

BioPAX – Biological Pathways Exchange Language

Level 2, Version 0.92 (Release Candidate 2)

Documentation

BioPAX Release Candidate Recommendation, August 22, 2005

The BioPAX data exchange format is the joint work of the BioPAX workgroup: Mirit Aladjem, Gary D. Bader, Eric Brauner, Michael P. Cary, Kam Dahlquist, Emek Demir, Peter D'Eustachio, Ken Fukuda, Frank Gibbons, Marc Gillespie, Robert Goldberg, Chris Hogue, Michael Hucka, Geeta Joshi-Tope, David Kane, Peter Karp, Teri Klein, Christian Lemer, Joanne Luciano, Elgar Pichler, Debbie Marks, Natalia Maltsev, Elizabeth Marland, Eric Neumann, Suzanne Paley, John Pick, Jonathan Rees, Aviv Regev, Alan Ruttenberg, Andrey Rzhetsky, Chris Sander, Vincent Schachter, Imran Shah, Andrea Splendiani, Mustafa Syed, Edgar Wingender, Guanming Wu, Jeremy Zucker

This document edited by Gary D. Bader and Michael P. Cary

Copyright © 2005 BioPAX Workgroup. Some rights reserved under the Creative Commons License (<http://creativecommons.org/licenses/by/2.0/>)

Abstract

At present, there are over 170 Internet-accessible databases that store biological pathway data. Biologists often need to use information from many of these to support their research, but since each has its own representation conventions and data access methods, integrating data from multiple databases is very difficult. A widely-adopted biological pathway data exchange format will help.

BioPAX (Biological Pathway Exchange - <http://www.biopax.org>) enables the integration of diverse pathway resources by defining an open file format specification for the exchange of biological pathway data. By utilizing the BioPAX format, the problem of data integration reduces to a semantic mapping between the data models of each resource and the data model defined by BioPAX. Widespread adoption of BioPAX for data exchange will increase access to and uniformity of pathway data from varied sources, thus increasing the efficiency of computational pathway research.

This document describes BioPAX Level 2, which expands the scope of BioPAX to include representation of molecular binding interactions, protein post-translational modifications, basic experimental descriptions, and hierarchical pathways. Adding coverage of these features will allow BioPAX to represent the bulk of the data in the PSI-MI Level 2 format

(<http://psidev.sourceforge.net/mi/rel2/doc/>) and lays the groundwork for better support of signal transduction and molecular states.

Scope of this document

This BioPAX documentation is targeted at bioinformaticians with an interest in biological pathway data. Those who are only interested in an overview of BioPAX are encouraged to read the introduction (section 1). It is expected that readers are familiar with one or more pathway databases and have a basic understanding of both bioinformatics and molecular and cellular biology. This background information is available in a number of textbooks^{1,2}.

This document provides an overview the BioPAX Level 2 ontology. This includes descriptions of the BioPAX ontology classes, sample use cases and best practice recommendations. This document does not provide a full definition of the BioPAX Level 2 ontology, which is given by the BioPAX Level 2, Version 0.9 OWL file, located here:

<http://www.biopax.org/Downloads/Level2v0.9/biopax-level2.owl>

New Features in BioPAX Level 2

This section provides a description of the additional features found in BioPAX Level 2 and the rationale for each.

Molecular Binding Interactions

As large amounts of molecular interaction data are being produced from proteomics experiments, early BioPAX development discussions identified the importance of representing molecular interaction data and it was prioritized for Level 2 inclusion.

Unlike most metabolic pathway data, which tends to represent interactions in a high level of detail, molecular interaction datasets rarely capture causal or temporal aspects of interactions. As a result, molecular binding interactions are often considered a “low resolution” form of pathway data. BioPAX Level 2 captures molecular binding interactions at a relatively high level in the ontology class hierarchy, reflecting the fact that any given binding interaction may be a low-resolution, or more abstract, view of a more specific type of interaction.

For example, a signaling database would likely capture the interaction between MEK1 and ERK1 as a catalysis event (MEK1 catalyzes the phosphorylation of ERK1). A molecular interaction database, on the other hand, would likely store the interaction using a simpler abstraction, such as a binary protein-protein interaction. BioPAX Level 2 supports both of these representations. Furthermore, since the catalysis class is a descendent of the physical interaction class (in which molecular binding interactions are stored), BioPAX Level 2 also implicitly captures the fact that the catalysis instance involving MEK1 and ERK1 may be a more specific representation of the corresponding physical interaction instance.

Sequence Features

BioPAX Level 2 adopts the mechanism used by the PSI-MI format to represent sequence features. Any protein, RNA, or DNA participant may contain a sequence feature. An example sequence feature for a protein is a phosphorylation site or other post-translational modification. It is recommended that users define separate participant instances for each interaction and, within each participant, define only those sequence features that are relevant to the interaction at hand.

Evidence

Since many molecular interactions in existing databases are derived from experiments with high false-positive rates, it is important to be able to capture the experimental evidence supporting these interactions. This is not as important for metabolic pathways, which have historically been better studied. The PSI-MI format allows detailed descriptions of experiments, which may be associated with one or more interactions. BioPAX Level 2 directly use the PSI-MI evidence data model and expands it to allow evidence codes, such as those developed by GO or BioCyc, to be attached as evidence for interactions and pathways.

Hierarchical Pathways

A pathway can be composed of another pathway (including itself). For example, the cell cycle is often decomposed into stages that are each considered their own pathway. To allow hierarchical pathways and recursive pathways, BioPAX Level 2 expands the pathway class to allow participation of pathways as well as interactions.

Miscellaneous

Utility class organization

With the increased number of utility classes in Level 2, a number of new organizational classes in the utility class tree were created to partition the utility class hierarchy into more easily navigable subdivisions.

InChI

BioPAX Level 2 allows small molecule structures to be represented using the new IUPAC-NIST Chemical Identifier (InChI) format. This format is used to describe small molecules by NCBI's PubChem resource, among others.

Representation Styles Supported in BioPAX Level 2

Different pathway representation styles are in common use for different types of pathway information. Each style is tailored to make representation of the specific type of pathway data easier. This section details the representation styles supported in BioPAX Level 2 and mentions representation styles that will be supported in the future.

Metabolic pathways

Metabolic pathways mostly involve biochemical reactions where protein enzymes convert small molecule reactants to small molecule products. While there are many exceptions to this general statement, the majority of metabolic pathway data in databases is covered. BioPAX Level 1 introduced support for this pathway data type.

Proteomics style molecular interactions

Molecular interactions typically present in proteomics and functional genomics databases involve mainly binary and set interactions between proteins (protein-protein interactions), DNA (protein-DNA interactions) and sometimes other molecules. Experimental description is important for this pathway data type, but otherwise the molecular interactions are known at a low level of detail. BioPAX Level 2 introduces support for this pathway data type.

Future Supported Styles

Signaling pathways mostly involve cascades of protein and other molecule chemical modifications to implement information transfer across the cell. An important difference between these pathways and metabolic or proteomics data is the central role of protein post-translational modifications (and other molecular modifications) that are not stored in any primary database. Chemically modified proteins are created and reused often in these pathways and must be tracked and reused within the ontology. This is not the case for metabolic pathways, where all important small molecule modifications are tracked in small molecule databases. This is also not the case for proteomics data, since modified proteins can be described, but are seldom reused between interactions (interactions are usually treated independently outside the context of a pathway). BioPAX Level 2 adds some support for this representation by allowing post-translationally modified proteins to be described (using PSI-MI data structures) in the `physicalEntityParticipant` class, but this class cannot be reused between interactions. This means that tracking interaction participants requires extra work. For example, creating a network representation of a signaling pathway, will involve comparing participant classes for equality. BioPAX Level 3 will better support for signaling pathways, with the availability of a state class that will allow participants to be efficiently reused between interactions and will eventually allow more complex states to be efficiently described (another important issue for signaling pathways).

Key definitions

BioPAX workgroup: Community group designing the BioPAX ontology and format.

BioPAX ontology: The abstract representation of biological pathway concepts and their relationships developed by the BioPAX workgroup. This is also called the object model.

BioPAX format: The file format implementation of the BioPAX ontology that defines the syntax of representation for data. The BioPAX format is currently implemented only in OWL, but other implementations, such as XML Schema may be developed in the future.

OWL: Web Ontology Language. OWL is an XML-based language defined by the World Wide Web Consortium (see <http://www.w3.org/TR/owl-guide/>). OWL can be used to both define an ontology and to store instance data that adheres to that ontology. It is intended that the BioPAX ontology is used to validate that a set of instances follows all BioPAX defined syntax rules. It is recommended that the BioPAX ontology be imported from its location on the biopax.org website, although it may also be defined directly within an instance data document.

Status of this document

This document is under review by BioPAX workgroup members and interested third parties. Comments on this specification may be sent to biopax-discuss@biopax.org; archives of the comments are available by subscribing to our mailing list here: <http://www.biopax.org/mailman/private/biopax-discuss/>.

This document and the BioPAX Level 2 OWL file will be updated over time, based on community input. The documentation for the latest version of BioPAX Level 2 version 1.x can always be found here: <http://www.biopax.org/release/biopax-level2-documentation.pdf>

Document changes since the previous version

Previous version of this document: <http://www.biopax.org/Downloads/Level2v0.5/biopax-level2-documentation.pdf>

The previous version of this document was an abbreviated description of BioPAX Level 2; it only described features that were new in Level 2 and referred readers to the Level 1 documentation for more information. The current version of this document provides a complete, stand-alone description of BioPAX Level 2.

Related documents

BioPAX Level 2, Version 0.9 OWL file:
<http://www.biopax.org/Downloads/Level2v0.9/biopax-level2.owl>
BioPAX Level 2, Version 0.9 HTML file:
<http://www.biopax.org/Downloads/Level2v0.9/biopax-level2.html>

BioPAX Namespace

The following URI is defined to be the BioPAX Level 2, version 1.x namespace:

<http://www.biopax.org/release/biopax-level2.owl#>

This namespace name (URI) will always be used to refer to the most recently released 1.x version of BioPAX; different URIs will be used for any and all other major versions of BioPAX Level 1 (e.g. versions 2.x, 3.x, etc.) See *Appendix B* for an explanation of the BioPAX release cycle and level and version numbers.

Table of contents

<i>BioPAX – Biological Pathways Exchange Language</i>	<i>1</i>
<i>Level 2, Version 0.9 (Draft Release) Documentation</i>	<i>1</i>
Abstract	1
Scope of this document	2
New Features in BioPAX Level 2	2
Molecular Binding Interactions	2
Hierarchical Pathways	3
Miscellaneous	3
Key definitions	4
Status of this document	5
Document changes since the previous version	5
Related documents	5
BioPAX Namespace	5
Table of contents	6
1 Introduction	9
How to Participate	9
2 BioPAX Ontology Class Structure	11
Top level entity classes	11
Entity (Root class of ontology)	12
Second level classes	13
Pathway	13
Interaction	14
Physical Entity	15
Interaction subclasses	15
Physical Interaction	16
Physical Interaction subclasses	17
Control	17
Conversion	19
Control subclasses	20
Catalysis	21
Modulation	22
Conversion subclasses	23
Biochemical Reaction	23
Complex Assembly	25
Transport	26
Transport with Biochemical Reaction	27
Summary of Interaction Class Structure	28
Physical Entity subclasses	28
DNA	28
RNA	29

Protein	30
Small Molecule	30
Complex	31
Utility classes	32
Top level utility classes	32
Chemical Structure	32
Confidence	33
Evidence	34
Experimental Form	34
External Reference Utility Class	35
Pathway Step	35
Physical Entity Participant	36
Sequence Feature	37
Sequence Location	37
External Reference Utility Class subclasses	38
BioSource	38
DataSource	38
Open Controlled Vocabulary	39
Xref	39
Xref subclasses	40
Publication Xref	40
Relationship Xref	41
Unification Xref	41
Physical Entity Participant subclasses	42
Complex Participant	42
DNA Participant	Error! Bookmark not defined.
Protein Participant	Error! Bookmark not defined.
RNA Participant	Error! Bookmark not defined.
Small Molecule Participant	Error! Bookmark not defined.
Sequence Location subclasses	43
Sequence Interval	43
Sequence Site	43
Summary of BioPAX Class Structure	44
3 Examples	45
4 Best Practices	46
Referencing External Objects	46
Using xrefs	46
Importance of unification xrefs	47
Using external controlled vocabulary terms	47
Reusing utility class instances	48
Pathways and Networks	48
Metabolic pathways	49
Interaction networks	49
Conversion Direction	49
Conventions for LEFT and RIGHT	50
Technical note: OWL and RDF Conventions	50
RDF ID	50

Document namespace	51
BioPAX and the Semantic Web	52
5 HOW-TO	54
Creating a knowledge-base using BioPAX and Protégé	54
Viewing Instances Graphically	55
6 Use Case Outlines	56
Data Sharing Between Databases	56
BioPAX as a knowledge-base (KB) Model	57
Pathway Data Warehouse	57
Pathway Analysis Software	57
Pathway Analysis Software Example: Molecular profiling analysis	57
Visualizing Pathway Diagrams	58
Pathway Modeling	58
Using BioPAX as metadata for SBML and CellML	58
Pathway analysis using logical inference	58
7 Glossary	60
Appendix B: Level and Version Numbers	62
Acknowledgements	62
References	62

1 Introduction

BioPAX (Biological Pathway Exchange) aims to facilitate the integration and exchange of data maintained in biological pathway databases. Traditionally, integrating data from a number of databases, diverse in form and content, has been a challenge in the field of Bioinformatics³. One solution is to define a mutually agreed upon file format as a standard way of representing a given type of data in a community. An example of such a standard is the DDBJ/EMBL/GenBank flat-file format, used to represent nucleic acid sequence data.

