

Menu of Proposals for SBML Level 2 Version 2 or Level 3

Andrew Finney

University of Hertfordshire, UK
Molecular Sciences Institute, USA



This presentation follows
proposal document emailed to
sbml-discuss mailing list recently

Overview

- Introduction
- Use of `Id` Attribute on `SimpleSpeciesReference`
- Dependant Variable attribute on Algebraic Rules
- Nested Unit Definitions
- `SpeciesType`
- Diagram Layout
- Controlled Vocabularies
- Assertions
- Variable and Flux Bounds
- Conclusion

Introduction

- A menu of new features
 - Each feature is independent of the others
 - A given feature could be incorporated into Level 2 Version 2 or Level 3 or not at all
 - None of the features forms a large module

`id` on `SimpleSpeciesReference`

- Create new `id` attribute on `SimpleSpeciesReference`
 - Has type `sid`
- Required by diagram layout proposal
- Value is unique amongst all `id` values declared at the global level
- Optional `name` attribute of type `string` as well
- In addition to existing `species` attribute that refers to a species

Dependant Variables

- It can be important for reproducing a given numerical solution to specify explicitly the dependent variable on an algebraic rule.
- Given more than one algebraic rule it is often ambiguous which variable is the dependent variable of a rule.
- Application can decide which variable is the dependent variable of each rule
- Different decisions can unfortunately give different results.
- New optional `sid` field `dependentVariable` is added to `AlgebraicRule`
 - Makes explicit the variable whose value should be calculated using the rule.
 - Contains a variable symbol declared by a `Species`, `Compartment` or `Parameter` structure.
 - The field cannot refer to a constant symbol.
 - The use of this field should be avoided in best practice.

Nested Unit Definitions

- In Level 2 Version 1 all unit definitions have to be composed from fundamental units
 - For example
 - You can define minutes as 60 seconds
 - But not hours as 60 minutes
- Fix this by allowing the `kind` field of `Unit` structures be able to refer to other unit definitions as well as the `unitkind` enumeration

SpeciesType

- Class proposed in multicomponent species proposal
- In this proposal the `model` class is extended to have a list of species types
- `SpeciesType` has only
 - one mandatory attribute `id`
 - one optional attribute `name`
- `Species` for each compartment must explicitly be declared and can optionally refer to the `SpeciesType`
 - Relaxed later

Diagram Layout

- Should we include the diagram proposal into the new version of SBML?
- As a module?
- Should the module exist in its own namespace?

Controlled Vocabularies: Requirements

- Often features are requested where a single attribute can contain terms from a controlled vocabulary (CV)
 - examples:
 - role on `SimpleSpeciesReference`
 - type on `KineticLaw`
 - type on `AlgebraicRule`

Controlled Vocabularies Requirements

Aspects of Typical Controlled Vocabularies

- Term assignment doesn't affect the mathematical interpretation of the model
- The CVs are typically application domain specific
- The CVs appear to be incomplete or obviously extensible and thus likely to change more rapidly than the core SBML standard.
- The CVs are not complex ontologies in fact they are not likely to be more complex than terms linked to form a directed acyclic graph (DAG) and will often consist of a list of alternative terms.
- The CVs are designed for use within an arbitrarily defined group for example in the whole community, the BioSpice consortium or with in a single laboratory.

Controlled Vocabularies

Required Features

- Allow the CVs to be developed independently of SBML with a clear boundary between the CVs and core SBML
- Allow the relationship between an SBML object and CV terms to be defined in a generic fashion
 - When referring to a Gene Ontology (GO) process term from a reaction we state that the reaction is *'part of'* the referenced process
 - In another context we may say the GO process is *'inhibited'* by a species
 - CellML bioentity metadata does not allow for this
 - implied 'is a' relationship
- Allow users in model capture tools to select terms from a CV definition and associate them with a selected SBML object.
 - The CV definition is located at a URL and the tool is not aware in advance the CV's existence.
 - The CV format is sufficiently simple to allow for the straightforward implementation of this use case.
- Allow the GO CV to be used seamlessly with SBML
- In the future allow us to use more than one CV format

Controlled Vocabularies

Alternative Implementation Strategies

- Pure 'Custom' XML scheme
 - Akaza Research whilst consulting for Science's STKE
 - Sachs' presentation at 8th Forum, St Louis
 - Controlled Vocabulary format not extensible
 - Can't refer directly to GO
- RDF based approaches
 - OWL
 - Slots
 - Difficult for simple implementations to browse
 - Restricted to form without slots
 - BioPAX
 - Implemented in OWL
 - Limited depth(?) and scope with respect to modeling
 - Hasn't solved controlled vocabulary issue either
 - GO RDF/XML
 - Limited tool support
 - Not primary format for GO itself
 - Simple
 - DAG only
 - No Slots

W3C RDF

- Semantic Web Technology
- Allows the incremental distributed construction of information
- Consists of statements of the form
 - Subject Verb Object
 - Verb often called Predicate
 - Can form any kind of graph
- **All three parts are uniquely identified**
 - Subject and Object are **URIs**
 - They don't have to physically exist
 - Try going to a GO term URI
 - Getting a list of relevant subjects and objects is not addressed by RDF
 - Predicate Element tag
 - Uniquely defined by element name and namespace
 - Not nice from perspective of schema validation (?)

