Putting Semantics into the Semantic Web: How Well Can It Capture Biology?

Toni Kazic

Pacific Symposium on Biocomputing 11:140-151(2006)

PUTTING SEMANTICS INTO THE SEMANTIC WEB: HOW WELL CAN IT CAPTURE BIOLOGY?

TONI KAZIC
Dept. of Computer Science
University of Missouri
Columbia, MO 65201
toni@athe.rnet.missouri.edu

Could the Semantic Web work for computations of biological interest in the way it's intended to work for movie reviews and commercial transactions? It would be wonderful if it could, so it's worth looking to see if its infrastructure is adequate to the job. The technologies of the Semantic Web make several crucial assumptions. I examine those assumptions; argue that they create significant problems; and suggest some alternative ways of achieving the Semantic Web's goals for biology.

1. Introduction

Imagine you are interested in purine salvage. You go to KEGG's maps [1] and see that the reactions

$$\text{deoxyadenosine} \rightleftharpoons \text{adenine} \tag{1}$$

$$\text{deoxyinosine} \rightleftharpoons \text{hypoxanthine} \tag{2}$$

are both catalyzed by EC 2.4.2.4. When you click on the link for that EC number, you discover the name of the enzyme is thymidine phosphorylase, and its reaction is

$$\text{thymidine} + \text{phosphate} \rightleftharpoons \text{thymine} + \text{2-deoxy-}\alpha\text{-D-ribose 1-phosphate} \tag{3}$$

Thymidine isn't a purine nucleoside, so how can it catalyze the cleavage of deoxyadenosine and deoxyinosine? Maybe KEGG's chart is mixed up, so you go to to the web site of the Joint Committee on Biochemical Nomenclature, which is the international body in charge of classifying enzymes [2]. Same result — EC 2.4.2.4 is thymidine phosphorylase — but now you notice a comment saying the enzyme can catalyze reactions like those of EC

2.4.2.6, nucleoside deoxyribosyltransferase,

$$\text{2-deoxy-D-ribosyl-base}_1 + \text{base}_2 \rightleftharpoons \text{2-deoxy-D-ribosyl-base}_2 + \text{base}_1 \quad (4)$$

under some circumstances. That's a fundamentally different reaction than that shown for EC 2.4.2.4; one cannot logically substitute a nucleotide base for a phosphate or the ribosyl moiety of nucleosides. If one rewrote the KEGG reactions to fit that of nucleoside deoxyribosyltransferase, *e. g.*,

$$\text{deoxyadenosine} + \text{thymine} \rightleftharpoons \text{thymidine} + \text{adenine}, \quad (5)$$

the result is still not the reaction shown either for EC 2.4.2.4 or on KEGG's map. Adding to the confusion, you notice that two synonyms for the enzyme's name are "blood platelet-derived endothelial cell growth factor" and "gliostatins". Statins stop things and growth factors stimulate growth; endothelial cells are not the same as glial cells; so how can the same enzyme stimulate the growth of one cell and inhibit the growth of another? And why would an enzyme of nucleotide salvage (you're still not sure it's the right enzyme) be involved in cell growth anyway?

Your ability to check, understand, and reconcile apparently contradictory information depends on understanding the semantics of terms such as thymidine, purine, statin, growth factor, glia, and endothelial cell; questioning the apparent contradictions; and synthesizing information from multiple sources. The hope for the Semantic Web is that it would do just these things automatically, accurately, and transparently on the Internet. Simple questions, usually the hardest, would be simply, rapidly, and at least plausibly answered. The net would become a connected engine of knowledge and inference [3].

This powerful and alluring vision has stimulated great excitement. But the utility of the Semantic Web to address biological questions depends as much on the adequacy of its infrastructure as it does on the passion of its advocates. Computations that return biologically incorrect or misleading answers are not helpful, especially if automation returns more "results". To ensure the *scientific* validity of the Semantic Web's computations, it must sufficiently capture and use the semantics of the domain's data and computations: for example, it mustn't confuse reactions of enzymes with reactions to drugs (unpleasant side-effects). Accurate semantics are even more important if the goal of the computation is to return plausible answers, since plausibility depends on persuading someone that the implicit relationships among rather disparate facts are strong enough to form a reasonable hypothesis (*e. g.*, reactions 1 and 3 are the same). Since the Semantic Web's

fundamental technologies will be the foundation for any domain-specific extensions, it's important to ask how well their structure fits the semantics of biology [4].