Currently, there is no file format standard broadly applicable to biological pathway data, despite the presence of this data in over 170 different internet accessible databases³. While previous work has been done to standardize specific types of pathway data, notably, the protein-protein interaction database community has developed a format called PSI-MI⁴, there is no format capable of representing all of the most frequently used types of pathway data. **The goal of the BioPAX project is to provide a data exchange format for pathway data that will represent the key elements of the data models from a wide range of popular pathway databases.** To achieve this goal, the BioPAX ontology was designed to support the data models of a number of existing pathway databases, such as [BioCyc](#)^{5,6}, [BIND](#)⁷, [WIT](#)⁸, [PATIKA](#)⁹, [Reactome](#), [aMAZE](#)¹⁰, [KEGG](#)¹¹ and others. When designing the BioPAX ontology, the BioPAX workgroup endeavored to balance the many different representational needs of these and other biological pathway databases by adhering to design principles that promote interoperability. These design principles include flexibility, extensibility, optional encapsulation of frequently used external data, compatibility with other standards and computability (see Appendix 1: Design Principles).

Because pathway data are complex and can be represented at many levels of detail, the BioPAX group is using a leveled development approach, similar to that of SBML¹². While the overall framework of the BioPAX ontology, i.e. the root class structure, has been designed with the entire pathway data space in mind, representation of specific types of pathway data are the focus of individual levels. **BioPAX Level 1 was designed to represent metabolic pathway data.** Representing other types of pathway data with BioPAX Level 1 is possible but may not be optimal. **BioPAX Level 2 expands the scope of Level 1 to include representation of molecular binding interactions and hierarchical pathways.** Future levels will enhance coverage for additional types of pathway data, such as signal transduction pathways and genetic regulatory networks; see the BioPAX Roadmap, located here:

http://www.biopax.org/Docs/BioPAX_Roadmap.html

How to Participate

Since a data exchange format is only useful if it is widely adopted, the BioPAX project aims to promote the use of the format by as many data sources as possible. This is achieved partly

* <http://www.cbio.mskcc.org/prl>

through community outreach at conferences and workshops, and partly through active participation in the project by data providers and consumers.

There are many ways to participate in BioPAX development: one can participate directly in its design, provide feedback to the BioPAX group, provide data in the BioPAX format, develop software tools that support the BioPAX format, provide sponsorship for BioPAX activities, and encourage participation by others.

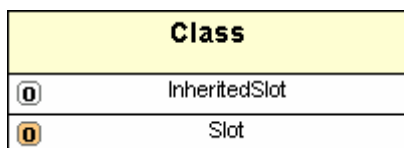
BioPAX participation is currently on a volunteer basis and members have typically paid their own expenses. The US Department of Energy (DOE) has provided some funding for holding meetings and will support additional meetings in the future.

More details are available on the www.biopax.org web site.

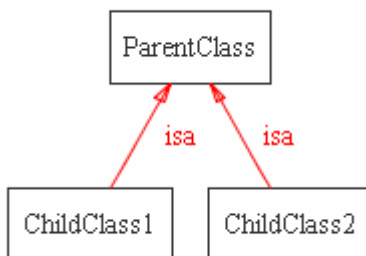
2 BioPAX Ontology Class Structure

This section provides an overview of the BioPAX Level 2 class structure. Full definitions are found in the BioPAX Level 2 OWL document (<http://www.biopax.org/release/biopax-level2.owl>). Text definitions of classes are provided along with synonyms, comments and examples, where possible, to help the reader understand the definition and intended use of each class.

Additionally, classes are shown graphically using the ezOWL Protégé plugin (see HOW-TO section below). Classes are shown in a sub-divided box, with the name of the class in the pale top yellow box and the properties of the class in white boxes below. A white **O** denotes an inherited property and an orange **O** denotes a property defined in this class.



Interspersed throughout this section are diagrams generated by the Ontoviz Protégé plugin that show the parent/child relationships between selected classes of the BioPAX ontology. In these diagrams, red “isa” arrows connect child classes to their parents.



Top level entity classes

The BioPAX ontology defines 4 basic concepts in the ontology: the root level **entity** class and three subclasses: **pathway**, **interaction** and **physicalEntity**.

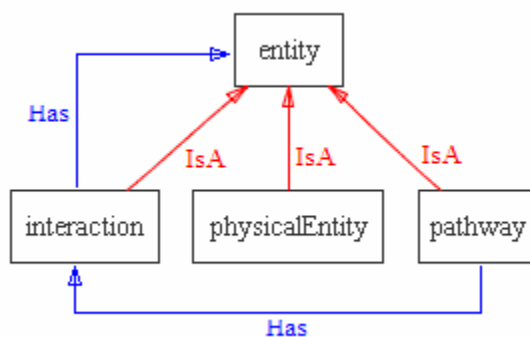


Table 1: Analogies of the root BioPAX ontology structure (first and second level classes) to other conceptual areas.			
	Linguistic	Graph representation	Pathway shorthand
Entity	Noun (Subject or Object)	Node	A, B, C
Relationship	Verb	Edge	\rightarrow , \rightarrow
Interaction	Phrase/Sentence	Either a node set by itself (member of a set relationship) or a node set connected to another node set by an edge (relationship between sets)	$A \rightarrow B$, $B \rightarrow C$
Pathway	Paragraph	Graph	$A \rightarrow B \rightarrow C$

Entity (Root class of ontology)

Definition: A discrete biological unit used when describing pathways.

Comment: This is the root class for all biological concepts in the ontology, which include pathways, interactions and physical entities. As the most abstract class in the ontology, instances of the entity class should be created rarely, if ever.

Synonyms: thing, object, bioentity.

Properties:

AVAILABILITY - Describes the availability of this data (e.g. a copyright statement).

COMMENT - Comment on the data in the container class. This property should be used instead of the OWL documentation elements (rdfs:comment) for instances because information in COMMENT is data to be exchanged, whereas the rdfs:comment field is used for metadata about the structure of the BioPAX ontology.

DATA-SOURCE - A free text description of the source of this data, e.g. a database or person name. This property should be used to describe the source of the data. This is meant to be used by databases that export their data to the BioPAX format or by systems that are integrating data from multiple sources. The granularity of use (specifying the data source in many or few instances) is up to the user. It is intended that this property report the last data source, not all data sources that the data has passed through from creation.

NAME - The preferred full name for this entity.

SHORT-NAME - An abbreviated name for this entity, preferably a name that is short enough to be used in a visualization application to label a graphical element that represents this entity. If no short name is available, an xref may be used for this purpose by the visualization application.

SYNONYMS - One or more synonyms for the name of this entity. This should include the values of the NAME and SHORT-NAME property so that it is easy to find all known names in one place.

XREF - Values of this property define external cross-references from this entity to entities in external databases.

entity	
0	SYNONYMS
0	COMMENT
0	DATA-SOURCE
0	SHORT-NAME
0	AVAILABILITY
0	NAME
0	XREF

Second level classes

Pathway

Definition: A set or series of interactions, often forming a network, which biologists have found useful to group together for organizational, historic, biophysical or other reasons.

Comment: It is possible to define a pathway without specifying the interactions within the pathway. In this case, the pathway instance could consist simply of a name and could be treated as a 'black box'.

Synonyms: network

Examples: glycolysis, valine biosynthesis

Properties:

EVIDENCE - Scientific evidence supporting the existence of the entity as described.

ORGANISM - An organism, e.g. 'Homo sapiens'. This is the organism that the pathway is found in. A pathway may not have an organism associated with it, for instance, reference pathways from KEGG.

PATHWAY-COMPONENTS – The set of interactions and/or pathwaySteps in this pathway/network. Each instance of the pathwayStep class defines: 1) a set of interactions that together define a particular step in the pathway, for example a catalysis instance and the conversion that it catalyzes; 2) an order relationship to one or more other pathway steps (via the NEXT-STEP property). Note: This ordering is not necessarily temporal - the order described may simply represent connectivity between adjacent steps. Temporal ordering information should only be inferred from the direction of each interaction (see section on biochemical reaction direction in Section 4).

pathway	
<input type="radio"/>	NAME
<input type="radio"/>	SHORT-NAME
<input type="radio"/>	SYNONYMS
<input type="radio"/>	DATA-SOURCE
<input type="radio"/>	XREF
<input type="radio"/>	AVAILABILITY
<input type="radio"/>	COMMENT
<input checked="" type="radio"/>	ORGANISM
<input checked="" type="radio"/>	EVIDENCE
<input checked="" type="radio"/>	PATHWAY-COMPONENTS

Interaction

Definition: A single biological relationship between two or more entities. An interaction cannot be defined without the entities it relates.

Comment: Since it is a highly abstract class in the ontology, instances of the interaction class should never be created. Instead, more specific classes should be used. Currently this class only has subclasses that define physical interactions; later levels of BioPAX may define other types of interactions, such as genetic (e.g. synthetic lethal).

Naming rationale: A number of names were considered for this concept, including “process”, “synthesis” and “relationship”; Interaction was chosen as it is understood by biologists in a biological context and is compatible with [PSI-MI](#).

Examples: protein-protein interaction, biochemical reaction, enzyme catalysis

Properties:

EVIDENCE - Scientific evidence supporting the existence of the entity as described.

PARTICIPANTS - This property lists the entities that participate in this interaction. For example, in a biochemical reaction, the participants are the union of the reactants and the products of the reaction. This property has a number of sub-properties, such as LEFT and RIGHT in the biochemicalInteraction class. Any participant listed in a sub-property will automatically be assumed to also be in PARTICIPANTS by a number of software systems, including Protégé, so this property should not contain any instances if there are instances contained in a sub-property.

interaction	
<input type="radio"/>	NAME
<input type="radio"/>	SHORT-NAME
<input type="radio"/>	SYNONYMS
<input type="radio"/>	DATA-SOURCE
<input type="radio"/>	XREF
<input type="radio"/>	AVAILABILITY
<input type="radio"/>	COMMENT
<input checked="" type="radio"/>	PARTICIPANTS
<input checked="" type="radio"/>	EVIDENCE

Physical Entity

Definition: An entity with a physical structure.

Comment: This class serves as the super-class for all physical entities, although its current set of subclasses is limited to molecules. This list may be expanded to include photon, environment, cell and cellular component in later levels of BioPAX, depending on community need. As a highly abstract class in the ontology, instances of the physicalEntity class should be created rarely, if ever.

Synonyms: part, interactor, object

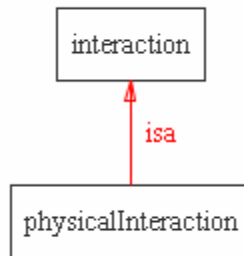
Naming rationale: It's difficult to find a name that encompasses all of the subclasses of this class without being too general. E.g. PSI-MI uses 'interactor', BIND uses 'object', BioCyc uses 'chemicals'. physicalEntity seems to be a good name for this specialization of entity.

Examples: protein, small molecule, RNA

physicalEntity	
<input type="radio"/>	SYNONYMS
<input type="radio"/>	COMMENT
<input type="radio"/>	DATA-SOURCE
<input type="radio"/>	SHORT-NAME
<input type="radio"/>	AVAILABILITY
<input type="radio"/>	NAME
<input type="radio"/>	XREF

Interaction subclasses

The interaction class has one subclass in BioPAX Level 2: physicalInteraction. In future BioPAX levels, additional subclasses will likely be added, such as a class to store genetic interactions (see the BioPAX Roadmap).



Physical Interaction

Definition: An interaction in which at least one participant is a physical entity, e.g. a binding event.

Comment: This class should be used by default for representing molecular interactions, such as those defined by PSI-MI level 2. The participants in a molecular interaction should be listed in the PARTICIPANTS slot. Note that this is one of the few cases in which the PARTICIPANT slot should be directly populated with instances (see comments on the PARTICIPANTS property in the interaction class description). If sufficient information on the nature of a molecular interaction is available, a more specific BioPAX interaction class should be used.

Example: Two proteins observed to interact in a yeast-two-hybrid experiment where there is not enough experimental evidence to suggest that the proteins are forming a complex by themselves without any indirect involvement of other proteins. This is the case for most large-scale yeast two-hybrid screens.

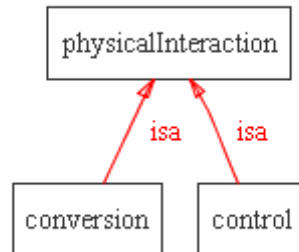
Properties:

INTERACTION-TYPE - External controlled vocabulary characterizing the interaction type, for example "phosphorylation". See the controlled vocabulary section in Section 4 for more detail.

physicalInteraction	
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	INTERACTION-TYPE

Physical Interaction subclasses

Two classes exist under physicalInteraction: control and conversion.



Control

Definition: An interaction in which one entity regulates, modifies, or otherwise influences another. Two types of control interactions are defined: activation and inhibition.

Comment: In general, the targets of control processes (i.e. occupants of the CONTROLLED property) should be interactions. Conceptually, entities are involved in interactions (or events) and the events should be controlled or modified, not the entities themselves. For example, a kinase activating a protein is a frequent event in signaling pathways and is usually represented as an ‘activation’ arrow from the kinase to the substrate in signaling diagrams. This is an abstraction that can be ambiguous out of context. In BioPAX, this information should be captured as the kinase catalyzing (via an instance of the catalysis class) a reaction in which the substrate is phosphorylated, instead of as a control interaction in which the kinase activates the substrate. Since this class is a superclass for specific types of control, instances of the control class should only be created when none of its subclasses are applicable.

Synonyms: regulation, mediation

Examples: A small molecule that inhibits a pathway by an unknown mechanism controls the pathway.

Properties:

CONTROL-TYPE - Defines the nature of the control relationship between the CONTROLLER and the CONTROLLED entities.

The following terms are possible values of the CONTROL-TYPE property:

ACTIVATION: General activation

The following term can not be used in the catalysis class:

INHIBITION: General inhibition

The following terms can only be used in the modulation class (these definitions from EcoCyc):

INHIBITION-ALLOSTERIC

Allosteric inhibitors decrease the specified enzyme activity by binding reversibly to the enzyme and inducing a conformational change that decreases the affinity of the enzyme to its substrates without affecting its V_{MAX} . Allosteric inhibitors can be competitive or noncompetitive inhibitors, therefore, those inhibition categories can be used in conjunction with this category.

INHIBITION-COMPETITIVE

Competitive inhibitors are compounds that competitively inhibit the specified enzyme activity by binding reversibly to the enzyme and preventing the substrate from binding. Binding of the inhibitor and substrate are mutually exclusive because it is assumed that the inhibitor and substrate can both bind only to the free enzyme. A competitive inhibitor can either bind to the active site of the enzyme, directly excluding the substrate from binding there, or it can bind to another site on the enzyme, altering the conformation of the enzyme such that the substrate can not bind to the active site.

