BioPAX: An OWL Early Adopter Perspective

W3C Workshop on Semantic Web for Life Sciences
Position Paper

Authors: BioPAX Workgroup (Bader GD, Brauner E, Cary MP, Goldberg R, Hogue C, Karp P, Klein T, Luciano JS, Maltsev N, Marks D, Marland E, Neumann E, Paley S, Pick J, Regev A, Rzhetsky A, Schachter V, Shah I, Zucker J, Sander C)
www.BioPAX.org

INTRODUCTION

The completion of the human genome gives us completeness at the molecular level. For the first time, we have a full list of parts of the cell and we can think about how they fit together to form functional units. Connecting molecules to function requires knowledge representation tools that can capture key biological processes. A useful organizing concept in this context is the pathway, which relates biological molecules to each other and to a specific biological process. Many efforts are underway to map pathways. These efforts have produced almost 150 "pathway" databases that capture the molecular interactions involved in biological processes[a]. Pathway databases typically use their own non-standard data model to represent pathway data, making it a challenge to integrate data from multiple sources.

OBJECTIVES AND SCOPE

The main objective of the BioPAX initiative is to develop a data exchange format for biological pathway data that is flexible, extensible, optionally encapsulated, and compatible with other standards. The scope of BioPAX is all pathways relating to cellular and molecular biology. Initially, BioPAX will focus on metabolic, signal transduction, gene regulatory pathways and genetic interactions, as most existing data fall into one of these four categories.

USE CASES

The primary function of the BioPAX data exchange format will be to facilitate data sharing among pathway databases such as aMAZE[1], BIND[2], DIP[3], EcoCyc[4,5], IntAct[6], KEGG[7], Reactome and WIT[8]. BioPAX could also facilitate the creation of a shared public repository for pathway data. The desire for such a repository was one of the driving forces behind formation of the BioPAX effort. Another intended use of the BioPAX format is to provide a standard format for future pathway databases and the software tools that must access pathway data.

STATUS

Level 1, Version 1 of BioPAX was released in July 2004. Level 2, which will extend the BioPAX format to cover molecular interaction data, is under active development and targeted for release in early 2005. The features of Level 3, which will extend BioPAX to cover signaling pathways, are currently being planned.

POSITION

The OWL specification of BioPAX was originally developed using the GKB Ontology Editor and subsequently using Protégé. The BioPAX team favored OWL for its knowledge representation capabilities, despite the more mature toolsets of XML Schema and UML.

---

[a] http://www.cbio.mskcc.org/prl/index.php

As early adopters of OWL, Protégé and other semantic web technologies in the life sciences/computational biology community, we have found a number of barriers to more widespread adoption. A large part of it is simply that a lot of tools to support these technologies are still immature. But part of it comes from the fact that knowledge representation schemes can range from purely syntactic systems, in which no concern is given to the meaning of the knowledge that is being represented, all the way to purely semantic systems, in which there is no unified form. From our perspective, XML-based tools are better at providing support for one end of the spectrum, and the OWL-based tools are better at providing support for the other. Yet, for practical issues, such as data integration, the sweet spot may be somewhere in between. What we desire are tools that make it easy to specify the right level of representation for any given problem.

OWL (Web Ontology Language)

1. Tool Support

The complexity of OWL necessitates substantial tool support to make it easy to use.

a. Language support: Other languages, in addition to Java must be supported. Many bioinformatics labs, even large ones, only use Perl and Python. Perl OWL parsing tools must be supported for widespread adoption in the life sciences.

b. Utility Software:

i. XML-like tools:

Tools similar to those available for XML Schema are needed. Before these tools are implemented, very robust automatic conversion to XML Schema would be useful.

1. XML Spy like editing, visualization
2. Automatic object to relational mapping (e.g. Apache OJB)
3. Automatic object code generation in multiple languages, but at least Java, combined with OWL I/O functionality that is specific to an OWL defined ontology without the instance data. (Similar to Sun's JAXB or Apache's Castor).
4. XML Schema-like data validators (Validator Light vs. Validator Full) for users who do not use reasoning technology. E.g. validation of strings and validation of domain constraints on properties.

ii. `diff` functionality: `diff`ing an OWL file is essential when building an ontology. This is because the current editing tools are still under development and contain bugs and may change the OWL output format produced by different versions of the editor. It is helpful to `diff` to see what changes have been made. Currently we `diff` by saving to n-triple format, loading the output into spreadsheet, sorting and manually `diff`ing. This is messy and time-consuming.

c. Reasoning:

Better reasoning over instance data or availability of a query language similar to what biologists are used to such as SQL. E.g. find all pathways that contain a gene of interest in a given BioPAX OWL document or in the various BioPAX OWL documents that can be found on the web.

