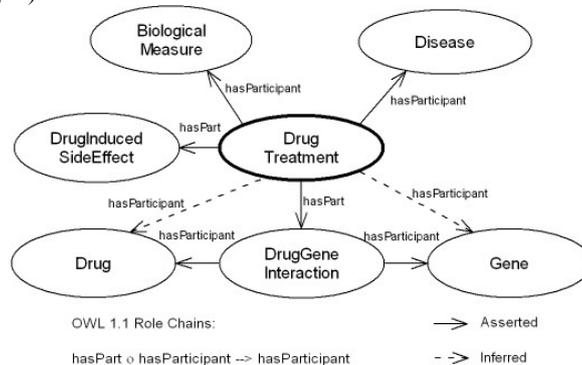


Fig. 1. Main concepts and their relations in the Pharmacogenomics Ontology.

Measurement values, clinical or otherwise, are reused from the Biological Measure Ontology¹², which contains 123 qualitative and quantitative biological measures (e.g. BloodPressure, GeneExpressionMeasure and BodyMassIndex) and can be specified along with the necessary units. Every measurement value may be reported with the *hasValue* datatype property, and units can be further specified with *hasUnit* to an

¹⁰ <http://ontology.dumontierlab.com/atom-complex-disjoint>

¹¹ <http://ontology.dumontierlab.com/pharmacogenomics-complex>

¹² <http://ontology.dumontierlab.com/biological-measure-primitive>

instance of the Unit Ontology¹³. New object properties were added to describe the more specific relations required by (but not restricted to) this domain. For instance, *isVariantOf* (symmetric, transitive, and irreflexive) can be used to identify allelic variants or genes with particular SNPs. It defines necessary and sufficient condition on the GeneVariant class such that all variants of a particular gene that include SNP descriptions while excluding the inference of a gene being a variant of itself.

5 Additional Requirements and New Investigations

5.1 Annotations

While some of the features added in OWL 1.1 increase the expressivity of our knowledge bases, the current specification¹⁴ **does not provide annotation properties, which are necessary and widely used** in the scientific domain. We support the incorporation of annotation properties into future revisions.

5.2 Antisymmetric properties

Since antisymmetric properties play an important role in the constructions of mereologies, as described in section 2, we encourage the addition of antisymmetric properties in future OWL specifications, if reasoning with SROIQ and antisymmetric properties is decidable.

5.3 Ontology Versioning

Our ontology versioning strategy publishes both the versioned document and the most current version with different HTTP URIs. This allows users to either use the most current set of compatible ontologies or select a set that is invariant over time. The OWL 1.0 specification provides the ability to keep track of previous ontology versions (*owl:priorVersion*) and further specify whether they are compatible (*owl:backwardCompatibleWith*) or incompatible (*owl:incompatibleWith*). However, **it is not currently possible to indicate a newer compatible or incompatible version** for any versioned ontology. This behavior is particularly important for agents to discover more recent versions and automatically determine whether the ontologies are compatible, and whether the user should “upgrade” to the newer version.

¹³ <http://ontology.dumontierlab.com/unit-individuals>

¹⁴ <http://www.w3.org/TR/owl11-syntax/>

5.4 N-ary predicates

There exists a proposal¹⁵ to incorporate n-ary *data* predicates into OWL. We believe that these would be tremendously valuable, as demonstrated by the set of use cases linked from the proposal page, and should be further extended towards the description of mathematical expressions (compatibility with MathML a certain plus). This will further enable us to express information that pertains to the representation of complex biological systems using differential equations, as has been established with SBML.

We would also urge some progress in the direction of providing support for **n-ary object predicates**, as has been previously described for DLR [12]. Support for DLR will then open the door to **temporal description logics** [13, 14], and enable us to express and reason about temporal models of knowledge. Currently, we query our temporal models via transitive object predicates linking explicit time intervals (months, years), and this enables us to ask about what has happened before or after some interval of time [7]. Clearly, keeping track of all instances is tedious, and would be better handled via reasoning over temporal axioms.

5.5 Description Graphs for Structured Objects

We find that OWL's class descriptions under specify structured objects. For instance, we cannot currently describe cycle-forming i) molecular structure and ii) chemical functional groups [4]. Recent work [15] describes that if some object in a model is an instance of a concept, then there must be graph around it. This will be useful for unambiguous definition of molecular structure, even with those containing cycles. However, should the inverse be possible, that is, to identify all individuals that satisfy a graph, then we could use it to discover which atoms are part of a chemical functional group. We expect that the constraints provided by description graphs will also improve reasoner performance.