INHIBITION-IRREVERSIBLE

Irreversible inhibitors are compounds that irreversibly inhibit the specified enzyme activity by binding to the enzyme and dissociating so slowly that it is considered irreversible. For example, alkylating agents, such as iodoacetamide, irreversibly inhibit the catalytic activity of some enzymes by modifying cysteine side chains.

INHIBITION-NONCOMPETITIVE

Noncompetitive inhibitors are compounds that noncompetitively inhibit the specified enzyme by binding reversibly to both the free enzyme and to the enzyme-substrate complex. The inhibitor and substrate may be bound to the enzyme simultaneously and do not exclude each other. However, only the enzyme-substrate complex (not the enzyme-substrate-inhibitor complex) is catalytically active.

INHIBITION-OTHER

Compounds that inhibit the specified enzyme activity by a mechanism that has been characterized, but that cannot be clearly classified as irreversible, competitive, noncompetitive, uncompetitive, or allosteric.

INHIBITION-UNCOMPETITIVE

Uncompetitive inhibitors are compounds that uncompetitively inhibit the specified enzyme activity by binding reversibly to the enzyme-substrate complex but not to the enzyme alone.

INHIBITION-UNKMECH

Compounds that inhibit the specified enzyme activity by an unknown mechanism. The mechanism is defined as unknown, because either the mechanism has yet to be elucidated in the experimental literature, or the paper(s) curated thus far do not define the mechanism, and a full literature search has yet to be performed.

ACTIVATION-UNKMECH

Compounds that activate the specified enzyme activity by an unknown mechanism. The mechanism is defined as unknown, because either the mechanism has yet to be elucidated in the experimental literature, or the paper(s) curated thus far do not define the mechanism, and a full literature search has yet to be performed.

ACTIVATION-NONALLOSTERIC

Nonallosteric activators increase the specified enzyme activity by means other than allosteric.

ACTIVATION-ALLOSTERIC

Allosteric activators increase the specified enzyme activity by binding reversibly to the enzyme and inducing a conformational change that increases the affinity of the enzyme to its substrates without affecting its VMAX.

CONTROLLED - The entity that is controlled, e.g., in a biochemical reaction, the reaction is controlled by an enzyme. **CONTROLLED** is a sub-property of **PARTICIPANTS**.

CONTROLLER - The controlling entity, e.g., in a biochemical reaction, an enzyme is the controlling entity of the reaction. **CONTROLLER** is a sub-property of **PARTICIPANTS**.

control	
<input type="checkbox"/>	INTERACTION-TYPE
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	CONTROL-TYPE
<input checked="" type="checkbox"/>	CONTROLLER
<input checked="" type="checkbox"/>	CONTROLLED

Conversion

Definition: An interaction in which one or more entities is physically transformed into one or more other entities.

Comment: This class is designed to represent a simple, single-step transformation. Multi-step transformations, such as the conversion of glucose to pyruvate in the glycolysis pathway, should be represented as pathways, if known. Since it is a highly abstract class in the ontology, instances of the conversion class should never be created.

Examples: A biochemical reaction converts substrates to products, the process of complex assembly converts single molecules to a complex, transport converts entities in one compartment to the same entities in another compartment.

Properties:

LEFT - The participants on the left side of the conversion interaction. Since conversion interactions may proceed in either the left-to-right or right-to-left direction, occupants of the LEFT property may be either reactants or products. LEFT is a sub-property of PARTICIPANTS.

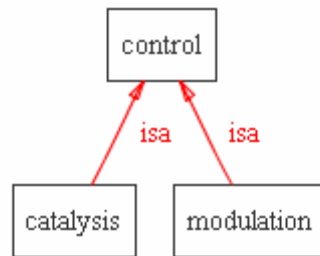
RIGHT - The participants on the right side of the conversion interaction. Since conversion interactions may proceed in either the left-to-right or right-to-left direction, occupants of the RIGHT property may be either reactants or products. RIGHT is a sub-property of PARTICIPANTS.

SPONTANEOUS - Specifies whether a conversion occurs spontaneously (i.e. uncatalyzed, under biological conditions) left-to-right, right-to-left, or not at all. If a conversion is not spontaneous, or if the spontaneity is not known, the SPONTANEOUS property should be left empty. See the section on reaction direction in Section 4 for how this property can be used to infer direction.

conversion	
<input type="checkbox"/>	INTERACTION-TYPE
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	SPONTANEOUS
<input checked="" type="checkbox"/>	LEFT
<input checked="" type="checkbox"/>	RIGHT

Control subclasses

Two types of control processes exist under the control class: catalysis and modulation.



Catalysis

Definition: A control interaction in which a physical entity (a catalyst) increases the rate of a conversion interaction by lowering its activation energy. Instances of this class describe a pairing between a catalyzing entity and a catalyzed conversion.

Comment: A separate catalysis instance should be created for each different conversion that a physicalEntity may catalyze and for each different physicalEntity that may catalyze a conversion. For example, a bifunctional enzyme that catalyzes two different biochemical reactions would be linked to each of those biochemical reactions by two separate instances of the catalysis class. Also, catalysis reactions from multiple different organisms could be linked to the same generic biochemical reaction (a biochemical reaction is generic if it only includes small molecules). Generally, the enzyme catalyzing a conversion is known and the use of this class is obvious. In the cases where a catalyzed reaction is known to occur but the enzyme is not known, a catalysis instance should be created without a controller specified (i.e. the CONTROLLER property should remain empty).

Synonyms: facilitation, acceleration.

Examples: The catalysis of a biochemical reaction by an enzyme, the enabling of a transport interaction by a membrane pore complex, and the facilitation of a complex assembly by a scaffold protein. Hexokinase -> (The “Glucose + ATP -> Glucose-6-phosphate +ADP” reaction). A plasma membrane Na⁺/K⁺ ATPase is an active transporter (antiport pump) using the energy of ATP to pump Na⁺ out of the cell and K⁺ in. Na⁺ from cytoplasm to extracellular space would be described in a transport instance. K⁺ from extracellular space to cytoplasm would be described in a transport instance. The ATPase pump would be stored in a catalysis instance controlling each of the above transport instances. A biochemical reaction that does not occur by itself under physiological conditions, but has been observed to occur in the presence of cell extract, likely via one or more unknown enzymes present in the extract, would be stored in the CONTROLLED property, with the CONTROLLER property empty.

Properties:

COFACTOR - Any cofactor(s) or coenzyme(s) required for catalysis of the conversion by the enzyme. COFACTOR is a sub-property of PARTICIPANTS.

DIRECTION - Specifies the reaction direction of the interaction catalyzed by this instance of the catalysis class. Possible values of this property are: REVERSIBLE: Interaction occurs in both directions in physiological settings. PHYSIOL-LEFT-TO-RIGHT PHYSIOL-RIGHT-TO-LEFT The interaction occurs in the specified direction in physiological settings, because of several possible factors including the energetics of the reaction, local concentrations of reactants and

products, and the regulation of the enzyme or its expression. IRREVERSIBLE-LEFT-TO-RIGHT IRREVERSIBLE-RIGHT-TO-LEFT For all practical purposes, the interactions occurs only in the specified direction in physiological settings, because of chemical properties of the reaction. (This definition from EcoCyc)

catalysis	
<input type="checkbox"/>	CONTROL-TYPE
<input type="checkbox"/>	CONTROLLER
<input type="checkbox"/>	CONTROLLED
<input type="checkbox"/>	INTERACTION-TYPE
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	DIRECTION
<input checked="" type="checkbox"/>	COFACTOR

Modulation

Definition: A control interaction in which a physical entity modulates a catalysis interaction. Biologically, most modulation interactions describe an interaction in which a small molecule alters the ability of an enzyme to catalyze a specific reaction. Instances of this class describe a pairing between a modulating entity and a catalysis interaction.

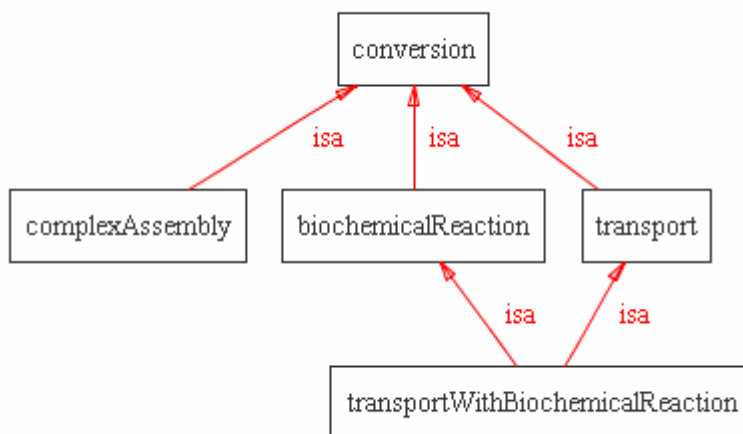
Comment: A separate modulation instance should be created for each different catalysis instance that a physical entity may modulate and for each different physical entity that may modulate a catalysis instance. A typical modulation instance has a small molecule as the controller entity and a catalysis instance as the controlled entity.

Examples: Allosteric activation and competitive inhibition of an enzyme's ability to catalyze a specific reaction.

modulation	
⓪	CONTROL-TYPE
⓪	CONTROLLER
⓪	CONTROLLED
⓪	INTERACTION-TYPE
⓪	PARTICIPANTS
⓪	EVIDENCE
⓪	NAME
⓪	SHORT-NAME
⓪	SYNONYMS
⓪	DATA-SOURCE
⓪	XREF
⓪	AVAILABILITY
⓪	COMMENT

Conversion subclasses

Four types of conversion processes exist under the conversion class: biochemical reaction, complex assembly, transport and transportWithBiochemicalReaction.



Biochemical Reaction

Definition: A conversion interaction in which one or more entities (substrates) undergo covalent changes to become one or more other entities (products). The substrates of biochemical reactions are defined in terms of sums of species. This is convention in biochemistry, and, in principle, all of the EC reactions should be biochemical reactions.

Examples: $\text{ATP} + \text{H}_2\text{O} = \text{ADP} + \text{P}_i$

Comment: In the example reaction above, ATP is considered to be an equilibrium mixture of several species, namely ATP^{4-} , HATP^{3-} , $\text{H}_2\text{ATP}^{2-}$, MgATP^{2-} , MgHATP^- , and Mg_2ATP . Additional species may also need to be considered if other ions (e.g. Ca^{2+}) that bind ATP are

present. Similar considerations apply to ADP and to inorganic phosphate (Pi). When writing biochemical reactions, it is important not to attach charges to the biochemical reactants and not to include ions such as H⁺ and Mg²⁺ in the equation. The reaction is written in the direction specified by the EC nomenclature system, if applicable, regardless of the physiological direction(s) in which the reaction proceeds. Polymerization reactions involving large polymers whose structure is not explicitly captured should generally be represented as unbalanced reactions in which the monomer is consumed but the polymer remains unchanged, e.g. glucose + glucose = glycogen.

Properties:

DELTA-G - For biochemical reactions, this property refers to the standard transformed Gibbs energy change for a reaction written in terms of biochemical reactants (sums of species), delta-G^o.

$$\text{delta-G}^{\circ} = -RT \ln K' \text{ and } \text{delta-G}^{\circ} = \text{delta-H}^{\circ} - T \text{ delta-S}^{\circ}$$

delta-G^o has units of kJ/mol. Like K', it is a function of temperature (T), ionic strength (I), pH, and pMg (pMg = -log₁₀[Mg²⁺]). Therefore, these quantities must be specified, and values for DELTA-G for biochemical reactions are represented as 5-tuples of the form (delta-G^o T I pH pMg). This property may have multiple values, representing different measurements for delta-G^o obtained under the different experimental conditions listed in the 5-tuple. (This definition from EcoCyc)

DELTA-H - For biochemical reactions, this property refers to the standard transformed enthalpy change for a reaction written in terms of biochemical reactants (sums of species), delta-H^o. delta-G^o = delta-H^o - T delta-S^o (This definition from EcoCyc)

DELTA-S - For biochemical reactions, this property refers to the standard transformed entropy change for a reaction written in terms of biochemical reactants (sums of species), delta-S^o. delta-G^o = delta-H^o - T delta-S^o (This definition from EcoCyc)

EC-NUMBER - The unique number assigned to a reaction by the Enzyme Commission of the International Union of Biochemistry and Molecular Biology. Note that not all biochemical reactions currently have EC numbers assigned to them.

KEQ – This quantity is dimensionless and is usually a single number. The measured equilibrium constant for a biochemical reaction, encoded by the property KEQ, is actually the apparent equilibrium constant, K'. Concentrations in the equilibrium constant equation refer to the total concentrations of all forms of particular biochemical reactants. For example, in the equilibrium constant equation for the biochemical reaction in which ATP is hydrolyzed to ADP and inorganic phosphate: K' = [ADP][Pi]/[ATP], The concentration of ATP refers to the total concentration of all of the following species: [ATP] = [ATP⁴⁻] + [HATP³⁻] + [H₂ATP²⁻] + [MgATP²⁻] + [MgHATP⁻] + [Mg₂ATP]. The apparent equilibrium constant is formally dimensionless, and can be kept so by inclusion of as many of the terms (1 mol/dm³) in the numerator or denominator as necessary. It is a function of temperature (T), ionic strength (I), pH, and pMg (pMg = -log₁₀[Mg²⁺]). Therefore, these quantities must be specified to be precise, and

values for KEQ for biochemical reactions may be represented as 5-tuples of the form (K' T I pH pMg). This property may have multiple values, representing different measurements for K' obtained under the different experimental conditions listed in the 5-tuple. (This definition adapted from EcoCyc)

biochemicalReaction	
<input type="checkbox"/>	SPONTANEOUS
<input type="checkbox"/>	LEFT
<input type="checkbox"/>	RIGHT
<input type="checkbox"/>	INTERACTION-TYPE
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	DELTA-G
<input checked="" type="checkbox"/>	DELTA-H
<input checked="" type="checkbox"/>	EC-NUMBER
<input checked="" type="checkbox"/>	KEQ
<input checked="" type="checkbox"/>	DELTA-S

Complex Assembly

Definition: A conversion interaction in which a set of physical entities, at least one being a macromolecule (protein, RNA, or DNA), aggregate via non-covalent interactions. One of the participants of a complexAssembly must be an instance of the class complex.