One can think of the Semantic Web and the scientific databases and algorithms it would call as a collection of languages that denote information. To translate among them, their semantics must be adequately specified. Two of the most important requirements for a system to scale are that its operations are automatic and that its methodologies are distributed. Here I examine several of the fundamental assumptions of the Semantic Web's infrastructure to estimate their limits *in the context of computations of biological interest*. With respect to semantics, I consider the structure of RDF and its denotational semantics; and for scalability, the topology and automation of semantic translation. While the only solutions I can suggest are as partial in their ways as the current approaches (see Section 6), I hope to stimulate broader consideration of the technical foundations of the Semantic Web's application to scientific computing. Thus, this paper is in the spirit of BioMOBY's goals of figuring out what's needed for the scalable big picture [5].

2. What are the Assumptions?

- ***A simple syntax is sufficient.*** The non-logical relationships among concepts are ultimately captured in the RDF specification of the terms, usually called an RDF Schema. The assumption is that the <subject> <predicate> <object> syntax is sufficient [6]; I argue in Section 3 that it is not.
- ***An implicit semantics is effective.*** The semantics of a term are given in a natural language comment and in the applications that use the term. Neither of these forms can be computationally inspected to determine the semantics of the term; the Semantic Web relies on humans reading the comments and code to determine the semantics. I argue in Section 4 that this implicit semantics is less effective than an explicit, by which I mean computationally determinable, semantics.
- ***Bilateral mappings, manual translation, and automated inference are just right.*** Because the tasks of definition and implementation are distributed to the community, and DAML+OIL-powered inference engines would translate among different definitions using manually constructed bilateral mappings, the claim is

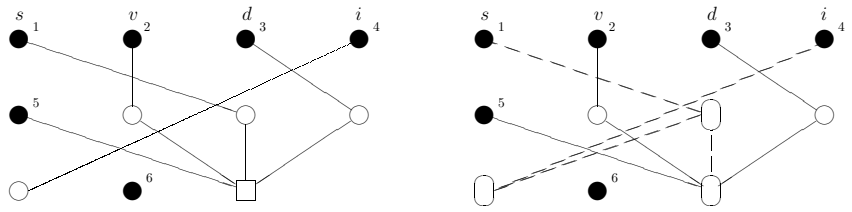

Figure 1. Structural inadequacy of RDF's syntax. The sentences are read from left to right. Each column is a part of speech (labelled s, subject; v, verb; d, direct object; i, indirect object). (left) Successive transformations of a complex sentence to fit RDF's syntax by nesting the subject, direct object, and indirect object of the original sentence together. Introduction of terms implied in the original sentence enables the nesting. Unique terms are denoted by numbered closed circles; open circles denote terms unchanged by syntactic transformation; the square denotes the nested, complex object in the final transformation. Solid lines indicate a term's syntactic transformations and the direction of transformation is from the top down. (right) Differences in term semantics can be concealed by the syntax. The enzyme denoted by term 4 has bound the substrates denoted by term 1, but the syntactic transformations needed to fit the original sentence to RDF give no clues about this change in the semantics of term 4 to another application. The ovals denote terms whose properties are not uniformly inherited. The dashed lines indicate the inheritance relationships and inheritance is from the top down. The terms are: 1, "thymidine and phosphate"; 2, "*convert*"; 3, "*thymine and 2-deoxy-α-D-ribose 1-phosphate"; 4, "*thymidine phosphorylase"; 5, "*reaction"; 6, "catalyzes".

that the Semantic Web will scale. I argue in Section 5 that scalability and precision will be very poor.

3. Is the Structure Sufficient?

RDF captures information in <subject> <predicate> <object> phrases. Thus, "<A> <is an element of> <B>" and "<A> <parses> <B>" fit into this syntax. When the syntax one would naturally use is more complex — say, "thymidine and phosphate are converted into thymine and 2-deoxy-α-D-ribose 1-phosphate by the enzyme thymidine phosphorylase" — one approach is to group the terms through nesting, then use that in a second phrase. This is illustrated in the left panel of Figure 1. Thus, thymidine, phosphate, thymine, and 2-deoxy-α-D-ribose 1-phosphate are ultimately compressed into a phrase that serves as a direct object: "<thymidine phosphorylase> <catalyzes> <*reaction *convert* thymidine and phosphate *thymine and 2-deoxy-α-D-ribose 1-phosphate>" (*s are the omitted prepositions, articles, and auxiliary verbs).

