

18 May 2001 CellML 1.0 Specification Errata

1 Introduction

This document records known errors in the 18 May 2001 Final Draft of the CellML 1.0 specification, which is available at the following URI:

http://www.cellml.org/public/specification/20010518/cellml_specification.html

The errors listed below have been fixed in the developer's "unstable" version of the CellML 1.0 specification, which is available at the following URI:

http://www.cellml.org/private/unstable_cellml_spec/cellml_specification.html

The latest stable version of the CellML 1.0 specification is available at the following URI:

<http://www.cellml.org/public/specification/index.html>

2 Known Errors

2.1 Known errors as of 10 August 2001

- [Specification Wide](#)¹
 - The description of how metadata content is to be included within CellML elements needs to be changed document-wide from “metadata framework elements, as described in Section 8” to “**<RDF>** elements in the RDF namespace”.
 - Where a CellML element may only contain metadata content, the “which may appear in any order” clause should be removed from the content model rule.
- [Section 1.2: Structure of the CellML Specification](#)²
 - The description of the Mathematics section should be changed to “This section describes how mathematical expressions are defined in CellML documents using MathML, and defines the CellML subset of MathML elements.”
 - The last sentence, regarding the list of changes, should be removed from the description of the Appendices.
- [Section 2.3: Examples](#)³
 - In the caption for Figure 2, “electro-physiological” shouldn't be hyphenated.
 - In Figure 3, the order of the namespace declarations on the **<model>** element should be changed so that **cellml** appears before **app**.
 - In the caption for Figure 3, the word “URI” should be added to the second sentence, giving: “The extension namespace URI was invented for demonstration purposes only.”

¹http://www.cellml.org/public/specification/20010518/cellml_specification.html

²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_intro_spec_structure

³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_examples

- [Section 3.2.2: Definition of components](#)⁴
 - In the second sentence of the first paragraph, “electro-physiological” shouldn’t be hyphenated.
- [Section 3.2.3: Definition of variables](#)⁵
 - In the first sentence of the paragraph after the bulleted list, “set” should be changed to “sets”, giving: “... so the value of the **initial_value** attribute sets the starting condition for a simulation ...”
- [Section 3.2.4: Definition of connections](#)⁶
 - In paragraph two, the fourth sentence (“It is not necessary ...”) should be removed. The “However” at the start of the fifth sentence should be removed, giving: “The interface attributes ...”
 - The sentence deleted from paragraph two should be added to paragraph five, along with the clause “although this will typically be the case”, giving: “It is not necessary for the variables that are to be mapped to each other to have the same name, although this will typically be the case.”
- [Section 3.3: Examples](#)⁷
 - In the first sentence, “encoding” should be changed to “description”, giving: “Figure 4 contains a portion of the CellML description of the Hodgkin-Huxley squid axon model ...”
 - In the second sentence of the second paragraph, the word “belons” is not a new term, so should not be emphasised. It should remain in double quotes however, giving: “It has a **public interface** attribute value of "out", which indicates that the variable “belongs” to this component ...”
 - In Figure 4, the **<plus>** element in the MathML block should actually be a **<minus>** element.
 - In Figure 4, the value of the **id** attribute on the first **<apply>** element in the MathML block should be "membrane_voltage_diff_eq".
 - In the caption for Figure 4, “definition” should be changed to “description”, giving: “A small portion of the CellML description of the Hodgkin-Huxley squid axon model ...”
 - In paragraph four, the third sentence “The full set of equations ...” should be removed. In the latest version of the Hodgkin-Huxley model, the equation in the example is the only equation in the membrane component.
 - In Equation (1), the numerator on the RHS should have a minus sign in front.
- [Section 3.4.1: The **<model>** element](#)⁸
 - In the first rule, the ordering of the list of the elements in the CellML namespace that are valid children should be changed to “**<units>**, **<component>**, **<group>**, and **<connection>** elements”
 - In the explanation for the first rule, “Recommended practice” should be changed to “The recommended best practice”.

⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_component_element

⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_variable_element

⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_connection_element

⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_examples

⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_spec_rules_model_element

- [Section 3.4.2: The `<component>` element](#)⁹
 - In the explanation for the first rule, “Recommended practice” should be changed to “The recommended best practice”.
- [Section 3.5.1: Mapping of variables](#)¹⁰
 - The text of this section defined the semantics of the `<connection>`, `<map_components>` and `<map_variables>` elements. Element semantics have already been defined in the discussion section, and should not be defined in the processor behaviour section. This section can be used to introduce the topic of units conversion however, with the following text: “For interoperability, CellML processing software should take into account the units definitions referenced by any two variables that are mapped together. If the units references are not equivalent, as defined in Appendix C.2.1, then a conversion may be required. An algorithm for performing this conversion is proposed in Appendix C.3.5.”
- [Section 4.2.1: Definition of mathematics](#)¹¹
 - In the second sentence, “A `<mathml:math>` element ...” should be changed to read “`<mathml:math>` elements ...”
- [Section 4.2.2: MathML’s presentation and content markup elements](#)¹²
 - The third sentence in the second paragraph should be re-arranged for clarity, giving: “There is one exception: model authors may associate rendering information with a particular expression by placing MathML presentation markup elements inside a `<mathml:annotation-xml>` element.”
- [Section 4.2.7: Definition of scripts](#)¹³
 - In the last bullet point, all instances of the word “should” should be replaced with the word “must”, giving: “Functions must be side-effect free. That is, a function must not assign values to variables that are not local to that function. In particular, functions must not alter the values of their arguments or global variables.”
- [Section 4.5.3: Treatment of annotations](#)¹⁴
 - The first and second sentences should be switched to give better right-margin justification in the PDF version.
- [Section 5.2.2: User defined units](#)¹⁵
 - In the first sentence of the last paragraph, “The `offset` attribute ...” should be changed to “An `offset` attribute ...”

⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_spec_rules_component_element

¹⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_behaviour_rules_mapping

¹¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_bs_definition_of_mathematics

¹²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_presentation_vs_content

¹³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_scripting

¹⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_rules_treatment_of_annotations

¹⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_user_defined_units

- [Section 7.2: Basic Structure](#)¹⁶
 - In the sixth sentence of paragraph three, “must only” should be changed to “may”, giving: “The optional **direction** attribute may be used on **<role>** elements in reversible reactions.”
- [Section 7.4.3.1: Allowed use of the **<role>** element](#)¹⁷
 - In the second bullet, “**delta_variable**” and “**direction**” should be switched to give better right-margin justification in the PDF version.
- [Section 7.4.3.3: Proper use of the **role** attribute](#)¹⁸
 - The wording of the first rule should be re-arranged to give: “A **<reaction>** element must contain no more than one **<variable_ref>** element with a **<role>** element with a **role** attribute with a value of “rate”.”
 - In the third rule, the comma after “**delta_variable**” should be removed.
- [Section 7.4.3.5: Proper use of the **direction** attribute](#)¹⁹
 - The wording of the first rule is confusing and should be changed to: “A **direction** attribute on a **<role>** element that is inside a **<reaction>** element with a **reversible** attribute value of “no” must have a value of “forward”.”
 - The second rule should start with “A **direction** attribute ...”, not “The **direction** attribute ...”
- [Section 7.4.3.8: Proper use of the **delta_variable** attribute](#)²⁰
 - The first rule should start with “A **delta_variable** attribute ...”, not “The **delta_variable** attribute ...”
 - The last sentence of the explanation for the second rule should be re-arranged to give better right-margin justification in the PDF version, giving: “If the **stoichiometry** attribute is absent, the relationship between the variable assigned the role of “rate” and the variable named in the **delta_variable** attribute must be defined using MathML.”
- [Section 7.5.5: Math implied by the **delta_variable** and **stoichiometry** attributes](#)²¹
 - In the first paragraph, a clause should be added explaining the expected value of the rate variable, giving: “The use of the **delta_variable** and **stoichiometry** attributes on a **<role>** element implies the following mathematical relationship between the declared delta variable and the rate variable (where the variable representing the reaction rate will have a negative value when the reaction is proceeding in the forward direction)”.
 - In the second sentence of the last paragraph, “cannot” should be replaced with “must not”, giving: “Therefore, a modeller must not declare ...”

¹⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_basic_structure

¹⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

¹⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

¹⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

²⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

²¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_implied_math

- [Section 7.5.6: Meaning of mathematics in reactions](#)²²
 - The following sentence should be added to the first bullet describing reaction rate: “Conventionally, the variable representing the reaction rate will have a negative value when the reaction is proceeding in the forward direction.”

- [Appendix A.6: The CellML 1.0 DTD](#)²³
 - The content models for the `<connection>` and `<group>` elements indicate an ordering of sub-elements, where the specification specifically states that they may appear in any order. These should be changed to:

```
<!ELEMENT connection (map_components | map_variables)+>
<!ELEMENT group (relationship_ref | component_ref)+>
```

- [Appendix C.3.2: Units-based restrictions on the use of MathML operators](#)²⁴
 - The “Note that” should be removed from the start of the second sentence in the second paragraph to give better right-margin justification in the PDF version.

2.2 Known errors as of 2 August 2001

- [Appendix A.6: The CellML 1.0 DTD](#)²⁵
 - In the header comment of the CellML 1.0 DTD, the line starting “PUBLIC IDENTIFIER” should be removed, and the line starting “COPYRIGHT” be changed to “COPYRIGHT : (2001) Bioengineering Research Group, University of Auckland”. The comment and URI following “DESCRIPTION” should be changed to reflect the “10 August 2001 Specification for CellML 1.0”, available at “http://www.cellml.org/public/specification/20010810/index.html”.
 - In the CellML 1.0 DTD, the “mathml_dtd_path” entity should be changed to the online URI for the MathML 2.0 DTD, so that it need not be reset by users of the CellML 1.0 DTD. This gives:

```
<!ENTITY % mathml_dtd_path
    "'http://www.w3.org/TR/MathML2/dtd/mathml2.dtd'">
```

- In the CellML 1.0 DTD, the attribute list for the units element is wrong. The `base_units` attribute should be added, giving:

```
<!ATTLIST units
    %cellml_common_attributes;
    name                CDATA                #REQUIRED
    base_units          (yes|no)              #IMPLIED
>
```

- In the CellML 1.0 DTD, the content model for the `<variable_ref>` element is wrong. `<math>` elements may not be placed inside `<variable_ref>` elements, and this should be removed from the content model, giving:

```
<!ELEMENT variable_ref (role+)>
```

²²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_use_of_math

²³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_cellml_1_0_dtd

²⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_rules_restrictions_on_the_use_of_mathml_ope

²⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_cellml_1_0_dtd

- [Appendix A.4: Use of MathML within CellML](#)²⁶

- By changing the value of the “mathml_dtd_path” entity to point directly to the MathML 2.0 DTD (see below) we need to expand the first paragraph to include an example demonstrating how the MathML 2.0 DTD can be referenced on the W3C website, as follows:

The CellML 1.0 DTD can be used in conjunction with the [MathML 2.0 DTD](#)²⁷ to validate any MathML content within a CellML document. The CellML DTD contains a reference to the MathML 2.0 DTD, which is maintained at the [World Wide Web Consortium’s website](#)²⁸, inside a conditional section, which by default is not included in the CellML DTD. CellML document authors may cause the conditional section to be included by setting the value of the “use_mathml_dtd” entity to “INCLUDE” inside the internal subset of their CellML document, as shown below:

```
<!DOCTYPE model SYSTEM "http://www.cellml.org/cellml/cellml_1_0.dtd" [
  <!ENTITY % use_mathml_dtd "INCLUDE">
]>
```

If copies of both the CellML 1.0 DTD and the MathML 2.0 DTD are available on the local filesystem, it is possible to conveniently override the reference to the MathML 2.0 DTD by setting the value of the “mathml_dtd_path” entity to the appropriate relative path from the CellML DTD to the MathML DTD, as shown below:

```
<!DOCTYPE model SYSTEM "/my/local/copy/of/the/cellml_1_0.dtd" [
  <!ENTITY % mathml_dtd_path "'relative/path/to/mathml2.dtd'">
  <!ENTITY % use_mathml_dtd "INCLUDE">
]>
```

2.3 Known errors as of 1 August 2001

- [Section 1.2: Structure of the CellML Specification](#)²⁹

- At the end of this section, the statement about all elements being in the CellML namespace unless otherwise stated from Section 3.2.1 should be inserted. This sentence should read: “Throughout the CellML specification, all XML elements and attributes that occur in the text are in the CellML namespace unless explicitly stated otherwise.”

- [Section 2.1: Introduction](#)³⁰

- There is a superfluous comma in the first sentence of this section. The sentence should read: “This section of the CellML specification introduces some concepts that are used throughout the entire language and defines rules that apply to all or many of the other parts of the specification.”

- [Section 2.2.1 Definition of a valid CellML identifier](#)³¹

²⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_dtd_use_of_mathml

²⁷<http://www.w3.org/TR/MathML2/dtd/mathml2.dtd>

²⁸<http://www.w3.org/>

²⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_intro_spec_structure

³⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_introduction

³¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_basic_structure

- In the second paragraph, the third and fourth sentences both contain superfluous commas. Those sentences should read: “For this reason CellML identifiers must consist of only alphanumeric characters and the underscore character (“_”), and are subject to some additional constraints outlined below. These names will generally not be the most effective way of identifying the object to humans working with CellML models as it is not possible to include whitespace or formatting.”
- The last paragraph should be removed as it might be misinterpreted to mean that if the definition of SBML changes, the definition of CellML would change.
- [Section 2.2.2: Namespaces in CellML](#)³²
 - In the third paragraph, the fourth sentence contains a superfluous comma and should read: “MathML is particularly important to CellML because content in this namespace is considered as fundamental as content in the CellML namespace.”
 - In the third paragraph, the fifth sentence has two superfluous commas and needs to specifically state that it is an **id** attribute that links metadata to a CellML object. It should read: “Metadata is defined using elements in the RDF namespace and linked to CellML elements using an **id** attribute in the CellML Metadata namespace as described in Section 8.”
- [Section 2.2.3: Extending CellML documents](#)³³
 - In the first paragraph, the last sentence has a superfluous comma, and should read: “Extension elements and extension attributes may appear anywhere in a CellML document as long as the result is well-formed XML.”
 - In the second paragraph, no comma is needed in the first sentence after “For interoperability”. The complete sentence should read: “For interoperability CellML processing software should respect the extension elements and attributes of other applications.”
 - In the second paragraph, adding the word “then” before the second “application B” makes the sentence read easier. The complete sentence should read: “If a model is created in application A, which adds its own extension elements, and is subsequently edited in application B, then application B should attempt to include application A’s extension elements in its output, even if these extension elements are now invalid.”
- [Section 2.3: Examples](#)³⁴
 - In the second paragraph, the second sentence should read “The root element sets the default namespace to the CellML namespace URI and explicitly maps the CellML namespace URI to the **cellml** prefix.”
 - In Figure 3, the namespace URI mapped to the **app** prefix should be changed to "<http://www.physiology.org>". A namespace URI in the "**software.com**" domain is probably not a good idea.
- [Section 2.5.3: Treatment of extension namespaces](#)³⁵
 - In the first rule, the second sentence of the explanation has a superfluous comma and should read: “Polite software should attempt to store non-CellML data so that it can write it out again when it exports the document.”

³²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_namespaces

³³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_extending_cellml

³⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_examples

³⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_processor_rules_extension_namespaces

- [Section 3.2.1: Definition of a model](#)³⁶
 - In the first paragraph, the second sentence, which deals with all elements being in the CellML namespace unless stated otherwise, should be moved into the introduction, as it actually applies for the entire document, and not just this section.
- [Section 3.2.3: Definition of variables](#)³⁷
 - In the first paragraph, the words “investigate” and “biological” should be substituted in place of “simulate” and “physiological”, respectively, giving: “Models are usually developed to investigate the behaviour of a number of variables that have biological significance.”
 - The bullet point relating to the **initial_value** attribute should be changed to read:
 - * **initial_value** — This attribute provides a convenient means for specifying the value of a scalar real variable when all independent variables in the model have a value of 0.0. Independent variables are those with respect to which another variable is differentiated or integrated.
 - Immediately after the bulleted list, the following paragraph explaining the **initial_value** attribute in more detail should be added:

The name of the **initial_value** attribute derives from the fact that, in a model with only one independent variable, this would generally correspond to time, and so the value of the **initial_value** attribute set the starting condition for a simulation which progressed from time equals 0.0. The initial values of variables need not be set in the model definition at all. When multiple simulations are to be run using the same model, initial and boundary conditions are most conveniently set in an external simulation configuration file loaded separately by CellML processing software.
 - In the last paragraph, the word “singular” in the last sentence should be changed to read “scalar real”, giving: “At the present time, all variables must have scalar real values.”
- [Section 3.2.4: Definition of connections](#)³⁸
 - In the first paragraph, the second, third, and fourth sentences should be run together into the following sentence: “The mapping of variables involves the transfer of a variable’s value from one component to another, a process which may involve a conversion to ensure the units match.”
- [Section 3.3: Examples](#)³⁹
 - In Figure 4 the following comment regarding units definitions should be added before the first **<component>** element: “Units definitions which could be referenced from the **<variable>** elements would typically be inserted here. Units are discussed in Section 5.”
 - As Figure 4 is now quite long, the last sentence (“See the text for more information”) should be removed from the caption.
- [Section 4.1: Introduction](#)⁴⁰

³⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_model_element

³⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_variable_element

³⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_connection_element

³⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_examples

⁴⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_introduction

- In the first paragraph, specifying should be specify in the first sentence, giving: “CellML allows modellers to unambiguously specify the underlying mathematics of a cellular model.”
- In the first sentence, a “2.0” should be added after “Mathematical Markup Language” in the first sentence.
- In the second paragraph, the full history of MathML should be omitted, as too much mention of MathML 1.0 is likely to confuse people. Only the first and last sentences need be retained.
- [Section 4.2.4: Ordering of expressions](#)⁴¹
 - The first draft of this paragraph has been judged long-winded and innaccurate, and should be changed to the following:

The mathematics in a model defined using CellML 1.0 consist of a static system of expressions, which are distributed over a network of components. CellML does not define the order of evaluation of equations, as this is simulation information rather than model information.
- [Section 4.2.5: Scope of expressions](#)⁴²
 - An introductory section should be added after Section 4.2.4 entitled “Scope of expressions” with the following content:

Within a CellML model, all expressions are assumed to have unlimited scope with respect to the independent variables unless explicitly stated using MathML’s `<piecewise>` construct or some other form of conditional expression. This means that if the initial conditions for a variable, the value of which is determined by a differential equation, are to be specified using an equality, the two equations should have their scope limited so that they do not contradict each other.
- [Section 4.5.1: Ordering of expressions](#)⁴³
 - The second sentence should be removed from this section, as it hints at a particular implementation.
- [Section 4.5.2: Scope of expressions](#)⁴⁴
 - A new CellML processing software rule needs to be added after Section 4.5.1 entitled “Scope of expressions” with the following content:

CellML processing software must make no assumptions about the scope or domain of a mathematical expression defined within a model. Unless explicitly stated, all expressions hold for any and all combinations of independent variables.
- [Section 5.2.2: User defined units](#)⁴⁵
 - In Table 3 the symbol columns (the 3rd and 6th columns) should be removed as they don’t add anything and most likely to confuse people into thinking they can use the symbols listed in place of the full prefix. The last sentence of the caption, which discusses the symbol columns, should also be removed.

⁴¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_cellml_subset

⁴²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_cellml_subset

⁴³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_rules_order_of_calculations

⁴⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_rules_order_of_calculations

⁴⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_user_defined_units

- [Section 6.1: Introduction](#)⁴⁶
 - A sentence should be added to the end of the first paragraph to indicate that CellML’s grouping mechanism can be used for non-hierarchical grouping. Something like: “This grouping scheme is general enough that it can be used within CellML documents to specify non-hierarchical grouping relationships between components.”
 - In the second paragraph, the last sentence incorrectly states that “a component must only appear once within a given hierarchy”. This should be changed to read: “a component must only appear once within a set of hierarchies of a given type ...”.
 - In the second paragraph, the following sentence should be appended to ease the transition to the third paragraph: “CellML defines two types of relationship for use within the grouping scheme: *encapsulation* and *containment*.”
 - In the last paragraph, the word “anonymous” should be used in place of “imaginary”.
- [Section 6.2.1: Definition of groups](#)⁴⁷
 - In the last paragraph, the phrase “not be useful” is too vague and should be replaced with “be an error”, giving: “... it would generally be an error for an encapsulating component to be physically inside one of its encapsulated child components.”
- [Section 6.2.2: The *encapsulation* relationship type](#)⁴⁸
 - In the second paragraph, the word “anonymous” should be used in place of “imaginary”.
 - In the body and caption of Table 4, the word “anonymous” should be used in place of “imaginary”.
- [Section 6.2.3: The *containment* relationship type](#)⁴⁹
 - In the last sentence, “restrict” should be replaced with “influence”, giving: “Containment relationships do not influence any aspect of model definition or behaviour.”
- [Section 6.4.2.1: Allowed use of the `<relationship_ref>` element](#)⁵⁰
 - In the second sentence of the explanation after the second rule, the unclear antecedent “this” should be changed to “This restriction has been included to”, giving: “This restriction has been included to prevent conflicts with future versions of the CellML specification ...”
- [Section 7.1: Introduction](#)⁵¹
 - In the third sentence, “It will always be” should be replaced with “It is”, giving: “It is possible to specify a model purely in terms of mathematics, ...”
 - In the fourth sentence, “in reducing” should be changed to “when reducing”, giving: “... information is lost when reducing the model to pure mathematics.”

