CellML Metadata Framework Core Specification 2.0

December 2011

Authors

Michael T. Cooling (m.cooling@auckland.ac.nz)

Contributors

Randall Britten
James R. Lawson
Andrew K. Miller
David P. Nickerson
Poul F. M. Nielsen
Tommy Yu

Status of this Document

Discussion draft.

Introduction

The CellML Metadata 2.0 Framework describes how annotations should be connected to elements within CellML 1.1 model documents. The framework is designed to be modular. It comprises a Core specification (this document), accompanied by one or more satellite specifications. The satellite specifications are each designed to cater for annotation of models for a specific domain or purpose. Examples include the Citation Specification and the Licensing Specification, which cater for adding metadata about citable works, and licenses pertaining to the model, respectively. The modular specification framework allows great flexibility through the addition of satellite specifications for dealing with new domains of interest, and incremental development of annotation pertaining to specific domains.

Scope of the CellML Metadata Framework

- The CellML Metadata Framework 2.0 is designed specifically for the CellML Language 1.1 (http://www.cellml.org/specifications/cellml_1.1).
- This framework deals with annotations that relate solely to a CellML model document. Annotations that pertain to the model itself (as in the abstract entity held in people's heads), or that depend on the CellML model document in some context (for example, the curation status of the model in some repository, or the use of the model document in some virtual experiment), are not within the scope of this framework.
- In reality, annotations may be made inside, or outside of a CellML model document. This framework deals with those that are made inside a CellML model document. Where annotations are made outside of a model document, consistency with the framework is recommended in order to provide the greatest potential of using framework-compatible tools.

- This metadata specification pertains to adding metadata to CellML elements as per the CellML 1.1 language specification. It is possible to define extension elements in addition to the elements defined in the 1.1 language specification. Metadata attached to extension elements is not explicitly recognised by this framework.

When annotating or attempting to use the annotation provided by this framework, the following points should be observed:

- This framework holds an 'open-world' assumption. That is, not all real-world relationships are necessarily documented as annotation. It is not to be assumed for a given model document that every possible relationship is be annotated.
- Annotation is in no way guaranteed to be correct, or up-to-date. Even assuming an annotation was true when it was made, is no guarantee that the annotation will remain true over any period of time. Nor is a given annotation within a model document guaranteed to be consistent with any other annotation within that document, or in any other document. The CellML Metadata Specification Framework 2.0 details how annotations are to be made but gives no information as to the annotations' validity or truthfulness at a particular point in time. Care should be taken not to assign multiple contradictory annotations to a particular model element, including possible contradictory annotations between parent and children elements.

Satellite Specification Principles

In order to ensure a consistent and functional framework, the following principles should be borne in mind when defining new, or updating existing, satellite specifications:

- Each satellite metadata specification must be compatible with this Core Metadata Specification.
- Ontologies used for defining relationships in satellite specifications should be derived from existing 'standard' ontologies where possible (as ratified by appropriate communities, where possible).
- Relationship types to be encoded as part of a new specification, or an update to an existing specification, should be chosen so as to be either direct subsets of, or orthogonal to, the relationship types advocated in existing satellite specifications. Other satellite specifications may need to be updated to ensure that this is the case.
- Relationship types should be chosen so as to avoid 'use-mention confusion'. An example of this confusion would be a relationship type 'is a' when used with a CellML *component* and a database record representing a biological entity. This is incorrect, because a CellML component is not a database record. It is not even a biological entity. In fact, the component represents a biological entity that is represented by the aforementioned database record. While in some limited domains, assumptions of this type are intrinsic and untangling is taken 'as read', errors of this type make working with annotations from multiple ontologies problematic, particularly for machine processing, and should be avoided.
- Where possible, the collection of advocated predicates should be defined. For example, a specific versioned ontology would be preferable to an evolving ontology whose future members maybe be undefined.

- Predicates should be in the form of nouns, where possible. This is for maximum compatibility with the Realisation Strategy (see below). For example ‘part’ is preferable to ‘isPart’, since resolving the RDF to English yields ‘Subject A has an isPart whose values is Object B’ in the first case, which is not as natural as ‘Subject A has a part whose value is Object B’.