However, the reaction equations and our model sentence have a much richer set of connotations, and these connotations don't readily fit into

the syntax. For example, enzymes bind their substrates before catalyzing the reaction, and this binding partitions the populations of molecules into rapidly exchanging, random subpopulations of enzyme-substrate complexes. This information is expressible in natural language. But when compressed into RDF's syntax, the phrasal transformation has separated the interacting molecules (thymidine phosphorylase, thymidine, phosphate), concealing that the semantics of "thymidine phosphorylase" *in the reaction* are different from the semantics of the unbound enzyme (Figure 1, right panel). So another application may be able to unify its "thymidine phosphorylase" with this one, but the semantics of the two instances can differ and the rest of the phrase will not necessarily provide any clues as to the difference. Building a tree of phrases to emulate binding (*e. g.*, "thymidine phosphorylase binds thymidine or phosphate", "the complex of thymidine phosphorylase and thymidine or phosphate binds phosphate or thymidine", *etc.*) forces one to say explicitly something one may not know (*e. g.*, whether the binding is random or sequential, what the order of any sequential binding is, how many substrates are bound per enzyme). By expanding the detail to accommodate the phrasal structure, essential and useful ambiguities have been lost; the task of deciding where to unify an instance of "thymidine phosphorylase" from another application has been complicated; and the ease of description has vanished.

4. Are an Implicit Semantics Effective?

Like other languages, the semantics of the Semantic Web depend on those of its grammar and terms, their "context", and the applications that use them [7]. The terms are named either directly in an RDF document or in a referenced ontology. By "context" of a term, the Semantic Web workers generally mean the URI at which the term is found; if the same term is used in different URIs, its semantics are assumed to be different unless explicitly stated otherwise. But the more common meaning of context is especially important to the Semantic Web, because the *way* a term is used helps bound its semantics. In the Semantic Web, this usage is intended to be by programs. The only way a program can "know" that it is doing something biologically meaningful to the data retrieved by a term from another application is if it can check that the application's definition of the term and its definition of the term are identical and unambiguous.

Terms used in an RDF *may* be defined, in English, in the comments (more often these are partial descriptions of the denoted concepts rather

than definitions) [4, 8, 9]. In this situation, the semantics of the language are implicit in three important senses. First, any definitions are in natural language, which remains notoriously difficult for machines to understand. Second, much of the semantics of a construct are carried in the applications — easy for machines to process, but less accessible to humans trying to understand exactly what a program means by a "gene" or "reaction". Third, some of the semantics are intensional, relying on automated reasoning systems such as DAML+OIL, the logical features of OWL, or the predicate calculus to draw (so far relatively simple) inferences.

An implicit, non-machine computable, semantics raises three problems. The first is that people must do the job of reading, understanding, and reifying the connotations of a term or a program before they can implement any resource (*e. g.*, see references 3, 8). How many people are really willing or able to do this, especially when the domain is as specialized as biology? Reactions 3 and 4 are stated to be related, but it requires much more semantics than just knowing that thymidine is a member of the set of 2-deoxy-D-ribosyl-bases to determine how to map these, and there are probably more biologists and chemists who know that information than developers. The second problem is the fact that humans interpret constructs differently; we don't all know what the words mean because we vary so much in the way we use natural languages (for some biological examples, see reference 10; broader cultural examples are found in references 11, 12). Terms, whether natural language words or RDF properties, merely point to a variable set of connotations [13]. Are reactions 1 and 3 the "same"? The answer depends on whether you think KEGG's map has omitted other reactants and your willingness to believe thymidine phosphorylase is broadly specific.

The third problem is that most of the semantics are pushed onto the applications. While reading other people's code can be frustrating, the fundamental problem is that the semantics are far less transparent than they could be. Suppose one wants to test whether thymidine phosphorylase could catalyze reaction 1, and there's a resource that retrieves "related" reactions. One's interpretation of the output will be determined by how that resource defines "related", which will be determined by its code and any underlying databases. Just seeing the result won't divulge the resource's notion of "related", and returning the expected result doesn't test the hypothesis of relationship unless one knows how the test was made. Lack of semantic transparency limits reuse, since each developer must inspect the code of possible components for him or herself.

5. Will It Scale?

Scaling the current model of the Semantic Web requires enough pairs of resources that translate between them, and robust enough inference engines, so that local subnets can be automatically connected. The assumption is that enough translating pairs would spontaneously arise so that a suitably equipped inference engine could compute a directed path between any two resources. The path is directed because one can assume only that the translations will be asymmetric; symmetric systems would be multigraphs, and very welcome too.