⁴⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_grouping_introduction

⁴⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_grouping_bs_syntax

⁴⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_grouping_bs_encapsulation

⁴⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_grouping_bs_containment

⁵⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_grouping_spec_rules_relationship_ref_element

⁵¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_introduction

- The second sentence should be split into two, and the unnecessary word “rather” removed, giving: “Therefore, its basic structure is general. Models are primarily specified by explicitly defining mathematics using MathML.”
- [Section 7.1.2: Qualitative vs. quantitative pathway models](#)⁵²
 - The title of this section should be changed to “Qualitative and quantitative pathway models” — this isn’t a competition.
- [Section 7.4.3.2: Allowed values of the **role** attribute](#)⁵³
 - In the second sentence of the "**catalyst**" bullet point, the comma is superfluous and should be removed, giving: “In biochemical pathways such a species will almost always be an enzyme ...”
- [Section 7.4.3.3: Proper use of the **role** attribute](#)⁵⁴
 - The fourth rule and its explanation should be deleted. It was decided that a species may act as both a reactant and product within the same reaction.
- [Section 7.4.3.5: Proper use of the **direction** attribute](#)⁵⁵
 - The second rule and its explanation should be changed to the following: “The **direction** attribute on a **<role>** element for which the **role** attribute has a value of "**rate**", "**reactant**" or "**product**" must have a value of "**forward**". [Variables representing the reaction rate, a reactant or a product should always be labelled as such in the forward direction. To do otherwise would cause the implicit mathematics defined by the **delta_variable** and **stoichiometry** attributes on the reactant and product variables to be erroneous.]”
 - The third rule should be deleted. It is poorly written, and in any case completely unnecessary given the second rule.
- [Section 7.5.5: Math implied by the **delta_variable** and **stoichiometry** attributes](#)⁵⁶
 - In Figure 15, the CellML is incorrect: the **<math>** elements should be inside the **<role>** elements. The separation between the two examples is highlighted by adding comments, and removing excess whitespace.
- [Section 8.3: Examples](#)⁵⁷
 - In the last comment in Figure 16, “a” should be changed to “an”, giving: “Some metadata content, such as an annotation describing limitations of this representation of the membrane”.
- [Appendix A.2: The CellML DOCTYPE declaration](#)⁵⁸

⁵²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_qualitative_models

⁵³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

⁵⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

⁵⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

⁵⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_implied_math

⁵⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_metadata_examples

⁵⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_dtd_doctype_declaration

- The text makes it sound like all CellML documents should contain DOCTYPE declarations of the form shown. This should be changed to read: “CellML documents may reference the version of the CellML 1.0 DTD maintained on the CellML website with a DOCTYPE declarations of the following form:”
- [Appendix B.4: Preferred scripting language](#)⁵⁹
 - In the first paragraph, “when” should be changed to “if” in the second sentence, giving: “It is anticipated that, if scripting functionality is officially endorsed by a future version of CellML, ...”
- [Appendix B.6: Effects of scripts](#)⁶⁰
 - All instances of the word “should” should be replaced with the word “must”, giving: “Functions must be side-effect free. That is, a function must not assign values to variables that are not local to that function. In particular, functions must not alter the values of their arguments or global variables.”
- [Appendix C.3.2: Units-based restrictions on the use of MathML operators](#)⁶¹
 - In the bottom-right cell of Table 5, the operand is wrongly defined. The definition should be removed, and the first two sentences condensed into one, giving: “The operand of this operator and the value of the **<bvar>** qualifier element, if present, may have any units.”
- [Appendix C.3.3: Applying operators to units definitions](#)⁶²
 - In the bottom-right cell of Table 6, the terms in the quotient are in the wrong order. These should be reversed, giving: “The result of this operator has units that are the quotient of the units of the operand over the units of the term in the **<bvar>** qualifier element ...”
- [Appendix C.3.5: Conversion between units definitions](#)⁶³
 - In the second paragraph, there is a duplicate “the”, which should be removed, giving: “... then there is a one-to-one mapping between the variable’s value in both components.”

2.4 Known errors as of 29 June 2001

- [Section 2.4: Rules for CellML Documents](#)⁶⁴
 - The 18 May 2001 Final Draft did not restrict model authors from defining plain text inside CellML elements. Without proper specification, it is not clear what CellML processing software should do when it encounters this. The simplest approach is to make documents containing CellML elements that contain text nodes invalid. We add the following rule and explanation, entitled “2.4.4 Text nodes within CellML elements” with text:

⁵⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_use_of_ecmascript

⁶⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_execution_of_scripts

⁶¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_rules_restrictions_on_the_use_of_mathml_operators

⁶²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_rules_applying_operators_to_units_definitions

⁶³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_rules_conversion_between_units_definitions

⁶⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_document_rules

Any characters that occur immediately within elements in the CellML namespace must be either space (`#x20`) characters, carriage returns (`#xA`), line feeds (`#xD`), or tabs (`#x9`).

[All of the elements in the CellML 1.0 namespace contain no text content. The characters listed above correspond to the definition of whitespace given in [Section 2.3 of the XML 1.0 Recommendation](#)⁶⁵. Text content may still be included in extension elements inside CellML elements.]

- [Section 4.4.1: The `<mathml:math>` element](#)⁶⁶

- The first rule in this section states that “The `<mathml:math>` element must only appear as a child of `<cellml:component>` or `<cellml:role>` elements.” This is a duplication of the content model definitions for `<cellml:component>` and `<cellml:role>`, and does not allow the `<mathml:math>` element to appear within extension elements. This rule should be amended to read as follows: “The `<mathml:math>` element must only appear as a child of the following elements in the CellML namespace: `<cellml:component>` and `<cellml:role>`.” The following sentence should be added to the explanation to clarify this: “The `<mathml:math>` element may appear inside elements in the RDF and CellML Metadata namespaces if permitted by the relevant specifications, and may be used inside extension elements. When MathML elements occur within extension elements, CellML processing software is free to ignore them.”
- The sentence regarding the CellML subset of MathML elements in the explanation after the second bullet point should be removed and made its own bullet point. This lends the MathML subset a bit more importance, and the use of “For interoperability” and “should” ensures that the rule should really be regarded more as a recommendation. The full text of this bullet should read “For interoperability, all elements in the MathML namespace that are within a `<mathml:math>` element, and not within a `<mathml:annotation>` or `<mathml:annotation-xml>` element should be taken from the CellML subset of MathML content markup elements defined in Figure 5. [The CellML subset is discussed further in Section 4.2.3. Note that this is an interoperability recommendation and not a firm rule.]”