Controlled Vocabularies Implementation Overview

- Understanding of semantic web technology emerging
- To represent ontology
 - Use Gene Ontology (GO) XML RDF form or OWL without slots?
 - In SBML would this be best practice or standardized?
- Have optional list of URLs that refer to control vocabularies at SBML element level
 - In best practice should physically exist
- Work around RDF predicate namespace issue

Example of SBML element including CVs

```
<sbml
  level="2" version="2"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.sbml.org/sbml/level2">

  <listOfControlledVocabularies>
    <controlledVocabulary
      xlink:type="simple"
      xlink:href="http://www.gsk.com/SBML-
resources/speciesCV.xml"/>
  </listOfControlledVocabularies>

  ...
</sbml>
```

sBase Extension

- List of RDF statements about object
- Subject is the sBase object
- Whilst following RDF standard follows non-typical form
 - The verb or predicate is a resource and **not** in an alternative namespace
 - Not best practice
 - Overcomes namespace issues

Example of Species with Associated CV Terms

```
<species metaid="x11" id="S" compartment="C">
  <statements>
    <rdf:RDF xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-
ns#">
      <rdf:Description rdf:about="#x11">
        <statement>
          <rdf:Description>
            <term
rdf:resource="http://biospice.org/BIOSPICE:0003673"/>
            <predicate
rdf:resource="http://biospice.org/BIOSPICE:0005454"/>
            </rdf:Description>
          </statement>
        </rdf:Description>
      <rdf:Description rdf:about="#x11">
        <statement>
          <rdf:Description>
            <term
rdf:resource="http://www.gsk.com/GSK:0000010"/>
            <predicate
rdf:resource="http://www.gsk.com/GSK:0000001"/>
            </rdf:Description>
          </statement>
        </rdf:Description>
      </rdf:RDF>
    </statements>
  </species>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE go:go>
<go:go
  xmlns:go="http://www.geneontology.org/dtds/go.dtd#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:RDF>
    <go:term rdf:about="http://biospice.org/BIOSPICE:0003673">
      <go:accession>BIOSPICE:0003673</go:accession>
      <go:name>Protein</go:name>
      <go:definition>a polymer of amino acids</go:definition>
      <go:isa
        rdf:resource="http://biospice.org/BIOSPICE:0003674" />
    </go:term>
    <go:term rdf:about="http://biospice.org/BIOSPICE:0003674">
      <go:accession>BIOSPICE:0003674</go:accession>
      <go:name>Polymer</go:name>
      <go:definition>
        a compound made up of a linked series of repeated monomers
      </go:definition>
    </go:term>
    <go:term rdf:about="http://biospice.org/BIOSPICE:0005454">
      <go:accession>BIOSPICE:0003674</go:accession>
      <go:name>isa</go:name>
      <go:definition>
        the subject is a subtype of the object
      </go:definition>
    </go:term>
  </rdf:RDF>
</go:go>

```

Example of CV in GO XML/RDF

Assertions - Concept

- **Concept**
 - Define constraints that enable the detection of internal inconsistencies in a model and/or external perturbations of variables and parameters which render a model invalid.
 - Simply math expressions that are either true or false given some subset of variables and constant defined in the model.
 - Assertions are not structured to facilitate particular types of analysis in fact they are not meant to form the core of any given analysis but instead identify error conditions.
- **Requirement**
 - Assertions must be clearly separated from other structures that are used directly in analysis.
 - To facilitate this type of use of assertions a error string should be optionally associated with the assertion.
- **Example**
 - Define quantitatively the assumption in a rate law that the product concentration is much lower than that of the enzyme
 - If the product concentration becomes large enough to render the rate law invalid during a simulation the simulator can notify the user

Assertion - Example

```
<listOfRules>
  <assertion>
    <math xmlns="http://www.w3.org/1998/MathMathML">
      <apply>
        <lt/>
        <ci> P </ci>
        <cn> 1 </cn>
      </apply>
    </math>
    <listOfMessages>
      <message lang="en">
        The product concentration, P, is too high for
        the model assumptions to hold.
      </message>
    </listOfMessages>
  </assertion>
  ...
</listOfRules>
```

Flux and Variable Bounds

- Its clear that there are non-spatial analyzes of biochemical reaction networks which can be driven by mathematical structures that are not yet part of the SBML standard.
- These analyzes can use the reaction network contained in a SBML model as a starting point yet they require additional information as input.
- Some analyzes of biochemical reaction networks operate on bounds of various properties of a network rather than absolute values or expressions.
- Bounds need to be placed on initial values, variables, parameters and reaction rates.
- See UML diagram in document
- For flux we allow reaction ids to be a symbol
 - In MathML as well