

# The CellML Metadata Framework 2.0 Specification

Michael T. Cooling<sup>1\*</sup> and Peter Hunter<sup>1</sup>

<sup>1</sup>Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

## Summary

The CellML Metadata Framework 2.0 is a modular framework that describes how semantic annotations should be made about mathematical models encoded in the CellML ([www.cellml.org](http://www.cellml.org)) format, and their elements. In addition to the Core specification, there are several satellite specifications, each designed to cater for model annotation in a different context. Basic Model Information, Citation, License and Biological Annotation specifications are presented.

---

\* To whom correspondence should be addressed. Email: [m.cooling@auckland.ac.nz](mailto:m.cooling@auckland.ac.nz)

## Contents

- 1 Introduction
- 2 Core Specification 2.0
- 3 Basic Model Information Specification 2.0
- 4 Licensing Specification 2.0
- 5 Citation Specification 2.0
- 6 Biological Annotation Specification 2.0
- 7 Acknowledgements

## 1 Introduction

The CellML Metadata 2.0 Framework describes how annotations should be connected to CellML 1.1 model document elements. The framework is designed to be modular. It comprises a Core specification, 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 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. The Core and satellite specifications will be discussed in turn.

## 2 Core Specification 2.0

### 2.1 Scope of the 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](http://www.cellml.org/specifications/cellml_1.1)).
- This framework deals with annotations that relate solely to a CellML model document or its elements. 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.

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 should not be assumed that every possible relationship is annotated for a given model document.
- Annotation is not, in general, guaranteed to be correct, or up-to-date. Even assuming an annotation was true when it was made, there is no guarantee that the annotation will remain true over any period of time. Nor is a given annotation about some element of a model document guaranteed to be consistent with any other annotation about any other elements in 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 allow multiple contradictory annotations about a particular model element to become relevant, including possible contradictory annotations between parent and children elements.

## 2.2 Realisation Strategy

Annotations will be made using RDF (<http://www.w3.org/RDF/>) triples. For a conceptual primer on RDF, please see the W3C RDF Primer (<http://www.w3.org/TR/rdf-primer/>). 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. An RDF statement typically links a model element to an external entity via some relationship, thereby providing an annotation that the model element has that relationship.

There are several common formats for RDF statements. In keeping with CellML, which is serialized as an XML document, it is recommended that framework annotations be serialized using RDF/XML (<http://www.w3.org/TR/rdf-syntax-grammar/>). Even within RDF/XML, there are multiple valid ways to express the same relationship. This Metadata Framework documentation typically demonstrates a single simple method for each example, but often there are several alternatives that are equally valid.

In contrast to the CellML Metadata Specification 1.0, in 2.0 annotations should never be made within the CellML model document to which they pertain. Perhaps the simplest (but not the only) way to do this is to place the annotation in a separate XML document, housed its own file, which in turn has the ".xml" extension - assuming the format is RDF/XML.

Model elements can be referenced by RDF statements if they have an appropriate ID. An attribute "cmeta:id" can be created on any CellML element inside its CellML model document. 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](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 is shown below. The example presupposes that there exists, in the same folder as the annotation file, a CellML model document "model.cellml" that has, on its <model> tag, a cmeta:id "model\_example" defined. That provides a hook for the RDF Subject to "point to". In this case the model "model\_example" is annotated with a simple description, following the CellML Metadata Framework Basic Model Information Specification 2.0.

---

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/">

  <rdf:Description rdf:about="./model.cellml#model_example">
    <dcterms:description>
      This is an example description that the model_example is annotated with.
    </dcterms:description>
  </rdf:Description>