Realisation Strategy

Annotations will be made using RDF (http://www.w3.org/RDF/) Statements. Briefly, an RDF statement has three parts, a Subject, a Predicate and an Object. The Subject of an annotation will generally be a CellML model element, and the Predicate will generally be some kind of relationship type. The Object will generally be an external entity such as an identifier for a publication, or perhaps a record in a database of known genes. The RDF statement such as that outlined above links a model element to an external entity, thereby providing an annotation. For a conceptual primer on RDF, please see the following web document http://www.w3.org/TR/rdf-primer/.As CellML is serialized as an XML document, Framework annotations will be serialized using RDF/XML (http://www.w3.org/TR/rdf-syntax-grammar/). As per RDF/XML standards, in order for the documents to be valid, namespaces for the XML attributes through which annotations are encoded will need to be declared. Specifically, a core namespace has been defined specifically to encompass annotations made in a CellML model document:

Prefix	Namespace URI
cmeta	`"http://www.cellml.org/metadata/2.0#"`

(note the “2.0” towards the end of the Namespace URI, which pertains to the 2.0 Metadata Specification, as opposed to “cmeta” as used in the previous 1.0 Metadata Specification [http://www.cellml.org/specifications/metadata/cellml_metadata_1.0]). Additional namespaces for particular relationship types, where required, must be defined in the associated satellite specifications.

Model elements can be referenced by RDF statements if they have an ID. An attribute “cmeta:id” can be created on any CellML element. The “id” is required to be unique across all attributes of type ID inside a CellML model document. Making annotations to MathML inside a CellML model document is similarly catered for using MathML’s existing “math:id” optional attribute. The creation of “id”s is covered in more detail in the CellML 1.1 language specification, section 8 (http://www.cellml.org/specifications/cellml_1.1/#sec_metadata).

RDF annotation linking to CellML model elements should be enclosed within RDF tags. A simple example with one set of RDF tags is shown below. The example shows the basic declaration of a CellML model document, the core annotation namespace and RDF tags. A cmeta:id defined on the <model> tag (or any other tag as per above) can be used as a ‘hook’ to attach annotation to (in this case, the ID is declared is ‘model_example’).

```
<?xml version="1.0"?>
<model       xmlns="http://www.cellml.org/cellml/1.0#"
             xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             cmeta:id="model_example"
```

```
            name="model_example" >
```

…other elements…

```
<rdf:RDF>
  <rdf:Description rdf:about="#model_example">
       …annotation about the <model> element goes here
  </rdf:Description>
</rdf:RDF>
```

…other elements…

```
</model>
```

In keeping with RDF/XML, a basic RDF Subject declaration is shown above, by means of the `rdf:Description` tag. The `rdf:about` attribute uses the URI reference fragment delimiter (‘#’) to declare that the Subject is the element with the ID of ‘model_example’, within the current document.

For additional examples of the RDF/XML that goes between the RDF tags, please see one or more of the CellML Metadata Framework 2.0 satellite specifications.

An alternative method of attaching metadata to CellML model elements is to use, as values of the `rdf:about` attribute, URIs with XPATH 1.0 expressions (see http://www.w3.org/TR/1999/REC-xpath-19991116/ for details) as fragments. XPATH is attractive because it can specify not only XML tags that have IDs, but any XML node within the document – such as attributes of tags, or tags without IDs.

The ability to use complex expressions as fragments for URIs that reference XML documents is provided by the XPointer Framework W3C Recommendation (http://www.w3.org/TR/2003/REC-xptr-framework-20030325/). At the time of writing, the part of the XPointer Framework (namely the ‘xpointer() Scheme’) allowing XPATH expressions is not yet a W3C Recommendation, but a Working Draft (see http://www.w3.org/TR/2002/WD-xptr-xpointer-20021219/ for details). The requirements here are a subset of those catered for by that Draft, and by other xpointer Schemes currently in the xpointer Scheme Registry (http://www.w3.org/2005/04/xpointer-schemes/). Therefore a simpler scheme called the' xpointernode() Scheme’ is defined here. The syntax of the xpointernode() Scheme is defined as a subset of XPATH 1.0 expressions (http://www.w3.org/TR/1999/REC-xpath-19991116/) - specifically those that are evaluated to ultimately yield an object of type XPATH ‘node-set’ (thus excluding those expressions that ultimately evaluate to objects of other XPATH data types), where the node-set contains exactly one element.

Some examples of such fragments as values of `rdf:about` attributes are

```
rdf:about=”#xpointernode(component[@name=‘calcium'])”
```

making the component with the name of ‘calcium’ the RDF Subject of the annotation, and

```
rdf:about=”#xpointernode(component
[@name=‘calcium']/variable[@name=‘concentration’]/@initial_value)”
```

making the ‘initial_value’ attribute of the variable whose name is ‘concentration’, inside the component with the name of ‘calcium’, the RDF Subject of the annotation.