Obviously this scaling by $n(n-1)$ is more labor-intensive than if applications refer to a common semantic middle layer, and there have been enough pleas to keep URIs stable and to reuse ontologies to warn the naïve that the machinery could break for trivial reasons. Similarly, people reading and reconciling the system's semantics from ontologies and code is slow. A more fundamental problem is that automatically drawn inferences can explode all too easily. The "signal" – useful or novel inferences — is often lost in the "noise" of the huge collection of trivial inferences (for example, see reference 14). Why wouldn't this happen in the Semantic Web? The usual answer is that declarations that site A "trusts" only site B for certain kinds of information will sufficiently bound the inferences. Perhaps; but then for the Semantic Web to be a web, site C must decide to trust A, rather than site D; and once site E trusts B and D, forming a web, E will have to decide what to do with contradictions, incomplete information, and semantic inconsistencies. (Here I push the Semantic Web beyond its stated plan of not worrying about the accuracy of inferences [3], because automating incorrect scientific inference, *e. g.*, thymidine phosphorylase is not an enzyme of purine salvage because thymidine is not a purine nucleoside, is not a step forward.) As the topology of the Semantic Web changes from disconnected small graphs to larger connected components, the number of paths among the possibly relevant URIs in a component will also increase explosively [15]. One might even run into resource issues, in the sense of available cycles to compute all those inferences and paths.

6. Towards Solutions

This list is not exhaustive, and none of these suggestions will solve all aspects of the problems raised. Indeed, some could interfere with the results of others. But each offers at least one alternative to the problems of the present infrastructure.

Enrich the Allowed Syntax If RDF's syntax isn't robust enough, why not enrich it? Adding more parts of speech, such as adjectives and indirect objects, and permitting its trees to become networks, might well solve the problems of Section 3. Rather than efforts to map more complex relations onto RDF, why not let RDF accommodate these constructs directly, for example by semantic networks [16, 17]? One might begin by watching biologists diagram and explain the relationships among concepts in research papers, which will often be among sentences.

Let Biologists Build Since biologists know the semantics — even if they disagree — one way to develop applications is to let them define the semantics *in a structured way.* Exploiting biological expertise is the fundamental power of the UMLS, the GO, nomenclature committees, and similar efforts, and spreading the effort is central to the Semantic Web's philosophy [17–19]. One possible advantage of a richer syntax is that it might enable faster definition by biologists. But to prevent cacophony, we must either all agree on the semantics of the terms or find a way to translate among them. Agreement is slow, socially difficult, and scientifically inflexible; translation is hard.

Make the Semantics Finely Grained One reason agreement is hard is because we tend to focus on relatively "big" ideas rather than on their component notions. As a crude example, rather than arguing over whether reactions 1 and 4 are similar, one could define different types of similarities, and then allow each person or application to mix and match those types to suit their needs. (The example is crude because in practice there are many much more finely grained notions underneath, such as the tautomerism of the bases.) In an ideal world, the most finely grained ideas would be so axiomatic as to be uncontroversial; and for mixing and matching to be unambiguous, the semantics of the axioms and the symbols denoting them would have to be unique. Two problems would likely arise: ensuring that the axioms, their denotations, and semantics were unique; and deciding how to scope the axioms so that they were truly elementary for the scientific domain in question. For example, whether one maps a molecule's name to SMILES string or a Hamilitonian depends strongly on one's domain.

Very fine granularity exacerbates a problem ontology developers have already experienced: keeping track of the terms so that relevant ones are easily found and all their semantics are disjoint. Solving this tracking problem would let us to avoid meetings and arguments and help suppress synonymy among the terms.

Make the Semantics Computable The obvious solution to implicit

```
<owl:AnnotationProperty rdf:about="&foo;purine"/>

<owl:Class rdf:about="#nucleoside">
  <rdfs:label>nucleoside</rdfs:label>
  <foo:purine>deoxyadenosime</foo:purine>
</owl:Class>
```

Figure 2. A mythical nucleoside ontology *foo*.

semantics is to make them computationally explicit. At its most basic, this would mean a set of signs — not terms in a natural language, but computational data structures — such that *each one's semantics can be algorithmically determined*. Then every application, database, and query are self-describing by referring to the signs; and the semantics of the descriptions can be computed by any other application.