</rdf:RDF>

```

---

For additional examples of possible annotations, please see the CellML Metadata Framework 2.0 satellite specifications.

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 facilitate annotations made in a CellML model document:

---

Suggested prefix	Namespace URI
cmeta	"http://www.cellml.org/metadata/2.0#"

---

Annotations may also validly be facilitated with the “cmeta” namespace as defined in the CellML Metadata 1.0 Specification ([https://www.cellml.org/specifications/metadata/cellml\\_metadata\\_1.0](https://www.cellml.org/specifications/metadata/cellml_metadata_1.0)). This means that Metadata Framework 2.0 annotations could be made about CellML models and their elements without modification of the CellML, if Metadata 1.0 cmeta:ids already exist. Care should be taken that all annotations intended for processing remain relevant and are not internally inconsistent when taken together.

It is often desirable to annotate model elements that are part of the model via CellML <import>s. Units and components that are imported into a model can be referenced in RDF statements via their <units> and <component> children (respectively) of the CellML <import> tag where the import is declared. These child tags can be considered as proxies for the actual unit and component definition, which otherwise exist only virtually. Variables in imported components, whether they are accessible in the CellML sense via their CellML interfaces or not, are not given proxies any 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. If these are given cmeta:ids, care should be taken with the predicates used - it is important to bear in mind that since the mapping is being annotated the annotation is one level removed from the variable itself.

### 2.3 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 specification.

- Ontologies used for defining relationships in satellite specifications should be derived from existing ‘standard’ ontologies, 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’, defined literally, when used with a CellML component and a database record representing a biological entity. This would be 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 and should be avoided.
- Where possible, the collection of advocated predicates should be defined. 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’.
- Additional namespaces (such as “determs” above) for constructs such as particular relationship types (predicates) or objects, where required, must be detailed in the appropriate Metadata Framework satellite specification.

## 3 Basic Model Information Specification 2.0

### 3.1 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 / organisation or artefact).
- 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. The creator may be part of an organization and recording that relationship may also be desirable. 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 specification describes how these should be annotated for a given CellML model document.

### 3.2 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-formatted 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
- Agent - superclass of Person and Group. May also describe other types of Agents not given their own explicit subclass, such as software packages
- Group - A collective of Agents

#### 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

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.

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

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’, used 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, potentially, the purpose of a model (which would be encoded with the `cmeta:id` of the CellML `<model>` tag as the RDF Subject), should be encoded using the Dublin Core 'description' term, as text.

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

---

Suggested prefix	Namespace URI
foaf	"http://xmlns.com/foaf/0.1/"
dcterms	"http://purl.org/dc/terms/"

---

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

### 3.3 Examples

#### 1. Representing a person, a group and a software package

---

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Person>
    <foaf:givenName>Joe</foaf:givenName>
    <foaf:familyName>Bloggs</foaf:familyName>
  </foaf:Person>

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

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

</rdf:RDF>
```

---

## 2. Specifying members of a group

This could be done ‘inline’ as follows:

---

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Group>
    <foaf:name>Auckland Bioengineering Institute</foaf:name>
    <foaf:member>
      <foaf:Person>
        <foaf:name>Joe Bloggs</foaf:name>
      </foaf:Person>
    </foaf:member>
  </foaf:Group>

</rdf:RDF>
```

---

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:

---

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

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Person rdf:nodeID="joe_bloggs">
    <foaf:givenName>Joe</foaf:givenName>
    <foaf:familyName>Bloggs</foaf:familyName>
  </foaf:Person>

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

</rdf:RDF>
```

---

## 3. Attributing creator, timestamp and purpose descriptions to a CellML model

---

```
<?xml version="1.0"?>
<rdf:RDF 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/">

  <foaf:Person rdf:nodeID="joe_bloggs">
    <foaf:givenName>Joe</foaf:givenName>
    <foaf:familyName>Bloggs</foaf:familyName>
  </foaf:Person>

  <rdf:Description rdf:about="./model.cellml#model_example">
    <foaf:maker rdf:nodeID="joe_bloggs"/>
    <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 Metadata
      Specification Framework.
    </dcterms:description>
  </rdf:Description>

</rdf:RDF>
```

---

The above example shows the construction of a FOAF Person object, which becomes the RDF subject of a ‘maker’ relationship for the CellML 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 annotation document.

#### 4. Adding creator and timestamp elements to a model element

Here we assume that the model element we wish to annotate is defined in the “model.cellml” file as

---

```
<component name="model_parameters" cmeta:id="parameters">
  ...other elements...
</component>
```

---

An RDF Description to annotate the component with the cmeta:id of “parameters” could be (using the FOAF Person defined in example 2)

---

```
<?xml version="1.0"?>
<rdf:RDF 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/">
  <rdf:Description rdf:about="./model.cellml#parameters">
    <foaf:maker>joe_bloggs</foaf:maker>
    <dcterms:created rdf:datatype="http://purl.org/dc/terms/W3CDTF">
      2010-11-07
    </dcterms:created>
  </rdf:Description>
</rdf:RDF>
```

---

#### 5. Adding a text comment to a model element

The variable for this example, and for examples 6 and 7, would be defined in a cellml model (in a “model.cellml” file) as follows

---

```
<variable cmeta:id="vi_variable" initial_value="0.025" name="vi" public_interface="out"
  units="flux"/>
```

---

The annotation could be as follows

---

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/">

  <rdf:Description rdf:about="./model.cellml#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>

```

---

## 6. Annotating a comment

Here the comment annotation “vi\_comment” is itself annotated with a creator, timestamp and text comment of its own.

---

