A simple scheme for annotating SBML with references to controlled vocabularies and database entries

Nicolas Le Novère, Andrew Finney
lenov@ebi.ac.uk,afinney@caltech.edu

December 8, 2005

Contents

Abstract

The SBML community requires a standardized mechanism for annotating SBML models with terms taken from controlled vocabularies and from other external bioinformatics resources. The metadata framework copied from CellML into SBML Level 2 Version 1 turned out to be very complex, and therefore poorly supported by the community. The present document proposes a more detailed subset standard of this framework, sufficient for most used needs. Mainly based on the use of references to external resources, rather than embedding the information in the SBML files themselves, it results in simpler and more easely parsable files. This framework is already used by the database BioModels and the SBMLeditor to annotate the main components of the models. We hope that this proposal will stimulate debate as to the best practice in this area in which there is a obvious need but as yet no consensus.

1 Aims

The exchange and composition of models require not only standards to exchange models and understand their structure, but also ways of exchanging additional information in order to make sense of the model elements and interpret the results of simulations. The SBML community therefore needs a standardized mechanism for annotating SBML models with terms taken from controlled vocabularies and from other external bioinformatics resources. Such an additional information permits to precisely identify model components such as species, and to replace them into the biological context.

SBML Level 1 (Hucka et al., 2003b,a) did not include any formal representation for such metadata. SBML Level 2 Version 1 (Finney and Hucka, 2003b,a) adopted the CellML (CellML; Lloyd et al., 2004) metadata frameworkL (Cuellar et al.) based on RDF (RDF) and using a mix of additional standards, such as Dublin Core (DublinCore), vCard (vCard) and OpenBQS (Senger). The resulting specification turned out to be quite complex. In addition, the flexibility of the scheme had to be paid by a lack of standardisation. For instance, there are several ways of describing the authors of an article, based on vCard or BQS. As a consequence, the CellML metadata framework was so far poorly supported by the SBML community.

In addition to the difficulty to implement software support, other problems of SBML Level 2 Version 1 metadata come from the storage of information in the model itself. The most serious issue is the possible obsolescence of the information. Most of the data resources are constantly evolving, the information they contain improving over time. Quite often, only a unique identifier is perenial. Examples of changing information are UniProt (UniProt; Apweiler et al., 2004) names (equivalent to Swiss-Prot ID) or Gene Ontology (GO; Ashburner et al., 2000) terms. Another burden to carry with the local storage of information is the redundancy of information (cf examples below). This procedure results in verbose structures, increasing unecessarily the size of the SBML files.

This document proposes a simple syntax to use within the SBML Level 2 Version 1 standard, for the annotation of models with references to bioinformatics resources. The idea is to precisely identify a piece of information in an external bioinformatics resource that has a well defined relationship with a given SBML element, rather than copy part of the resource within the SBML file. The approach described is designed to be simple, highly extensible and *compliant with the existing SBML Level 2 metadata specification*.

2 Syntax for Referring to Controlled Vocabulary Terms and Database Identifiers

In this section we describe the proposed format for referring to controlled vocabulary terms and database identifiers. The annotation of an element is located in an RDF block. The content of an `rdf:RDF` element must conform to the RDF/XML Syntax Specification recommendation from the W3C (RDFsyntax). The syntax consists of Dublin Core `relation` elements, embedded in RDF `Description` elements that refer to the SBML elements where they are embedded. The Dublin Core elements follow the Guidelines for implementing Dublin Core in XML (Powell and Johnston).

We will present two flavours of annotations: with or without qualifications.

2.1 Non-qualified dublin-core annotation

The first level of annotation uses unqualified Dublin Core annotation (Beckett et al.). The SBML component is linked to a precise external piece of information, but nothing specifies how exactly the annotation relates to the SBML component. The external information is described by a Unique Resource Identifier (URI) (Berners-Lee et al.) contained in the `rdf:resource` attribute of an element `dc:relation`, itself contained in an RDF block present in the `annotation` element of a component of the model. For instance a species representing a protein could be annotated with a reference to the database UniProt by the `http://www.uniprot.org/#P12999` identifier, identifying exactly the protein described by the species.