Comment: This class is also used to represent complex disassembly. The assembly or disassembly of a complex is often a spontaneous process, in which case the direction of the complexAssembly (toward either assembly or disassembly) should be specified via the SPONTANEOUS property.

Synonyms: aggregation, complex formation

Examples: Assembly of the TFB2 and TFB3 proteins into the TFIIF complex, and assembly of the ribosome through aggregation of its subunits.

Note: The following are not examples of complex assembly: Covalent phosphorylation of a protein (this is a biochemicalReaction); the TFIIF complex itself (this is an instance of the complex class, not the complexAssembly class).

complexAssembly	
<input type="checkbox"/>	SPONTANEOUS
<input type="checkbox"/>	LEFT
<input type="checkbox"/>	RIGHT
<input type="checkbox"/>	INTERACTION-TYPE
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT

Transport

Definition: A conversion interaction in which an entity (or set of entities) changes location within or with respect to the cell. A transport interaction does not include the transporter entity, even if one is required in order for the transport to occur. Instead, transporters are linked to transport interactions via the catalysis class.

Comment: Transport interactions do not involve chemical changes of the participant(s). These cases are handled by the transportWithBiochemicalReaction class.

Synonyms: translocation.

Examples: The movement of Na⁺ into the cell through an open voltage-gated channel.

transport	
<input type="checkbox"/>	SPONTANEOUS
<input type="checkbox"/>	LEFT
<input type="checkbox"/>	RIGHT
<input type="checkbox"/>	INTERACTION-TYPE
<input type="checkbox"/>	PARTICIPANTS
<input type="checkbox"/>	EVIDENCE
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT

Transport with Biochemical Reaction

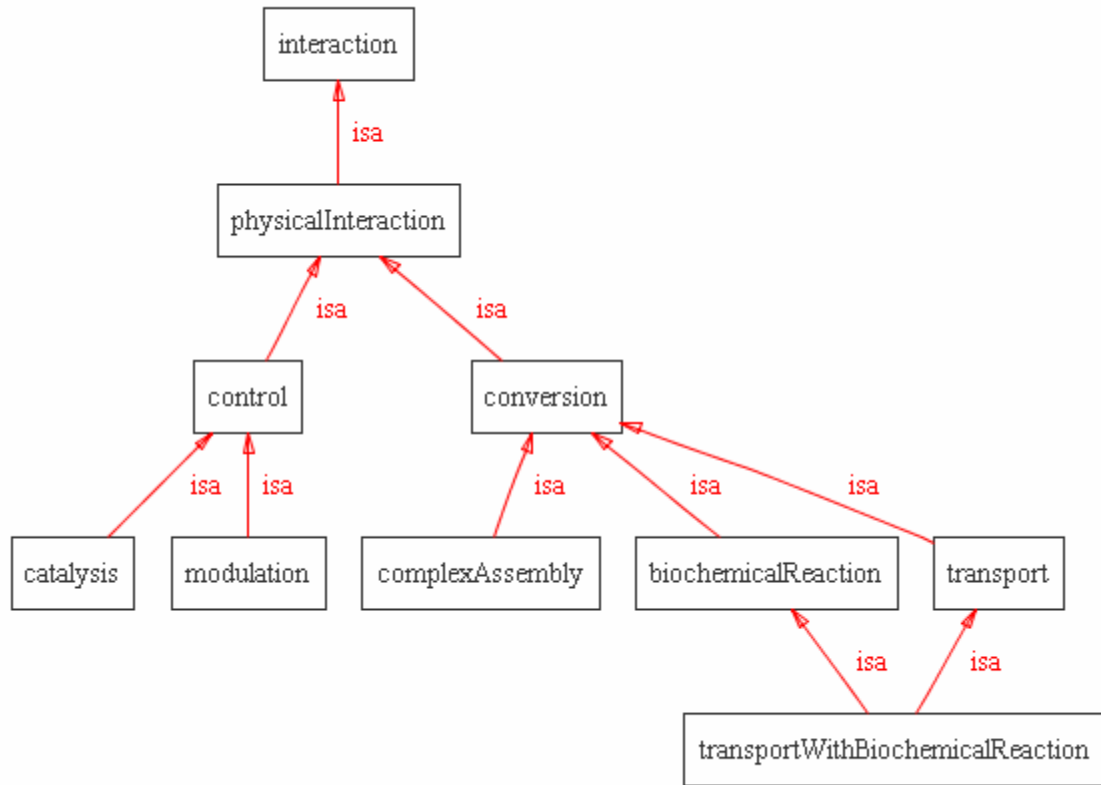
Definition: A conversion interaction that is both a biochemicalReaction and a transport. In transportWithBiochemicalReaction interactions, one or more of the substrates change both their location and their physical structure. Active transport reactions that use ATP as an energy source fall under this category, even if the only covalent change is the hydrolysis of ATP to ADP.

Comment: This class was added to support a large number of transport events in pathway databases that have a biochemical reaction during the transport process. It is not expected that other double inheritance subclasses will be added to the ontology at the same level as this class.

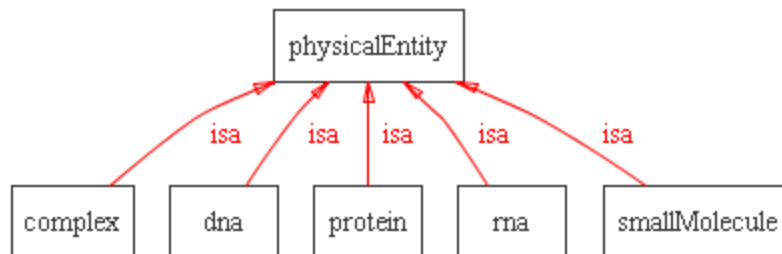
Examples: In the PEP-dependent phosphotransferase system, transportation of sugar into an E. coli cell is accompanied by the sugar's phosphorylation as it crosses the plasma membrane.

transportWithBiochemicalReaction	
<input type="radio"/>	DELTA-G
<input type="radio"/>	DELTA-H
<input type="radio"/>	EC-NUMBER
<input type="radio"/>	KEQ
<input type="radio"/>	DELTA-S
<input type="radio"/>	SPONTANEOUS
<input type="radio"/>	LEFT
<input type="radio"/>	RIGHT
<input type="radio"/>	INTERACTION-TYPE
<input type="radio"/>	PARTICIPANTS
<input type="radio"/>	EVIDENCE
<input type="radio"/>	NAME
<input type="radio"/>	SHORT-NAME
<input type="radio"/>	SYNONYMS
<input type="radio"/>	DATA-SOURCE
<input type="radio"/>	XREF
<input type="radio"/>	AVAILABILITY
<input type="radio"/>	COMMENT

Summary of Interaction Class Structure



Physical Entity subclasses



DNA

Definition: A physical entity consisting of a sequence of deoxyribonucleotide monophosphates; a deoxyribonucleic acid.

Comment: This is not a 'gene', since gene is a genetic concept, not a physical entity. The concept of a gene may be added later in BioPAX.

Examples: a chromosome, a plasmid. A specific example is chromosome 7 of Homo sapiens.

Properties:

ORGANISM - An organism, e.g. 'Homo sapiens' that this DNA molecule was found in. An xref to a DNA database may be present that contains organism information, in which case the information should be consistent with the value for ORGANISM.

SEQUENCE - Polymer sequence in uppercase letters, usually A,C,G,T letters representing the nucleosides of adenine, cytosine, guanine and thymine, respectively.

dna	
<input type="checkbox"/>	NAME
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	XREF
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	SEQUENCE
<input checked="" type="checkbox"/>	ORGANISM

RNA

Definition: A physical entity consisting of a sequence of ribonucleotide monophosphates; a ribonucleic acid.

Examples: messengerRNA, microRNA, ribosomalRNA. A specific example is the let-7 microRNA.

Properties:

ORGANISM - An organism, e.g. 'Homo sapiens' that this RNA molecule was found in.

SEQUENCE - Polymer sequence in uppercase letters, usually A, C, U, G.

rna	
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	COMMENT
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	NAME
<input type="checkbox"/>	XREF
<input checked="" type="checkbox"/>	ORGANISM
<input checked="" type="checkbox"/>	SEQUENCE

Protein

Definition: A physical entity consisting of a sequence of amino acids; a protein monomer; a single polypeptide chain.

Examples: The epidermal growth factor receptor (EGFR) protein.

Properties:

ORGANISM - An organism, e.g. 'Homo sapiens' that this protein was found in.

SEQUENCE - Polymer sequence in uppercase letters, corresponding to the 20 letter IUPAC amino acid code.

protein	
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	COMMENT
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	NAME
<input type="checkbox"/>	XREF
<input checked="" type="checkbox"/>	ORGANISM
<input checked="" type="checkbox"/>	SEQUENCE

Small Molecule

Definition: Any bioactive molecule that is not a peptide, DNA, or RNA. Generally these are non-polymeric, but complex carbohydrates are not explicitly modeled as classes in this version of the ontology, thus are forced into this class.

Comment: Recently, a number of small molecule databases have become available to cross-reference from this class.

Examples: glucose, penicillin, phosphatidylinositol

Properties:

CHEMICAL-FORMULA - The chemical formula of the small molecule. Note: chemical formula can also be stored in the **STRUCTURE** property (in CML). In case of disagreement between the value of this property and that in the CML file, the CML value takes precedence.

MOLECULAR-WEIGHT - Defines the molecular weight of the molecule, in daltons.

STRUCTURE - Defines the chemical structure and other information about this molecule, using an instance of class `chemicalStructure`.

smallMolecule	
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	COMMENT
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	NAME
<input type="checkbox"/>	XREF
<input checked="" type="checkbox"/>	CHEMICAL-FORMULA
<input checked="" type="checkbox"/>	STRUCTURE
<input checked="" type="checkbox"/>	MOLECULAR-WEIGHT

Complex

Definition: A physical entity whose structure is comprised of other physical entities bound to each other non-covalently, at least one of which is a macromolecule (protein, DNA, or RNA). Complexes must be stable enough to function as a biological unit; in general, the temporary association of an enzyme with its substrate(s) should not be considered or represented as a complex. A complex is the physical product of an interaction (`complexAssembly`) and is not itself considered an interaction.

Comment: In general, complexes should not be defined recursively so that smaller complexes exist within larger complexes, i.e. a complex should not be a **COMPONENT** of another complex (see comments on the **COMPONENT** property below). The boundaries on the size of complexes described by this class are not defined here, although elements of the cell as large and dynamic as, e.g., a mitochondrion would typically not be described using this class (later versions of this ontology may include a `cellularComponent` class to represent these). The strength of binding and the topology of the components cannot be described currently, but may be included in future versions of the ontology, depending on community need.

Examples: Ribosome, RNA polymerase II. Other examples of this class include complexes of multiple protein monomers and complexes of proteins and small molecules.

Properties:

COMPONENTS - Defines the physicalEntity subunits of this complex. This property should not contain other complexes, i.e. it should always be a flat representation of the complex. For example, if two protein complexes join to form a single larger complex via a complex assembly interaction, the COMPONENTS of the new complex should be the individual proteins of the smaller complexes, not the two smaller complexes themselves. Exceptions are black-box complexes (i.e. complexes in which the COMPONENTS property is empty), which may be used as COMPONENTS of other complexes because their constituent parts are unknown / unspecified. The reason for keeping complexes flat is to signify that there is no information stored in the way complexes are nested, such as assembly order. Otherwise, the complex assembly order may be implicitly encoded and interpreted by some users, while others created hierarchical complexes randomly, which could lead to data loss.

Additionally, the physicalEntityParticipants used in the COMPONENTS property are in the context of the complex, thus should not be reused between complexes. For instance, a protein may participate in two different complexes, but have different conformation in each.

ORGANISM - An organism, e.g. 'Homo sapiens' that this complex is found in.

complex	
<input type="checkbox"/>	SYNONYMS
<input type="checkbox"/>	COMMENT
<input type="checkbox"/>	DATA-SOURCE
<input type="checkbox"/>	SHORT-NAME
<input type="checkbox"/>	AVAILABILITY
<input type="checkbox"/>	NAME
<input type="checkbox"/>	XREF
<input checked="" type="checkbox"/>	COMPONENTS
<input checked="" type="checkbox"/>	ORGANISM

Utility classes

A number of properties in the ontology accept instances of utility classes as values. In BioPAX Level 2, a number of new organizational classes in the utility class tree have been added to partition the utility class hierarchy into more easily navigable subdivisions.

Top level utility classes

The BioPAX Level 2 ontology defines 9 direct subclasses of utilityClass: **chemicalStructure**, **confidence**, **evidence**, **externalReferenceUtilityClass**, **pathwayStep**, **physicalEntityParticipant**, **sequenceFeature**, and **sequenceLocation**.

Chemical Structure

Definition: Describes a small molecule structure. Structure information is stored in the property STRUCTURE-DATA, in one of three formats: the CML format¹³ (see URL www.xml-cml.org),

the SMILES format¹⁴ (see URL www.daylight.com/dayhtml/smiles/) or the InChI format (<http://www.iupac.org/inchi/>). The STRUCTURE-FORMAT property specifies which format is used.

Comment: By virtue of the expressivity of CML, an instance of this class can also provide additional information about a small molecule, such as its chemical formula, names, and synonyms, if CML is used as the structure format.




Examples: The following SMILES string, which describes the structure of glucose-6-phosphate:

`'C(OP(=O)(O)O)[CH]1([CH](O)[CH](O)[CH](O)[CH](O)O1)'`.

Properties:

STRUCTURE-DATA - This property holds a string of data defining chemical structure or other information, in either the CML or SMILES format, as specified in property Structure-Format. If, for example, the CML format is used, then the value of this property is a string containing the XML encoding of the CML data.

STRUCTURE-FORMAT - This property specifies which format is used to define chemical structure data.

chemicalStructure	
	COMMENT
	STRUCTURE-FORMAT
	STRUCTURE-DATA

Confidence

Definition: Confidence that the containing instance actually occurs or exists *in vivo*, usually a statistical measure. The xref may contain a publication that describes the method used to determine the confidence. There is currently no standard way of describing confidence values, so any string is valid for the confidence value. In the future, a controlled vocabulary of accepted confidence values could become available, in which case it will likely be adopted for use here to describe the value.