```

<?xml version="1.0"?>
<rdf:RDF 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/">

  <rdf:Description rdf:about="./model.cellml#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="joe_bloggs"/>
    <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>

```

---

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.

## 7. A variable with a commented timestamp

Here the “vi\_variable” variable is annotated with a timestamp, which in turn is commented with the timestamp, and a text comment

---

```

<?xml version="1.0"?>
<rdf:RDF 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/">

  <rdf:Description rdf:about="./model.cellml#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="joe_bloggs"/>
    <dcterms:description>This date may be plus or minus 2 days</dcterms:description>
  </rdf:Description>

</rdf:RDF>

```

---

## 4 Licensing Specification 2.0

### 4.1 Introduction

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

### 4.2 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 RDF Objects are to be 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 then that document that should be specified as the RDF Object of an annotation statement. The recommended Dublin Core namespace is as follows:

---

Suggested prefix	Namespace URI
dcterms	"http://purl.org/dc/terms/"

---

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

## 4.3 Examples

### 1. A URI reference to a license

---

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/">

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

</rdf:RDF>
```

---

### 2. Both a license URI and a literal text of the license

---

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/">

  <rdf:Description rdf:about="./model.cellml#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>

</rdf:RDF>
```

---

## 5 Citation Specification 2.0

### 5.1 Introduction

This specification describes how to make 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.

### 5.2 Realisation Strategy

Citable works are linked to a CellML model document or an element of that document by making the `cmeta:id` of the element an RDF Subject, and the reference to 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 that labelled as '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 namespaces are as follows:

---

Suggested prefix	Namespace URI
bibo	"http://purl.org/ontology/bibo/"
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 Metadata Framework Basic Information Specification 2.0 for details), hence using those properties in a BIBO object may also require the specification of additional namespaces, if they are not already defined.

## 5.3 Examples

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

---

```
<?xml version="1.0"?>
<rdf:RDF 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:Description rdf:about="./model.cellml#ip3_model">
    <bqmodel:description rdf:resource="http://identifiers.org/pubmed/17693463"/>
  </rdf:Description>

</rdf:RDF>
```

---

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

---

```
<?xml version="1.0"?>
<rdf:RDF 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:Description rdf:about="./model.cellml#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>
      Perturbations 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>
```

---

## 3. Extending the previous example: the Article declaration contains an ordered list of authors

---

```
<?xml version="1.0"?>
<rdf:RDF 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:Description rdf:about="./model.cellml#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>
      Perturbations 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>
```

---

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

---

```

<?xml version="1.0"?>
<rdf:RDF 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:Description rdf:about="./model.cellml#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>

```

---

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

---

```

<?xml version="1.0"?>
<rdf:RDF 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#">

  <rdf:Description rdf:about="./model.cellml#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:duration>
    <dcterms:title>Marsh Warbler Symposium 2010</dcterms:title>
  </bibo:Conference>

</rdf:RDF>

```

---

## 6. Biological Annotation Specification 2.0

### 6.1 Introduction

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

### 6.2 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’.

The recommended Biological qualifier namespace is shown in the table below.

**Suggested prefix    Namespace URI**

bqbiol	"http://biomodels.net/biology-qualifiers/"
--------	--

In keeping with the Biomodels framework, the RDF Objects of the annotation 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. The recommended (sub-)ontologies are the following:

- Protein Ontology
- UniProt
- ChEBI
- Gene Ontology:cell component
- Cell Ontology
- Mouse Adult Gross Anatomy
- Foundational Model of Anatomy
- Ontology of Physics for Biology

Where possible, Identifier.org URIs (<http://identifiers.org>) should be used as the RDF Object URIs, as these are persistent and easily dereferenced.

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.

## 6.3 Examples

### 1. A component represents a particular protein

The protein is listed in the Protein Ontology as ID:000005120.

```
<?xml version="1.0"?>
<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="./model.cellml#c">
    <bqbiol:hypernym rdf:resource="http://identifiers.org/obo.pr/PR:000005120" />
  </rdf:Description>

</rdf:RDF>
```

### 2. An equation contains two terms that deal with different biological processes

The biological processes are represented by Gene Ontology records 0051603 and 0042398.

---

```
<?xml version="1.0"?>
<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="./model.cellml#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>
```

---

## 7 Acknowledgements

The authors wish to thank Randall Britten, David Brooks, Richard Christie, James R. Lawson, Caton Little, Catherine M. Lloyd, Andrew K. Miller, Maxwell L. Neal, David P. Nickerson, Poul F. M. Nielsen and Tommy Yu for fruitful discussions.