Ontologies and Repositories

Table of Contents

Ontologies and Repositories

Contents

Introduction

The Physiome project is about *organising, disseminating, and collaborating* research.

We tend to:

> *organise* by stuffing things in databases or shared filesystems
>
> *disseminate* by physical publication or stuffing it into public databases
>
> *collaborate* by email

We know this isn't what people are saying they should do anymore.

Enter the *Semantic Web*

> A connected system of resources that have machine interpretable meaning and machine level interfaces for processing.

When we talk about this we talk about *standards, repositories, and ontologies*

Repositories

1. The picture on the box
2. The puzzle as it is now

This is kind of backwards because it's only through getting to the latter that we have any real idea of the picture.

A brief view of goals and interfaces. A bit outdated.

The picture on the box: Physiome workspace

Objectives

Adding workflow, shared resources, and visibility to daily work-cycle

Where is all the work?

Facets:

Collections of data - declarative models, numerical data sets, publications

Levels of access: permissions and roles.

Changes over time: new versions and variants.

Workflow: workspace may involve more than one person, different roles may be relevant - e.g. author, reviewer, tester.

Application of semantics: consistent and replete

Collaboration

More than one person working on the same model or set of models

Facets:

Separation of concerns: e.g. encapsulation and importing in CellML

Changes over time: versions or variants

Signaling changes

Tracking issues

Software integration: exchange of models between systems

Intention and extension - evolving the semantics of the model(s).

Publication

Composite Models

Combining models or reusing parts of models

Facets:

Fitting sub-models (other models or parts of models) together

Locate sub-models in a library

Validate new construction

Find relevant best practices

Tracking sub-models over time: request stable or volatile versions.

Feedback to authors of submodels

Curation

A library of models

Facets:

Adding new models

Determining the relationship to other models (versions, variants, biological, mathematical, authorship ... etc)

Establishing provence

Validating - biological, physical, numerical.

Publication

Related Data

Models are not built in isolation

Built from other models

Are improved versions of older ones

Are new variants of other ones

Produce simulation data in particular environments

Have graphs or illustrations demonstrating its results

Has an associated physical and biological model

Solutions

Validation

pass the tests

Facets:

MIRIAM

(and it should be only one word!)

- but it's not a standard
- it isn't mandated by publishers
- it needs to be reviewed by us who will try to use it
- I don't agree with the annotation specification - it should be oriented towards what tests it would pass
- How do Andre's thoughts gel with this?
- perhaps we should work through more detail on this in Edmund's session

Software Interfaces

Adding CellML to your software

Facets:

Exchange CellML models

Create and Modify models

Submit to a repository

Receive events from repository

Search/Query repository

Search/Query model metadata

Share model between running applications

Conversion

Exchanging models with other language environments

Facets:

Lossless on round-trip

Read and Write - what happens to metadata?

What is there:

- ♦ CellML2SBML XSLT - only works on reaction based models
    - ◊ Biopax annotation of CellML models makes conversion of non reaction based models to SBML feasible
- ♦ XSLT for RDF form of CellML
- ♦ what about MML and VCMDL and SimBio?

Versions and Variants

Mathematical models are like programming code - they need versioning

Facets:

Track latest version or set to particular version

A version remains unchanged for all time

A variant is a new model closely related to an existing model.

implementation:

- ◊ Version and variant identifiers are part of the URI and part of the model metadata
- ◊ Refactoring to follow subversion rules and webdav REST rules for URI formation (see http://en.wikipedia.org/wiki/REST)

Privacy

Repositories need security

Facets:

Role based permissions integrated with workflow

Public, private, shared group spaces.

Anonymous, Author, Reviewer, Administrator

Current implementation:

- ♦ Use Plone CMS system - contains these features by default, simple to customise. But still need to expose these needs at the software level and capture the state in the metadata of models.
- ♦ Need to formalise some workflows for curation of models in the repository.

Repository Services

API interface

Features:

Search and Query across models

Download and upload

Inspect changes in repository

Generate C-Code

Validate

Run simulation

Receive events about changes in the repository

The puzzle as it is now : cellml.org/models

CellML is one area of the physiome workspace

Important to maintain the focus community it has

The repository itself - link

Where's tommy?

- can't upload cellml 1.1 models
- extend metadata editing facility
- fix speed problems
- user interface improvements
- validate metadata that currently exists - RDF syntax check and against schemas
- merge documentation xml into the rdf metadata

He wants:

- new repository that supports proper versioning and branches (variants).
- whatever that ontology thing is that Matt keeps talking about

TingTing

Amongst the technologies there are:

- databases
- Web/API services for software developers

TingTing will be scoping, researching, and implementing these

This means reviewing current technologies and finding active users. Please volunteer to be a software level user.