- [Section 4.4.3.2: Allowed values of the `cellml:units` attribute](#)⁶⁷

- The CellML namespace prefix should be added to the three CellML elements mentioned in the bullet point, giving: “The value of the `cellml:units` attribute must be taken from the standard dictionary of units given in Section 5.2.1, or be the value of the `name` attribute on a `<cellml:units>` element defined in the current `<cellml:component>` or `<cellml:model>` element.”

- [Section 2.4.2: The `<relationship_ref>` element](#)⁶⁸

- The specification does not currently prevent the model author from referencing the same relationship type more than once within the same group element. Rather than specify software behaviour when encountering this, it should just be outlawed. So the following text should be added under the section heading “6.4.2.5 Proper use of the `relationship` and `name` attributes”:

⁶⁵<http://www.w3.org/TR/2000/REC-xml-20001006#sec-common-syn>

⁶⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_spec_rules_mathml:math_element

⁶⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_spec_rules_mathml:cn_element

⁶⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_grouping_spec_rules_relationship_ref_element

[The following rules together prevent the model author from referencing the same hierarchy more than once within a given `<group>` element.]

- * A `<group>` element must not contain two or more `<relationship_ref>` elements that define a `relationship` attribute in a common namespace with the same value and that have the same `name` attribute value.
- * A `<group>` element must not contain two or more `<relationship_ref>` elements that define a `relationship` attribute in a common namespace with the same value and do not define `name` attributes.

- [Section 7.4.3.5: Proper use of the `direction` attribute](#)⁶⁹

- There is no reason why modellers shouldn't be able to define `direction` attributes with a value of "forward" (the default value) on `<role>` elements with a `role` attribute value of "rate". The third bullet point should read: "". The second rule in Section 7.4.3.5 should then be changed to read: "The `direction` attribute on a `<role>` element for which the `role` attribute has a value of "rate", "reactant" or "product" must have a value of "forward".".
- There is no reason why modellers shouldn't be able to define `direction` attributes with a value of "forward" (the default value) inside reactions which are not reversible. The first rule in this section should thus be changed to read: "A `direction` attribute must only have a value of "reverse" or "both" if defined on a `<role>` element contained in a `<reaction>` element on which the `reversible` attribute has a value of "yes"."

2.5 Known errors as of 22 June 2001

- [Section 2.4.3: Extension namespaces](#)⁷⁰

- The word "set" should be deleted, giving: "For interoperability, elements in the CellML namespace should not be defined inside extension elements."
- A third rule should be added to this section to make it clear that CellML attributes may (but should not) be defined on extension elements. The text should be: "For interoperability, attributes in the CellML namespace should not be defined on extension elements."

2.6 Known errors as of 20 June 2001

- [Section 1.1.1: Purpose and scope of CellML](#)⁷¹

- In paragraph two, sentence two, it is not clear what "the links between models, operating systems and programming languages" really are. This should be rewritten as: "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."
- In paragraph two, sentence three, the first clause ends with a preposition. This clause should be changed to read: "CellML also allows the generation of equations for publishing from the same definition upon which the solution method is based, [...]"

⁶⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_role_element

⁷⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_rules_namespaces

⁷¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_intro_purpose_and_scope

- [Section 1.1.3: Terminology](#)⁷²
 - In paragraph one, “various useful model classifications” should be “two useful model classifications”.
 - In the definition of error, “It is recommended that software make information about errors [...]” should be changed to “The recommended best practice is for software to make information about errors [...]”.
 - “CellML compliant software” should be “CellML conformant software”. The use of both terms was considered confusing.
- [Section 1.2: Structure of the CellML Specification](#)⁷³
 - The description of Section 7 (Reactions) should read: “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 2.2.2: Namespaces in CellML](#)⁷⁴
 - In paragraph one, in sentence two, a comma should be added after the word “allowing” to make the sentence read better, giving: “XML namespaces add a second level of naming to elements and attributes, allowing processing software to distinguish between elements and attributes from different languages.”
 - In paragraph one, the fourth sentence should be split into two shorter sentences for clarity, giving: “The value of a namespace URI need have nothing to do with the XML document that uses it. However, it typically points to a document that defines the rules for the language.”
 - In paragraph four, an extra “the” should be removed: “For interoperability, the root element of any CellML document should set the default namespace and map the `cellml` prefix ...”
- [Section 2.2.3: Extending CellML documents](#)⁷⁵
 - In paragraph one, the third sentence should read: “Any attribute in an extension namespace is an *extension attribute*.”
- [Section 2.4.1: Valid CellML identifiers](#)⁷⁶
 - The first sentence of the explanation should become the rule itself, as it is simple and in plain English. The EBNF definition should be retained but demoted to secondary status. In the EBNF fragment, name (which was borrowed from SBML) should be replaced with `identifier`. The section should thus have the following content:

A valid CellML identifier must consist of only letters, digits and underscores, and must contain at least one letter or digit. This can be written using Extended Backus-Naur Form (EBNF) notation as follows:

⁷²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_intro_terminology

⁷³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_intro_spec_structure

⁷⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_namespaces

⁷⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_extending_cellml

⁷⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_rules_identifiers

```

letter      ::= 'a' ... 'z' , 'A' ... 'Z'
digit      ::= '0' ... '9'
identifier ::= ('_')* ( letter | digit ) ( letter | '_' | digit )*

```

[The variant of EBNF used above is defined in [Section 6 of the XML 1.0 Recommendation](#)⁷⁷.]

