CellML Specification

Draft — 30 September 2003

1 Introduction

1.1 Introduction to CellML

This document specifies CellML^{TM} 1.1, an XML-based language for describing and exchanging models of cellular and subcellular processes. CellML is being developed by scientists at the University of Auckland (in the Bioengineering Institute) and at Physiome Sciences, Inc. The development of CellML is guided by an advisory board drawn from many different areas of biological modelling (see the project team[1] page on the CellML website for more information). CellML is being developed as an open standard, and all interested parties are encouraged to send feedback to info@cellml.org, or to the cellml-discussion[2] mailing list.

1.1.1 Purpose and scope of CellML

CellML is intended to support the definition of models of cellular and subcellular processes. CellML facilitates the reuse of models and parts of models by using a component-based architecture. Models are split into logical sub-parts called components that are connected together to form a model.

CellML separates the specification of the underlying mathematics of a model from the implementation of the model's solution. This makes a model independent of a particular operating system or programming language and allows modellers to easily integrate parts of other peoples' models into their own models. CellML also allows the generation of equations for publishing from the same definition upon which the solution method is based, removing inconsistencies between the model and associated results in academic papers, and allowing others to reliably reproduce these results.

The scope of the CellML language is specifically limited to the definition of model structure. All other types of information that modellers need or want to include in a model document are incorporated using other languages. For instance, mathematics is included in CellML documents using Mathematical Markup Language[3] (MathML). Metadata may be included using the Resource Description Framework[4].

1.1.2 What is XML?

The CellML language is defined in terms of a meta-language called eXtensible Markup Language (XML)[5]. XML is a standard published by the World Wide Web Consortium[6], the organisation responsible for defining many internet-related standards, including HTML. XML is essentially a means of adding structure to text documents, allowing machines to unambiguously associate text or binary data with a particular component in a document's data model.

XML is an appropriate medium for CellML because it is both human and machine readable. A model author can create a CellML document with a text editor or with CellML authoring software. XML is a well-defined and widely used specification. Many free software utilities and libraries for the processing of XML already exist, simplifying the development of CellML processing software. XML has also been designed

[1] http://www.cellml.org/public/about/project_team.html

[2] http://www.cellml.org/public/mailing_lists/discussion.html

[3] http://www.w3.org/Math/

[4] http://www.w3.org/RDF

[5] http://www.w3.org/XML/

[6] http://www.w3.org/

to be usable over the internet, making CellML suitable for the interchange of models between software and databases at different locations.

A quick introduction to XML[7] is available in the examples section of the CellML website.

1.1.3 Terminology

A model is a representation of the rules that govern the behaviour of a system. The terms in the following list provide two useful model classifications.

- **qualitative model**

  A model that defines the relationships between objects in the system, without defining any mathematics that represent the behaviour of those objects.

- **quantitative model**

  A model that defines the relationships between objects in the system, including the mathematics that represents the behaviour of those objects.

The terms defined in the following list are used in specifying the conformance of CellML documents and processing software to this specification.

- **may**

  Conforming CellML documents are permitted but not required to adhere to the limitation described. Conforming CellML software is permitted but not required to behave as described.

- **must**

  Conforming CellML documents must adhere to the limitation described. Conforming CellML software must behave as described.

- **for interoperability**

  A non-binding recommendation included to increase the chances that CellML documents will be processed in a consistent manner by different applications.

- **error**

  A violation of the rules of this specification; results are undefined. Conforming CellML software may detect and report an error and may recover from it. The recommended best practice is for software to make information about errors available to the user.

- **valid CellML document**

  A document that conforms to all of the rules in this specification.

- **valid CellML subset document**

  A valid CellML document that only uses MathML elements from the CellML subset defined in Section 4.2.3[8].

- **CellML conformant software**

  CellML processing software that will interpret any valid CellML subset document according to the language semantics and processor rules defined in this specification.

- **fully MathML capable software**

  Software that can correctly interpret the full set of MathML content markup elements.

[7] http://www.cellml.org/examples/introduction/xml_guide.html

[8] http://www.cellml.org/public/specification/20030930/cellml_specification.html#sec_math_cellml_subset

1.2 Structure of the CellML Specification

The CellML specification is divided into several sections, each of which discusses a particular aspect of CellML:

- Section 1 — **Introduction** — This section introduces CellML, XML, the terminology used throughout the specification, and the structure of the specification.
- Section 2 — **Fundamentals** — This section explains concepts used in all other sections of the specification, such as the definition of a valid CellML identifier and the use of XML namespaces in CellML.
- Section 3 — **Model Structure** — This section describes how models are organised in CellML. It includes an explanation of the use of a network of components to define a model and a discussion of variables in CellML.
- Section 4 — **Mathematics** — This section describes how mathematical expressions are defined in CellML documents using MathML and defines the CellML subset of MathML elements.
- Section 5 — **Units** — This section explains the requirements for units in CellML and describes how a modeller can define arbitrary sets of units.
- Section 6 — **Grouping** — This section explains how a model can be organised into logical encapsulation and geometric containment hierarchies by grouping components.
- Section 7 — **Reactions** — This section introduces CellML syntax that allows the modeller to classify the involvement of the participants in the chemical expressions that make up reaction/pathway models.
- Section 8 — **Metadata Framework** — This section describes how RDF is used in CellML documents to define metadata and associate it with models, model components, and other CellML elements.
- Section 9 — **Importing Models** — This section explains how a model may be built on existing models. The import feature also allows a modeller to create an incomplete model, with the expectation that the necessary components and connections may be included in the future.
- **Appendices** — The appendices cover advanced and technical topics including the CellML DTD, recommendations for adding scripts to CellML documents, and units processing algorithms.

A valid CellML model can be created using nothing beyond the material covered in the fundamentals, model structure, mathematics, and units sections of the specification. The concepts in the remaining sections of the specification allow modellers to build more meaningful models.

Each section of the specification is further divided into five subsections:

- **Introduction** — This subsection explains the purpose of the elements covered in the current section.
- **Basic Structure** — This subsection describes the new elements and attributes introduced in the current section of the specification and how they are combined.
- **Examples** — This subsection provides one or two basic examples of the correct use of the elements and attributes introduced in the current section of the specification. More extensive examples can be found in the examples section[9] of the CellML website.
- **Rules for CellML Documents** — This subsection provides formal rules for the use of the elements and attributes introduced in the current section to create valid CellML documents. These rules are specified as bulleted lists. Each rule may have an associated explanation, which appears directly after the rule in square brackets ([]).
- **Rules for Processor Behaviour** — This subsection provides some rules for correct CellML processor behaviour with regards to the elements and attributes introduced in the current section.

[9] http://www.cellml.org/examples/introduction/index.html

Throughout the CellML specification, all XML elements and attributes that occur in the text are in the CellML namespace unless explicitly stated otherwise.

E-mail questions, criticism, submissions or info to info@cellml.org
Input document last modified : Tue Sep 30 14:43:07 NZST 2003