Ontologies in CellML

Ontologies are a specification of two main things:

1. How to represent knowledge
2. The knowledge itself

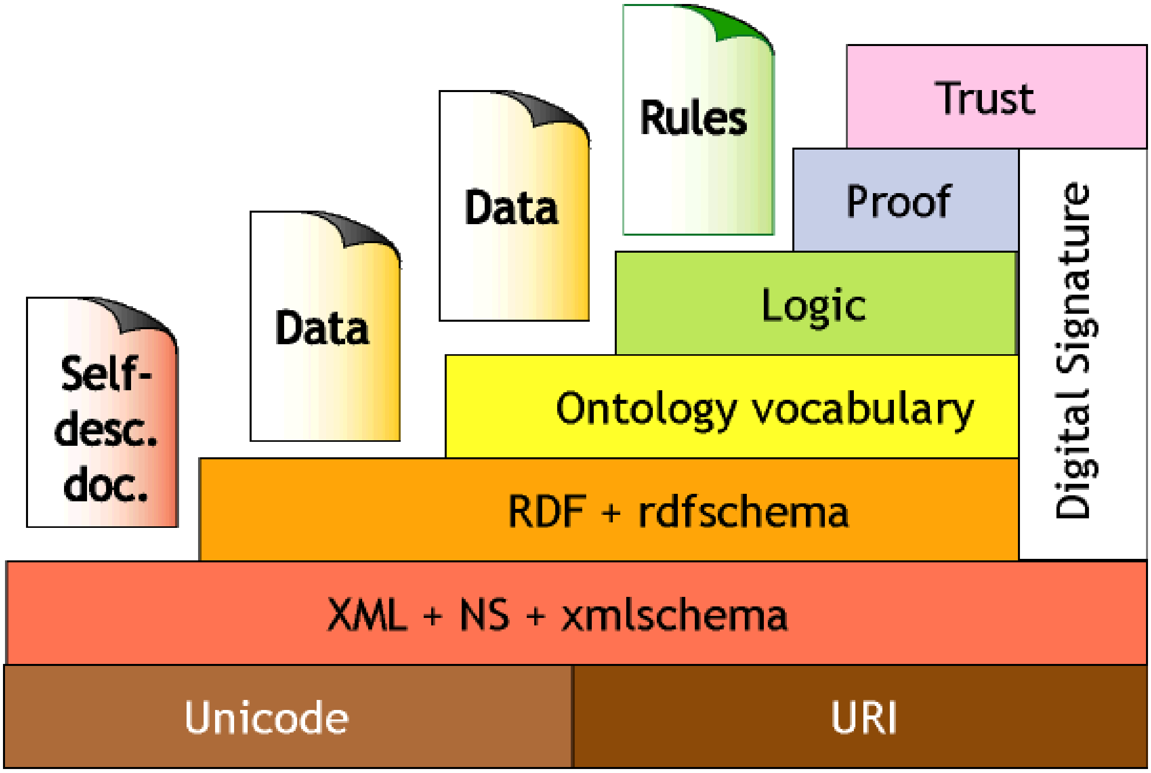
Put another way:

> **Giving your data meaning outside of the data itself**

We have chosen RDF and OWL

Why?

- use existing standards
- the semantic web layer cake:



The advantages of using RDF:

- It provides a common attribute=value data model for the metadata
- It provides an extensible method for storing metadata of increasing complexity
- It makes it possible for applications that don't know anything about CellML to understand our metadata
- There are tools out there that use RDF

How does it work?

- RDF forms a graph, not a tree.

Curation data

The nice view:

```
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

<rdf:Description rdf:about="#cellml_element_id">
  <dc:creator rdf:parseType="Resource">
    <vCard:N rdf:parseType="Resource">
      <vCard:Family>Flintstone</vCard:Family>
      <vCard:Given>Fred</vCard:Given>
    </vCard:N>
  </dc:creator>
  <dc:creator rdf:parseType="Resource">
    <vCard:N rdf:parseType="Resource">
      <vCard:Family>Brown</vCard:Family>
      <vCard:Given>Charlie</vCard:Given>
    </vCard:N>
  </dc:creator>
  <dc:creator rdf:parseType="Resource">
    <vCard:N rdf:parseType="Resource">
      <vCard:Family>Doo</vCard:Family>
      <vCard:Given>Scooby</vCard:Given>
    </vCard:N>
  </dc:creator>
</rdf:Description>
</rdf:RDF>
```

That's only creator. Other recommended metadata provided in specification

The list is:

Model Creator

stores information about the person or persons who coded the model into CellML. (Most models)

Contributor

a person contributed to a resource but did not actually create it, such as an editor. (Some models)

Publisher

the person or organisation responsible for providing the model, model component, or other CellML element. A given CellML element can have multiple publishers. (Most models)