Using XPATH expression predicates that rely on a particular ordering of model elements is possible in this Scheme, but strongly discouraged when used for CellML models. Model element order in CellML has no particular meaning and can often change when a model is serialized by different tools, complicating the ability of such tools to keep annotations attached appropriately. Instead, elements should be identified as explicitly as possible within XPATH expressions, by making use of name and/or id attributes where applicable.

Finally, note that the standardised method of using XPATH within URI fragments facilitates the attachment of metadata to CellML model elements outside of the current XML document, for example in the following RDF Subject declaration:

```
rdf:about=”http://www.cellml.org/somewhere/otherdocument.cellml#xpointernode(
component[@name=‘calcium'])”
```

which references a particular component in a file accessible on the Web. xpointernode() fragments can also be used with other URI types that also point to XML documents, such as 'file://' URIs.

CellML Basic Model Information Metadata Specification 2.0

June 2011

Authors

Michael T. Cooling (m.cooling@auckland.ac.nz)

Contributors

Richard Christie
David P. Nickerson
Tommy Yu
David Brooks

Status of this Document

Discussion draft.

Dependencies

This specification is dependent on the CellML Core Metadata Specification 2.0.

Introduction

Regardless of the type of model encoded in a CellML model document, there are three attributes that apply:

> Every model document has a creator (whether human or by some other entity or artifact). This creator may be part of an organization and recording that fact may be desirable.
>
> Every model document is created at a point in time.
>
> Every model document will be created for some reason or purpose, of varying significance (for example: as part of a scientific mode of enquiry, or as a side-effect of some automated process).

The same could be said about any CellML model element. In addition, it is useful to be able to add descriptive, free-form text annotation to model elements for the general purpose of adding comments about them.

This ability to add author, timestamp information and a comment or description is also extended to annotation elements. One could add annotation forming a 'comment' about a piece of CellML code, then annotate that comment to form a comment about that comment etc.

Together, these annotations are considered 'Basic Model Information'. This document describes how these should be annotated for a given CellML model document, should someone wish to do so.

Realisation Strategy

People, organizations and other entities (such as software packages or physical artefacts capable of performing some action) will be described using a subset of an external emerging (currently version 0.1) standard known as Friend-Of-A-Friend (FOAF). At the time of writing, the current FOAF specification

(Vocabulary Specification 0.98 “Marco Polo Edition”) can be found here: http://xmlns.com/foaf/spec/20100809.html.

In FOAF, the aforementioned entities are collectively known as Agents. CellML model document annotations will describe Agents using RDF/XML serializations of FOAF objects. Most often, a FOAF object will become the RDF Objects of some RDF Subject-Predicate-Object triple. For example: the Agent which creates a model or model element.

In order to describe Agents, the following subset of the FOAF Core is considered part of this specification

FOAF objects

- Person – describing a person
- Group – A collective of Agents
- Agent – superclass of Person and Group. May also describe other types of Agents not given their own explicit subclass, such as a software package

FOAF properties

- name – a name for one of the FOAF objects above
- familyName – a family name, used for Person objects
- givenName – a given name, used for Person objects
- member – used to declare that some Agent is a member of a Group

FOAF properties and objects not listed above are not considered part of the CellML Basic Model Information Metadata Specification 2.0.

The relationship to specify the creator of a model document can be made by forming an RDF triple where the Subject is the model tag (via a cmeta:id), the Object is the Agent who is the creator, and the Predicate is described using the ‘maker’ property from FOAF.

The point in time at which a model element was created should be encoded using the Dublin Core (http://dublincore.org/documents/2010/10/11/dcmi-terms/) term ‘created’, annotated similarly to ‘maker’ above. The time point itself should be encoded in the W3C’s Time and Date Formats Specification (http://www.w3.org/TR/NOTE-datetime). This allows the specification of the time point at an encoder-chosen level of precision.

A general comment on a model element, including the purpose of a model (which would be encoded with the <model> tag as the RDF Subject), should be encoded using the Dublin Core ‘description’ term, as text.

The recommended namespace declarations for using FOAF and the Dublin Core in the context of this specification are as follows:

```
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
```

Dublin Core terms not explicitly mentioned above are not considered part of the CellML Basic Model Information Metadata Specification 2.0.

In order to make comments about annotation elements, including other comments, RDF 'reification' is used. Conceptually, this involves making the RDF statement that links some model element with an annotation into a statement with an identifier that can itself be referenced in further RDF statements. In pure RDF, this involves making a set of 4 RDF statements (known as an 'RDF reification quad') that together describe the original annotation statement and provide it with an identifier (more details on reification can be found in http://www.w3.org/TR/rdf-primer/). This is in addition to the RDF that actually makes the original annotation statement. In pure RDF this can become verbose, particularly if the original annotation statement has a lengthy Object. Fortunately, in XML/RDF there is a shorthand version, where predicates are assigned an rdf:ID, which can then be used as the RDF Subject of the comment annotation. This Specification recommends that approach, although full specification of the 'RDF reification quad' is also acceptable.