- [Section 2.4.3: Extension namespaces](#)⁷⁸
 - In the explanation of the second bullet point, “that” should be changed to “which”, giving: “Specifically, applications should not define [...] information within extension elements, which other applications are free to ignore.”
- [Section 2.5.3: Treatment of extension namespaces](#)⁷⁹
 - The second bullet point should be changed to read: “CellML processing software may ignore the attributes and content of extension elements.” The XML specification uses the phrase “element content” rather than “element contents”.
- [Section 3.2.1: Definition of a model](#)⁸⁰
 - In paragraph one, “recommended method” should be changed to “recommended best practice”.
 - In paragraph three, sentence two, “recommended practice” should be changed to “recommended best practice”.
 - In the bulleted list, it is stated that “each component contains variables [...] and mathematics [...]”, suggesting that this is required. This should be changed to “each component may contain variables [...] and/or mathematics [...]”.
- [Section 3.2.2: Definition of components](#)⁸¹
 - In paragraph four, sentence two, “recommended practice” should be changed to “recommended best practice”.
- [Section 3.2.4: Definition of connections](#)⁸²
 - In the last sentence of paragraph two, the word “following” is used twice. The first instance should be changed to “tracing”.
- [Section 3.3: Examples](#)⁸³
 - In the second sentence of the last paragraph, the “which” should be changed to “since this”, giving: “The value of the **variable_1** attribute on each **<map_variables>** element references the corresponding variable in the **membrane** component, since this is the component referenced by the **component_1** attribute on the **<map_components>** element.”

⁷⁷<http://www.w3.org/TR/2000/REC-xml-20001006#sec-notation>

⁷⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_rules_namespaces

⁷⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_fundamentals_processor_rules_extension_namespaces

⁸⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_model_element

⁸¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_component_element

⁸²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_connection_element

⁸³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_examples

- [Section 3.4: Rules for CellML Documents](#)⁸⁴
 - An “and” and an “elements” should be deleted from the single sentence in this section. The sentence should read: “The following are the rules for using the `<model>`, `<component>`, `<variable>`, `<connection>`, `<map_components>`, and `<map_variables>` elements.”
- [Section 3.4.2.1: Allowed use of the `<component>` element](#)⁸⁵
 - A `<component>` element may contain `<reaction>` elements in the CellML namespace. The first sub-bullet under the first bullet should have the text: “`<units>`, `<variable>` and `<reaction>` elements in the CellML namespace.”
- [Section 4.1: Introduction](#)⁸⁶
 - In the first sentence of paragraph one, “is aimed at” should be replaced with “allows modellers to”, giving: “CellML allows modellers to unambiguously specifying the underlying mathematics of a cellular model.”
- [Section 4.2.1: Definition of mathematics](#)⁸⁷
 - In paragraph one, the last sentence should include “`<cellml:reaction>`” in the list of elements introduced in Section 7. Also “first introduced” should be changed to “described in detail” — technically, those elements are first introduced in that sentence. The result is: “The `<cellml:role>`, `<cellml:variable_ref>` and `<cellml:reaction>` elements mentioned in this section are described in detail in Section 7 of this specification.”
 - In paragraph two, in the first sentence, “in terms of” should be “relating”, giving: “[...] arbitrary expressions relating the variables [...]”.
- [Section 4.2.2: MathML’s presentation and content markup elements](#)⁸⁸
 - In paragraph one, sentence two, “are oriented towards describing” should be changed to “describe”, giving: “The presentation markup elements describe the visual rendering of [...]”.
 - In paragraph one, sentence three, “aim to capture” should be changed to “specify”, giving: “The content markup elements specify the underlying meaning of [...]”.
- [Section 4.2.3: The CellML subset of MathML content elements](#)⁸⁹
 - In paragraph one, sentence one, “follow” should be “follows”. That sentence should read: “*Valid CellML documents* may contain any MathML content markup elements within a `<mathml:math>` element, as long as the arrangement of these elements follows the rules defined in the [MathML 2.0 Recommendation](#)⁹⁰.”
 - In paragraph one, in the last sentence, “compliant” should be “conformant”. The last sentence should read: “CellML processing software may only call itself *CellML conformant* if it is able to correctly interpret all of the MathML elements in the CellML subset according to the semantics defined in the MathML 2.0 Recommendation.”

⁸⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_spec_rules

⁸⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_structure_spec_rules_component_element

⁸⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_introduction

⁸⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_bs_definition_of_mathematics

⁸⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_presentation_vs_content

⁸⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_cellml_subset

⁹⁰<http://www.w3.org/TR/2001/REC-MathML2-20010221>

- [Section 4.2.5: Definition of scripts](#)⁹¹

- In paragraph one, sentence one, “any means with” should be changed to “a standard method by”, giving: “CellML 1.0 does not define a standard method by which model authors can embed scripts in CellML documents in a portable way.”.
- In paragraph one, sentence three, the sentence should start with “However” as it’s a change of direction from the previous sentence, giving: “However, the use of scripts in CellML is strongly discouraged.”.
- In the final bullet point, it is not clear if the second sentence is an explanation of “side-effect free” or some additional requirement. That bullet point should be changed to read as follows: “Functions should be side-effect free. That is, a function should not assign values to variables that are not local to that function. In particular, functions should not alter the values of their arguments or global variables.”

