The CellML Metadata 1.0 Specification
Working Draft - 2 November 2001

2 Overview

The CellML development team has decided to use existing standards wherever possible to describe metadata for reasons listed below. This section describes our use of RDF, Dublin Core, vCard, BQS, and our own CellML Metadata.

2.1 Resource Description Framework

Information about RDF can be found on the W3C's Resource Description Framework (RDF) Page[1].

2.1.1 What is RDF?

RDF, which stands for the "Resource Description Framework", is the W3C's recommendation for handling metadata on the web. The Resource Description Framework is just that: it is a framework that allows you to store descriptions (i.e., metadata) about resources. A resource can be literally *anything*. For the purposes of CellML, resources can be the model document, the model itself, or components in the model.

2.1.2 The Advantages of Using RDF

RDF by itself does not allow people to store metadata. It merely provides a standard framework onto which various groups can hang their metadata vocabularies. Some benefits of having this standard framework are:

- **It provides a common *attribute*=*value* data model for the metadata**. All metadata expressed in RDF can be presented as a series of attributes (i.e., properties of the resource) and their values. For instance, one attribute:value pair for a CellML model might be `species=Mus musculus`. Furthermore, the W3C provides SiRPAC[2], a Simple RDF Parser and Compiler which returns a graphical representation of the RDF code it is fed to aid in visualization of the attribute:value pairs.
- **It provides an extensible method for storing metadata of increasing complexity**. Some metadata properties will have simple values, such as the species property shown above. Other metadata properties will have complex values. In the latter case, the value of the metadata property is itself considered a resource, and additional metadata properties are stored about it. This is made more clear by an example. Consider the case of the `model creator` property. This could be given a simple value of the creator's name, such as **`John Doe`**. However, it is more powerful to consider the value of the creator property to be a new resource (the person identified by the name "John Doe"). This allows the person's name to be stored as metadata about the new resource. Why bother doing this? Because it also allows additional metadata to be stored about the person, such as the person's mailing address, phone number, etc. Most importantly, we don't have to know ahead of time what sorts of metadata processing software might want to store about the person. If a particular application wants to store the person's favorite colour, it can do so. Other applications might not recognise the meaning of the particular element that stores the favourite colour, but they will be able to understand that it is some sort of property about the resource (i.e., person) that is the creator of the model. This allows the application to handle the unknown metadata gracefully (most likely, many applications would at least be able to present the *attribute*=*value* pair to the user).

[1] http://www.w3.org/RDF/
[2] http://www.w3.org/RDF/Validator/

- **It makes it possible for applications that don't know anything about CellML to understand our metadata**. Though not a reality yet, it is part of Tim Berners-Lee's vision of a semantic web. Eventually, search engine tools could become RDF capable. In that case, people would be able to perform much more powerful searches for information on the web. If someone wants to find all web resources created by John Doe, he/she could search explicitly for resources where `creator=John Doe`, instead of just searching for resources that contain the string "John Doe".
- **There are tools out there that use RDF**. It is true that RDF is still a fledgling technology. However, there are tools out there that parse RDF and tools that actually use RDF to build databases, knowledge stores, and other such things. See the W3C's RDF project list[3] for a list of tools and projects using RDF.

2.1.3 Multiple Methods for Expressing Metadata in RDF

The RDF Model and Syntax Specification[4] allows multiple methods for expressing the same information. Perhaps the reason it does so has to do with the fundamental database design principles of generalize whenever possible and overload your classes as much as possible. This is effective database design because generalizing and overloading are both useful techniques to create a flexible data model that will handle new types of information as they become necessary. Mapping each type of information to a column in a database table (or an element in an XML schema) requires that new types of information be stored in new columns (or elements). A generalized structure that handles an entire class of information allows the handling of any type of information without changing a previously specified schema.

The CellML development team has picked one way of expressing metadata in RDF and sticks with the same method throughout the CellML Metadata 1.0 Specification. This is the recommended way of implementing RDF in CellML, but it is not the only way of representing metadata. From here on, the **`rdf`** prefix will be used to indicate that elements and attributes are in the RDF namespace.

2.2 The Dublin Core

Information about the Dublin Core can be found on the Dublin Core's website[5].

2.2.1 What is the Dublin Core?

The Dublin Core is a group of metadata properties. These properties were identified as "common" across a large range of resources by a group of library science and knowledge management folks. These properties are things like creator, publisher, subject, and date. A full list, with definitions, can be found in the Dublin Core Metadata Element Set, Version 1.1: Reference Description[6].

The Dublin Core Metadata Initiative group has also provided a standard set of "qualifier" elements. These elements add information to the basic elements. Qualifier elements either provide *type* information or *scheme* information. Type information classifies the basic element. For instance, the date element can have a type of created, modified, valid, available, or issued. Scheme information indicates how the content of the element is encoded. For instance, the date element can have a scheme of W3C-DTF or DCMI Period. A full list of qualifiers and their allowed values can be found in the Dublin Core Qualifiers document[7].

It is important to note that Dublin Core does not have to be expressed in RDF. The Dublin Core elements are not elements in the XML sense. They are simply standard names and definitions for common types of

[3] httP://www.w3.org/RDF/#projects
[4] http://www.w3.org/TR/REC-rdf-syntax/
[5] http://dublincore.org/
[6] http://dublincore.org/documents/dces/
[7] http://dublincore.org/documents/dcmes-qualifiers/

metadata. However, the Dublin Core Metadata Initiative has published two articles that suggest a method for implementing an RDF representation of Dublin Core elements and qualifiers: Expressing Simple Dublin Core in RDF/XML[8] and Expressing Qualified Dublin Core in RDF/XML[9], respectively.