Examples

1. Representing a Person

```
<foaf:Person>
      <foaf:givenName>Mike</foaf:givenName>
      <foaf:familyName>Cooling</foaf:familyName>
</foaf:Person>
```

2. Representing a research institute or other group

```
<foaf:Group>
      <foaf:name>Auckland Bioengineering Institute</foaf:name>
</foaf:Group>
```

3. Representing a software package

```
<foaf:Agent>
      <foaf:name>CellML API v1.8</foaf:name>
</foaf:Agent>
```

4. Specifying members of a Group

This could be done 'inline' as follows:

```
<foaf:Group>
      <foaf:name>Auckland Bioengineering Institute</foaf:name>
      <foaf:member>
            <foaf:Person>
                  <foaf:name>Mike Cooling</foaf:name>
            </foaf:Person>
      </foaf:member>
</foaf:Group>
```

Or, where an Agent might be involved in several annotations within the CellML model document it is recommended to define the Agent separately and use an rdf:nodeID as follows:

```
<foaf:Person rdf:nodeID=”mike_cooling”>
      <foaf:givenName>Mike</foaf:givenName>
      <foaf:familyName>Cooling</foaf:familyName>
</foaf:Person>

<foaf:Group>
      <foaf:name>Auckland Bioengineering Institute</foaf:name>
      <foaf:member rdf:nodeID="mike_cooling"/>
</foaf:Group>
```

5. Adding creator, timestamp and purpose descriptions to a CellML model.

```
<?xml version="1.0"?>
<model xmlns="http://www.cellml.org/cellml/1.0#"
      xmlns:cmeta=”http://www.cellml.org/metadata/2.0#”
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"

      cmeta:id="model_example"
      name="model_example"
>

<rdf:RDF>

<foaf:Person rdf:nodeID="mike_cooling">
      <foaf:givenName>Mike</foaf:givenName>
      <foaf:familyName>Cooling</foaf:familyName>
</foaf:Person>

<rdf:Description rdf:about="#model_example">
      <foaf:maker  rdf:nodeID="mike_cooling"/>
      <dcterms:created rdf:datatype=" http://purl.org/dc/terms/W3CDTF">
            2011-02
      </dcterms:created>
      <dcterms:description>This model was constructed as an example model for
      the CellML 1.1 Metadata Specification Framework.
      </dcterms:description>
</rdf:Description>

</rdf:RDF>
```

…other elements…

```
</model>
```

The above example shows the construction of a FOAF Person object, which becomes the RDF subject of a ‘maker’ relationship for the model document. The ‘created’ predicate is used to specify that this particular model was created during February 2011, and the ‘description’ predicate describes the purpose of the model’s creation. In the above example all three ‘Basic model information’ statements are made together, which is recommended, but there is no reason why one or more cannot be absent, or specified as separate statements in the model document, e.g.

```
<rdf:Description rdf:about="#model_example">
      <dcterms:created rdf:datatype=" http://purl.org/dc/terms/W3CDTF">
            2011-02
      </dcterms:created>
</rdf:Description>
```

…other elements…

```
<rdf:Description rdf:about="#model_example">
      <foaf:maker  rdf:nodeID="mike_cooling"/>
</rdf:Description>
```

6. Adding creator (using the FOAF Person defined in example 1) and timestamp elements to a model element (in this case, a CellML *component*)

```
<RDF>
<rdf:Description rdf:about="#parameters">
      <foaf:maker>mike_cooling</foaf:maker>
      <dcterms:created rdf:datatype=" http://purl.org/dc/terms/W3CDTF">
            2010-11-07
      </dcterms:created>
</rdf:Description>
</RDF>
```

…other elements…

```
<component name="model_parameters" cmeta:id=”parameters”>
```

…other elements…

```
</component>
```

7. Adding a comment to a model element (in this case, a CellML *variable*)

```
<variable cmeta:id="vi_variable" initial_value="0.025" name="vi"
public_interface="out" units="flux">
       <rdf:RDF>
           <rdf:Description rdf:about="#vi_variable">
                  <dcterms:description>This value of 0.025 comes from Fig 3
                  caption, page 9110 of the original
                  paper</dcterms:description>
           </rdf:Description>
       </rdf:RDF>
</variable>
```