Examples: The statistical significance of a result, e.g. “p<0.05”.

Properties:

CONFIDENCE-VALUE - The value of the confidence measure.

XREF - Values of this property define external cross-references from this entity to entities in external databases.

confidence	
0	COMMENT
0	CONFIDENCE-VALUE
0	XREF

Evidence

Definition: The support for a particular assertion, such as the existence of an interaction or pathway. At least one of CONFIDENCE, EVIDENCE-CODE, or EXPERIMENTAL-FORM must be instantiated when creating an evidence instance. XREF may reference a publication describing the experimental evidence using a publicationXref or may store a description of the experiment in an experimental description database using a unificationXref (if the referenced experiment is the same) or relationshipXref (if it is not identical, but similar in some way e.g. similar in protocol). Evidence is meant to provide more information than just an xref to the source paper.

Examples: A description of a molecular binding assay that was used to detect a protein-protein interaction.

Properties:

CONFIDENCE - Confidence in the containing instance. Usually a statistical measure.

EVIDENCE-CODE - A pointer to a term in an external controlled vocabulary, such as the GO or BioCyc evidence codes, that describes the nature of the support. See the section on controlled vocabularies in Section 4 for more information.

EXPERIMENTAL-FORM - The experimental forms associated with an evidence instance.

XREF - Values of this property define external cross-references from this entity to entities in external databases.

evidence	
0	COMMENT
0	XREF
0	EXPERIMENTAL-FORM
0	EVIDENCE-CODE
0	CONFIDENCE

Experimental Form

Definition: The form of a physical entity in a particular experiment, as it may be modified for purposes of experimental design.

Examples: A His-tagged protein in a binding assay.

Properties:

EXPERIMENTAL-FORM-TYPE - Descriptor of this experimental form from a controlled vocabulary. See the section on controlled vocabularies in Section 4 for more information.

PARTICIPANT - The participant that has the experimental form being described. This is a physicalEntityParticipant instance.

experimentalForm	
0	COMMENT
0	PARTICIPANT
0	EXPERIMENTAL-FORM-TYPE

External Reference Utility Class

Definition: A pointer to an external object, such as an entry in a database or a term in a controlled vocabulary.

Comment: This class is for organizational purposes only; direct instances of this class should not be created.

externalReferenceUtilityClass	
0	COMMENT

Pathway Step

Definition: A step in a pathway.

Comment: Multiple interactions may occur in a pathway step, each should be listed in the STEP-INTERACTIONS property. Order relationships between pathway steps may be established with the NEXT-STEP slot. This order may not be temporally meaningful for specific steps, such as for a pathway loop or a reversible reaction, but represents a directed graph of step relationships that can be useful for describing the overall flow of a pathway, as may be useful in a pathway diagram.

Example: A metabolic pathway may contain a pathway step composed of one biochemical reaction (BR1) and one catalysis (CAT1) instance, where CAT1 describes the catalysis of BR1.

Properties:

NEXT-STEP - The next step(s) of the pathway. Contains zero or more pathwayStep instances. If there is no next step, this property is empty.

STEP-INTERACTIONS - The interactions that take place at this step of the pathway.

pathwayStep	
0	COMMENT
0	STEP-INTERACTIONS
0	NEXT-STEP

Physical Entity Participant

Definition: Any additional special characteristics of a physical entity in the context of an interaction. These currently include stoichiometric coefficient and cellular location, but this list may be expanded in later levels.

Comment: PhysicalEntityParticipants should not be used in multiple interaction or complex instances. Instead, each interaction and complex should reference its own unique set of physicalEntityParticipants. The reason for this is that a user may add new information about a physicalEntityParticipant for one interaction or complex, such as the presence of a previously unknown post-translational modification, and unwittingly invalidate the physicalEntityParticipant for the other interactions or complexes that make use of it.

Example: In the interaction describing the transport of L-arginine into the cytoplasm in E. coli, the LEFT property in the interaction would be filled with an instance of physicalEntityParticipant that specified the location of L-arginine as periplasm and the stoichiometric coefficient as one.

Properties:

CELLULAR-LOCATION - A cellular location, e.g. 'cytoplasm'. This should reference a term in the Gene Ontology Cellular Component ontology. The location referred to by this property should be as specific as is known. If an interaction is known to occur in multiple locations, separate interactions (and physicalEntityParticipants) must be created for each different location. Note: If a location is unknown then the GO term for 'cellular component unknown' (GO:0008372) should be used in the LOCATION property. If the location of a participant in a complex is unspecified, it may be assumed to be the same location as that of the complex. In case of conflicting information, the location of the most outer layer of any nesting should be considered correct. Note: Cellular location describes a specific location of a physical entity as it would be used in e.g. a transport reaction. It does not describe all of the possible locations that the physical entity could be in the cell. See the section on controlled vocabularies in Section 4 for more information.

PHYSICAL-ENTITY - The physical entity annotated with stoichiometry and cellular location attributes from the physicalEntityParticipant instance.

STOICHIOMETRIC-COEFFICIENT - Each value of this property represents the stoichiometric coefficient for one of the physical entities in an interaction or complex. For a given interaction, the stoichiometry should always be used where possible instead of representing the number of participants with separate instances of each participant. If there are three ATP molecules, one ATP molecule should be represented as a participant and the stoichiometry should be set to 3.

physicalEntityParticipant	
0	COMMENT
0	CELLULAR-LOCATION
0	STOICHIOMETRIC-COEFFICIENT
0	PHYSICAL-ENTITY

Sequence Feature

Definition: A feature on a sequence relevant to an interaction, such as a binding site or post-translational modification.

Examples: A phosphorylation on a protein.

Properties:

FEATURE-LOCATION - Location of the feature on the sequence of the interactor. One feature may have more than one location, used e.g. for features which involve sequence positions close in the folded, three-dimensional state of a protein, but non-continuous along the sequence.

FEATURE-TYPE - Description and classification of the feature. See the section on controlled vocabularies in Section 4 for more information.

NAME - The preferred full name for this sequence feature.

SHORT-NAME - An abbreviated name for this entity. Preferably a name that is short enough to be used in a visualization application to label a graphical element that represents this entity. If no short name is available, an xref may be used for this purpose.

SYNONYMS - One or more synonyms for the name of this sequence feature.

XREF - Values of this property define external cross-references from this entity to entities in external databases.

sequenceFeature	
0	COMMENT
0	NAME
0	FEATURE-TYPE
0	SHORT-NAME
0	SYNONYMS
0	XREF
0	FEATURE-LOCATION

Sequence Location

Definition: A location on a nucleotide or amino acid sequence.

Comment: For organizational purposes only; direct instances of this class should not be created.

sequenceLocation	
①	COMMENT

External Reference Utility Class subclasses

BioSource

Definition: The biological source of an entity (e.g. protein, RNA or DNA). Some entities are considered source-neutral (e.g. small molecules), and the biological source of others can be deduced from their constituents (e.g. complex, pathway).

Examples: HeLa cells, human, and mouse liver tissue.

Properties:

CELLTYPE - A cell type, e.g. 'HeLa'. This should reference a term in a controlled vocabulary of cell types. See the section on controlled vocabularies in Section 4 for more information.

NAME - The preferred full name for this entity. E.g. “Homo sapiens”.

TAXON-XREF - An xref to an organism taxonomy database, preferably NCBI taxon. This should be an instance of unificationXref, unless the organism is not in an existing database.




TISSUE - An external controlled vocabulary of tissue types. See the section on controlled vocabularies in Section 4 for more information.

bioSource	
①	COMMENT
①	CELLTYPE
①	TISSUE
①	NAME
①	TAXON-XREF

DataSource

Definition: The direct source of this data. This does not store the trail of sources from the generation of the data to this point, only the last known source, such as a database. The XREF property may contain a publicationXref referencing a publication describing the data source (e.g. a database publication). A unificationXref may be used e.g. when pointing to an entry in a database of databases describing this database.

Examples: A database or person name.

dataSource	
	COMMENT
	NAME
	XREF




Open Controlled Vocabulary

Definition: Used to import terms from external controlled vocabularies (CVs) into the ontology. To support consistency and compatibility, open, freely available CVs should be used whenever possible, such as the Gene Ontology (GO)¹⁵ or other open biological CVs listed on the OBO website (<http://obo.sourceforge.net/>). See the section on controlled vocabularies in Section 4 for more information.

Properties:

TERM - The external controlled vocabulary term.

XREF - Values of this property define external cross-references from this entity to entities in external databases.

openControlledVocabulary	
	COMMENT
	XREF
	TERM

Xref

Definition: A reference from an instance of a class in this ontology to an object in an external resource.

Comment: As the most abstract xref class in the ontology, instances of the xref class should be created rarely, if ever.

Properties:

DB - The name of the external database to which this xref refers.

DB-VERSION - The version of the external database in which this xref was last known to be valid. Resources may have recommendations for referencing dataset versions. For instance, the Gene Ontology recommends listing the date the GO terms were downloaded.

ID - The primary identifier in the external database of the object to which this xref refers.

ID-VERSION - The version number of the identifier (ID). E.g. The RefSeq accession number NM_005228.3 should be split into NM_005228 as the ID and 3 as the ID-VERSION.

xref	
0	COMMENT
0	DB
0	DB-VERSION
0	ID
0	ID-VERSION

Xref subclasses

Publication Xref

Definition: An xref that defines a reference to a publication such as a book, journal article, web page, or software manual. The reference may or may not be in a database, although references to PubMed are preferred when possible. The publication should make a direct reference to the instance it is attached to.

Comment: Publication xrefs should make use of PubMed IDs wherever possible. The DB property of an xref to an entry in PubMed should use the string “PubMed” and not “MEDLINE”.

Examples: PubMed:10234245

Properties:

The following properties may be used when the DB and ID fields cannot be used, such as when referencing a publication that is not in PubMed. The URL property should not be used to reference publications that can be uniquely referenced using a DB, ID pair. One reason for this is that it is expected that DB, ID pairs are more stable than URLs.

AUTHORS - The authors of this publication, one per property value.

SOURCE - The source in which the reference was published, such as: a book title, or a journal title and volume and pages.

TITLE - The title of the publication.

URL - The URL at which the publication can be found, if it is available through the Web.

YEAR - The year in which this publication was published.

publicationXref	
0	DB
0	DB-VERSION
0	ID
0	ID-VERSION
0	COMMENT
1	TITLE
1	YEAR
1	AUTHORS
1	URL
1	SOURCE

Relationship Xref

Definition: An xref that defines a reference to an entity in an external resource that does not have the same biological identity as the referring entity.

Comment: There is currently no controlled vocabulary of relationship types for BioPAX, although one will be created in the future if a need develops.

Examples: A link between a gene G in a BioPAX data collection, and the protein product P of that gene in an external database. This is not a unification xref because G and P are different biological entities (one is a gene and one is a protein). Another example is a relationship xref for a protein that refers to the Gene Ontology biological process, e.g. 'immune response,' that the protein is involved in.

Properties:

RELATIONSHIP-TYPE - This property names the type of relationship between the BioPAX object linked from, and the external object linked to, such as 'gene of this protein', or 'protein with similar sequence'.

relationshipXref	
0	DB
0	DB-VERSION
0	ID
0	ID-VERSION
0	COMMENT
1	RELATIONSHIP-TYPE

Unification Xref

Definition: A unification xref defines a reference to an entity in an external resource that has the same biological identity as the referring entity¹⁶. For example, if one wished to link from a database record, C, describing a chemical compound in a BioPAX data collection to a record, C', describing the same chemical compound in an external database, one would use a unification

xref since records C and C' describe the same biological identity. Generally, unification xrefs should be used whenever possible, although there are cases where they might not be useful, such as application to application data exchange.

Comment: Unification xrefs in physical entities are essential for data integration, but are less important in interactions. This is because unification xrefs on the physical entities in an interaction can be used to compute the equivalence of two interactions of the same type. An xref in a protein pointing to a gene, e.g. in the LocusLink database¹⁷, would not be a unification xref since the two entities do not have the same biological identity (one is a protein, the other is a gene). Instead, this link should be captured as a relationship xref¹⁶. References to an external controlled vocabulary term within the OpenControlledVocabulary class should use a unification xref where possible (e.g. GO:0005737)

Examples: An xref in a protein instance pointing to an entry in the Swiss-Prot database, and an xref in an RNA instance pointing to the corresponding RNA sequence in the RefSeq database.

unificationXref	
<input type="checkbox"/>	DB
<input type="checkbox"/>	DB-VERSION
<input type="checkbox"/>	ID
<input type="checkbox"/>	ID-VERSION
<input type="checkbox"/>	COMMENT

Physical Entity Participant subclasses

sequenceParticipant

Definition: A DNA, RNA or protein participant in an interaction. See physicalEntityParticipant for more documentation.

sequenceParticipant	
<input type="checkbox"/>	PHYSICAL-ENTITY
<input type="checkbox"/>	CELLULAR-LOCATION
<input type="checkbox"/>	STOICHIOMETRIC-COEFFICIENT
<input type="checkbox"/>	COMMENT
<input checked="" type="checkbox"/>	SEQUENCE-FEATURE-LIST

SEQUENCE-FEATURE-LIST - Sequence features relevant for the interaction, for example binding domains or modification sites. Warning: this property may be moved into a state class in Level 3.

Sequence Location subclasses

Sequence Interval

Definition: Describes an interval on a sequence. All of the sequence from the begin site to the end site (inclusive) is described, not any subset.

Properties:

SEQUENCE-INTERVAL-BEGIN - The begin position of a sequence interval.

SEQUENCE-INTERVAL-END - The end position of a sequence interval.

sequenceInterval	
0	COMMENT
0	SEQUENCE-INTERVAL-END
0	SEQUENCE-INTERVAL-BEGIN

Sequence Site

Definition: Describes a site on a sequence, i.e. the position of a single nucleotide or amino acid.

Properties:

POSITION-STATUS - The confidence status of the sequence position. This could be:

UNCERTAIN: The site is not certain. This could be used to describe a post-translational modification that is known to occur on a protein, but the location is not known.

