

This document lists some topics for discussion for the SBML Level 3 Annotation Package workshop to be held at the EBI in May 2010. For more information, please see http://sbml.org/Events/Other_Events/Annotation_package_workshop_2010 .

General Points

The following might want to be kept in mind while we're designing the new annotation package.

1. I believe it is mandatory (or some would like it to be mandatory) that any model encoded in L3, once *stripped out* of all the package-related elements and attributes, remains a valid L3 core. No "conversion" needed.
2. Nicolas has called L3 annotations things which are meant to "describe precisely biological knowledge that does not affect the interpretation of the model". This should be kept in mind when thinking about the discussion points. Specifically, anything that affects the *interpretation* of a model must remain in the core, and not be put in the annotation package.
3. Nicolas has said in the past that he does not "think SBML should pre-emptively incorporate features that could be useful 'at some point'"

Workshop Topic

Topics for Discussion

Please note that some of the topics listed here may be unsuitable for the annotation package workshop: they may not be true annotation topics at all, or they may need further consultation with authors of other packages. If they're deemed out of scope, it would be useful to cross them out and provide an explanation as to why they're out of scope.

1. **The relationship of the annot package to the Groups package.** See http://sbml.org/Community/Wiki/SBML_Level_3_Proposals/Groups. I think we should add this link as part of the required reading for the workshop. There have already been a number of posts on the sbml-discuss mailing list wrt annotations and groups. Specifically, what does it mean to annotate a group over one of the members in the group? Some quotes, and links to their complete context in the mailing lists follow that are relevant to this point:
 - From the groups package page mentioned above: "Since *Group* is derived from *SBase* and *SBase* provides both the ability to attach SBO terms as well as MIRIAM annotations, the semantics of a given "group" in a model can be made more precise by reference to external controlled vocabularies and ontologies. (For instance, a group could be annotated as having an "isa" relationship to specific GO or SBO terms.)"
 - "the annotation of the group is not the annotation of the element. Whether a member is a version of the group or a part of the group (AFAIK this discussion has not been resolved and must be re-opened in the views of the new version), the annotation will be semantically different." My question is: what does it mean, semantically, in such

cases? Post: <http://sbml.org/Forums/index.php?t=tree&goto=5686&rid=0>

- Because any given sbml element (e.g. species, reaction) can be part of many different groups (to have different axes of annotation), the annotation in those groups will differ. How does that affect the members of the group, and what annotation would go into the species element, and what annotation belongs in its various groups that it belongs to?
 - The groups package docs seem to suggest that ListOfMembers inherits from SBase, even though I have heard from Mike that this inheritance might be removed. In a conversation between me and Mike, I mentioned the following, which he said had been brought up by others: annotation could be put in the group, which might then mean that all members have that relationship to each other, or could be put in the listofmembers element, which might mean something different - what, if any, would that difference be? However, if listofmembers does not inherit from sbase, no MIRIAM annotation would be allowed and such troubles would not occur.
 - "If we require a way to describe, using annotation, that several species are versions of the same class of species, this is perfectly achieved with the groups and we do not need any part of the annot package to do this." <http://sbml.org/Forums/index.php?t=tree&goto=5437&rid=0>
2. **Annotations about annotations.** History is only for model, or for everything? How limited are we in what information can be stored. Use of rdf reification, possibly.
 3. **Out of scope (almost): Linking different forms, e.g. phosphorylated and non-phosphorylated forms, to the same entity.**
 - "Controlled annotations currently have a limited semantics. At the moment we can only say that a species is a version of a protein, and that a species is phosphorylated. We actually know how we could say it is a phosphorylated version of a protein. This requires to put an id on an annotation. It was refused for L2V4 and postponed to L3, but finally did not make it to the core." Original post: <http://sbml.org/Forums/index.php?t=tree&goto=5427&rid=0>
 - Others in this same post (<http://sbml.org/Forums/index.php?t=tree&goto=5427&rid=0>) want "examples how to annotate variant sequences, post-translational modifications and different complex formations."
 4. Some people want to say with standard annotations: **We set this value to 5.34 instead of 5.21 because of this reference.**
 - This was from the sbml-discuss post entitled "allowing model history annotations anywhere", however, I can't find it in the forums. Perhaps it didn't get propagated? Anyway, most of the questions set out in that email seem to have been resolved, except I think the following may still need to be addressed.
 - This is useful (the part in **bold** above), but can be partially addressed, as Nicolas said at the time, by putting "on the parameter element a controlled annotation pointing to the pubmed/doi/arXiv entry with the qualifier bqmodel:isDescribedBy" Not perfect (can't say that the original aram was 5.21, for example), but partial. Would have to be careful how we structured that additional information.
 - Nicolas then adds: "I certainly believe this kind of approach would be useful in Systems Biology. However I have two issues here: 1) SBML currently encodes a frozen version of a model for exchange purposes. It is, currently, not aimed to be a database format to cover the whole life-cycle of a model. 2) As you wrote, while several groups are working on such systems, this is still a matter of cutting-edge research, and not a routine procedure, with agreed-upon approaches etc. I do not think

SBML should pre-emptively incorporate features that could be useful 'at some point'.