8. Extending the previous example so as to add a comment, and a timestamp, to a comment, the identifier on the predicate of the first statement shown in bold

```
<variable cmeta:id="vi_variable" initial_value="0.025" name="vi"
public_interface="out" units="flux">
       <rdf:RDF>
           <rdf:Description rdf:about="#vi_variable">
                  <dcterms:description rdf:ID=”vi_comment”>This value of
                  0.025 comes from Fig 3 caption, page 9110 of the original
                  paper</dcterms:description>
           </rdf:Description>

            <rdf:Description rdf:about="#vi_comment">
                <foaf:maker  rdf:nodeID="mike_cooling"/>
                  <dcterms:created
                  rdf:datatype="http://purl.org/dc/terms/W3CDTF">
                  2010-11-05</dcterms:created>
                <dcterms:description>Original author confirms Fig 3 is the
                best one to use.</dcterms:description>
           </rdf:Description>
       </rdf:RDF>
</variable>
```

Note that in this example the timestamp relates to the first comment (with an rdf:ID of “vi_comment”) only, and gives no information as to when the second (“Original author confirms…”) was made. If that second comment was itself given a nodeID, it could be further annotated with that information if desired.

9. A variable with a timestamp, where the timestamp is additionally commented with the timestamper, and a textual comment

```
<variable cmeta:id="vi_variable" initial_value="0.025‟  name="vi‟
    public_interface="out" units="flux">
 <rdf:RDF>
       <rdf:Description rdf:about="#vi_variable">
<dcterms:created rdf:ID="vi_timestamp"
rdf:datatype="http://purl.org/dc/terms/W3CDTF">
                  2010-11-05</dcterms:created>
      </rdf:Description>
    <rdf:Description rdf:about="#vi_timestamp">
      <foaf:maker  rdf:nodeID="mike_cooling"/>
       <dcterms:description>This date may be plus or minus 2
days</dcterms:description>
   </rdf:Description>
</rdf:RDF>
</variable>
```

CellML Licensing Metadata Specification 2.0

June 2011

Authors

Michael T. Cooling (m.cooling@auckland.ac.nz)

Contributors

Catherine M. Lloyd
David P. Nickerson
Caton Little

Status of this Document

Discussion draft.

Dependencies

This specification is dependent on the CellML Core Metadata Specification 2.0.

Introduction

This document describes how to annotate a CellML model document with licensing information for that model document.

Realisation Strategy

Licensing information should be added using the Dublin Core (http://dublincore.org/documents/2010/10/11/dcmi-terms/) term ‘license’, as the RDF Predicate, with the model document’s <model> tag as the RDF Subject. The Object could be a URI to the licensing information, or a literal text version of the license, or a combination of the two. If both are desired, use of the RDF container ‘Alt’ is recommended. See http://www.w3.org/TR/rdf-primer/#containers for more information on RDF containers.

It may be desirable to specify a multi-license for the model. If the objects of RDF statements are the alternative licenses, then RDF Container 'rdf:Alt' is not recommended because it is open - it does not specify that those are the ONLY alternatives. The RDF Collection construct may be used, however here the semantic meaning of being alternatives is lost, implying that all licenses should apply. Hence in this situation, it is recommended to define a license document that specifies the multi-license intent and links to the finite number of alternatives. It is that initial document that should be specified as the RDF Object of an annotation statement.

The recommended Dublin Core namespace declaration is as follows:

```
xmlns:dcterms="http://purl.org/dc/terms/"
```

Dublin Core terms not explicitly mentioned above are not considered part of the CellML Licensing Metadata Specification 1.1.

Examples

1. A URI reference to a license.

```
<?xml version="1.0"?>
<model xmlns="http://www.cellml.org/cellml/1.1#"
      xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
      cmeta:id="model_example"
      name="model"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dcterms="http://purl.org/dc/terms/"
>

<rdf:RDF>
<rdf:Description rdf:about="#model_example">
      <dcterms:license rdf:resource="http://exampleweb.org/licenses/2.0/" />
</rdf:Description>
</rdf:RDF>
```

… other model elements …

```
</model>
```

2. Both a license URI and a literal text of the license (using the rdf:Alt construct)

```
<rdf:RDF>
<rdf:Description rdf:about="#model_example">
      <dcterms:license>
        <rdf:Alt>
            <rdf:li rdf:resource="http://exampleweb.org/licenses/2.0/" />
            <rdf:li>
            Example License
            THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS
            LICENSE. THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER
            APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED
            IS...UNAUTHORIZED.
            </rdf:li>
        </rdf:Alt>
      </dcterms:license>
</rdf:Description>
```

CellML Citation Metadata Specification 2.0

October 2011

Authors

Michael T. Cooling (m.cooling@auckland.ac.nz)

Status of this Document

Discussion draft.

