

A survey of requirements for automated reasoning services for bio-ontologies in OWL

C. Maria Keet¹, Marco Roos², and M. Scott Marshall²

¹ Faculty of Computer Science, Free University of Bozen-Bolzano, Italy

² Faculty of Science, University of Amsterdam, the Netherlands

keet@inf.unibz.it, {roos,marshall}@science.uva.nl

Abstract. There are few successful applications of automated reasoning over OWL-formalised bio-ontologies, and requirements are often unclearly formulated. Of what is available, usage and prospective scenarios of automated reasoning is often different from the straightforward classification and satisfiability. We list nine types of scenarios and specify the requirements in more detail. Several of these requirements are already possible in practice or at least in theory, others are in need of further research, in particular regarding the linking of the OWL ontology to data, reasoning over roles, and dynamically linking modular ontologies.

1 Introduction

Ontology development for the life sciences is driven by an active community that is marked by the development of the Gene Ontology (GO) in 1998 [14], which was primarily motivated by the drive for interoperability among the gene databases of the so-called model organisms, such as *Saccharomyces cerevisiae* (baker's yeast), *Drosophila melanogaster* (fruitfly), and *Mus musculus* (mouse) (more joined later, see <http://www.geneontology.org/GO.consortiumlist.shtml>). For this purpose, a structured controlled vocabulary in the GO-native format, obo, is already in wide use primarily for annotations [13, 36]. With the development of the Semantic Web, however, development of *formal* ontologies gains momentum, and in particular the prospect of *reasoning* over the ontologies is promoted as an important benefit outweighing the challenges of developing one. Over the past two years, bio-ontologies and taxonomies listed on the Open Biomedical Ontologies (OBO) website [38] have been converted into OWL format and many others of considerable size, such as the Cell Cycle Ontology [32] and Protein Ontology [42] have been developed using OWL or offer an OWL version of the ontology. However, there are as of yet few successful implementations that demonstrate the benefits of automated reasoning over formal bio-ontologies in the life sciences (comprising both basic biology and applied sciences such as medicine and agriculture). It may be that the time since W3C recommendation of OWL (2004) has been too short to realise “the killer application”. From literature on software implementations [4, 20, 22, 23, 30, 33] and requirements analysis with biologists (a.o., [26, 44]), however, current reasoning services are not always used as they were intended to be used by logicians. Furthermore, some

functionality from reasoning services that is expected by biologists is either not available in the expected form, not (yet) possible, or are available under another name. Based on literature research, further interviews, and our own experiences, we generalised from the examples and sorted them into a list of reasoning scenarios for bio-ontologies. Informally, and partially formulated in biologist and bio-ontologist parlance, they are as follows.

1. Supporting the ontology development process;
2. Classification;
3. Model checking (violation);
4. Finding gaps in an ontology and discovering new relations;
 - a. Deriving (TBox) types and relations from (ABox) instance-level data;
 - b. Computing derived relations at the type level;
5. Comparison of two ontologies ([logical] theories);
6. Reasoning with mereological parthood and other (part-whole) relations;
7. Using (including finding inconsistencies in) a hierarchy of relations;
8. Reasoning across linked ontologies;
9. Complex queries.

Some of the types of scenarios (e.g. 1-3) fall within extant capabilities of tools such as Racer, Pellet, and FaCT++, although the actual usage can be different from the intended usage and therefore could benefit from additional user friendliness, whereas e.g. item 5 is not feasible with OWL-formalised ontologies. Although a domain ontology in any subject domain could employ these types of reasoning scenarios, items 4-5 are quite specific to the life sciences.

The remainder of this paper explains and discusses the specifics of these scenarios in the subsections of section 2, where the main aims are a) to provide an overview of usage of and requirements for reasoning services and b) to contribute to foster convergence of the logic-based approach toward automated reasoning and biologists' expectations. We conclude in section 3.

