# Context and Joins in the Semantic Web

Kenneth Baclawski
College of Computer and Information Science
Northeastern University
Boston, MA 02115
Ken@Baclawski.com

February 7, 2007

**Abstract**

Databases for the life sciences, especially those associated with bioinformatics, are currently very large and complex, and there is every reason to expect that they will continue to increase in size and complexity in the future. Processing such massive databases requires high performance and scalability. Modern distributed relational databases can provide the necessary performance and scalability, but at the cost of using a different logical foundation for the data. While the Semantic Web has been successfully applied to many domains of the life sciences, it has some limitations that could impede further progress in its use, especially with respect to performance and scalability. This position paper discusses two features that would make the Semantic Web more compatible with high performance databases without changing its logical foundations. The two features are a notion of context and the ability to specify joins. This paper discusses these features in more detail as well as suggesting how they could be incorporated into the Semantic Web.

## 1 Context

Databases for the life sciences are currently very large and complex. As a result, they are commonly implemented using relational databases. While

the Semantic Web has been successfully applied to many domains of the life sciences, it differs from relational databases with respect to how it interprets the meaning of the known facts about the world. Databases assume a *closed world:* if a table does not include a record, then the corresponding fact does not hold. In other words, each table represents everything that is true about the mathematical relation which the table implements. The Semantic Web, on the other hand, assumes an *open world:* the absence of a fact cannot be used to make inferences. In other words, the Semantic Web accepts the possibility that what is currently known about a relation might not be complete.

A logical system that assumes an *open world* is said to be *monotonic*, while a logical system that assumes a *closed world* is nonmonotonic. Although the Semantic Web is monotonic, it has a construct which is closed. This is the concept of a *List*. RDF has two mechanisms for specifying collections: Containers and Lists. Containers are open: one can add a new resource to a container at any time. By contrast, lists are closed: one can only add a new resource to a list by retracting some previously asserted statements. In general, it is not possible to determine the number of elements in a container, because one can never be sure that one knows all of the elements. By contrast, one can always determine the number of elements in any list.

The fact that there is a construct in RDF and OWL which is closed, without violating the monotonicity of RDF or of OWL, shows that it is possible to introduce closed constructs to a monotonic logic.

We propose to add a notion of *context* or *situation* to OWL. Situations have been introduced in logic by Barwise[3]. More recently situations have been used for assisting in achieving *situation awareness* when operating complex machinery such as nuclear power plants, ships and aircraft [4], as well as for emergency response teams and military operations. A core ontology for situation awareness has been developed [2, 5].

Semantically, a context is a set of statements. Roughly speaking, a context generalizes the concept of a reified statement which is already part of RDF and OWL to a set of statements. However, the statements in a context are real statements, i.e., they are not reified. Contexts also differ from reified statements syntactically. A context would be specified by giving the sources containing the statements, in the same way that an ontology imports other ontologies. A context would be an instance of a Context class, and its sources would be specified using a list as follows:

```
<owl:Context rdf:ID="microarray-dataset-1">
  <owl:sources parseType="Collection">
    <owl:Source rdf:resource="file://ma-exp1.xml"/>
    <owl:Source rdf:resource="file://ma-exp2.xml"/>
  </owl:sources>
  <owl:reasoning rdf:resource="&owl;DatabaseLogic"/>
</owl:Context>
```

The reasoning property specifies the kind of logic that should be used within the context.

# 2 Joins

The second issue considered by this paper is the increasing complexity of information in the life sciences. RDF and OWL are capable of representing this information, but in complex data structures, closely related entities can be distant from one another in the RDF graph. Both RDF and OWL further exacerbate this problem by restricting to binary relations. In order to synthesize higher-order relations using RDF, it is necessary to reify the relations. As a result, entities that are directly related to one another via a higher-order relation become only indirectly related via a pair of binary relations.