- [Section 4.4.1: Allowed use of the `<mathml:math>` element](#)⁹²

- In the final bullet point, “contents” should be changed to “content”, giving: “The content of a `<mathml:math>` element must conform to [...]”. (The XML specification uses content in the singular.)

- [Section 4.5.1: Ordering of expressions](#)⁹³

- It is currently not clear what the behaviour of processing software should be with regard to the ordering of equations, a topic currently only dealt with in Section 4.5.1. The following sentence should be added: “It is recommended that CellML processing software evaluate expressions in an order which minimises the number of unknown variables in each expression.” A new introductory section should be added after Section 4.2.3 entitled **Ordering of expressions** with the following content:

The mathematics in a model defined using CellML 1.0 consist of a static system of expressions, which are distributed over a network of components. CellML provides no facilities for specifying the order in which these expressions should be evaluated, as this is simulation information rather than model information. CellML processing software must not assume that the ordering of expressions within a component or within a CellML document has any significance, and may evaluate equations in any order when running simulations. For interoperability, it is recommended that CellML processing software evaluate expressions in an order which minimises the number of unknown variables in each expression.

- [Section 4.5.2: Treatment of annotations](#)⁹⁴

- In both the first and second sentences, the word “contents” should be replaced with “content”. This means that, in the second sentence, “define” should be changed to “defines”. The total text is thus: “CellML processing software may ignore the content of `<mathml:annotation>` and `<mathml:annotation-xml>` elements. CellML processing software must assume that the content of the first child of a `<mathml:semantics>` element defines an expression describing the mathematical behaviour of the model.”.

⁹¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_scripting

⁹²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_sec_math_rules_mathml:math_element

⁹³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_rules_order_of_calculations

⁹⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_rules_treatment_of_annotations

- [Section 5.2.2: User defined units](#)⁹⁵
 - In paragraph four, the word “can” should be replaced with “may” giving: A **<units>** element may contain a set of **<unit>** elements [...]
 - In paragraph five, “has no content” should be replaced with “must not contain any elements in the CellML namespace,” giving: “A **<unit>** element may not contain any elements in the CellML namespace, but may have up to five attributes.”
- [Section 5.4.2.2: Allowed values of the **units** attribute](#)⁹⁶
 - In the explanation after the second bullet point, “the base SI units” should be changed to read “SI and user-defined base units”.
- [Section 7.2: Basic Structure](#)⁹⁷
 - In paragraph one, sentence five should be “However, this makes it difficult to re-use the individual reactions, and is therefore not the recommended best practice.”
 - In paragraph two, sentence two, “recommended practice” should be “recommended best practice”.
 - In paragraph three, the word “empty” should be removed from the first sentence. The second sentence should be changed to read: “A **<role>** element must not contain any elements in the CellML namespace, but may have up to four attributes.”
 - In paragraph four, sentence two, “recommended practice” should be “recommended best practice”.
- [Section 7.4.1.1: Allowed use of the **<reaction>** element](#)⁹⁸
 - In the explanation on bullet point two, “recommended practice” should be “recommended best practice”.
- [Section 7.5.5: Math implied by the **delta_variable** and **stoichiometry** attributes](#)⁹⁹
 - In paragraph two, “recommended practice” should be “recommended best practice”.
 - In paragraph four, “recommended practice” should be “recommended best practice”.
 - In the caption of Figure 15, “recommended method” should be “recommended best practice”.
- [Section 7.5.7: Resolution of inconsistencies](#)¹⁰⁰
 - In paragraph three, “CellML-compliant” should be “CellML conformant”.
- [Section 8.2: Basic Structure](#)¹⁰¹
 - In paragraph four, “recommended practice” should be “recommended best practice”.

⁹⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#

⁹⁶http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_units_spec_rules_unit_element

⁹⁷http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_basic_structure

⁹⁸http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_spec_rules_reaction_element

⁹⁹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_implied_math

¹⁰⁰http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_rxn_inconsistencies

¹⁰¹http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_metadata_basic_structure

- [Section 8.4.2.1: Allowed use of the <rdf:RDF> element](#)¹⁰²
 - In the explanation after the first bullet point, “recommended practice” should be “recommended best practice”.
- [Appendix A.1: Introduction](#)¹⁰³
 - In paragraph one, “CellML compliant” should be “CellML conformant”.
- [Appendix B.6: Effects of scripts](#)¹⁰⁴
 - It is not clear if the second sentence is an explanation of “side-effect free” or some additional requirement. The complete text should be changed to read as follows: “Functions should be side-effect free. That is, a function should not assign values to variables that are not local to that function. In particular, functions should not alter the values of their arguments or global variables.”
- [Appendix D.1: Changes between 2 March 2001 Draft and the 18 May 2001 Final Draft](#)¹⁰⁵
 - A fullstop should be added to the end of the **Namespaces** paragraph.
 - Whitespace should be added after the third link in the **Mathematics** paragraph.
 - Whitespace should be added after the third link in the **Grouping** paragraph.
 - A spelling mistake should be corrected (amd, and) in **Conformance Levels** and **Terminology** paragraph.
 - In the **Conformance Levels** and **Terminology** paragraph, “CellML compliant” should be “CellML conformant”.

E-mail questions, criticism, submissions or info to info@cellml.org
Input document last modified : Tue Nov 27 15:25:15 GMT+13:00 2001

¹⁰²http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_metadata_spec_rules_rdf:RDF_element

¹⁰³http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_dtd_introduction

¹⁰⁴http://www.cellml.org/public/specification/20010518/cellml_specification.html#sec_math_execution_of_scripts

¹⁰⁵http://www.cellml.org/public/specification/20010518/cellml_specification.html#20010302_changes