This identifier maps to a unique entry in UniProt which is never deleted from the database. In the case of UniProt, this is the "accession" of the entry. When the entry is merged with another one, both "accession" are conserved. Similarly in a controlled vocabulary resource, each term is associated with a perenial identifier. The UniProt entry also possess an "entry name" (the Swiss-Prot "identifier"), a "protein name", "synonyms" etc. Only the "accession" is perennial and should be used.

The value of the rdf:resource attribute is a URI that both uniquely identifies the resource, and the data in the resource. The resource constraining the identifier precedes the '#' symbol and the term or database identifier follows the '#' symbol. In the present example, the resource http://www.uniprot.org/ includes the entry P12999.

Note that the value of the rdf:resource attribute is a URI, not a URL; as such, a URI does not have to reference a physical web object but just identifies a controlled vocabulary term or database object (think of a URI as a label that, in this case, just happens to look like a URL). For instance, the URL http://www.uniprot.org/entry/P12999 would correspond to the URI http://www.uniprot.org/#P12999.

Note that nothing specifies how a tool is to interpret a URI. In the case of a transformation into a physical URL, there could be several solutions. For instance, the URI http://www.geneontology.org/#GO:0007268 can be translated into:

http://www.ebi.ac.uk/ego/DisplayGoTerm?selected=GO:0007268
http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details\&query=GO:0007268
http://www.informatics.jax.org/searches/GO.cgi?id=GO:0007268

Similarly the URI http://www.ebi.ac.uk/intenz/#EC3.5.4.4 can refer to:

http://www.ebi.ac.uk/intenz/query?cmd=SearchEC\&ec=3.5.4.4
http://www.expasy.org/cgi-bin/nicezyme.pl?3.5.4.4
http://www.chem.qmul.ac.uk/iubmb/enzyme/EC3/5/4/4.html
http://www.genome.jp/dbget-bin/www_bget?ec:3.5.4.4
etc.

To enable interoperability, the community will have to agree on a set of standard valid URI syntaxes. These URIs will always be composed as "resource#id". URI syntax rules would not be a fixed part of the SBML standard but would be extendable independently from specific SBML Levels and Versions. We will set up a web page available through sbml.org that points to a new website, biomodels.net, where we will list URI syntaxes and possible physical links to controlled vocabulary and databases. This list will simply enumerate the set of strings that could precede the '#', and for each member of this list there would be a brief summary of the syntax for the identifier following the '#'. A subset of this forthcoming list is presented at the end of this document. An API would be set-up so that a tool could retrieve automatically valid URL(s) corresponding to a given URI. The list would evolve with the evolution of databases and resources. Note that this annotation scheme doesn't require such a list to operate.

The use of elements rdf:Bag allows multiple links to external resources for a given SBML object as shown in the following example, describing the two components of the human complex calcium/calmodulin kinase II by their UniProt and KEGG compound accessions and a Gene Ontology term.

```
<species id="CaCaMKIIspecies" metaid="z666">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
    >
      <rdf:Description rdf:about="#z666">
        <dc:relation>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.uniprot.org/#P62158"/>
            <rdf:li rdf:resource="http://www.uniprot.org/#Q9UQM7"/>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/compound/#C00076"/>
            <rdf:li rdf:resource="http://www.geneontology.org/#GO:0005954"/>
          </rdf:Bag>
        </dc:relation>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

The value of the attribute rdf:about present in the the element rdf:Description should match the value of the corresponding attribute metaid in the SBML element it refers to. Technically, the use of metaid and rdf:about attributes enables any number of RDF elements to be placed anywhere in the SBML document. However, we recommend as a best practice to place the rdf:Description element for an SBML element within the annotation element for that SBML element (this enables the editing and deletion of the annotation to be managed in a straightforward manner). Ideally an annotation element should contain only one nested sequence of rdf:RDF, rdf:Description, dc:relation and rdf:Bag elements.

The annotation can be located at different depths within a model component. For instance the following reaction contains “direct” annotation, but its kineticlaw is also annotated, referring to the forthcoming ratelaw control vocabulary.