GREATER-THAN: The site is greater than the SEQUENCE-POSITION.

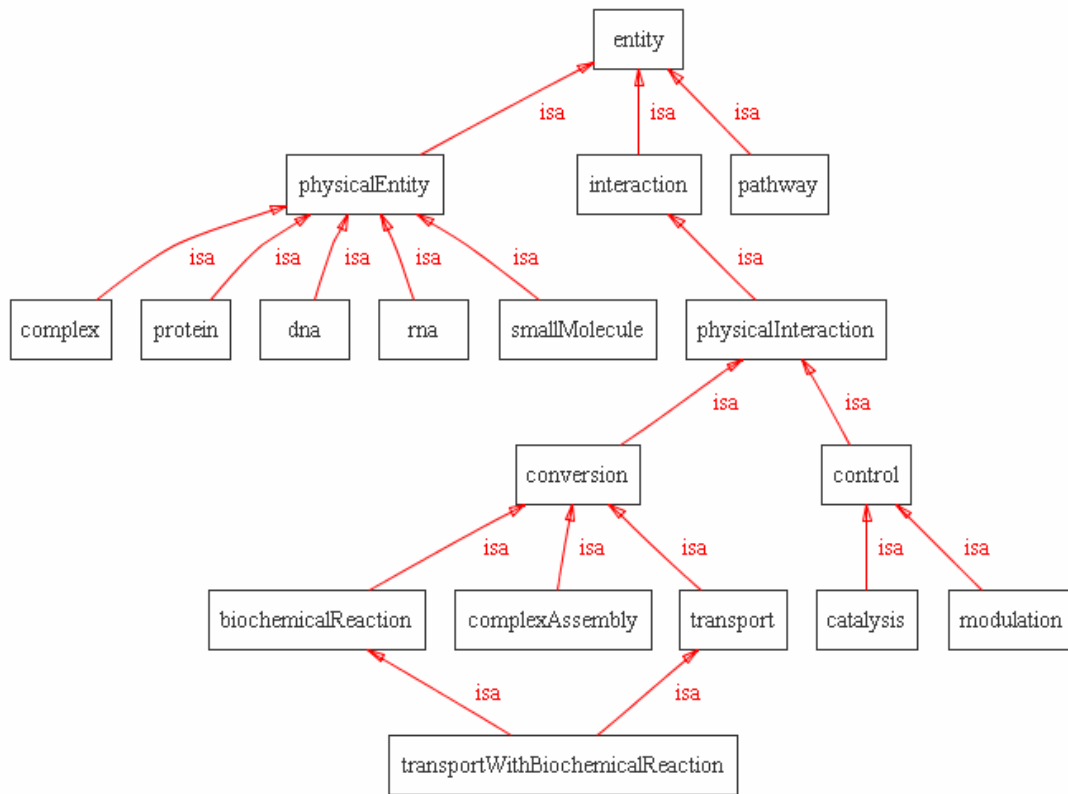
LESS-THAN: The site is less than the SEQUENCE-POSITION.

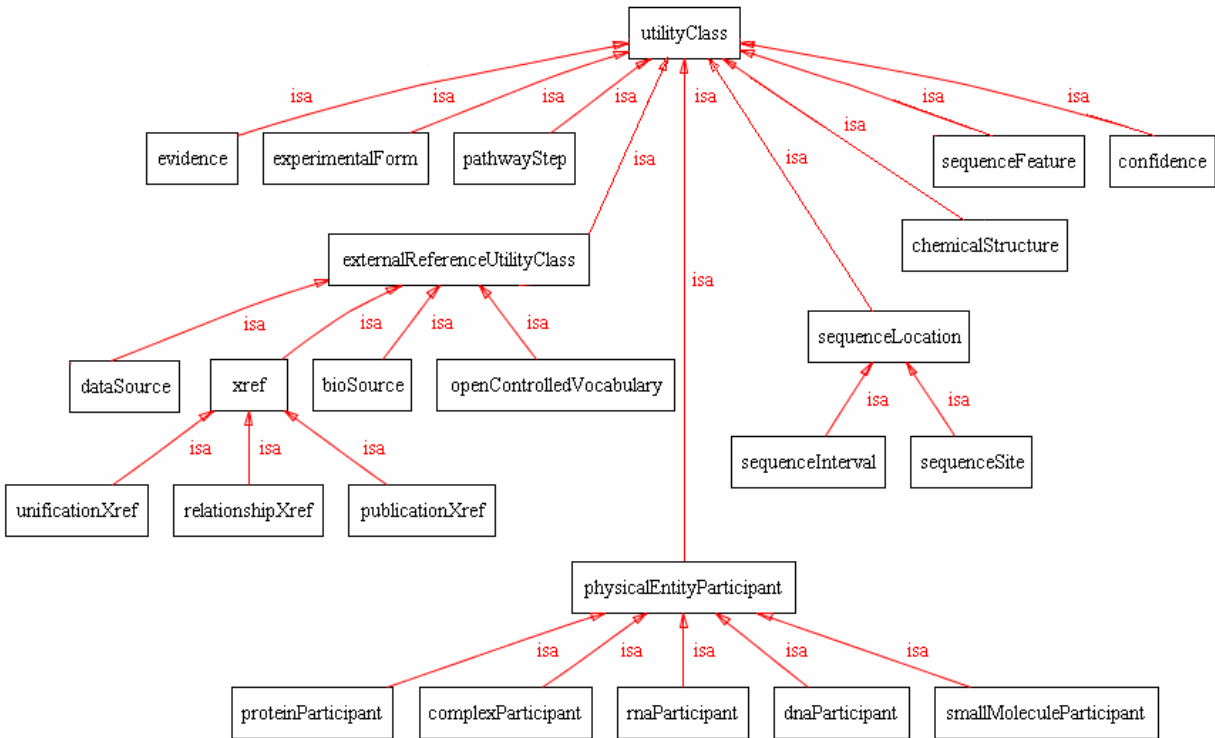
CERTAIN: The SEQUENCE-POSITION is known with certainty.

SEQUENCE-POSITION - The integer listed gives the position. The first base or amino acid is position 1. In combination with the numeric value, the property 'POSITION-STATUS' allows to express fuzzy positions, e.g. 'less than 4'.

sequenceSite	
0	COMMENT
0	POSITION-STATUS
0	SEQUENCE-POSITION

Summary of BioPAX Class Structure





3 Examples

A number of examples of pathways in the BioPAX format are available for download from the BioPAX homepage (<http://www.biopax.org/>) and from databases that support BioPAX.

4 Best Practices

While the BioPAX ontology imposes many logical constraints so that data encoded make sense for the use cases envisioned, some parts of the ontology have the potential for encoding data in multiple ways or there may be multiple options for treating data. This section recommends best practices in the use of the ontology for data exchange between groups. It supplements recommendations made in the class and property definitions provided above. It is expected that major data providers follow these recommendations to ensure compatibility of their data with other BioPAX data.

Users of BioPAX who are not exchanging data between groups, e.g. using BioPAX as an internal data model for their software, might find alternate representations to the ones recommended here more useful for their purposes.

Referencing External Objects

BioPAX is focused on representing interactions and pathways, but links to many different types of information. It is important to maintain these links, since Biological objects in external databases, such as proteins and small molecules, should be referenced via instances of the `xref` class, and external controlled vocabulary terms, such as those defined by the Gene Ontology Consortium or the PSI-MI initiative, should be referenced via instances of the `openControlledVocabulary` class.

BioPAX does not currently support a general mechanism to use RDF IDs to seamlessly point to external biological objects and controlled vocabulary terms on the semantic web, since these mechanisms are not yet mature. When these become more widespread, BioPAX may support them (see section on BioPAX and the Semantic Web, below).

Using xrefs

External references (Xrefs) relate elements within a BioPAX document to external data. Xrefs are more than just identifiers, as they contain the name of the data source the identifiers are part of and potentially version information as well. They exist to uniquely point to a record in an external data source (e.g. a bioinformatics database). For example, a pointer from a protein instance in BioPAX to a record in a database describing the protein would be established with an xref. The creator of the BioPAX document likely does not have control over all external references. Note: Xrefs are NOT related to RDF IDs.

Within any xref, database names (in the `DB` property) should be from a controlled vocabulary to avoid data integration problems that arise when different people use different spellings of database names. The PSI-MI has created a database name controlled vocabulary, a copy of which may be found on the BioPAX website (<http://www.biopax.org>). If it is not possible to use this controlled vocabulary for a specific database, be careful to use the database name exactly as spelled on the database website (e.g. Use “Swiss-Prot” instead of `swissprot`, `SWP` or other frequent spellings). Also, suggest that the database name you want to use be added to the PSI-MI vocabulary. Similarly, the `ID` property should use the primary key of the target object, e.g.

“P54352” instead of “HUMAN_P53”. Software should be able to use xref information to construct a web hyperlink to the database record being pointed to.

Xrefs to database accession numbers that contain version information should keep the version information separate from the ID, e.g. the accession number “CAA61361.2” should be stored as “ID=CAA61361”, and “ID-VERSION=2”. This is to enable computer software to easily identify the accession or the version without having to be aware of all possible ways of encoding the version in the accession number. If you are unsure how to encode the ID, think of how easy it would be to build a web hyperlink to the referenced database record from the xref using the information provided.

Importance of unification xrefs

Abundant use of unification xrefs, where possible, is highly recommended, especially in `physicalEntity` instances. These xrefs allow a user to understand that two independent instances from different BioPAX documents are actually the same entity (as long as they share one or more unification xrefs).

When exporting data from a database with primary keys, those keys should generally be encoded as unification xrefs. For example, if a database contains biochemical reactions with IDs for both the reactions and the small molecules that participate in those reactions, unification xrefs containing these IDs should appear in the corresponding BioPAX instances generated by the database. In general, the original data record from which an instance was generated should be pointed to via a unification xref. The exception to this rule occurs when the native class of the data is not completely synonymous with the BioPAX class to which it is mapped. In these cases, the resulting BioPAX instances should point back to the original data records via relationship xrefs.

Caution: Complications with unification xrefs can arise when the database that is being pointed to contains redundant information or contains more than one type of record. If a database contains redundant information, such as GenBank or Chemical Abstracts Service (CAS), it is possible to reference the same physical entity in the same database, but use IDs of different redundant records. In this case, unification xrefs can not be guaranteed to be useful in determining if two physical entities are the same across multiple BioPAX documents. More information about database record relationships will be required. Also, if a database contains different types of records, such as mRNA and protein records in GenBank or chemical structures with and without R groups in CAS, then it may be impossible to determine the type of record referenced, which may lead to unification xrefs that point to molecules of a different type than the referencing physical entity. Care in creating unification xrefs should be taken when linking to these types of databases so that the link is unambiguous.

Using external controlled vocabulary terms

A number of slots in the BioPAX ontology reference the `openControlledVocabulary` class. Some of these properties are referred to as “mission-critical” because the information they provide is very important to most users of pathway data and to enable software to make simplifying assumptions about which vocabularies to expect. These include: `CELLULAR-LOCATION`, `EVIDENCE-CODE`, `EXPERIMENTAL-FORM-TYPE`, and `FEATURE-TYPE`. It is strongly

recommended that the following external controlled vocabularies (CVs) be used for these mission-critical properties:

CELLULAR-LOCATION: Gene Ontology Cellular Component

EVIDENCE-CODE: BioCyc Evidence Ontology, PSI-MI interaction detection CV (Note: the PSI-MI CV will likely be extended to contain all BioCyc evidence codes. When this occurs, the PSI-MI CV will be preferred.)

EXPERIMENTAL-FORM-TYPE: PSI-MI experimental form type CV

FEATURE-TYPE: PSI-MI Feature Type CV

DB (in xref): PSI-MI Database Name CV where possible

Note: EVIDENCE-CODE allows multiple CV terms to be included. Because it is mission critical, at least one term should be from the above recommended CV.

Several other properties, CELLTYPE, INTERACTION-TYPE, and TISSUE, also make use of the openControlledVocabulary class. These non-mission-critical properties serve simply to provide additional annotation. The following CVs are suggested for these properties:

CELLTYPE: <http://obo.sourceforge.net/cgi-bin/detail.cgi?celltype>

INTERACTION-TYPE: PSI-MI Interaction Type CV

TISSUE: MeSH Tissue CV (<http://www.nlm.nih.gov/mesh/meshhome.html>)

Note: The PSI-MI CVs may be found here: <http://psidev.sourceforge.net/mi/controlledVocab/>

Mission critical CVs that are relatively stable are packaged with BioPAX as a convenience and are periodically updated with BioPAX releases, though developers should always consult the latest versions listed above for potential changes.

Reusing utility class instances

Utility classes store structured bits of information in the context of the main ontology classes ('entity' and its subclasses). As such, they are not guaranteed to make sense out of the context of the classes they are used in. Utility classes should be reused carefully to avoid making improper statements out of context. For example, consider a proteinParticipant instance that was used by multiple interactions. If new information became available for one of those interactions, e.g. the necessity of a particular phosphorylated residue, addition of this additional information to the proteinParticipant class could invalidate it for all of the other interactions that refer to it.

Due to the potential problems, it should not be assumed that utility class instances will be re-used in a BioPAX file. Software implementations must be aware of this if instance equality is important, so that equality statements are made based on all content of utility class instances.

Pathways and Networks

In BioPAX, a pathway is defined using interactions and/or pathwayStep instances. This provides sufficient flexibility to support two main representation conventions, the typical biochemical

pathway, composed of a set of pathway steps, and the typical molecular interaction network, composed of a set of interactions not involving pathway steps.

Metabolic pathways

For most metabolic pathways, the contents of the PATHWAY-COMPONENTS property should be a set of pathwayStep instances, one for each step of the pathway. When possible, a pairing of a biochemical reaction and its corresponding catalysis instance should be stored in a pathway step. In rare cases, such as if a pathway has not been completely elucidated, the topology of the pathway may not be fully known. These interactions may be stored directly in the PATHWAY-COMPONENTS property without being wrapped inside pathwayStep instances. The PATHWAY-COMPONENTS property may also be left completely empty, in which case the metabolic pathway would simply have a name and could be treated as a black box. This use is valid according to the BioPAX ontology, but is not encouraged as the intent of BioPAX is to represent pathways with a high degree of detail, if known.