In the Semantic Web, the computations are either ontological on the term (this term has some relationship to another term) or applications acting on the data denoted by the term. For example, some of the key relations in OWL are those of set theory (*e. g.*, *subClassOf*, *sameAs*, *cardinality*, *disjointWith*) and annotation tags (*e. g.*, *AnnotationProperty*, *OntologyProperty*, *isDefinedBy*, *subPropertyOf*) [9, 20]. The best candidate for describing the semantics of biology is likely to be the *AnnotationProperty* [21]. For example, one might have a reference ontology *foo* that described nucleosides (see Figure 2). One could compute that deoxyadenosine is a member of the nucleosides using this ontology. But if the ontology's relations were incomplete — for example, that some nucleosides have the structure of 2-deoxy-D-ribosyl-base — then one would need other code to compare the structures of thymidine and adenosine to decide if thymidine is a 2-deoxy-D-ribosyl-base. The *semantics* of the biology are given by the *structural relationships* of the molecules, not the words used to denote particular molecules or classes of molecules. Conversely, the semantics of OWL's literals are given by natural language definitions — that is, they are implicit.

In contrast, I'm suggesting one determine the *definition* of a computational structure denoting a biological idea from that structure and rules governing its formation from very finely-grained axioms. Semantically, this is what compilers do. Several years ago I developed a formal language to do that for biology, called *Glossa*, and we demonstrated this idea works for the semantically most demanding queries we found in a relational database of maize genetics [22, 23]. *Glossa*'s capabilities haven't yet been thoroughly

tested for significant areas of biology; it suffers from the tracking problem even more acutely than the ontologies do; and it's even less user-friendly than ontologies. So it's premature to believe *Glossa* will be the solution. Description Logics may offer still another route if they can capture the biology sufficiently and escape the limitations of OWL and *Glossa* [24].

Constrain Inferences to Improve Scaling Making semantics computably explicit might decrease scaling by facilitating automatic inferences. Schemes that scoped inferences by semantics ("thymidine phosphorylase accelerates an enzyme reaction, not a drug reaction") obviously depend on being able to compute something about semantics. Or one might permit inferences over the entire web, testing (perhaps at each inferential step) by some set of quality metrics. For example, one might prefer rarer inferences built over longer chains of reasoning; inferences whose components had the fewest number of direct contradictions (or for contrarians, the maximum number); or the most frequent inferences that appear within N reasoning steps. Or perhaps a more explicitly theorem-proving approach is desirable: if a proposition can be proved it is of interest. One can imagine a smorgasbord of such metrics, and it would be fun to compare them. Sequential inference over structured resources is hard enough, but often the most important answers come from unstructured context. For example, it is very easy to find web pages about EC 2.4.2.4, but much harder to see why being both a growth factor and a statin is very provocative, let alone how this occurs. Right now, determining that means reading a large number of retrieved documents. Once again, we're back in the realm of natural language processing.

7. Prospects

So is the Semantic Web the wrong vision for biology? Perhaps not; but there are some fundamental gaps between the infrastructure of the Semantic Web and the needs of distributed computations for biology.

Whatever infrastructure is developed, one problem I've ignored so far — usage — will affect how well the Semantic Web ultimately facilitates scientific computation. At present, only manual inspection can tell if a term retrieves or produces semantically identical types of data within, let alone among, resources. Even when people read the directions (in this case, definitions), consistently implementing them is extremely hard, especially as the volume of data increases. One's view of the meaning of the definitions changes as more instances are worked through; usually more than

one person builds a resource, such as a database, and they all have slightly different ideas; and people make mistakes. Ideally, a device would use the definitions to check the usage of terms in each contextual instance, making the Semantic Web self validating. Since many clues to inappropriate usage come from the connotations stored in a wide knowledge of the field, the device would have to somehow compare usage with the definitions and that knowledge — bringing us full circle to why the Semantic Web is an important idea worthy of effort.

Acknowledgments Frank Olken, Robert Stevens, Olivier Bodenreiter, and Daniel McShane encouraged me and gave me an excuse to go out on this particular limb (they are blameless). Jonathan W. King beautifully demonstrated the thymidine phosphorylase example to me. Armani Valvo prompted many useful walks. The reviewers' incisive comments considerably improved the paper. I am grateful to all of you. This work is supported by a grant from the U. S. National Institutes of Health (GM-56529).