5.6 Reflection: Modularity, Resolution, and Distribution

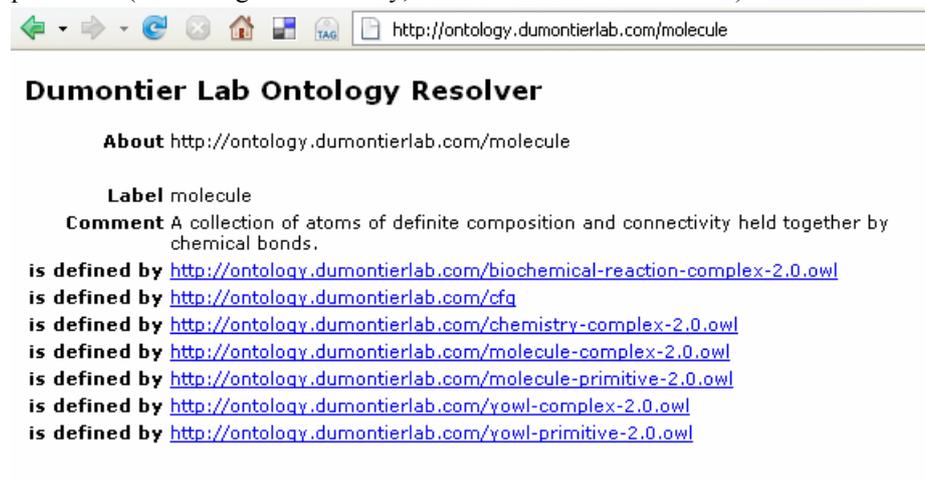
We believe that, in the long term, semantic web ontologies will compete in a dynamic marketplace in order to provide meaning to named entities. We anticipate that many logical derivations and deviations will exist for named entities, and that it will be up to reasoning-capable agents to determine which of these are compatible in order to satisfy some query. Towards this eventuality, we ensure that our ontologies¹⁶ and entities are web resolvable¹⁷. Information about ontological entities is provided by a web application which generates a RDF/XML document with the human readable label (rdfs:label) and definition (rdfs:comment) along with links (rdfs:isDefinedBy) to ontology documents that refer to or make statements about it (Fig. 2). An XML stylesheet makes the RDF/XML document viewable as HTML when using a web

¹⁵ http://www.w3.org/2007/OWL/wiki/N-ary_Data_predicate_proposal

¹⁶ <http://dumontierlab.com/?page=ontologies>

¹⁷ Ontologies and entities are named from the base URI - <http://ontology.dumontierlab.com/>

browser. We consider the natural language definition to be that which will ultimately guide ontology users in whether they should use the terminology. However, since the current mechanism of OWL imports is at the document level, users may choose that document which provides them with a set of axioms that they feel are necessary (constraints for data input) or useful (will generate valuable inferences). While some documents are substantially more expressive than others they can also be unnecessarily computationally expensive for certain queries. In the absence of automatic techniques to modularize ontologies in order to satisfy queries, we try to provide alternate documents to import so as to promote maximal reuse of domain ontologies. Our ontologies are generally composed of 3 layers that separate taxonomically organized domain terminology from i) disjointness, ii) complex expressions that define a world view, and iii) application specific requirements that impose a specific data model for data exchange and document validation [16]. In this way, some documents (particularly those labeled as “complex”) will import the primitives (containing the taxonomy, as well as label and definition).



About <http://ontology.dumontierlab.com/molecule>

Label molecule

Comment A collection of atoms of definite composition and connectivity held together by chemical bonds.

is defined by <http://ontology.dumontierlab.com/biochemical-reaction-complex-2.0.owl>

is defined by <http://ontology.dumontierlab.com/cfg>

is defined by <http://ontology.dumontierlab.com/chemistry-complex-2.0.owl>

is defined by <http://ontology.dumontierlab.com/molecule-complex-2.0.owl>

is defined by <http://ontology.dumontierlab.com/molecule-primitive-2.0.owl>

is defined by <http://ontology.dumontierlab.com/yowl-complex-2.0.owl>

is defined by <http://ontology.dumontierlab.com/yowl-primitive-2.0.owl>



Fig. 2 Entity resolution with reference to multiple ontology documents

Choice of which ontology to import will ultimately depend on what inferences are expected or demanded. The ontology documents are designed to i) provide *necessary* axioms and ii) reduce the computational complexity involved in importing all sorts of linked but unnecessary things. For instance, reasoning over the 118 disjoint atoms in the Atom Ontology is very expensive, but most of the atoms are not referred to, nor will they be queried. As such we derived an Ontology of Common Organic Atoms¹⁸ as a subset of the full atom ontology so as to speed up reasoning, but made more expressive with disjunction, unions and intersections, and qualified cardinality restrictions. Since the common Atom Ontology makes reference to the Atom URIs, it