Dependencies

This specification is dependent on the CellML Core Metadata Specification 2.0.

Introduction

This document describes how to add annotation linking the CellML model document, or an element of that document, to one or more citable works such as journal articles or conference presentations.

Realisation Strategy

Citable works are linked to a model document or an element of that document by making the element an RDF Subject, and the citable work the RDF Object, of an RDF statement. The Predicate for this relationship should be the Biomodels Qualifier (http://biomodels.net/qualifiers/) 'description'.

Where multiple citable works are required, multiple RDF statements could be made. Alternatively, multiple citable works could be specified using the RDF 'Bag' container. See http://www.w3.org/TR/rdf-primer/#containers for more information on RDF containers.

Citable works (the RDF Objects) should be specified as URIs wherever possible. If the work has a record in a persistent database such as Pubmed (http://www.ncbi.nlm.nih.gov/pubmed/), then the recommended method is to use an Identifiers.org URI (see http://identifiers.org/ for more on Identifiers.org). Where this is not applicable (if, for example, the work does not have a record, or perhaps is not published at all) another URI may be used (in some examples below, ISSN URNs are used). Care should be taken that the URI be as discoverable and long-lived as possible in order for someone to gain meaningful information from it.

If an appropriate URI is not available, an alternative method is to provide a BIBO (http://bibliontology.com/specification) Object in the RDF to represent the bibliographic resource. This object can then be linked to the model element of interest with 'description'. Any BIBO object may be encoded, and it is recommended that as many BIBO properties should be added as necessary so that someone can clearly identify the cited work. At the time of writing, the recommended BIBO version is the latest one, labelled 'Revision 1.3'. A browsable list of BIBO classes and properties can be found at http://bibotools.googlecode.com/svn/bibo-ontology/trunk/doc/index.html.

The recommended BIBO and Biomodels qualifer namespace declarations are as follows:

```
xmlns:bibo="http://purl.org/ontology/bibo/"
xmlns:bqmodel="http://biomodels.net/model-qualifiers"
```

Several of the BIBO properties are ‘borrowed’ from other collections of properties such as the Dublin Core, and FOAF which are also used elsewhere in the CellML Metadata Framework 2.0 (see the CellML Basic Information Metadata Specification 2.0 for details), hence using those properties in a BIBO object may also require the specification in a CellML Model Document of additional namespaces, if they are not already defined. Several cases of this are shown in the Examples below.

Examples

1. A model is described in a published journal article that is identified in Pubmed.

```
<?xml version="1.0"?>

<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
cmeta:id="ip3_model"
name="ip3_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
xmlns:bibo="http://purl.org/ontology/bibo/"
>

<rdf:RDF>
 <rdf:Description rdf:about="#ip3_model">
      <bqmodel:description
      rdf:resource="http://identifiers.org/pubmed/17693463"/>
 </rdf:Description>
</rdf:RDF>
```

...other elements...

```
</model>
```

2. A model is described in a published journal article which lacks an identifier in Pubmed.

```
<?xml version="1.0"?>

<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
cmeta:id="ip3_model"
name="ip3_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
xmlns:bibo="http://purl.org/ontology/bibo/"
xmlns:dcterms="http://purl.org/dc/terms/"
>

<rdf:RDF>
 <rdf:Description rdf:about="#ip3_model">
      <bqmodel:description rdf:resource="#example_article"/>
 </rdf:Description>

<bibo:Article rdf:ID="example_article">
      <dcterms:creator>Fred Bagg</dcterms:creator>
      <dcterms:issued>1981</dcterms:issued>
```

```
      <dcterms:title>Pertubations in calcium signaling activate immune system
      function</dcterms:title>
      <bibo:volume>66</bibo:volume>
      <bibo:issue>10</bibo:issue>
      <bibo:pageStart>1102</bibo:pageStart>
      <bibo:pageEnd>1111</bibo:pageEnd>
      <dcterms:isPartOf rdf:resource="urn:issn:0027-8128"/>
</bibo:Article>
</rdf:RDF>
```