```
<reaction id="calciumBinding" metaid="jb007">
  <annotation>
    <rdf:RDF
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    >
      <rdf:Description rdf:about="#jb007">
        <dc:relation>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.geneontology.org/#GO:0042166"/>
          </rdf:Bag>
        </dc:relation>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
  <listOfReactants>
    <speciesReference species="calmodulin"/>
    <speciesReference species="calcium" stoichiometry="4"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="calcium/calmodulin"/>
  </listOfProducts>
  <kineticLaw metaid="hal9000">
    <annotation>
      <rdf:RDF
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      >
        <rdf:Description rdf:about="#hal9000">
          <dc:relation>
            <rdf:Bag>
              <rdf:li rdf:resource="http://www.biomodels.net/CVs/rateLaw/#SBO:0000001"/>
            </rdf:Bag>
          </dc:relation>
        </rdf:Description>
      </rdf:RDF>
    </annotation>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times/>
        <ci>kon</ci>
        <ci>calmodulin</ci>
        <apply>
          <power/>
          <ci>calcium</ci>
          <cn>4</cn>
        </apply>
      </apply>
    </math>
    <listOfParameters>
      <parameter id="kon" value="1"/>
    </listOfParameters>
  </kineticLaw>
</reaction>
```

2.2 Qualified Dublin-Core annotation

Although useful in most of the cases, the unqualified annotation is sometimes not sufficient. For instance, how is the following annotation to be interpreted?

```
<species id="species" metaid="r2d2">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
    >
      <rdf:Description rdf:about="#r2d2">
```

```
          <dc:relation>
            <rdf:Bag>
              <rdf:li rdf:resource="http://www.uniprot.org/#P04551"/>
              <rdf:li rdf:resource="http://www.uniprot.org/#P10815"/>
              <rdf:li rdf:resource="http://www.uniprot.org/#P40380"/>
            </rdf:Bag>
          </dc:relation>
        </rdf:Description>
      </rdf:RDF>
    </annotation>
  </species>
```

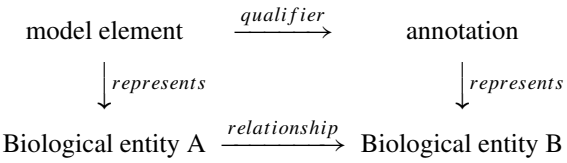
How can a parser tell the difference between the previous and the following example.

```
<species id="species" metaid="c3po">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
    >
      <rdf:Description rdf:about="#c3po">
        <dc:relation>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.uniprot.org/#P02942"/>
            <rdf:li rdf:resource="http://www.uniprot.org/#P07017"/>
            <rdf:li rdf:resource="http://www.uniprot.org/#P05704"/>
          </rdf:Bag>
        </dc:relation>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

In the former case, the species represents a complex between the human proteins cdc2, cdc13 and rum1. The annotations point to components, parts of the species. In contrast, in the second case, the species represent a chemotaxis receptor, and the annotations point to homologous receptors. They can all be represented by the species, but not at the same time.

To refine the annotation, the Dublin Core consortium introduced terms that refine the relation. This qualification can be expressed in RDF (Kokkelink and Schwänzl). The qualifiers are listed in the main DCMI Metadata Terms (Board).

In our case, the qualifier of an annotation should reflect the relationships between the biological objects represented by the model element and the annotation:



SBML annotation can only make use of five of those qualifiers.

hasPart	The described component includes the subject of the referenced resource, either physically or logically
isPartOf	The described component is a physical or logical part of the subject of the referenced resource
isVersionOf	The described component is a version, an instance of the subject of the referenced resource
hasVersion	The subject of the referenced resource is a version, an instance of the described component
isReferencedBy	The described component is referenced, cited, or pointed to by the referenced resource

In addition, the Dublin Core `relation` element, already used in the previous section, takes now a more precise meaning, that is perfect match: "The described component is the subject of the referenced resource".