2 Scenarios

2.1 Supporting the ontology development process

A high-level purpose-oriented requirement is to aid the ontologist to develop ontologies. This includes *any* type of automated reasoning service that can provide some guidance to the developer. From the domain expert's perspective, it is an obvious requirement that automated reasoners should find the 'correct errors', i.e., the source errors and not those that are merely wrong due to logical consequences of the source errors (see also §2.5 on role boxes). In addition, it facilitates the learning process of how to represent knowledge formally by intentionally introducing an error in the ontology and where the domain expert examines the reasoner output. It is also useful for distributed ontology development and to ensure no contradictory information remains in the ontology when an ontology becomes too large to comprehend. Finding mistakes and providing meaningful error messages to the user using "glass box techniques" as with Pellet and the development tool SWOOP [24] are useful improvements for this purpose.

2.2 Classification and model checking (and violation)

The classical DL reasoning services of classification of concepts and satisfiability have been sparingly used to date. One demonstration of the advantages of having a formal ontology, is the classification of protein phosphatases by [30], where also novel knowledge of biological interest was discovered that was entailed in the extant information but hitherto unknown. Problematic from the perspective of a biologist, is that this reasoning service needs DL-concept's properties, whereas in several bio-ontologies concepts lack conditions (partially because it is very difficult to identify necessary and sufficient conditions for natural kinds). Bandini and Mosca [4] pushed satisfiability reasoning a step in another direction. To constrain the search space of candidate rubber molecules for tire production, [4] defined in the TBox the constraints that all types of molecules for tires must satisfy, treated each candidate molecule as an instance in the ABox, and performed model checking on the knowledgebase: each instance inconsistent with respect to the TBox was discarded from the pool of candidate-molecules. Overall, the formal representation with model violation reduced the amount of costly laboratory research, because the number of candidate-molecules satisfying the desired properties was greatly reduced. Thus, the more inconsistencies, the better. This approach could be effectively employed in pharminformatics when searching for drug candidates.

Another approach to classification and model checking is to test the ontology (at the type-level) against instance data that ought to conform to the logical theory. In terms of realist ontologists: an ontology of universals has to represent reality, universals exist in reality and “[a] universal is an entity which is multiply located in space and time through its instances. It is what these instances share in common with each other” [28], therefore, for each DL-concept in the OWL-formalised ontology (representing a universal), there has to be at least one ABox instance (as representation of the entity in reality). Inconsistencies or concepts that do not happen to have instances associated with the concept may indicate an error in the definition of the concept or a ‘redundant’ concept in the ontology because it does not have representations of entities in reality.

2.3 Dealing with gaps and finding new relations and types

This requirement may be particular to the life sciences, where research is focussed on increasing understanding of nature and making discoveries. From the perspective of biology, proving the complexity class of a DL language is not a discovery, but finding out that the causative agent of stomach ulcers is the bacterium *Helicobacter pylori* is, or that each instance x of disease of type X with symptom y of type Y is always preceded by infection by z of species Z in all of its patients suffering from X . The idea is that the combination of bio-ontologies, instances, and automated reasoning services somehow can find either the missing relations, or the types, or both (e.g., [25, 22] for histones and the GO, respectively). But such novel knowledge is not (known to be) declared in the ontology. How can one find what is, or may, not be in the ontology but ought to be there? Necessarily, we have to break down this requirement into several components.

Computing derived relations The easiest version of ‘finding new relations’ is already a supported reasoning service, but not yet implemented in all Ontology Development Environments (ODE, which is the combination of ontology editor & automated reasoner). One can derive relations among types after an ontology developer has declared several types, relations, and other properties—but not everything. The reasoner takes the declared knowledge and returns relations that are logically implied by the formal ontology. From a user perspective, such a derived relation may be perceived as a ‘new’ relation, even though the relation was already entailed in the ontology, and can be welcome additions already; e.g., an OWL ontology based on biological knowledge described in 60000+ publications on apoptosis [19] is most likely ‘incomplete’ regarding declared relations.