…other elements…

```
</model>
```

3. Extending the previous example so that the Article declaration contains an ordered list of authors.

```
<?xml version="1.0"?>

<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
cmeta:id="ip3_model"
name="ip3_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
xmlns:bibo="http://purl.org/ontology/bibo/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
>

<rdf:RDF>
 <rdf:Description rdf:about="#ip3_model">
      <bqmodel:description rdf:resource="#example_article"/>
 </rdf:Description>


<bibo:Article rdf:ID="example_article">

      <bibo:authorList>
            <rdf:Seq>
                  <rdf:li rdf:resource="#fred_bagg"/>
                  <rdf:li rdf:resource="#joe_fligs"/>
            </rdf:Seq>
      </bibo:authorList>

      <dcterms:issued>1981</dcterms:issued>
      <dcterms:title>Pertubations in calcium signaling activate immune system
      function</dcterms:title>
      <bibo:volume>66</bibo:volume>
      <bibo:issue>10</bibo:issue>
      <bibo:pageStart>1102</bibo:pageStart>
      <bibo:pageEnd>1111</bibo:pageEnd>
      <dcterms:isPartOf rdf:resource="urn:issn:0027-8128"/>
</bibo:Article>

<foaf:Person rdf:ID="fred_bagg" foaf:name="Fred Bagg"/>
```

```
<foaf:Person rdf:ID="joe_fligs" foaf:name="Joe Fligs"/>

</rdf:RDF>
```

…other elements…

```
</model>
```

4. The model that a *component* represents is described in a book chapter which is not in Pubmed.

```
<?xml version="1.0"?>

<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
cmeta:id="some_model"
name="some_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
xmlns:bibo="http://purl.org/ontology/bibo/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
>
```

...other elements...

```
<component cmeta:id="example_component">

<rdf:RDF>
 <rdf:Description rdf:about="#example_component">
      <bqmodel:description rdf:resource="#the_chapter"/>
 </rdf:Description>


<bibo:chapter rdf:ID="the_chapter">
      <dcterms:isPartOf rdf:resource="#the_book"/>
      <dcterms:creator rdf:resource="#sam_smith"/>
      <bibo:chapter>14</bibo:chapter>
      <dcterms:title>Marsh-warbler feeding calls</dcterms:title>
      <bibo:pageStart>160</bibo:pageStart>
      <bibo:pageEnd>164</bibo:pageEnd>
</bibo:chapter>

<bibo:EditedBook rdf:ID="the_book">
      <dcterms:publisher rdf:resource="#the_publisher"/>
      <bibo:editorList>
            <rdf:Seq>
            <rdf:li rdf:resource="#hamish_wang"/>
            <rdf:li rdf:resource="#fred_ming"/>
            <rdf:li rdf:resource="#gertrude_brown" />
            </rdf:Seq>
      </bibo:editorList>
      <dcterms:issued>September, 2010</dcterms:issued>
      <bibo:isbn>3273876876K</bibo:isbn>
</bibo:EditedBook>
```

```
<foaf:Organisation rdf:ID="the_publisher" foaf:name="Marsh Animals Press" />

<foaf:Person rdf:ID="hamish_wang" foaf:name="Hamish Wang"/>
<foaf:Person rdf:ID="fred_ming" foaf:name="Fred Ming"/>
<foaf:Person rdf:ID="gertrude_brown" foaf:name="Gertrude Brown"/>
<foaf:Person rdf:ID="sam_smith" foaf:name="Sam Smith"/>

</rdf:RDF>
```

...other elements...

```
</model>
```

5. The model that a component represents is described in a presentation performed at a conference

```
<?xml version="1.0"?>
<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
cmeta:id="some_model"
name="some_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
xmlns:bibo="http://purl.org/ontology/bibo/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:event="http://purl.org/NET/c4dm/event.owl#"
xmlns:timeline="http://purl.org/NET/c4dm/timeline.owl#"
>

...other elements...

<component cmeta:id="example_component">

<rdf:RDF>
 <rdf:Description rdf:about="#example_component">
<bqmodel:description rdf:resource="#the_presentation"/>
 </rdf:Description>

<bibo:Slideshow rdf:ID="the_presentation">
      <dcterms:creator rdf:resource="#sam_smith"/>
      <dcterms:date>16-April-2010</dcterms:date>
      <dcterms:title>Marsh Warblers I have known</dcterms:title>
      <bibo:presentedAt rdf:resource="#the_conference" />
</bibo:Slideshow>

<foaf:Person rdf:ID="sam_smith" foaf:name="Sam Smith"/>

<bibo:Conference rdf:ID="the_conference" >
      <event:place rdf:resource="http://sws.geonames.org/2193733/" />
<timeline:at rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2010-
10-25T12:00:00</timeline:at>
<timeline:duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">PT5D</timeline:durat
ion>
   <dcterms:title>Marsh Warbler Symposium 2010</dcterms:title>
</bibo:Conference>
```

```
</rdf:RDF>

...other elements...

</model>
```

CellML Biological Annotation Metadata Specification 2.0

December 2011

Authors

Michael T. Cooling (m.cooling@auckland.ac.nz)

Status of this Document

Discussion draft.