For example, Gene Ontology (GO) associations can involve a series of relationships between the term and an associated reference[1]. Here is an excerpt that illustrates this:

```
<go:term rdf:about="http:...">
  <go:accession>GO:0016209</go:accession>
  <go:name>antioxidant activity</go:name>
  ...
  <go:dbxref rdf:parseType="Resource">
    <go:database_symbol>SP_KW</go:database_symbol>
    <go:reference>Antioxidant</go:reference>
  </go:dbxref>
  ...
```

In this example, the traversal from go:term to a go:reference involves traversing at least two intermediate statements, involving two properties,

and some references require even more traversals. The path from the GO accession number to each the first go:reference requires traversing three edges in the RDF graph. A join that directly associates the GO term with its dbxref references would be defined as follows:

```
<owl:ObjectProperty rdf:ID="dbxref_reference">
  <owl:joinOf parseType="Collection">
    <rdf:Property rdf:about="&go;dbxref"/>
    <rdf:Property rdf:about="&go;reference"/>
  </owl:joinOf>
</owl:ObjectProperty>
```

Using the notation of the OWL Semantics and Abstract Syntax [6], the join of properties $c$ and $d$ is the property $e$ defined as follows:

$$\langle x, y\rangle \in EXT_I(c) \text{ and } \langle y, z\rangle \in EXT_I(d) \text{ implies } \langle x, z\rangle \in EXT_I(e)$$

for any interpretation $I$.

One can join more than two properties, and one can invert properties if necessary. A join that relates each GO accession number with its dbxref references would be defined as follows:

```
<owl:ObjectProperty rdf:ID="accession_dbxref">
  <owl:joinOf parseType="Collection">
    <owl:inverseOf>
      <rdf:Property rdf:about="&go;accession"/>
    </owl:inverseOf>
    <rdf:Property rdf:about="&go;dbxref"/>
    <rdf:Property rdf:about="&go;reference"/>
  </owl:joinOf>
</owl:ObjectProperty>
```

Joins play a fundamental role in rules, and introducing joins to OWL would make it possible to express the most important rules directly in OWL rather than separately in another file. One of the purposes of the Semantic Web is to give a well defined meaning to Web based data. Rules are an important part of the meaning of data and should therefore be part of the ontology rather than an independent construct.

It is easy to show that adding joins to OWL would make it an undecidable language. For this reason joins should only be available in OWL Full which is already undecidable.

# 3 Conclusion

This position paper introduces two features that would make the Semantic Web more suitable for life sciences applications that require high performance and complex data structures. The two features are a notion of context and the ability to specify joins. The proposed features would add important functionality to the Semantic Web without modifying its logical foundations. In particular, contexts would add closed world reasoning without violating monotonicity, and joins would only be added to OWL Full which is already undecidable.

# 4 Acknowledgments

The author would like to acknowledge the support of Jarg Corporation and the Division of Preventive Medicine of Brigham and Women's Hospital, The Harvard Medical School.

# References

[1] M. Ashburner and S. Lewis. On ontologies for biologists: the Gene Ontology–untangling the web. *Novartis Found. Symp.*, 247:66–80, 2002. Discussion 80-83, 84-90, 244-252.

[2] K. Baclawski, M. Kokar, C. Matheus, J. Letkowski, and M. Malczewski. Formalization of situation awareness. In H. Kilov and K. Baclawski, editors, *Practical Foundations of Behavioral Semantics*, pages 25–40. Kluwer Academic, Dordrecht, Netherlands, 2003.

[3] J. Barwise. Scenes and other situations. *J. Philosophy*, 77:369–397, 1981.

[4] M. Endsley and D. Garland. *Situation Awareness, Analysis and Measurement*. Lawrence Erlbaum, Mahwah, NJ, 2000.

[5] C. Matheus, M. Kokar, and K. Baclawski. A core ontology for situation awareness. In *Proc. Sixth Intern. Conf. on Information Fusion FUSION'03*, pages 545–552, July 2003.

[6] P. Patel-Schneider, P. Hayes, and I. Horrocks. OWL web ontology language semantics and abstract syntax, 2004. www.w3.org/TR/owl-semantics/.