Finding gaps yet to be filled A second scenario that is still relatively easy with respect to theory and technology, is where many concepts and relations have been declared, but one wants to find out where relations that are known by the developer have not yet been added to the ontology¹. For instance, the Foundational Model of Anatomy (FMA) has about 72000 concepts and 1.9 million relations among them [21, 34]—and is known to be incomplete in particular at the cellular and sub-cellular levels of granularity. Browsing an ontology of this size is not realistic, therefore, targeted queries by domain experts can help to find gaps. For instance, there are 17 types of **Macrophage** (types of cells of the immune system) in the FMA, which must be part of or contained in something. Declaring a recursive query in OQAFMA [21], gives as answer that **Hepatic macrophage** is part of **Liver** [17]. An informed user knows it cannot be the case that the other 16 types of macrophage are not part of anything. One probably cannot expect from a reasoner to second-guess the domain expert by asking “those 16 other types of macrophage are rather isolated, do you want them to have more relations to other anatomical structures?”; this is a manual assessment by the ontology developer to fill this gap—adding the missing relations—by developing those cell-level sections of the ontology. On the other hand, any concept in an ontology ought to have at least one relation to another concept, other than the subsumption relation, and providing an option to flag concepts without any property should be feasible to implement and will be of use; e.g., finer-grained colour-coding of DL-concepts in ODEs. In addition, reasoning services to detect ‘incompleteness’ requires inclusion of some *ontological* notions in new algorithms; e.g., if there is a root concept “Whole” (which may coincide with owl:Thing), then each concept in the TBox must have a path to that root, analogous to a subsumption hierarchy without ‘orphan concepts’² in the TBox.

¹ With centralised management of the ontology development process, one may be able to avoid this problem, but is more likely to occur in collaborative distributed ontology development.

² The notion of orphan concept is a left-over from informal ontologies in obo format where some concepts were (mainly accidentally) not included in the DAG, which, when converted into OWL-format, are classified directly under owl:Thing.

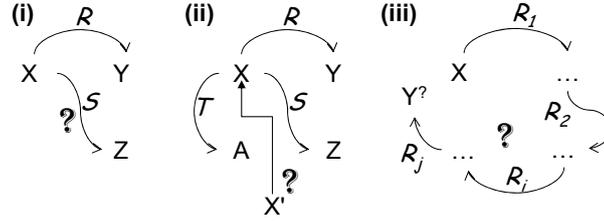


Fig. 1. Graphical depictions of the three query patterns over an ontology linked to the data source to find ‘new’ concepts or roles. (i): correlation between two roles; (ii): examining if the data supports some subconcept X' or a quaternary role; (iii): path query that may be of arbitrary length, with any roles and concepts that relate to X .

Finding new types of relations and DL-concepts by using instances

Combining TBox and ABox (be it as ABox, Instance Store [6] or some other RDBMS) is certainly more challenging and, to the best of our knowledge, no successful implementations have been achieved of the scenarios described further below in this section. In any case, instances will justify type-level declared knowledge and type-level knowledge is used for instances³, thereby the requirements essentially are those for a knowledge base (TBox & ABox), blurring the (philosophical) line between an ontology of types versus a database with instances. Let a section of some ontology be as depicted in Fig.1, lower-case letters denote instance-level entities (objects, tokens, omitted from the diagram to avoid clutter) of the types (DL-concepts, universals) in upper-case letters, then the following three *patterns* of discovering implicit knowledge are realistic queries (bio-examples follow afterward).

- i. “for each $x:X, y:Y, r:R, XRY$, does there exist a $z:Z, s:S$, such that there exist ≥ 1 x and xsz ?” , which is querying for the ‘known unknown’ Z , even if one adds a further constraint that, say, at least 50% of all instances of X that participate in XRY must also participate in instances of XSZ .
- ii. “for each $x:X, y:Y, r:R, XRY$, does there exist an xsz and an xta where $z:Z, s:S, a:A, t:T$ hold?” Logically, this is an extension of (i), but it is also simple rendering of the hypothesis that there may be a relation among the three roles R, S, T and, hence, among A, X, Y , and Z . Put differently: should there be a quaternary role instead of three binaries, or maybe a subtype X' that satisfies all the conditions?
- iii. Find-me-anything-you-have, that is, “for each $x:X$, return any r_1, \dots, r_n , their type of role and the concepts Y_1, \dots, Y_n they are related to” (with or without successive percentages as in item (i)).