References

1. Minoru Kanehisa, Susumu Goto, Hiroyuki Ogata, Hiroko Ishida, Sanae Asanuma, Toshi Nakatani, Saeko Adachi, Kana Matsumoto, Noriko Man, Rumiko Okada, Hidemasa Bono, Kazushige Sato, Toyoko Katrurada, Tomomi Kamiya, Mayuko Egoshi, Wataru Fujibuchi, Hiromi Adachi, Takaaki Nishioka, and Atshuhiro Oka. *From Sequence to Function. An Introduction to the KEGG Project*. Univeristy of Kyoto, Uji, Japan, `http://www.genome.ad.jp/kegg/kegg1.html`, 1996–present.
2. G. P. Moss. *Biochemical Nomenclature. International Union of Pure and Applied Chemistry and International Union of Biochemistry and Molecular Biology, IUPAC-IUBMB Joint Commission on Biochemical Nomenclature, and Nomenclature Commission of IUBMB Home Page*. Department of Chemistry, Queen Mary and Westfield College, `http://www.chem.qmw.ac.uk/iubmb/`, 1996.
3. Tim Berners-Lee. *Semantic Web Road Map*. W3C, `http://www.w3.org/DesignIssues/Semantic.html`, 1998.
4. Patrick Hayes, editor. *RDF Semantics*. W3C, `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`, 2004.
5. BioMOBY.org. *Moby*. BioMOBY.org, `http://www.biomoby.org//`, 2005.
6. Dave Beckett, editor. *RDF/XML Syntax Specification (Revised)*. W3C, `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`, 2004.
7. Dan Brickley and R. V. Guha, editors. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C, `http://www.w3c.org/TR/rdf-schema`, 2004.
8. Frank van Harmelen, Peter F. Patel-Schneider, and Ian Horrocks, editors. *Reference Description of the DAML+OIL Ontology Markup Language (March 2001)*. W3C, `http://www.daml.org/2001/03/reference.html`,

2001.
9. Mike Dean and Guus Schreiber, editors. *OWL Web Ontology Language Reference*. W3C, `http://www.w3.org/TR/owl-ref/`, 2004.
10. Toni Kazic. Representation, reasoning and the intermediary metabolism of *Escherichia coli*. In Trevor N. Mudge, Veljko Milutinovic, and Lawrence Hunter, editors, *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences*, volume 1, pages 853–862, Los Alamitos CA, 1993. IEEE Computer Society Press.
11. Kate Fox. *Watching the English. The Hidden Rules of English Behaviour*. Hodder and Stoughton, London, 2004.
12. Deborah Tannen. *That's Not What I Meant!* Ballantine Books, New York, 1986.
13. Umberto Eco. *Semiotics and the Philosophy of Language*. Indiana University Press, Bloomington IN, 1984.
14. Larry Wos. *Automated Reasoning: Introduction and Applications*. McGraw-Hill Book Company, New York, second edition, 1992.
15. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, 1979.
16. John F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Publishing Co., Reading MA, 1984.
17. Betsy L. Humphreys and Donald A. B. Lindberg. The UMLS project: making the conceptual connection betweeen users and the information they need. *Bull. Med. Libr. Assoc.*, 81:170–177, 1993.
18. Michael Ashburner, C. A. Ball, J. A. Blake, David Botstein, H. Butler, John M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, Suzanna Lewis, J. C. Matese, Jane E. Richardson, M. Ringwald, Gerald M. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.
19. International Union of Biochemistry and Molecular Biology. *Enzyme Nomenclature. Recommendations (1992) of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology*. Academic Press, Inc., London, 1992.
20. Jeremy J. Carroll and Jos De Roo, editors. *OWL Web Ontology Language Test Cases*. W3C, `http://www.w3.org/TR/2004/REC-owl-test-20040210/`, 2004.
21. Peter F. Patel-Schneider and Ian Horrocks, editors. *OWL Web Ontology Language Semantics and Abstract Syntax*. W3C, `http://www.w3.org/TR/-owl-semantics/`, 2004.
22. Toni Kazic. Semiotes — a semantics for sharing. *Bioinformatics*, 16:1129–1144, 2000. Also at `http://www.biocheminfo.org/repository/-semiotes.ps`.
23. Phani Chilukuri and Toni Kazic. *Semantic Interoperability of Heterogeneous Systems*. University of Missouri Bioinformatics Technical Report 2004-01, `http://www.biocheminfo.org/repository/parser.ps`, 2004.
24. Carsten Lutz, editor. *Description Logics*. dl.kr.org, `http://dl.kr.org`, 2005.