Metabolic pathwayStep instances should contain at most one conversion, typically one catalysis, and any number of modulation instances. Exceptions to this rule are cases in which a conversion is known to be catalyzed by multiple enzymes. In these cases, each separate catalysis instance should be included in the pathwayStep (providing each occurs within the context of the pathway). The opposite case of one enzyme catalyzing different reactions, maybe using different cofactors, should be represented as a separate pathway step for each reaction, following the above rule.

A pathway step should not be listed in the NEXT-STEP property of another pathwayStep if the intersection of the entities in the PARTICIPANTS properties of their interactions is empty. Typically, at least one product of the conversion in each preceding pathwayStep should participate either as a CONTROLLER or as a substrate to the conversion interaction of a pathwayStep. Note: The NEXT-STEP property is meant only to represent pathway topology, not order of events (see definition of the pathwayStep class in section 2 for more information). Holes in the pathway are allowed, for instance, if intermediate steps are not known.

Interaction networks

Often, molecular interaction datasets do not contain any notion of a pathway, but instead simply store a collection of binary or higher order interactions between molecules. For these datasets, instances of the pathway class are not necessary, though may be used to store sub-networks of the overall interaction network that are part of a pathway.

Conversion Direction

Multiple places exist in BioPAX for providing information on the direction in which a conversion interaction proceeds. The DIRECTION property of the catalysis instance, if specified, should override all other sources of direction information. If the conversion is not catalyzed, or the DIRECTION property is empty, the SPONTANEOUS property of the conversion should be used as the source of direction information. If a conversion is spontaneous, then it will occur in the specified direction without any catalyst (although, in the cell, the reverse may happen by unknown processes). If values for neither DIRECTION nor SPONTANEOUS are specified, it may be possible to infer direction given the thermodynamic constants in the

biochemical reaction, if specified and if assumptions about the conditions in the cell are made. It may be possible to infer direction using other computational techniques, such as flux-balance analysis¹⁸. The topology information from the pathwayStep instances in the pathway class should not be used for direction information, nor should it be assumed that the default direction of conversions is in either the LEFT-to-RIGHT or RIGHT-to-LEFT directions.

Conventions for LEFT and RIGHT

As stated above, substrates and products of a conversion may be placed in either the LEFT or the RIGHT properties as these are not used to determine the direction of a conversion. However, in order to ease data integration, it is preferable that users adhere to the same conventions for the contents of these properties. We therefore recommend the following, in order of precedence:

1. If the conversion has an Enzyme Commission (EC) number or a Transport Commission (TC) number, store the participants in the LEFT and RIGHT properties such that they mirror the EC/TC reaction.
2. For complex assemblies, store the subunits in the LEFT property and the complex in the RIGHT property.
3. For transport instances, store the outermost participants (relative to the interior of the cell or organelle) in the LEFT property and the innermost participants in the RIGHT property.
4. If none of the above are applicable, store the participants from left-to-right in the order that the conversion occurs or is suspected to occur in the pathway.

Technical note: OWL and RDF Conventions

A typical set of BioPAX data consists of many instances of various BioPAX classes. Each of these instances must be given an RDF ID that is unique within the document to be a valid OWL/RDF document. These IDs are used to reference instances from other parts of the OWL document. When combined with a unique document namespace, these IDs form a URI that can provide a globally unique identifier for each BioPAX instance.

RDF ID

In an OWL document, such as BioPAX, each instance of a class must have an RDF ID. This comes from the Resource Descriptor Framework standard (<http://www.w3.org/RDF/>). These IDs must be unique and are used to reference instances within a document. An RDF ID exists within a namespace, which can be explicitly appended before the RDF ID. If not explicit, the RDF ID exists in the default namespace of the document. Like anchors in HTML, a pointer to an RDF ID defined elsewhere in the document is denoted with a hash mark (“#”) in front of the RDF ID.

Example

```
<protein rdf:ID="protein76">  
  <XREF rdf:resource="#xref1146"/>  
</protein>
```

It is recommended that RDF IDs do not encode any semantics and be composed of the class name followed by a unique positive integer (e.g. “protein76”) or some other naming convention that guarantees unique names within the file. Some applications that use OWL, such as Protégé, and some examples of OWL from the main OWL website, use human readable names for the

RDF IDs. As long as these names are unique, a BioPAX document will be valid, but the use of human readable names as RDF IDs might encourage people to rely on information stored in them and is thus not recommended. RDF IDs may not persist after certain data processing operations, such as integrating data from two separate BioPAX files.

Please note that in the Protégé tool, the RDF ID of an instance is referred to as its Name. This should not be confused with the BioPAX NAME (all letters capitalized) property, which is meant to provide the human readable name for biological entities (Figure 1). Protégé can be configured to display the value of the NAME property (or another field value) instead of the RDF ID. Use the Display Slot pull-down menu in the Individuals tab to select the property to display.

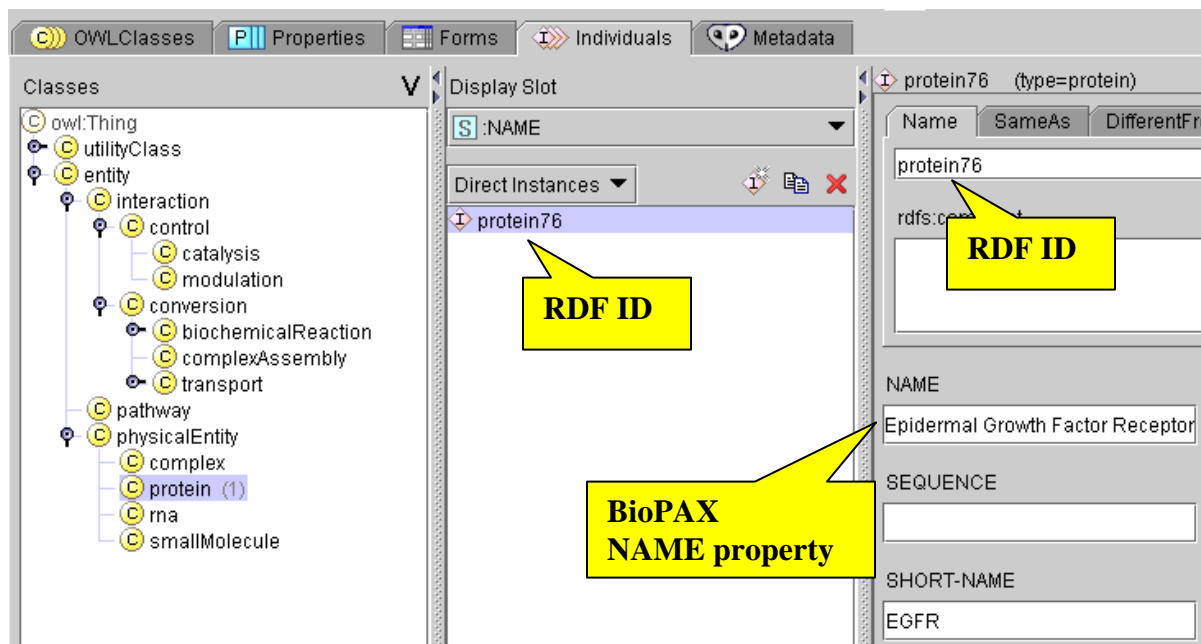


Figure 1: The difference between name and RDF ID shown in Protégé.

Document namespace

OWL XML documents require a default namespace. The creator of the BioPAX document should create a namespace and encode it in the BioPAX document. The namespace and the RDF ID may be used together to reference instances in a document from an external document (explicit use of namespace). This reference mechanism is part of the basis of the planned Semantic Web (<http://www.w3.org/2001/sw/>). If a BioPAX document is going to be on the Semantic Web, it should have a unique namespace. Since there is no namespace naming authority, it is not possible to guarantee unique namespaces across the internet, but following these recommendations will reduce the chances of naming collisions.

Technically, any string without spaces is allowed (see [namespace rules](#)) as a namespace. Operationally, a URL (or more generally a URI) should be used. This does not have to be a 'real' URL that resolves to a web page, but it should be related to the organization of the creator and a

registered domain name owned by the organization is useful to include e.g. “<http://biocyc.org/ontology/biopax/#>”.

Use of the `xmlns` and `xml:base` attributes to specify the namespace for any BioPAX documents created is recommended. The BioPAX ontology definition should be imported and the BioPAX namespace should be defined using the ‘bp’ string (if it does not conflict with other existing namespaces called ‘bp’) e.g. `xmlns:bp=http://www.biopax.org/release/biopax-level1.owl`, so that elements in the file appear like this: `<bp:pathway></bp:pathway>`.