³ Observe that there is a difference in approach between usage and intentions by biologists and medicine: medicine is generally more focussed on the instance-level *using* type-level knowledge (see [3] for algorithms that may support this task), whereas in the basic life sciences, one aims to *generalise* from instances to the type level, hence, discovering new type-level relations and concepts based on instance-level data.

Queries of pattern (i) are straightforward in relational databases, but challenging when one wants to query the data through an OWL ontology that contains concepts X , Y , and Z and relations R , S , and the instances are stored either in the ABox or in a relational database. Posing such a query on an electronic health record referent tracking information source with some 50 million tuples, one is necessarily limited to a simpler ontology language than OWL, such as DL-Lite [8, 9]; For instance, X could be **Patient**, R as **has disease**, Y as **Lactose intolerance**, S as **has symptom**, and Z **Nausea**, thereby querying if more than 50% of the patients that suffer from lactose intolerance also have the symptom of being nauseous. Performing type (ii) queries with brute force leads to a combinatorial explosion if it is not constrained by the domain expert *a priori*. For instance, one could demand from the ontologist to select only a few DL-concepts in the ontology to consider in the query. Observe that this requirement of searching for such stronger constraints that hold at the type level, which are logically entailed in instance-level data, is not supported by the current DL-based reasoners. Naively, some cues to solve this problem might be gleaned from database reverse engineering, whose algorithms already can detect concepts, relations, and mandatory and uniqueness constraints from the table definitions and data in the tables. With the third pattern, the aim is to find type-level knowledge analogous to fact finding in databases, but then the query answer also should include the concept the instance(s) belong to and the table name corresponding to the relation in an EER conceptual model. This still can be tractable if one considers only concepts directly related to X , but exploring the search space of sequences of conjunctive queries of not-predefined arbitrary length may not be realistic; for instance, to establish if, say, **G_s protein** is in some way related to anything else. A more realistic, and restricted, query is to discover the relationships between **Histone code**, **DNA sequence**, and **Gene expression regulation** [25], where at least the end points of the paths are given.

Other examples than diseases and symptoms, are, say, a plant specimen of a botanist who wants to locate the type (species), or who tries to find a new plant species by querying if there is a concept in the TBox that satisfies at least the necessary, but possibly also sufficient, conditions as provided by the botanist; if the query result is empty, the botanist consults the instances to examine if there are recorded specimen that satisfies the manually identified conditions. While it is theoretically feasible to query an OWL ontology to find such knowledge, we are not aware of ODEs that support it.

2.4 Comparing ontologies

There are many articles dealing with matching ontologies to achieve ontology integration (merging), or to have at least an approximation mapping between the ontologies. Similar technology could be used for a different purpose: comparing ontologies, where the aim is not ironing out differences, but the differences themselves are of interest. One can treat a formal ontology as a rendering of a scientific theory, and any discrepancies between two representations of ‘the same’ theory or two competing theories can provide an impetus for experimentation

to resolve the issues. Thus, one explicitly does not want to accept approximations on hyper-/hyponyms and near-synonyms of the terms. A variation on this theme is biological pathway comparison. Pathways can be similar across species but one would want to know the differences, or the pathway is the same except for one molecule or step due to localization (e.g., **cAMP pathway** in cells in the intestine versus in liver cells), or the comparison between the canonical healthy pathway versus changes due to toxins or genetic defects. In its simplest form, this amounts to checking for sub-graph isomorphisms [29], but it does not seem feasible with NExpTime-complete OWL-DL, thereby requiring some engineering solution where an OWL-formalised ontology can be simplified to a lite version to perform basic pathway comparisons.