d. Better documentation

i. Better automatic documentation generation (like that of GKB or XML Spy)

ii. Expanded tutorials on use of tools for OWL similar in scope to those available for open source Java programming e.g. Wiley's *Open Source Java Programming* (ISBN: 0471463620).

e. Clear best practices
   i. Namespace: Namespace options are not clearly defined. When should a new namespace be defined? Who should define it in the following contexts?
      1. BioPAX documents created by databases for the purpose of distributing their data
      2. BioPAX documents created by databases via a web service in response to a user query
      3. BioPAX documents created by users for posting on their website
   ii. RDF ID: How should RDF IDs be chosen by users in the above situations? The idea of using globally unique RDF IDs currently imposes too much overhead on our users. We currently recommend that RDF IDs are only unique within a document, don't have to be globally unique and can change from document to document. Should LSIDs be used directly as RDF IDs or are better placed as data elements in a property value? We realize this doesn't completely support the semantic web, but we feel that we are forced to do things this way until the technology matures. Later someone can build an agent for BioPAX documents to convert our external references that we store as data to RDF IDs and synonym tables in the context of a semantic web application.
   iii. When and how to use various OWL and RDFS specific tags (e.g. ontology description, version information)

f. More examples. We were unable to find examples of the use of important OWL constructs e.g.
   i. Use of `oneOf` in a property restriction. Additionally, we were unable to decide if this OWL constructs is OWL DL or OWL Full, since different OWL validators disagreed.
   ii. Use of instances and reasoning over instance data
   iii. Validator consistency: validators disagree sometimes on OWL Full vs. OWL DL and report different errors and warnings.
   iv. Expanded tutorials considering reasoners and instance data. A list of recommended/best of breed/approved tools.

2. Easier use of specific features in OWL

a. Controlled vocabularies, either flat, hierarchical or DAG-like (directed acyclic graph), like the Gene Ontology (GO)[9], are commonly used in Bioinformatics applications. Flat controlled vocabularies map to `oneOf` lists of string literals in OWL. DAG-like ontologies like GO partly map to the value partition design pattern. The problem with the value partition is that it creates high overhead, since many classes are created, sometimes more than the number of classes in the main ontology. These classes are never expected to have properties, so why make classes? Classes require the user to create an instance to be used, which is more difficult than using literals. It would be helpful to have a construct that addressed this usability issue.

b. Aliases: It would be helpful to have aliases on properties so that they can be named differently at different levels of the class hierarchy. This issue crops up numerous times in biology. For instance, as classes become more specific down the class

hierarchy, names of properties should become more specific as well, since biology usually has generally accepted names for these more specific properties. Specifically, in BioPAX, a `Control` class is present with a subclass called `Conversion`. In `Control`, the property that describes the concept of a controller is called `CONTROLLER`. A biologically recognized name for this in the `Conversion` subclass is `ENZYME`, but we have to maintain `CONTROLLER`. There should be an aliasing feature that is natural to use in these cases. `Rdfs:label` may be the right tool, but it is not clear and current tools don't seem to make good use of it.

LSID (Life Science Identifier)

BioPAX recommends the use of `LSID`s as external references for biological entities such as proteins in addition to the database external references commonly used.

1. LSID requires the following services to be useful

a. A set of universal and up-to-date web services for LSID resolution and synonym matching for all existing databases, much like DNS servers.

b. Support for synonyms: Realization that there are many valid synonyms for many biological concepts and availability of a web service that, given an LSID, will provide a list of known synonymous LSIDs. This would solve a major problem in bioinformatics and would drive adoption of LSIDs. This information could be made available on the semantic web using the OWL `sameAs` element in various OWL documents.

As early adopters, we wish to help make the semantic web technologies more useful in the life sciences community by relaying our experiences, reporting bugs, and suggesting features.

1. Lemer, C. et al. The aMAZE LightBench: a web interface to a relational database of cellular processes. *Nucleic Acids Res* **32 Database issue**, D443-8 (2004).
2. Bader, G. D., Betel, D. & Hogue, C. W. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res.* **31**, 248-250 (2003).
3. Xenarios, I. et al. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **30**, 303-305 (2002).
4. Karp, P. D. et al. The EcoCyc Database. *Nucleic Acids Res.* **30**, 56-58 (2002).
5. Karp, P. D., Riley, M., Paley, S. M. & Pellegrini-Toole, A. The MetaCyc Database. **30**, 59-61 (2002).
6. Hermjakob, H. et al. IntAct: an open source molecular interaction database. *Nucleic Acids Res* **32**, D452-5 (2004).
7. Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y. & Hattori, M. The KEGG resource for deciphering the genome. *Nucleic Acids Res* **32 Database issue**, D277-80 (2004).
8. Overbeek, R. et al. WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. **28**, 123-125 (2000).
9. The_Gene_Ontology_Consortium. Gene ontology: tool for the unification of biology. **25**, 25-29 (2000).