(NB: It is to be noted that those qualifiers were defined by the Dublin Core in a slightly different purpose. A relation “hasPart” meant that the described document physically contained the referenced document, and a relation “isVersionOf” meant that the described element was derived from the referenced document. In the present system, the component of the model described by the element is linked to a piece of knowledge described in the reference

resource. Although the two concepts are on different logic level, the relationship between the object described and the object referenced are the same.)

The following example describes a species that represents a complex between the protein calmodulin and calcium ions:

```
<species id="Ca_calmodulin" metaid="cacam">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
    >
      <rdf:Description rdf:about="#cacam">
        <dcterms:hasPart>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.uniprot.org/#P62158"/>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/compound/#C00076"/>
          </rdf:Bag>
        </dcterms:hasPart>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

The following example describes a species that represents either “Calcium/calmodulin-dependent protein kinase type II alpha chain” or “Calcium/calmodulin-dependent protein kinase type II beta chain”. This is the case for instance in the comatic cytoplasm of striatal medium-size spiny neurons, where both are present, and one cannot functionaly differentiate them.

```
<species id="calcium_calmodulin" metaid="cacam">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
    >
      <rdf:Description rdf:about="#cacam">
        <dcterms:hasVersion>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.uniprot.org/#Q9UQM7"/>
            <rdf:li rdf:resource="http://www.uniprot.org/#Q13554"/>
          </rdf:Bag>
        </dcterms:hasVersion>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

Note that this approach should not be used to describe “any Calcium/calmodulin-dependent protein kinase type II chain”. Because in this case we need an annotation representing the other genes such as gamma or delta. One could enumerate all the known proteins, but such an approach would almost surely lead to inaccuracies due to the evolution of biological knowledge. What one should do instead is to refer to a generic information such as Ensembl family ENSF00000000194 “CALCIUM/CALMODULIN DEPENDENT KINASE TYPE II CHAIN” or PIR superfamily PIRSF000594 “Calcium/calmodulin-dependent protein kinase type II”.

The following two examples show how to use the qualifier `isVersionOf`. While with `HasVersion`, the described component could represent several alternative, with `isVersionOf` the described component is one of the alternative understated by the referenced resource.

A frequent example is the relationship between a reaction and an EC code. An EC code describe an enzymatic activity, and an enzymatic reaction involving a particular enzyme can be seen as an instance of this activity. For instance the following reaction represents the phosphorylation of a glutamate receptor by a complex calcium/calmodulin kinase II.

```
<reaction id="NMDAR_phosphorylation" metaid="thx1138">
  <annotation>
    <rdf:RDF
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
```

```
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    >
      <rdf:Description rdf:about="#thx1138">
        <dcterms:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/intenz/#EC 2.7.1.123"/>
          </rdf:Bag>
        </dcterms:isVersionOf>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
  <listOfReactants>
    <speciesReference species="NMDAR"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="P-NMDAR"/>
  </listOfProducts>
  <listOfModifiers>
    <modifierSpeciesReference species="CaMKII"/>
  </listOfModifiers>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times/>
        <ci>CaMKII</ci>
        <ci>kcat</ci>
        <apply>
          <quotient/>
          <ci>NMDAR</ci>
          <apply>
            </sum>
            <ci>NMDAR</ci>
            <ci>Km</ci>
          </apply>
        </apply>
      </apply>
    </math>
    <listOfParameters>
      <parameter id="kcat" value="1"/>
      <parameter id="Km" value="5e-10"/>
    </listOfParameters>
  </kineticLaw>
</reaction>
```

Another example is the complex between Calcium/calmodulin-dependent protein kinase type II alpha chain and Calcium/calmodulin, that is only one of the "calcium- and calmodulin-dependent protein kinase complexes" described by the Gene Ontology term GO:0005954.

```
<species id="CaCaMKII" metaid="C8H10N4O2">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
    >
      <rdf:Description rdf:about="#C8H10N4O2">
        <dcterms:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.geneontology.org/#GO:0005954"/>
          </rdf:Bag>
        </dcterms:isVersionOf>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

The previous case is different form the next one, although they could seem similar at first sight. The "Calcium/calmodulin-dependent protein kinase type II alpha chain" is a part of the abovementioned "calcium- and calmodulin-dependent protein kinase complex".