2.5 DL role properties, role hierarchies, and part-whole relations

Parthood relations in bio-ontologies are as important as the subsumption relation, and reasoning with part-whole relations has been proposed and investigated by many (e.g., [1, 7, 12, 15]), focussing on transitivity and other properties of roles necessary for representing mereological parthood. A system such as OpenGalen [39] adds a further request for considering role hierarchies, which is also present in more limited form in the FMA and OBO Relation Ontology [27]. At present, automated reasoners take a “role box” at face value (assumed to be correct) and never return an inconsistency in the role box, but which acts out as inconsistent or re-classified concepts in the TBox, even though a role may be defined incorrectly. However, mistakes can be made in the RBox and, analogously to classification in the TBox, one would expect this feature also for an RBox that permits role hierarchies, that is, for any OWL-formalised ontology. For instance, asymmetry implies irreflexivity, therefore any subrole of an asymmetric role should not be irreflexive only, because the subrole has to inherit the stronger notion of asymmetry from its parent role.

2.6 Reasoning across ontologies

Relating all bio-ontologies across their respective levels of granularity, envisaged –but not yet specified in detail– by the OBO Foundry [37], will result in an extremely large ontology that has to be in the most expressive OWL language that its most comprehensive ‘base’ ontology uses. We assume management of OWL-sublanguages for the various bio-ontologies [10] has been addressed and does not lead to undecidability. For instance, linking the following ontologies into one larger theory for reasoning across the respective ‘base’ ontologies makes the whole theory in $\mathcal{ALCHON}(\mathcal{D})$: MGED Ontology for microarray experiments is in $\mathcal{ALEOF}(\mathcal{D})$, BioPax for biological pathways in $\mathcal{ALCHON}(\mathcal{D})$, Cell type in $\mathcal{ALE}(\mathcal{D})$ and Mammalian Phenotype in just $\mathcal{AL}(\mathcal{D})$ (as of 10-2-2007, downloaded from OBO [38]). Although users want to pose cross-granular queries, one might expect that most automated reasoning will occur locally within the, possibly less expressive and separately developed, ‘base’ ontology. Users should be able to take advantage of more efficient reasoning over their simpler lite ontology

and benefit from improved performance. One only needs *on demand* access to content of other ontologies (sections of the combined ontology) to perform a cross-disciplinary query. For instance, where a biochemist wants to know more information about receptors (proteins in/on the cell wall) that reside in organs of several species, or combine pathological anatomical structures with an ontology of infectious organisms. This kind of coordinated modularization and linking of ontologies on demand will need some form of automation to compute (re-)connect points between the base ontologies (or, from the viewpoint of the integrated ontology: between ontology modules). Some work in this direction at the same level of granularity has been done with intersecting the GO cellular components with the GO biological processes to compute relations between the GO ontologies [11, 23]. Such work will need to be expanded across levels of biological granularity.

2.7 Other complex queries

Several types of queries already have been discussed in previous subsections, such as conjunctive queries and recursive queries. A variation on query pattern (ii) in section 2.3 was suggested in an application scenario by [5], where one can query the FungalWeb ontology [35] for any type of enzyme that “acts on” (degrade or modifies by other means) the type of molecule **Pectin**, hence, where X and (subtypes of) Y are known parameters but the type of relation is underspecified. Similarly, one should be able to extend this to a query to retrieve all biological pathways that contain e.g. **Ubiquitin** (which requires use of part-whole relations) or the place of **Hedgehog** in food webs (using cycles among the concepts in the ontology). Marshall et al’s interval join [20, 43] highlights another aspect of querying through an ontology by considering measurements (concrete values). Further, the query did not scale well when applied only to human genome data (11 million triples). Such queries will be more challenging to implement through an OWL-ontology with larger-scale genome data of several species.

3 Conclusions

There are few successful applications of automated reasoning over OWL bio-ontologies, and requirements are often unclearly formulated. Of what is available, usage and prospective scenarios of automated reasoning is often different from the straightforward classification and satisfiability. We have specified the requirements for automated reasoning services in more detail and highlighted the life scientists’ high-level goal-driven approach and desire to use OWL ontologies with reasoning services for discovering novel information about nature. Several reasoning requirements are already possible in practice or in theory but not yet implemented in the ODE tools. Some requirements need further research, in particular regarding the linking of the OWL ontology to data, reasoning over roles, and dynamically linking modular ontologies. Better explanations of currently supported services in terms of biologists’ use cases may induce realization of automated reasoning as well.