¹⁸ <http://ontology.dumontierlab.com/atom-common>

is then possible to import the full atom ontology at any point in time, when necessary. Similarly, as part of the yOWL project [5, 6], we derived a 161 concept subset (GOSLIM) ontology from the full Gene Ontology (with over 19,000 concepts) so as speed up query answering. While generating subsets can be laborious, we expect that the application of automatic modularization techniques will help. Moreover, flexible and decentralized methods to create, extract, map and publish ontologies (whether full or derived fragments) will model more accurately the dynamics of information of the web, creating new opportunities in distributed knowledge discovery.

6 Conclusion

From syntactic sugar to increased expressivity, the newly proposed OWL 1.1 features augment the possibilities for the representation of life science knowledge. Property characteristics applied to a set of basic relations will simplify knowledge representation, limit proliferation of domain specific relations, and guide their proper use. New features concerning entity annotation, ontology versioning, n-ary data and object predicates, temporal description logics and description graphs should also be investigated to provide additional avenues for improving knowledge representation and facilitate the development of an expressive, ontology-based semantic web.

Acknowledgments: The authors would like to thank the anonymous reviewers for providing constructive feedback that has improved the quality of this paper. This work was supported in part with CONACYT scholarship #150581 for Natalia Villanueva-Rosales and NSERC Discovery Grant for Michel Dumontier.

7 References

1. W3C: OWL Web Ontology Language Guide. In: Smith, M.K., Welty, C., McGuinness, D.L. (eds.): W3C Recommendation (2004)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
3. Horrocks, I.: Applications of Description Logics: State of the Art and Research Challenges.: ICCS2005, Kassel, Germany (2005) 78-90
4. Villanueva-Rosales, N., Dumontier, M.: Describing chemical functional groups in OWL-DL for the classification of chemical compounds. OWL Experiences and Design, Innsbruck, Austria. (2007)
5. Villanueva-Rosales, N., Osbahr, K., Dumontier, M.: Towards a semantic knowledge base for yeast biologists.: First International Workshop on Health Care and Life Sciences Data Integration for the Semantic Web (HCLS 2007), Banff, Alberta (2007)
6. Battista, A.D.L., Villanueva-Rosales, N., Palenychka, M., Dumontier, M.: SMART: A Web-Based, Ontology-Driven, Semantic Web Query Answering Application. International Semantic Web Conference, Busan, South Korea (2007)
7. Ferres, L., Dumontier, M., Villanueva-Rosales, N.: Semantic Query Answering with Time-Series Graphs. The 3rd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2007), Annapolis, USA (2007)
8. Dumontier, M., Faizan, M., Obeng, J., Villanueva-Rosales, N.: Modeling the pharmacogenomics of depression. Semantic Web for Health Care and Life Sciences Workshop, Beijing, China (2008)

9. Horrocks, I., Patel-Schneider, P., Sattler, U., Parsia, B., Motik, B., Bechhofer, S., Calvanese, D., Giacomo, G.d., Lutz, C.: OWL 1.1 Specification. (2006)
10. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A.L., Rosse, C.: Relations in biomedical ontologies. *Genome Biol* 6 (2005) R46
11. Grenon, P., Smith, B., Goldberg, L.: Biodynamic ontology: applying BFO in the biomedical domain. *Stud Health Technol Inform* 102 (2004) 20-38
12. Calvanese, D., Giacomo, G.D., Lenzerini, M.: Conjunctive query containment in Description Logics with n-ary relations. *Description Logic Workshop (1997)* 5–9
13. Artale, A., Franconi, E., Wolter, F., Zakharyashev, M.: A Temporal Description Logic for Reasoning over Conceptual Schemas and Queries. *Logics in Artificial Intelligence (2002)* 98-110
14. Artale, A., Franconi, E.: A survey of temporal extensions of description logics. *Ann. Math. Artif. Intell.* 30 (2000) 171-210
15. Motik, B., Grau, B.C., Sattler, U.: Structured Objects in OWL: Representation and Reasoning. 17th Int. World Wide Web Conference (WWW 2008). ACM Press, Beijing, China (2008) 169-182
16. Dumontier, M., Villanueva-Rosales, N.: Three-Layer OWL Ontology Design. Second International Workshop on Modular Ontologies (WOMO07), colocated with Knowledge Capture (KCAP2007), Whistler, Canada (2007)