- This leads to the question, what is an annotation referring to? If we do associate a PubMed reference to this element (a species, say), how do we know that this is referring to 5.34 (the initial concentration, say) and not the compartment="mitochondria" attribute? Essentially, we can only currently annotate elements, not attributes.

5. **Already resolved (restriction lifted): restriction on SBML namespace within the annotation element.**

- This question was sent to the SBML mailing list (<http://sbml.org/Forums/index.php?t=tree&goto=5333&rid=0>), but was never replied to. More information from the original post: "For L3, we're thinking of removing the (L2) restriction on <annotation> content, which is that content isn't allowed to come from the SBML namespaces. (Sec. 3.2.4 in the L2v4 spec.)"

6. **isVersionOf and hasPart when considering A->B, B->A, and A<=>B.**

- How do you deal with precedence of qualifier relationships?
- This is from a discussion with CellML that isn't on any mailing lists. This is a question to which Nicolas provided an answer (see next point), but which nobody seemed particularly happy with. Perhaps we can come up with a better solution in the workshop. If you want the full correspondence for this question, let me know and I can forward the relevant emails to you.
- "There is a blatant mis-use of nesting isVersion within hasPart" ... ". Nicolas then says "I believe one possible solution lies in the fact that several instances of the same qualifier are alternatives, effectively implicit "hasVersions". So if A -> B is annotated by

```
<bqbiol:isVersionOf>
  <rdf:Bag>
    <rdf:li rdf:resource="urn:miriam:ec-code:3.1.3.16"/>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0006470"/>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0045737"/>
  </rdf:Bag>
</bqbiol:isVersionOf>
```

and B -> A is annotated by

```
<bqbiol:isVersionOf>
  <rdf:Bag>
    <rdf:li rdf:resource="urn:miriam:ec-code:2.7.10.2"/>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0006468"/>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0045736"/>
  </rdf:Bag>
</bqbiol:isVersionOf>
```

A <=> B can be annotated by

```
<bqbiol:hasPart>
  <rdf:Bag>
    <rdf:li rdf:resource="urn:miriam:ec-code:3.1.3.16"/>
    <rdf:li rdf:resource="urn:miriam:ec-code:2.7.10.2"/>
  </rdf:Bag>
</bqbiol:hasPart>
<bqbiol:hasPart>
  <rdf:Bag>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0006470"/>
```

```

    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0006468"/>
  </rdf:Bag>
</bqbiol:hasPart>
<bqbiol:hasPart>
  <rdf:Bag>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0045737"/>
    <rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0045736"/>
  </rdf:Bag>
</bqbiol:hasPart>

```

I realise this is not ideal. And I hope that we will find something better while developing the SBMLG Level 3 annotation package." If you aren't sure why such annotation should be allowed in the first place, then here is some more background: the CellML discussion email was one where we were trying to figure out how to include MIRIAM annotations in CellML, and one of the CellML annotators had put the following into a CellML model, but he figured it was wrong. This was what prompted the comments above:

```

<rdf:Description rdf:about="#_rate_of_change_
of_M">
  <rdfs:comment>Reactions for production and degradation of active
cdc2 kinase.
  BIOMD0000000004 seperates this into two reactions and annotates
each reaction using
  isVersionOf. Without seperating the math, we have to use hasPart.
My interpretation is
  that each item in a bag of items for isVersionOf represents
alternatives. How do we represent
  such alternatives within a hasPart combination? The following is not
correct, but
  conveys what I am intending.
  </rdfs:comment>
  <bqbiol:hasPart>
    <rdf:Bag>
      <rdf:li rdf:parseType="Resource">

        <bqbiol:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="urn:miriam:ec-
code:3.1.3.16"/>
            <rdf:li
rdf:resource="urn:miriam:obo.go:GO%3A0006470"/>
            <rdf:li
rdf:resource="urn:miriam:obo.go:GO%3A0045737"/>
          </rdf:Bag>
        </bqbiol:isVersionOf>
      </rdf:li>
      <rdf:li rdf:parseType="Resource">

        <bqbiol:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="urn:miriam:ec-
code:2.7.10.2"/>
            <rdf:li
rdf:resource="urn:miriam:obo.go:GO%3A0006468"/>
            <rdf:li
rdf:resource="urn:miriam:obo.go:GO%3A0045736"/>