```
<species id="CaMKIIalpha" metaid="C10H14N2">
```

```
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
    >
      <rdf:Description rdf:about="#C10H14N2">
        <dcterms:isPartOf>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.geneontology.org/#GO:0005954"/>
          </rdf:Bag>
        </dcterms:isPartOf>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

Note that one can describe a component with several sets of qualified annotations. For instance, the following species represents a pool of guanosine phosphate, GMP, GDP and GDP. We annotate it with the three corresponding KEGG compound identifiers, but also with the three corresponding ChEBI identifiers.

```
<species id="GXP" metaid="GXP">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
    >
      <rdf:Description rdf:about="#GXP">
        <dcterms:hasVersion>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/#CHEBI:17345"/>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/#CHEBI:17552"/>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/#CHEBI:17627"/>
          </rdf:Bag>
        </dcterms:hasVersion>
        <dcterms:hasVersion>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/compound/#C00035"/>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/compound/#C00044"/>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/compound/#C00144"/>
          </rdf:Bag>
        </dcterms:hasVersion>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

The following example presents a reaction in a model that is actually the combination of three different elementary molecular reactions. We annotate it with the three corresponding KEGG reaction, but also with the three corresponding enzymatic activities.

```
<reaction id="adenineProd" metaid="adeprod">
  <annotation>
    <rdf:RDF
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    >
      <rdf:Description rdf:about="#adeprod">
        <dcterms:hasPart>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/intenz/#EC 2.5.1.22"/>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/intenz/#EC 3.2.2.16"/>
            <rdf:li rdf:resource="http://www.ebi.ac.uk/intenz/#EC 4.1.1.50"/>
          </rdf:Bag>
        </dcterms:hasPart>
        <dcterms:hasPart>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/reaction/#R00178"/>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/reaction/#R01401"/>
```

```
            <rdf:li rdf:resource="http://www.genome.jp/kegg/reaction/#R02869"/>
          </rdf:Bag>
        </dcterms:hasPart>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</reaction>
```

One can mix the type of annotations in a given set. The following represent two possible annotations of the human hemoglobine, one with ChEBI heme, the other with KEGG heme.

```
<species id="heme" metaid="heme">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
    >
     <rdf:Description rdf:about="#heme">
       <dcterms:hasPart>
         <rdf:Bag>
           <rdf:li rdf:resource="http://www.uniprot.org/#P69905"/>
           <rdf:li rdf:resource="http://www.uniprot.org/#P68871"/>
           <rdf:li rdf:resource="http://www.ebi.ac.uk/#CHEBI:17627">
         </rdf:Bag>
       </dcterms:hasPart>
       <dcterms:hasPart>
         <rdf:Bag>
           <rdf:li rdf:resource="http://www.uniprot.org/#P69905"/>
           <rdf:li rdf:resource="http://www.uniprot.org/#P68871"/>
           <rdf:li rdf:resource="http://www.genome.jp/kegg/compound/#C00032"/>
         </rdf:Bag>
       </dcterms:hasPart>
     </rdf:Description>
   </rdf:RDF>
  </annotation>
</species>
```

Finally, one can mix different qualified sets in the same annotation element. The following phosphorylation is annotated by its exact KEGG counterpart, and by the generis GO term “phosphorylation”.