Dependencies

This specification is dependent on the CellML Core Metadata Specification 2.0.

Introduction

This document describes how annotations can be added to elements of a CellML model document that declare what biological entities or processes those elements represent.

Realisation Strategy

To link CellML elements to biological concepts one should use RDF statements where the CellML element is the RDF Subject, and a URI to the concept the RDF Object. The RDF Predicate should be chosen from the Biomodels Biological Qualifiers (at the time of writing, a list can be found here: http://www.ebi.ac.uk/miriam/main/qualifiers/, under the heading 'biology-qualifiers'). In keeping with the principles in the CellML Core Metadata Specification 2.0, the 'noun' forms of the predicates must be used e.g. 'identity' rather than 'is', or 'part' rather than 'hasPart'.

In keeping with the Biomodels framework, the RDF Objects of the annotations statements should be URIs to biological concepts (not to instances of the biology themselves). Where possible these should link to a publically accessible ontology (or database, if no suitable ontology can be found) of such concepts such as the Gene Ontology (http://www.geneontology.org/), the PROTEIN ontology (http://pir.georgetown.edu/pro/) or UNIPROT (http://www.uniprot.org/). Where possible, Identifier.org URIs (http://identifiers.org) should be used as the RDF Object URIs, as these are persistent and easily dereferenced.

The recommended Biological qualifier namespace is as follows:

```
xmlns:bqbio="http://biomodels.net/biology-qualifiers/"
```

It is important to try to be as precise as possible. For example, it might seem useful to annotate a CellML *component* as representing a particular protein. However, if the *component* also represents other things, then better alternatives might include using 'part' instead, or annotating a specific CellML *variable* rather than the entire CellML *component* as being that particular protein. Similarly, it is important to try to be as specific as possible. Annotating a CellML *component* to the effect that it represents 'a protein' (which one?) is often not as useful as relating it to a particular protein.

Annotating Imported Model Elements

It is often desirable to annotate model elements that are part of the model via CellML <import>s. Units and components that are imported to the model can be referenced in RDF statements via their <units> and <component> children (respectively) of the <import> tag where the import is declared. These child tags can be considered as proxies for the actual unit and component definition, hence it is acceptable to consider them as possibly representing biological concepts and Biomodels Biological qualifiers can be used as RDF Predicates in statements about them.

Variables in imported components, whether they are accessible in the CellML sense via their CellML interfaces or not, are not given proxies in the current CellML model document. The only potential reference to them in the model is as part of one or more 'map_variable' elements in <connection>s, which themselves do not represent biology but are pointers to representations of biology. Therefore, RDF statements about the imported variable cannot be validly made using Biomodels Biological qualifiers as predicates. Therefore it is not possible to reference such variables in this Specification.

Examples

1. A *component* represents a particular protein (listed in the Protein Ontology ID:000005120)

```
<?xml version="1.0"?>

<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
cmeta:id="example_model"
name="the_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqbio=”http://biomodels.net/biology-qualifiers/">
```

…other elements…

```
 <component cmeta:id=”c” name=”c_component”>
<rdf:RDF>
  <rdf:Description rdf:about="#c">
      <bqbio:hypernym
rdf:resource="http://identifiers.org/obo.pr/PR:000005120" />
  </rdf:Description>
</rdf:RDF>
```

…other elements…

```
</component>
```

…other elements…

```
</model>
```

2. An equation contains two terms that deal with different biological processes (that are represented by Gene Ontology records 0051603 and 0042398).

```
<?xml version="1.0"?>

<model xmlns="http://www.cellml.org/cellml/1.1#"
xmlns:cmeta="http://www.cellml.org/metadata/2.0#"
```

```
cmeta:id="example_model"
name="the_model"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqbio=”http://biomodels.net/biology-qualifiers/">
```

…other elements…

```
<math id=”the_equation” xmlns="http://www.w3.org/1998/Math/MathML">
```

…other elements…

```
<rdf:RDF>
 <rdf:Description rdf:about="#the_equation">
        <bqbiol:part>
            <rdf:Bag>
            <rdf:li rdf:resource="http://identifiers.org/obo.go/GO:0051603"
            />
            <rdf:li rdf:resource="http://identifiers.org/obo.go/GO:0042398"
            />
            </rdf:Bag>
      </bqbiol:part>
  </rdf:Description>
</rdf:RDF>
```

…other elements…

```
</math>
```

…other elements…

```
</model>
```

Note: specifying exactly which terms of an equation encoded in MathML represent which biological processes may be achieved by making <apply> blocks the RDF Subjects, using the xpointernode() Scheme as described in the CellML Core Metadata Specification 2.0.