```

```

        </rdf:Bag>
        </bqbiol:isVersionOf>
    </rdf:li>
</rdf:Bag>
</bqbiol:hasPart>
</rdf:Description>

```

7. It is important to allow different bags for the same qualifier. I also know that the way the specification works seems to align with how the RDF specification would also interpret the bags/qualifier situation. However, **is there a better way to handle this** in the new version of the annotation package, or do we just want to leave it the same? Background: I started two threads on the topic of multiple bags for the same qualifier back in September (The first was me trying to understand how such a rule worked http://sourceforge.net/mailarchive/forum.php?thread_name=841ae00909151059m2c734416j4547189f399a9727%40mail.gmail.com&net-discuss ; the second was me finding the bug in libSBML that didn't implement that part of the spec correctly <http://sbml.org/Forums/index.php?t=tree&goto=5460&rid=0>). Originally, libSBML's behaviour did not match the SBML specification document: instead, irrespective of how the annotation was before passing through libSBML, all bags with the same qualifier were grouped together by the library. This is being changed by Sarah, but hasn't gone in yet. I have the changed code and I'll test it locally and let Sarah know if it's working. I believe she was going to add a flag to the addCVTerm() method on SBase to define whether the new CVTerm was to be added to an existing or a new bag. By the way, Nicolas has said in the past that the safest way of automatically adding annotation for hasPart and hasVersion (and maybe others) is in fact "If the annotation is added automatically, you should probably add them in different bags if you do not have additional information". For a webpage that describes how annotators of BioModels make use of this particular case of multiple bags of the same qualifier, see <http://www.ebi.ac.uk/biomodels-main/annotationtips> , section "Qualification of Annotation".
8. **Using unique identifiers across multiple models in SBML.** This is probably something that Dagmar would have more information on, but we should at least discuss *if it's in scope*. This is both the:
 - generic issue of stating connections and differences between two models, perhaps in a separate file or in a separate logical space
 - the more specific issue of identifying two objects from two models as semantically identical, as was discussed in <http://sbml.org/Forums/index.php?t=tree&goto=5970&rid=0>. The original use cases in this email were:
 - big model #1 is annotated with bells and whistles, and we create an updated version #2, also fully annotated. Now suppose someone does analysis on 1; the question is how we map these results onto the new network. Ideally, of course, metabolites (say) in each model would be annotated with the same ChEBI. But if we realise that the wrong ChEBI was used (and we don't rely on IDs), then any relationship between the models has been lost.
 - we decide to merge two big models #A and #B into one really big "consensus" model #C. A and B are not necessarily well-annotated. We'd like to ask questions like "how many reactions from A are present in C?" but are lacking the mapping needed.
9. **Links to generated data.** For instance, if a user wants to link his model to the simulation runs, or to any other kinds of data. There does not seem to be any defined MIRIAM uri for that sort of link (which may be to a private or a public data resource). Is this useful? Should it be considered as something for the new annot package?

10. Should **annotation about other packages**, e.g. about how different models have been imported with the multi package, be part of the annot package (which then the multi package could import) or something just for that other package to deal with? I'm leaning towards the latter, but we may want to think about this point... In addition, what about **propagation of annotations**: Is there any specific annotation scheme thought for the model composition. How will the model composed of other individual models inherit annotations? How do we define the connections of models? Should the model in the higher level overwrite the annotations in the lower level but also inherit the ones that are not provided in the higher hierarchy. Or is this entirely out of the scope of this package?
11. **Out of scope: see hierarchical modelling package discussions (may revisit in future if hierarchical modelling people ask questions): Bottom-up approach**: We can discuss about the annotation schemes that might be relevant for bottom-up modelling approach. Although the SBML models are used to understand systems in overall, the same mathematical equations, annotations can also be used for the bottom-up modelling. It would be nice to annotate mathematical models based on biological parts, devices and systems.
12. **Goksel will email examples (e.g. ontology terms) and we'll discuss Thursday PM: Chassis annotation**: Should we consider annotating the cell used for the model? One of the compartments in an SBML model is regarded as the actual cell and used to give the volume information to calculate the concentrations. How can we attach cell specific metadata to this specific compartment? Or should we? How do we know the compartment that actually represent the cell itself? Furthermore we can ask the question whether the cell exist or not in real life. The cell might be the minimal cell or a synthesised one. (<http://nar.oxfordjournals.org/content/vol38/issue8/cover.dtl>)
13. **Out of scope (but a good idea for some kind of synbio package): Synthetic biology visualisation**: As we can use the annotations to convert SBML models into other formats, is there anything we should be discussing for the visualisation of models for synthetic biology and the relevant annotation schemes? (http://dspace.mit.edu/bitstream/handle/1721.1/49523/RFC_16_SBOLv_Specification.pdf?sequence=1)
14. **Out of scope (we can't legislate certification, though could provide tests for people to conform to): Tool certification**: From my understanding software developers are free to choose the packages they want to support. Is there any sort of certification considered by the relevant SBML package team to verify that a software supports the package. This could be done as a whole and the tool can be certified for a list of packages it supports. Just an idea! (*Note from Allyson: I'm not sure if that's something we want to legislate - I think NLN may have an opinion on this one?*)
15. **Alternatives of annotations**: In the spec in section 6.5 it says that "When several relation elements with the same name are placed in the same SBML element's annotation, they represent alternatives." This assumption if known would work well for humans. However would not an RDF processor treat these two constructs as same? (see page 87 of the specification document)
16. **Out of scope (already using RDF, except perhaps for sboTerm attribute): What about RDFa?** See <http://sbml.org/Forums/index.php?t=tree&goto=5138&rid=0> for a previous discussion on this subject, especially Stefan Hoop's comment, which follows: "'> How might we leverage this in SBML?' We already do :) The notes are in XHTML format and RDFa specifies a set of attributes which allows the user to annotate the contents of XHTML using RDF. We need to be aware that it is the content of the XHTML document which is annotated and not the SBML object the notes are part of. This distinguishes it from the RDF/XML we are using for capturing MIRIAM compliant information. In my opinion we need to keep them separated. I propose to encourage the use of RDFa in the notes but I would be strongly opposed in putting any restrictions on it like we did it on the RDF/XML in the annotations. Doing this would lead to artificial constructs like the annotation example