```
<reaction id="phosphorylation" metaid="phosphorylation">
  <annotation>
    <rdf:RDF
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    >
      <rdf:Description rdf:about="#phosphorylation">
        <dc:relation>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.genome.jp/kegg/reaction/#R03313" />
          </rdf:Bag>
        </dc:relation>
        <dcterms:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="http://www.geneontology.org/#GO:0016310" />
          </rdf:Bag>
        </dcterms:isVersionOf>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</reaction>
```

2.3 Qualifiers missing in Dublin-Core annotation

Although useful in most cases, the current Dublin Core qualifiers cannot cope with all situations on could encounter when annotating models. For instance, there is no notion of homology. A model contains an enzymatic reaction in rat. I would like to annotate it with a Reactome entry (Joshi-Tope et al., 2003; Reactome), but it doesn’t exist. On the contrary the homologous reaction in man is present. The substrates are the same, the products are the same, and the rat

and human enzymes are descended from the same enzyme present in the common ancestor of rat and mouse. There is no element refinement `hasHomologue` or `isHomologueOf`.

We are looking for other missing qualification, and are welcoming any suggestion.

Several approaches can be envisioned to solve the problem:

1. Ignore the problem for now. We use an unqualified `dc:relation` to signify it is either semantically the same component, or the same in another species.
2. Try to get new element refinement in Dublin Core. Nicolas Le Novère contacted Dublin Core people to see how and when this would be possible.
3. Develop our own temporary qualifier in SBML namespace. This could be the only solution in the short-term. The advantage of this solution could be the possibility to enrich the vocabulary at will. For instance, we could have `hasHomologue`, `hasParalogue`, `hasOrthologue`.

3 Relationship to existing standards and proposals

This proposal is written in response to comments about a previous proposal for SBML Level 2 Version 2 discussed at the last SBML forum in Heidelberg (October 2004) see http://www.sbml.org/workshops/ninth/supplementary/sbml-level-2-version-2-proposal.pdf The scheme described in this document is better grounded in the RDF and Dublin Core standards than that initial proposal, is easier to parse and has the advantage that it can be adopted now as SBML L2 best practice. In addition it is already in use by at least one application software, SBMLeditor (Rodriguez et al.) and one database of models, BioModels database (BioModels-team).

3.1 Why not using CellML metadata?

SBML Level 2 Version 1 adopted the CellML metadata framework (Cuellar et al.). This specification is comprehense, but at the cost of complexity. In particular, in section 4.10 of the previous document. a "Biological Entity" is defined. In addition to the element `cmeta:bio_entity`, several other new CellML specific elements are created `cmeta:identifier`, `cmeta:identifier_scheme`, `cmeta:identifier_type`.

We propose that the CellML metadata `bio_entity` element is superseded by the scheme described here, based only on Dublin Core elements. We suggest that the CellML Metadata `bio_entity` element has an unnecessarily complex syntax scheme with resources refereed to via a fixed set of strings specific to CellML or alternatively via a URI. We believe that the scheme proposed here meets the needs of the community and is significantly simpler than the CellML `bio_entity` syntax.

For instance, the CellML example presented below describes "calmodulin", which is alternatively called "CaM" and related to the Swiss-Prot identifier "CALM_HUMAN". Not only are the two first pieces of information redundant (and useless - what if I call calmodulin "calm" instead of "CaM"?), but they are contained in the third one. In addition the Swiss-Prot "identifier" is not perenial (the "accession" is).

```
<rdf:RDF xmlns:cmeta="http://www.cellml.org/metadata/1.0#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:dcterms="http://purl.org/dc/qualifiers/1.0/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Description rdf:about="#element_metaid">
    <cmeta:bio_entity>
      <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
          <dc:title>calmodulin</dc:title>
          <dcterms:alternative>CaM</dcterms:alternative>
          <cmeta:identifier rdf:parseType="Resource">
            <cmeta:identifier_scheme>SWISS-PROT</cmeta:identifier_scheme>
            <rdf:value>CALM_HUMAN</rdf:value>
          </cmeta:identifier>
        </rdf:li>
      </rdf:Bag>
    </cmeta:bio_entity>
  </rdf:Description>
</rdf:RDF>
```

The equivalent with the scheme we present here would be:

```
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="#element_metaid">
    <dc:relation>
      <rdf:Bag>
        <rdf:li rdf:resource="http://www.uniprot.org/#P62158"/>
      </rdf:Bag>
    </dc:relation>
  </rdf:Description>
</rdf:RDF>
```

It is up to the application software to gather the human-readable information, such as name and definition, from a data resource understanding this accession (in this example, that could be a primary source, such as UniProt, PIR, Swiss-Prot or a secondary source, such as KEGG, Ensembl, etc.).

However, although more constrained, it is important to notice that the current annotation scheme is based on RDF and Dublin Core, and is compatible with the CellML metadata framework.

3.2 Relation with LSIDs

Life Science Identifiers (LSIDs) (Martin et al.) are Unique Resource Names (Moats) pointing to a unique piece of knowledge in the life science area. An example of LSID could be `urn:lsid:ebi.ac.uk:UniProt:P62158`. One can perfectly use LSIDs to fulfil our annotation purpose. Since a URN is effectively a URIs, one can directly use them in our dublin-core scheme. The previous example becomes:

```
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="#element_metaid">
    <dc:relation>
      <rdf:Bag>
        <rdf:li rdf:resource="urn:lsid:ebi.ac.uk:UniProt:P62158"/>
      </rdf:Bag>
    </dc:relation>
  </rdf:Description>
</rdf:RDF>
```

However, we feel that we should avoid LSIDs and go on with URIs for several reasons:

- LSIDs are attached to a perennial, unchangeable piece of knowledge. The described resource should always be the same, byte for byte. However, most of the data resource we want to use for SBML annotation are curated, and they get better all the time. We want to refer to a perennial, but perfectible piece of knowledge. Examples are UniProt entries, Gene Ontology terms etc. The identifiers used by those data resources are perennial, but the data they refer to is sometimes changed, or merged with another piece of knowledge etc.
- LSIDs use URNs rather than URIs. However, the latter are much more widespread than the former, and the toolbox richer. In addition, the root of our URIs generally bear a meaning, even if this is not always a physical location. Although this is not a fundamental advantage, it could play a role in the acceptance of the annotation system.
- LSIDs should be assigned and registered through "official" registrars. This would remove much of our control over the system and could impair community work. Finally, LSIDs are very much developed by private companies. Although there is nothing inherently wrong with that, the history told us that it was not the best prospect in terms of stability and long-term development (see the exemple of privately maintained web domain names versus government-backed ones).

Therefore, although we will support both URI schemes, we should be cautious with LSIDs.

4 Acknowledgements

Thanks for reading this far. Please let us know if you think the concepts described here are useful and should become part of SBML best practice and/or future SBML standards.

We'd like to acknowledge the assistance of Michael Hucka and Ben Bornstein in developing this proposal. Thanks to Marco Donizelli and Nicolas Rodriguez for their implementation of the proposal in BioModels database and SBM-Leditor.

References

R Apweiler, A Bairoch, CH Wu, WC Barker, B Boeckmann, S Ferro, E Gasteiger, H Huang, R Lopez, M Magrane, MJ Martin, DA Natale, C O'Donovan, N Redaschi, and LS Yeh. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res*, 32:D115–119, 2004.

M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25: 25–29, 2000.

Dave Beckett, Eric Miller, and Dan Brickley. Expressing simple dublin core in RDF/XML. Available via the World Wide Web at `http://dublincore.org/documents/dcmes-xml/index.shtml`.