Copyright

refers to the copyright that protects the CellML document, model, model component, or other CellML element (No models)

Creation Date

the date upon which the model or model part was coded into CellML. A given CellML element can have only one creation date. (All Models)

Modification History

lists changes that have been made to the CellML document (Some models)

Alternative Names

provides human-readable names for CellML elements (Some models)

Species

refers to the biological species (such as human, dog, pig, Palaemon affinis, etc.) for which an element is relevant. (Some models)

Sex

refers to the sex for which a CellML element is relevant. A given element may be relevant for more than one sex. (No models)

Biological Entity

for now it is simply a name or database unique identifier for a biological entity, such as an ion channel, signalling pathway, or specific cell type, that is represented by the model or model component. (Some models)

Mathematical Problem Type

a classification of the type of problem encoded in the math associated with the model or model component. It is specified using NIST's GAMS classification tree. Refer specifically the problem decision tree. (No models)

Description

a short description of the referenced resource. (No models)

Annotations

◊ comment : free-form comment by the person who coded the model into CellML. (Most models)

◊ limitation : brief description of the limitations/scope of the content of the CellML element. (No models)

◊ validation : description of the level of validation of the content of the CellML element. This may be a code. Note that validation codes are unlikely to be interoperable. (No models)

Citations

citing sources of publications relevant to the CellML element. (All Models have some subset of citation data)

◊ covers many types of publications (journal, book, web resource, patent)

◊ allows citations by unique identifier from a database, e.g.

```
<rdf:RDF
  xmlns:bqs="http://www.cellml.org/bqs/1.0#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:about="#cellml_element_id">
    <bqs:reference rdf:parseType="Resource">
       <bqs:Medline_id>97219925</bqs:Medline_id>
       <dcterms:abstract rdf:parseType="Resource">
          <dcterms:IMT>text/url</dcterms:IMT>
          <!--
            Note that the URI below, would not normally be split over two lines.
            It has been split so that it fits on a page
          -->
          <rdf:value>
            http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?
              cmd=Retrieve&db=PubMed&list_uids=9067300&dopt=Abstract
          </rdf:value>
       </dcterms:abstract>
    </bqs:reference>
  </rdf:Description>
 </rdf:RDF>
```

◊ can also provide our own summaries, keywords, subheadings etc for a citation

◊ spatial and temporal scope
◊ provide information about citations that have changed

A typical view in a model:

```
<rdf:RDF>
  <rdf:Seq rdf:ID="http://4suite.org/rdf/anonymous/0442f6e1-7a00-4451-8b8c-7bf05d38f66d">
    <rdf:li resource="http://4suite.org/rdf/anonymous/bd8aa1c3-47a3-4604-9d9e-f467c38c7389" />
  </rdf:Seq>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/25d1a305-fb14-44e9-ab27-f84d18f76fd
    <dcterms:W3CDTF>2005</dcterms:W3CDTF>
  </rdf:Description>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/bd8aa1c3-47a3-4604-9d9e-f467c38c738
    <rdf:type rdf:resource="http://www.cellml.org/bqs/1.0#Person" />
    <vCard:N rdf:resource="http://4suite.org/rdf/anonymous/85673b93-4f48-4d1e-97f8-965b7c7e85b3"
  </rdf:Description>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/85673b93-4f48-4d1e-97f8-965b7c7e85b
    <vCard:Given>Daniel</vCard:Given>
    <vCard:Family>Beard</vCard:Family>
    <vCard:Organisation>Biotechnology and Bioengineering Center, Department of Physiology, Medica
  </rdf:Description>
  <rdf:Description rdf:about="http://www.cellml.org/models/beard_2005_version01#beard_2005_versio
    <dc:publisher>
        The University of Auckland, Bioengineering Research Group
    </dc:publisher>
    <dc:title>
         A Biophysical Model of the Mitochondrial Respiratory System and Oxidative Phosphorylatio
    </dc:title>
    <bqs:reference>http://4suite.org/rdf/anonymous/56792768-20d5-4353-9cce-99413fab6eac</bqs:refe
    <dc:creator>http://4suite.org/rdf/anonymous/4c35bbd0-34c0-4370-bace-5b3d88cc0aee</dc:creator>
  </rdf:Description>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/df26ccd4-0b44-4b3f-9e16-3fc1eb9a40b
    <vCard:Given>Peter</vCard:Given>
    <vCard:Family>Villiger</vCard:Family>
    <vCard:Other>John</vCard:Other>
  </rdf:Description>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/4c35bbd0-34c0-4370-bace-5b3d88cc0ae
    <rdf:type rdf:resource="http://www.cellml.org/bqs/1.0#Person" />
    <vCard:N rdf:resource="http://4suite.org/rdf/anonymous/df26ccd4-0b44-4b3f-9e16-3fc1eb9a40b8"
  </rdf:Description>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/08c8eddd-b192-43e5-a507-fe17034025a
    <dc:title>
         A Biophysical Model of the Mitochondrial Respiratory System and Oxidative Phosphorylatio
    </dc:title>
    <dcterms:issued rdf:resource="http://4suite.org/rdf/anonymous/25d1a305-fb14-44e9-ab27-f84d18f
    <dc:creator rdf:resource="http://4suite.org/rdf/anonymous/0442f6e1-7a00-4451-8b8c-7bf05d38f66
  </rdf:Description>
  <rdf:Description rdf:about="http://4suite.org/rdf/anonymous/56792768-20d5-4353-9cce-99413fab6ea
    <bqs:JournalArticle rdf:resource="http://4suite.org/rdf/anonymous/08c8eddd-b192-43e5-a507-fe1
  </rdf:Description>
</rdf:RDF>
```