including Inchi code which Neil posted a short time ago. We should restrict ourselves to favorably mentioned RDFa in the specification of the notes."

17. **Out of scope: Sarah reports that example 2 is perfectly valid, and there was a bug in libSBML which has been fixed: Combining all the RDF annotations into one RDF node for an SBML element:**

From my understanding, we can have one RDF description element inside the annotation. Ideally I would like to combine any RDF into this rather than having a separate XML entity inside the annotation. For example, I would like to have an RDF as in the first example below. However both Example 1 and Example 2 would be read as Example 3 by the libsbml. As a result we end up as in example 4 if we want to have this piece of extra information. From the specification and the behaviour of the libsbml, it is not clear at the moment what is really meant. According to the specification we should be able to have the second example. Implementing the 4th solution separates the RDF into two chunks but they are still metadata written for the same purpose and adds another layer for RDF readers to check whether the other xml annotations are in fact contain RDF.

1)

```
<annotation>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:app="http://myapp.org/1.0#"
  xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
    <rdf:Description rdf:about="# Protein_metaid">
      <bqbiol:is>
        <rdf:Bag>
          <rdf:li rdf:resource="urn:miriam:uniprot:Pxxxxx"/>
        </rdf:Bag>
      </bqbiol:is>
      <app:Feature>xxxxxx</app:Feature >
    </rdf:Description>
  </rdf:RDF>
</annotation>
```

2)

```
<annotation>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:app="http://myapp.org/1.0#"
  xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
    <rdf:Description rdf:about="# Protein_metaid">
      <bqbiol:is>
        <rdf:Bag>
          <rdf:li rdf:resource="urn:miriam:uniprot:Pxxxxx"/>
        </rdf:Bag>
      </bqbiol:is>
    </rdf:Description>
    <rdf:Description rdf:about="#Protein_metaid">
      <app:Feature>xxxxxx</app:Feature >
    </rdf:Description>
  </rdf:RDF>
</annotation>
```

3)

```
<annotation>
```

```

    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:app="http://myapp.org/1.0#"
xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
    <rdf:Description rdf:about="# Protein_metaid">
        <bqbiol:is>
            <rdf:Bag>
                <rdf:li rdf:resource="urn:miriam:uniprot:Pxxxxx"/>
            </rdf:Bag>
        </bqbiol:is>
    </rdf:Description>
</rdf:RDF>
</annotation>

```

4)

```

<annotation>
    <MyApp xmlns="http://myapp.org/1.0#">
        <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:app="http://myapp.org/1.0#" >
            <rdf:Description rdf:about="#Protein_metaid">
                <app:Feature>xxxxx</app:Feature >
            </rdf:Description>
        </rdf:RDF>
    </MyApp >
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
        <rdf:Description rdf:about="# Protein_metaid">
            <bqbiol:is>
                <rdf:Bag>
                    <rdf:li rdf:resource="urn:miriam:uniprot:Pxxxxx"/>
                </rdf:Bag>
            </bqbiol:is>
        </rdf:Description>
    </rdf:RDF>
</annotation>

```