We plan to conduct an analysis on using OWL ontologies in conjunction with statistical querying, reasoning with uncertainty and other such application scenarios (see e.g., [18, 31]) to extract reasoning requirements that traditionally are not addressed by DL-based reasoners.

Acknowledgments The authors would like to thank Santiago Garcia-Vallvé, Benjamin Good, Kristina Hettne, and Barry Smith for their elaborations on requirements for reasoning over bio-ontologies.

References

1. Artale, A., Franconi, E., Guarino, N., Pazzi, L. Part-Whole Relations in Object-Centered Systems: an Overview. *Data & Knowledge Eng.*, 1996, 20(3):347-383.
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds). *Description Logics Handbook*, Cambridge University Press, 2003.
3. Baader, F., Ganter, B., Sertkaya, B., Sattler, U. Completing Description Logic knowledge bases using Formal Concept Analysis. In *Proc. of IJCAI 2007*.
4. Bandini, S., Mosca, A. Mereological knowledge representation for the chemical formulation. *2nd Workshop on Formal Ontologies Meets Industry 2006 (FOMI2006)*, 14-15 December 2006, Trento, Italy. 55-69.
5. Baker, C., Witte, R., Shaban-Nejad, A., Butler, G., Haarslev, V. The FungalWeb ontology: Application scenarios. *8th Bio-Ontologies Meeting 2005*, Detroit, USA.
6. Bechhofer, S., Horrocks, I., Turi, D. The OWL Instance Store: System Description. *Proceedings of CADE-20*, Lecture Notes in Computer Science, Springer-Verlag, 2005.
7. Bittner, T., Donnelly, M. Computational ontologies of parthood, componenthood, and containment, In: *Proc. of IJCAI 2005*, Kaelbling, L. (ed.), 382-387.
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M. A. Poggi, and R. Rosati. Linking data to ontologies: The description logic DL-Lite A. In *Proc. of the 2nd Workshop on OWL: Experiences and Directions (OWLED 2006)*, 2006.
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R. DL-Lite: Tractable description logics for ontologies. In: *Proc. of AAAI 2005*, 602-607.
10. Cuenca-Grau, B. Kutz, O. Modular ontology languages revisited. *IJCAI Workshop on Semantic Web for collaborative knowledge acquisition*. Hyderabad, India, 2007.
11. DeJongh, M., Van Dort, P., Ramsay, B. Linking Molecular Function and Biological Process Terms in the Gene Ontology for Gene Expression Data Analysis. *Proc. of IEEE Engineering in Medicine and Biology Society (EMBS)*, 2004, 2984-2986.
12. Donnelly, M., Bittner, T. and Rosse, C. A Formal Theory for Spatial Representation and Reasoning in Biomedical Ontologies. *AI in Medicine*, 2006, 36(1):1-27.
13. Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 2004, 32: D258-D261.
14. Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 2000, 25: 25-29.
15. Hahn, U., Schulz, S. Parthood as Spatial Inclusion - Evidence from Biomedical Conceptualizations. *Proc. of KR-2004*. Whistler, Canada, 2004.
16. Horrocks, I., Kutz, O., Sattler, U. The Even More Irresistible *SROIQ*. In: *Proc. of KR-2006*, Lake District, UK, 2006.
17. Keet, C.M. *Granular information retrieval from the Gene Ontology and from the Foundational Model of Anatomy with OQAFMA*. KRDB Research Centre Technical Report KRDB06-1, Free University of Bozen-Bolzano, 6 April 2006. 19p.