T Berners-Lee, R Fielding, and L Masinter. Uniform resource identifier (uri): Generic syntax. Available via the World Wide Web at `http://www.gbiv.com/protocols/uri/rfc/rfc3986.html`.

BioModels-team. Biomodels database, a database of annotated published models. Available via the World Wide Web at `http://www.ebi.ac.uk/biomodels/`.

DCMI Usage Board. Dcmi metadata terms. Available via the World Wide Web at `http://www.dublincore.org/documents/dcmi-terms/`.

CellML. The CellML language. Available via the World Wide Web at `http://www.cellml.org/`.

Automn A Cuellar, Melanie Nelson, and Warren Hedley. The CellML metadata 1.0 specification. Available via the World Wide Web at `http://www.cellml.org/public/metadata/cellml_metadata_specification.html`.

DublinCore. Dublin core metadata initiative. Available via the World Wide Web at `http://dublincore.org/`.

Andrew Finney and Michael Hucka. Systems biology markup language: Level 2 and beyond. *Biochem. Soc. Trans.*, 31:1472–1473, 2003a.

Andrew Finney and Michael Hucka. Systems biology markup language (sbml) level 2: Structures and facilities for model definitions. Technical report, California Institute of Technology, 2003b.

GO. Gene ontology. Available via the World Wide Web at `http://www.geneontology.org/`.

Michael Hucka, Hamid Bolouri, Andrew Finney, Herbert M Sauro, John C Doyle, Kitano Hiroaki, AP Arkin, Benjamin J Bornstein, Dennis Bray, AA Cuellar, S Dronov, Martin Ginkel, V Gor, II Goryanin, Warren J Hedley, TC Hodgman, Peter J Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, Nicolas Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, Poul F. Nielsen, T. Sakurada, Jim C. Schaff, Bruce E. Shapiro, Thomas Simon Shimizu, Hugh D. Spence, Jorg Stelling, Kouichi Takahashi, Masaru Tomita, John Wagner, and J. Wang. The systems biology markup language (sbml): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19:524–531, 2003a.

Michael Hucka, Andrew Finney, Herbert Sauro, and Hamid Bolouri. Systems biology markup language (sbml) level 1: Structures and facilities for model definitions. Technical report, California Institute of Technology, 2003b.

G Joshi-Tope, I Vastrik, G Gopinathrao, L Matthews, E Schmidt, M Gillespie, P D'Eustachio, B Jassal, S Lewis, G Wu, E Birney, and L Stein. The genome knowledgebase: A resource for biologists and bioinformaticists. *Cold Spring Harb Symp Quant Biol*, 68:237–243, 2003.

Stefan Kokkelink and Roland Schwänzl. Expressing qualified dublin core in RDF/XML. Available via the World Wide Web at `http://dublincore.org/documents/dcq-rdf-xml/index.shtml`.

CM Lloyd, MD Halstead, and PF Nielsen. Cellml: its future, present and past. *Prog Biophys Mol Biol*, 85:433–450, 2004.

Sean Martin, Mike Niemi, and Martin Senger. Life sciences identifiers rfp response. Available via the World Wide Web at http://www.omg.org/docs/lifesci/03-12-02.txt.

R Moats. Urn syntax. Available via the World Wide Web at http://www.ietf.org/rfc/rfc2141.txt.

Andy Powell and Pete Johnston. Guidelines for implementing dublin core in xml. Available via the World Wide Web at http://dublincore.org/documents/dc-xml-guidelines/index.shtml.

RDF. Resource description framework (RDF). Available via the World Wide Web at http://www.w3.org/RDF/.

RDFsyntax. Rdf/xml syntax specification. Available via the World Wide Web at http://www.w3.org/TR/rdf-syntax-grammar/.

Reactome. Reactome - a knowledgebase of biological processes. Available via the World Wide Web at http://www.reactome.org/.

Nicolas Rodriguez, Marco Donizelli, and Nicolas Le Novère. Sbml editor. Available via the World Wide Web at http://www.ebi.ac.uk/compneur-srv/SBMLeditor.html.

Martin Senger. Bibliographic query service. Available via the World Wide Web at http://industry.ebi.ac.uk/openBQS/.

UniProt. The universal protein resource. Available via the World Wide Web at http://www.uniprot.org/.

vCard. vcard: Your electronic business card. Available via the World Wide Web at http://www.imc.org/pdi/.