A typical header of an OWL XML document that uses the BioPAX ontology will look like this:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://www.myorganization.org/ontology#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xs="http://www.w3.org/2001/XMLSchema#"
  xmlns:bp="http://www.biopax.org/release/biopax-level1.owl#"
  xml:base="http://www.myorganization.org/ontology#"
>
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.biopax.org/release/biopax-level2.owl"/>
</owl:Ontology>
```

Where “<http://www.myorganization.org/ontology#>” defines the namespace for this document.

OWL XML documents that mix BioPAX definitions with those from other ontologies or extend BioPAX will have different ways of using namespaces, but those are not dealt with here. Extensions to the BioPAX ontology are not expected to be compatible with tools written specifically to support the BioPAX ontology.

BioPAX and the Semantic Web

The semantic web is a proposed next-generation world wide web which encodes data using RDF, which makes the encoded information more easy for software to read and make logical inferences on. This technology also promises to make semantically coherent data integration easy, though does not yet deliver on this promise outside of research projects. Once the semantic web is more mature and reliable services are provided for working with RDF IDs and distributed instance data, BioPAX will migrate to using semantic web conventions in addition to the classical use of xrefs in bioinformatics. As of 2005, the technology is not mature enough to rely upon for mission-critical pathway data exchange, thus bioinformatic conventions for data integration are used.

A number of biological object repositories and controlled vocabularies have been defined using OWL ontologies other than BioPAX. Using the URIs for these objects or terms as the values for BioPAX properties, instead of using BioPAX instances of the appropriate classes, is not

recommended. Although such a practice may be in accordance with semantic web conventions, it would make data exchange with BioPAX difficult.

The main reason for this is that software built to work with BioPAX data would have no idea how to resolve the reference. For example, suppose a software developer created a program to diagram BioPAX pathways. Suppose also that the developer wanted to label each interaction participant with its location, and extracted this information from the TERM property of the openControlledVocabulary instances found in participant LOCATION properties. If a provider of BioPAX data decided to use a URI to an object in an external ontology (e.g. an OWL version of the Gene Ontology) in the LOCATION property, instead of openControlledVocabulary instances, the visualization software would likely fail because there would be no TERM property to check, just a URI to an arbitrarily structured class.

5 HOW-TO

Creating a knowledge-base using BioPAX and Protégé

Protégé is an open-source ontology and knowledge-base editor from Stanford University. It can be used to view and edit the BioPAX ontology and to create a database of instances of BioPAX classes. The ezOWL plugin allows graphical viewing and editing of an ontology, but does not graphically show instances.

To use BioPAX in Protégé, it is necessary to download three components:

1. Protégé from <http://protege.stanford.edu/>
Downloading the current stable release and not the beta release is recommended.
2. The Protégé OWL Plugin from <http://protege.stanford.edu/plugins/owl/index.html>
This allows Protégé to understand the OWL format. (This is normally included with the 'Full' download of Protégé)
3. The Protégé ezOWL Plugin from <http://iweb.etri.re.kr/ezowl/>
Used to view and edit an OWL ontology graphically.

Follow the instructions for installing these applications provided on their respective web pages.

After installing the above software, create a new OWL Files project in Protégé. To load the BioPAX ontology, one may either: a) import the BioPAX OWL file from the web (recommended), or b) load the ontology from a local copy of the BioPAX OWL file.

To import from the web:

- 1) On the Metadata tab (on the main screen next to the Individuals tab) click on the + symbol near where it says "Namespace Prefixes". This will add a new namespace to the project. The new namespace must be manually changed from "http://www.domain2.com#" to "http://www.biopax.org/biopax-level1.owl#" and check the "Imported" checkbox. The prefix should be changed from "p1" to "biopax".
- 2) Save the project as an OWL file, then reload it (via the "Project→Import..." menu item).
- 3) Upon reloading, the BioPAX ontology will be visible (grayed-out) in the classes menu.
- 4) Use the Individuals tab to create instances.
- 5) To use ezOWL, check the "ezOWLtab" checkbox in the dialog box that appears under the "Project->Configure..." menu item. This allows viewing of the BioPAX ontology structure in the normal Protégé tab and in the ezOWL tab.

Note: This method of importing BioPAX into Protégé prevents inadvertently made changes to the imported BioPAX classes; changing the ontology is not recommended if the instance data are meant to be shared.

To import from a local copy of the BioPAX OWL file:

- 1) Load the BioPAX OWL file via the "File → Build new project..." menu item (Protégé v3.0). In the resulting dialog box, select "OWL Files" and browse to the BioPAX OWL

file on the local computer disk drive by clicking on the + symbol next to “OWL file name” and press “OK”. Protégé will load BioPAX.

- 2) Upon loading, the BioPAX ontology will be visible (not grayed-out) in the classes menu.
- 3) Use the Individuals tab to create instances.
- 4) To use ezOWL, check the “ezOWLtab” checkbox in the dialog box that appears under the “Project->Configure...” menu item. This allows viewing of the BioPAX ontology structure in the normal Protégé tab and in the ezOWL tab.

Note: This method of importing BioPAX into Protégé does not prevent inadvertently made changes to the imported BioPAX classes; changing the ontology is not recommended if the instance data are meant to be shared.

Protégé can be used as a full-fledged customizable database and data entry system, although it requires programming effort. For example, Reactome (<http://www.reactome.org>) uses Protégé as its backend system. If used this way, it may be desirable to modify the BioPAX ontology and create inverse properties for convenience. These properties should be removed in shared data files in order to make them compliant with the BioPAX standard.

Viewing Instances Graphically

Instances can be graphically viewed with a number of Protégé plugins that ship with the ‘Full’ protégé download. For instance, the Ontoviz plugin enables a highly customized view of instances in an OWL file.

6 Use Case Outlines

These use-cases were taken into account during the design of BioPAX. Other use-cases may be suggested via the biopax-discuss@biopax.org mailing list.

Data Sharing Between Databases

One of the primary intended functions of the BioPAX format is to facilitate data exchange between existing biological pathway databases. In order for this to happen, databases must develop the ability to write-to and read-from the BioPAX format. Typically, this will require the creation of in-house software. While a number of freely available software packages may help make this task easier (e.g. Jena, an open source Java API for RDF; see <http://jena.sourceforge.net/index.html> or the Protégé OWL API; see <http://protege.stanford.edu/plugins/owl/>), development of data translation software may nonetheless require a fair amount of programming time for each individual database.

The typical data transaction, i.e. passing a set of data from one database to another, will consist of a number of steps. These steps will vary depending on the particular situation, but in general they should consist of the following:

- 1) Convert a set of data into the BioPAX format. This step involves mapping the native data model to the BioPAX data model (i.e. the BioPAX ontology) and then creating a BioPAX OWL file that contains instances of the mapped classes. This step will almost always require developing software to perform the mapping.
- 2) Transfer the BioPAX file. There are many mechanisms by which this could be accomplished, e.g. the data provider could make the file available for download from an FTP or HTTP server.
- 3) Convert the BioPAX file into the native format of the receiving database (the reverse of step 1). Again, this will likely require new software to perform the data conversion.
- 4) Merge data sets and remove redundancies. Often, many instances in a BioPAX file may already exist in the target database (Note: these are only detectable if the redundant instances share one or more unification x-refs or if entire instances are compared). These instances should be merged with the existing data (if they contain additional information not present in the database) or removed from the data set being imported (if not) to prevent redundant entries from being created. Also, any pointers to such instances must be redirected to the existing database objects.

As more datasets become available in the BioPAX format, software utilities will be developed (by members of the BioPAX group and others) to ease data sharing. For example, a utility to integrate the data from two different BioPAX files would be useful. With such a utility, users could integrate new BioPAX data with their own by first outputting their data into BioPAX format, then running the utility to combine it with the new data, then translating the combined data set back into their own format. Thus, the need for system-specific data integration software (step 4 above) would be reduced.

BioPAX as a knowledge-base (KB) Model

The BioPAX ontology is readily usable as the data model for a pathway knowledge-base (KB) using a tool like Protégé (<http://protege.stanford.edu>). Building a new KB with the BioPAX ontology would save time and resources since it would eliminate the need to create a data schema from scratch and it would reduce the translation requirement for exporting and importing data to/from the BioPAX format (some custom semantic mapping and ID mapping might still be required to import data from another database).

Of course, some users may wish to extend the BioPAX ontology to suit their own needs. For example, many KBs use “inverse properties” – properties that are the reciprocal of other relationship properties – in order to speed up queries and facilitate browsing. Since such properties provide redundant information, they were left out of the BioPAX ontology. See the HOW-TO section for more information on creating a BioPAX KB. Note that instances adhering to an altered BioPAX ontology are not compatible with the official BioPAX standard.

Pathway Data Warehouse

The initial motivation for creating the BioPAX standard was that it was seen as a logical first step toward creating a central public repository for biological pathway data, a resource strongly desired by many members of the pathway community. If many databases provide access to their data in the BioPAX format, it should be relatively simple to aggregate this data in a central repository.

Pathway Analysis Software

Another intended function of BioPAX is to speed development time of software that makes use of pathway data. Currently, in order for pathway software to access pathway data from multiple sources it must either be programmed to understand each different format, or the data from each source must be translated into a format that the software understands. This can require significant development time and as a consequence most pathway software is run on only a few datasets, limiting its utility.

The presence of a standard format and object model for pathway data should alleviate this problem. With the lower barrier to data access, pathway software will be easier to develop and apply. Also, additional software that might not be practical without an agreed upon standard, e.g. a sophisticated pathway visualization tool, may be more likely to be developed if BioPAX becomes widely adopted.

Pathway Analysis Software Example: Molecular profiling analysis

Genomics and proteomics technologies, such as gene expression microarrays and mass spectrometers, are being used to generate large datasets of molecules present at a specific place and time in an organism (molecular profiling), among other types of data. Molecular profiling

experiments are often compared across two or more conditions (e.g. normal tissue and cancerous tissue). The result of this comparison is often a large list of genes that are differentially present in the tissue of interest. It is interesting and useful to analyze these lists of genes in the context of pathways. For instance, one could look for pathways that are statistically over-represented in the list of differentially expressed genes. The result is a list of pathways that are active or inactive in the condition of interest compared to a control. The list of pathways is often much shorter than the list of input genes, thus is easier to comprehend. BioPAX documents describing pathways could be supported by tools that perform pathway-based analysis.

Visualizing Pathway Diagrams

Pathway diagrams are useful for examining pathway data. A number of formats are available for these images, but only few available viewing tools link components in the image to underlying data. A mapping of BioPAX to a symbol library for pathway diagrams (such as Kohn maps - <http://discover.nci.nih.gov/kohnk/symbols.html>) could be the basis for a general BioPAX pathway diagram tool.

Pathway Modeling

Mathematical modeling to understand the dynamics of a pathway system is a frequent use of pathway information. Qualitative modeling requires information about components in the pathway and their connections, as well as some qualitative knowledge of rates (e.g. fast, slow) and concentrations of the components (e.g. high, medium, low). Quantitative modeling additionally requires such things as measured rate constants, stoichiometry and initial concentrations in order to quantitatively predict pathway behavior. Many tools are available for this type of modeling, and the SBML (<http://sbml.org>) and CellML (<http://www.cellml.org>) standards are available to describe the models, which many tools support. While BioPAX does not contain enough information to describe a pathway model as well as SBML and CellML, there are two envisioned use cases:

Using BioPAX as metadata for SBML and CellML

SBML and CellML, as model representation languages, focus on representing the structure, parameters and mathematical description of a pathway model. BioPAX focuses on molecule and interaction classification schemes and database cross-referencing for pathway components. BioPAX and SBML or CellML could be linked together when a user wants both a full model description and information about types of pathway components and database links. A hybrid XML document containing BioPAX and SBML or CellML elements that are tied together using the CellML metadata standards could be created that fills this need.

Pathway analysis using logical inference

One advantage of representing BioPAX pathway data in OWL format is the availability of logical inference tools that support OWL. These tools are useful for analyzing pathways. For example, given a metabolic network model for an organism in BioPAX format, a known minimal

nutrient media for that organism and the set of compounds essential for growth under one set of living conditions, then a transitive closure computation of the minimal nutrient set can be used to verify if the metabolic network model of the organism is sufficient to explain growth. If any essential compound is not reachable through the network from the minimal nutrient list, then the network model is incomplete.

7 Glossary

Some of the following definitions may only relate to BioPAX, thus may not be general.

Biological pathway: A working definition of a pathway is a series of molecular interactions and reactions (or other biological relationships), often forming a network. For molecular pathways, the start and end points are often defined by observation of a detectable phenotype after stimulation or perturbation, such as observing gene expression after stimulating the cell with a peptide growth hormone.

Class: Used in knowledge representation to represent a category of things. A specific member of a class is called an instance.

Data exchange format: Any data format, usually electronic, used to exchange data.

GKB Editor: Generic Knowledge Base Editor. A software tool to build an ontology and manage instances of classes defined in that ontology. <http://www.ai.sri.com/~gkb/>

Instance: An instance or particular member of a class. Known as 'individual' in OWL.

Ontology: A system for describing knowledge, a conceptualization of a domain of interest usually made up of any or all of the following: concepts (classes), relations, attributes, constraints, objects, values. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>

OWL: Web ontology language, a proposed W3C standard, is an extension of RDF to support ontologies. It provides semantics for classes and subclasses, instances, and relationships. <http://www.w3.org/TR/owl-features/>

Property: A 'field' or 'member' of a data structure. Known as a 'slot' in many knowledge representation systems.

Protégé: Protégé ontology and knowledge base editor. A software tool to build an ontology and manage instances of classes defined in that ontology. <http://protege.stanford.edu/>

RDF: Resource Description Framework, a proposed W3C standard, allows description of basic relationships between objects (subject-predicate-object semantics). <http://www.w3.org/TR/rdf-primer/>

Appendix A: Design Principles

Flexible: Biological pathway data are organized and represented in various ways depending on the type of data and its intended use. BioPAX must support the most frequently used representations to be widely accepted. Of course, there is a trade-off that must be considered: increased flexibility may increase data integration overhead. For example, the issue of semantic mapping between different representation styles must be dealt with when users wish to integrate BioPAX data sets that use different representations. Therefore, BioPAX should strike a reasonable balance between flexibility and rigidity by allowing multiple preferred representations and providing best practice recommendations to encourage consistent data representation.

Extensible: Biological pathway data are available in various forms and at varied levels of detail. BioPAX aims to initially support the most frequently used types of pathway data and levels of detail and to progressively broaden support for additional pathway data types and finer detail through a leveled approach. The class structure of BioPAX was designed to be extensible for this reason. Many parts of the BioPAX ontology, such as internal controlled vocabularies and many of the intermediate level classes, will be extended in future BioPAX levels. All efforts will be made to keep future levels backwards compatible.

Encapsulation: Pathway data depends on many primary databases of physical entities (e.g. proteins, small molecules, etc.). Many pathway data sets reference physical entities using database identifiers. Because of the varied nature of the physical entity databases, resolving these identifiers in a general way can be difficult, especially for the naïve user. Frequently used data about the physical entities (e.g. sequence for proteins, structure for small molecules) is optionally present (encapsulated) in the BioPAX format for convenience.

Compatible: BioPAX uses existing standards for encoding biological pathway information to avoid “re-inventing the wheel”. Specifically, pointers to the Gene Ontology (GO), and instances of Chemical Markup Language (CML) and the SMILES format are used in various properties in the ontology. Also, compatibility with other pathway standards, such as SBML, CellML, and PSI-MI has influenced the design of many BioPAX features.

Computable: BioPAX stores data in a format that supports many different types of computational analysis. Values are strongly typed and the class structure is clearly defined. A wide range of computational tasks, from simple reading and parsing of a BioPAX file to logical inference based on the data, are supported. The OWL version of the BioPAX ontology complies with the OWL-DL sublanguage and is thus interpretable by description logic software such as RACER (<http://www.sts.tu-harburg.de/~r.f.moeller/racer/>) and should be compatible with future software built to work with the Semantic Web.

Appendix B: Level and Version Numbers

BioPAX level numbers indicate the relative scope of the ontology. BioPAX Level 1 focuses on metabolic pathway data; subsequent levels will expand this scope to include other types of data such as molecular binding interactions and signal transduction pathways. BioPAX level numbers are always whole numbers (e.g. Level 1, version 1.0).

In addition to the level numbers, BioPAX version numbers indicate the relative stage of development of each level. Version numbers are a composite of two individual integers: the major version number and the minor version number separated by a decimal point to form the composite version number (e.g. Level 1, version 1.1). The major version number appears before the first decimal point and is only incremented when an update is likely to affect existing data. Releases in which the major version is 0 are early draft releases of their respective levels (e.g. Level 1, version 0.5).

The minor version appears after the first decimal point and is incremented when an update is unlikely to affect data that conforms to the prior version. Odd minor version numbers indicate beta versions, while even minor version numbers indicate release versions.

For example, the first non-draft release of every level is version 1.0 (major version 1, minor version 0). If this version would need to be updated, the first beta version of the update would be called version 1.1. When no further revisions were needed, a release version of the update would be created called version 1.2 if the revisions would not affect data that complied with version 1.0, or version 2.0 if they would.

All versions of BioPAX are available in the following directory on the BioPAX website:
<http://www.biopax.org/Downloads/>

The most recent major versions of each level of BioPAX are always available in this directory:
<http://www.biopax.org/release/>

Acknowledgements

The BioPAX workgroup thanks members of the community who have contributed to this work through discussions: Melissa Cline, Autumn Cuellar, Emek Demir and the PATIKA group, Andrew Finney, Ken Fukuda, Matt Halstead, Mike Hucka, Stan Letovsky, Peter Murray-Rust, the PSI-MI workgroup and others who have contributed through involvement in the biopax-discuss list, seminar participation and birds of a feather (BOF) sessions at conferences.

References

1. Baxevanis, A. D. & Ouellette, B. F. F. *Bioinformatics : a practical guide to the analysis of genes and proteins* (Wiley-Interscience, New York, 2001).

2. Alberts, B. *Molecular biology of the cell* (Garland Science, New York, 2002).
3. Stein, L. Creating a bioinformatics nation. *Nature* **417**, 119-120 (2002).
4. Hermjakob, H. et al. The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. *Nat Biotechnol* **22**, 177-83 (2004).
5. Karp, P. D. et al. The EcoCyc Database. *Nucleic Acids Res.* **30**, 56-58 (2002).
6. Krieger, C. J. et al. MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res* **32 Database issue**, D438-42 (2004).
7. Bader, G. D., Betel, D. & Hogue, C. W. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res.* **31**, 248-250 (2003).
8. Overbeek, R. et al. WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic Acids Res* **28**, 123-125 (2000).
9. Demir, E. et al. PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics.* **18**, 996-1003 (2002).
10. 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).
11. 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).
12. Hucka, M. et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524-31 (2003).
13. Murray-Rust, P. & Rzepa, H. S. Chemical markup, XML, and the World Wide Web. 4. CML schema. *J Chem Inf Comput Sci* **43**, 757-72 (2003).
14. Weininger, D. SMILES, a Chemical Language and Information System. *J.Chem.Inf.Comput.Sci.* **28**, 31-36 (1988).
15. The_Gene_Ontology_Consortium. Gene ontology: tool for the unification of biology. *Nat.Genet.* **25**, 25-29 (2000).
16. Karp, P. D. Database links are a foundation for interoperability. *Trends Biotechnol* **14**, 273-9 (1996).
17. Wheeler, D. L. et al. Database resources of the National Center for Biotechnology. *Nucleic Acids Res.* **31**, 28-33 (2003).
18. Edwards, J. S., Covert, M. & Palsson, B. Metabolic modelling of microbes: the flux-balance approach. *Environ Microbiol* **4**, 133-40. (2002).