18. Khatri, P., Draghici, S. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 2005, 21: 3587-3595.
19. Lazebnik, Y. Can a Biologist Fix a Radio? – or, What I Learned while Studying Apoptosis. *Cancer Cell*, 2002, 2: 179-182.
20. Marshall, M.S., Post, L., Roos, M., Breit, T.M. Using semantic web tools to integrate experimental measurement data on our own terms. *Workshop on Knowledge Systems in Bioinformatics (KSinBIT'06)*, Springer Verlag LNCS 4277, 2006, 679-688.
21. Mork, P., Brinkley, J.F., Rosse, C. OQAFMA Query Agent for the Foundational Model of Anatomy: a prototype for providing flexible and efficient access to large semantic networks. *Journal of Biomedical Informatics*, 2003, 36: 501-517.
22. Mungall, C.J. Obol: integrating language and meaning in bio-ontologies. *Comparative and Functional Genomics*, 2004, 5(6-7):509-520.
23. Myhre, S., Tveit, H., Mollestad, T., Lægreid, A. Additional Gene Ontology structure for improved biological reasoning. *Bioinformatics*, 2006, 22: 2020-2027.
24. Parsia, B., Sirin, E., Kalyanpur, A. Debugging OWL ontologies. *World Wide Web Conference (WWW 2005)*. May 10-14, 2005, Chiba, Japan.
25. Roos, M., Rauwerda, H., Marshall, M.S., Post, L., Inda, M., Henkel, C., Breit, T. Towards a virtual laboratory for integrative bioinformatics research. *CSBio Reader: Extended abstracts of CS & IT with/for Biology Seminar Series 2005*. Free University of Bozen-Bolzano, 2005, 18-25.
26. Ruttenberg, A., Clark, T., Bug, W., et al. Advancing translational research with the Semantic Web. *BMC Bioinformatics*, 2007, 8, Suppl. 2 (in press).
27. Smith, B., Ceusters, W., Klagges, B., Köhler, J., Kumar, A., Lomax, J., et al. Relations in biomedical ontologies. *Genome Biology*, 2005, 6:R46.
28. Smith, B., Kusnierczyk, W., Schober, D., Ceusters, W. Towards a reference terminology for ontology research and development in the biomedical domain. *Proc. of KR-MED 2006*, Baltimore, USA, 57-66.
29. Tian, Y., McEachin, R.C., Santos, C., States, D.J., Patel, J.M. SAGA: a subgraph matching tool for biological graphs. *Bioinformatics*, 2007, 23(2): 232-239.
30. Wolstencroft, K., Stevens, R., Haarslev, V. Applying OWL reasoning to genomic data. In: *Semantic Web: revolutionizing knowledge discovery in the life sciences*, Baker, C.J.O., Cheung, H. (eds), Springer: New York, 2007, 225-248.
31. Zhou, Y., Young, J.A., Santrosyan, A., Chen, K., Yan, S.F., Winzeler, E.A. In silico gene function prediction using ontology-based pattern identification. *Bioinformatics*, 2005, 21(7):1237-1245
32. Cell Cycle Ontology. <http://www.cellcycleontology.org>.
33. Debugging the Bug. http://bio.freelogy.org/wiki/Debugging_the_bug.
34. Foundational Model of Anatomy (FMA). <http://fme.biostr.washington.edu:8089/FME/index.html>.
35. FungalWeb Ontology. <http://www.cs.concordia.ca/FungalWeb/>.
36. Gene Ontology Consortium. <http://www.geneontology.org>.
37. OBO Foundry. <http://obofoundry.org>.
38. Open Biomedical Ontologies. <http://obo.sourceforge.net>.
39. OpenGalen Tutorial 9. <http://www.opengalen.org/tutorials/crm/tutorial9.html>.
40. OWL 1.1. (Editor's draft of 6-4-2007). <http://webont.org/owl/1.1/>.
41. OWL. <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/syntax.html>.
42. Protein ontology. <http://proteinontology.info>.
43. Semantic Data Integration for Histone Use Case Website. <http://integrativebioinformatics.nl/semanticdataintegration.html>.
44. W3C Semantic Web Health Care and Life Sciences Interest Group. <http://www.w3.org/2001/sw/hcls/>.