2.2.2 The Advantages of Using Dublin Core

Though not all of the Dublin Core elements are immediately applicable to CellML Metadata requirements, many of them do fulfill those requirements. This set of elements is widely referenced, and the W3C designed the Resource Description Framework with the Dublin Core in mind. General purpose tools are more likely to understand the Dublin Core metadata vocabulary than any other vocabulary. Also, it makes it more obvious that certain CellML Metadata properties (such as model creator) map directly to metadata properties that are found in other fields.

Henceforth, the prefixes **dc** and **dcq** will indicate that elements and attributes are in the Dublin Core and the Dublin Core Qualifiers namespaces, respectively.

2.3 vCard

It seems the only existing RDF definition of metadata about people is a note submitted to the W3C in February 2001 entitled Representing vCard Objects in RDF/XML[10]. (This note is the work of Renato Iannella working at the Distributed Systems Technology Centre at the University of Queensland and orginally appeared on their RDF project page[11].) This note's suggestions are fast becoming the standard method of referencing people in RDF.

As the vCard data model includes some elements that are not necessary for CellML Metadata, such as nickname and birthday, we will not require CellML processing software to recognize those elements. However, model authors are free to use them. That is, the use of vCard elements outside of the list defined in the CellML Metadata specification will not invalidate the metadata, but these elements may not necessarily be recognized by all CellML Metadata compliant processing software.

CellML Metadata compliant processing software is expected to recognize the following "vCard in RDF" elements that meet the information needs of CellML:

- **<vCard:N>** (the name construct), with all of its subelements:
  - **<vCard:Family>**: the person's family, or last name
  - **<vCard:Given>**: the person's given, or first name
  - **<vCard:Other>**: additional names, used for middle names and initials
  - **<vCard:Prefix>**: honorific prefixes, such as "Dr."
  - **<vCard:Suffix>**: suffixes such as "III" and "Jr."
- **<vCard:ADR>** (the mailing address construct), with all of its subelements:
  - **<vCard:Pobox>**: post office box
  - **<vCard:Street>**: street address
  - **<vCard:Locality>**: city, town, rural route, etc.
  - **<vCard:Region>**: state, etc.
  - **<vCard:Country>**: country

[8] http://dublincore.org/documents/dcmes-xml/
[9] http://dublincore.org/documents/dcq-rdf-xml/
[10] http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222
[11] http://www.dstc.edu.au/Research/Projects/rdf/

  - **<vCard:Pcode>**: postal code (such as the American zip code)
  - **<vCard:Extadd>**: extended address field. This is used to include the company or institution name.

- **<vCard:EMAIL>** (the e-mail address construct)
- **<vCard:TEL>** (the telephone number construct)
- **<vCard:ORG>** (the organization construct, which maps to the CellML requirement to be able to store a person's affiliation), with all of its subelements:
  - **<vCard:Orgname>**: the name of the organization (i.e., "The University of Auckland")
  - **<vCard:Orgunit>**: the division or department (i.e., "The Bioengineering Research Group")
- **<vCard:TITLE>**: the person's job title
- **<vCard:ROLE>**: the person's job role

The **<rdf:type>** element is used to specify "type parameters" on certain vCard elements. For instance, an address may be typed as domestic, international, postal, parcel, home, work, or preferred. Note that one address may be given more than one type. See section 3.3 of the vCard in RDF document[12] for more info.

Examples throughout the rest of this specification demonstrate the use of vCard elements in RDF. These elements are preceded by the **vCard** prefix to indicate that they are in the vCard namespace.

2.4 Bibliographic Query Service

No bibliographic standards yet exist within RDF/XML. However, the Object Management Group has published the Bibliographic Query Service Specification[13]. The **DSLSRBibObjects** Module from this specification presents an excellent general data model for bibliographic references. The CellML development team recommends an RDF serialization of this data model (henceforth referred to as the "BQS data model") described in detail in Section 5[14]. BQS metadata is designated by the namespace prefix **bqs** in this specification.

2.5 CellML Metadata

A CellML Metadata namespace has been created to include all metadata that has not been previously defined by the four standards listed above. These include biology-related attributes (such as species and bio-entities) as well as properties we felt were missing from other standards (such as annotations). We recommend CellML Metadata be designated by the namespace prefix **cmeta**.

2.6 Namespaces in CellML Metadata

Namespace URIs and recommended prefixes are given in Table 1.

[12] http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222

[13] http://industry.ebi.ac.uk/openBQS/

[14] http://www.cellml.org/public/metadata/20011102/cellml_metadata_specification.html#sec_bqs

Namespace Name	**Namespace URI**	**Recommended Prefix**
CellML Metadata	`"http://www.cellml.org/metadata/1.0#"`	**`cmeta`**
RDF	`"http://www.w3.org/1999/02/22-rdf-syntax-ns#"`	**`rdf`**
RDF Schema	`"http://www.w3.org/2000/01/rdf-schema#"`	**`rdfs`**
Dublin Core	`"http://purl.org/dc/elements/1.1/"`	**`dc`**
DC Qualifiers	`"http://purl.org/dc/terms/"`	**`dcq`**
BQS	`"http://www.cellml.org/bqs/1.0#"`	**`bqs`**

TABLE 1: The names, URIs and recommended prefixes of the namespaces referenced in this specification.