Some of this is saying:

```
http://www.cellml.org/models/beard_2005_version01#beard_2005_version01
    publisher : The University of Auckland, Bioengineering Research Group
    title : A Biophysical Model of the Mitochondrial Respiratory System and Oxidative Phosphoryla
    journal article : issued : 2005
                    : title : A Biophysical Model of the Mitochondrial
                              Respiratory System and Oxidative Phosphorylation
```

Ontologies and Repositories

: author : Daniel Beard
Biotechnology and Bioengineering Center, Department of Physiology,
College of Wisconsin

Simple data associated with a model component

```
<component cmeta:id="G6P" name="G6P">
  <rdf:RDF>
    <rdf:Description rdf:about="G6P">
      <dc:title>G6P</dc:title>
      <dcterms:alternative>glucose-6-phopshate</dcterms:alternative>
    </rdf:Description>
  </rdf:RDF>
  <variable units="millimolar" public_interface="out" name="G6P" />
  <variable units="flux" public_interface="in" name="V_PFK" />
  <variable units="flux" public_interface="in" name="V_HK" />
  <variable units="flux" public_interface="in" name="V_G6PDH" />
  <variable units="minute" public_interface="in" name="time" />
  <math xmlns="http://www.w3.org/1998/Math/MathML">
   ...
  </math>
</component>
```

Binding into a biopax model

- Sarala talked about this
- What is the intention?
    - Graph to Graph
    - a complete biological model for every cellml model
        - how do we deal with imports?
- How well is it working?

Draft metadata specifications

The specs

Binding into the current physiome ontology

- 201 classes (anatomy graph, pre-coordinated function, simple terms)
- 59770 instances
- everyone has an ontology!
    - what's important is that the intention is clear and equivalence is unambiguous (shared instance or logical equivalence)
- what's in it
    - anatomy concepts
    - physical entities
    - physiological processes ?
    - annotation terms (complex terms)
- binding of components and variables into ontological terms using instances for the terms and not repeating the same complex value for each instance:

        - ◊ there are pros and cons to this pro and con: all possible terms need to be defined globally
        - ◊ process to merge local definition using complex value into global repository
- small excerpt
- binding components doesn't make too much sense: it brought intention of the math, may as well bind the math instead (like the biopax binding)
- the ontological properties of a component would be the aggregate properties of its variables and math.
- need to build out the biopax models
- alternative graph representation to biopax

    - ♦ binding math to biopax processes may be vague without another step
    - ♦ purely process oriented
    - ♦ one to one with math and variables.
- what is the state of this ontology?

    - ♦ useful data, but lack of sensible structure
    - ♦ ontological terms are pre-coordinated
    - ♦ FMA - should we use this (ontological mismatch apparently)
    - ♦ integration into cellml repository nearly there (what is this implementation?)

Ontologies are only useful when used

- what do we need to do to get it into practice?
    - ♦ how can Duane use this?
        - ◊ anatomical topology eg: tissue : bone -> muscle
        - ◊ comfile data ... requires a specification (generate components of his RDF data resources)
    - ♦ how can CellML repository use this?
        - ◊ search interface
        - ◊ all model metadata integrated into the graph
        - ◊ Physiome Workspace relies on it
    - ♦ how do we publish and curate it?
        - ◊ contexts help - ownership of subgraphs of conjunctive graph

What got left out?

Best Practices

Too many constraints in the language would make it difficult to use.

Establish guide by example set of models

Provide tools to generate metrics that suggest better ways of representing a problem

Encourage reuse at every point

- imports
    - how to be import friendly
    - our database needs to promote it
- what is the repositories role in this?
    - decomposing models
    - making suggestions

Acknowledge authorship and establish clear provenance

Provide very good validation and feedback services

CellML imports and the repository library are core elements of promoting re-use and collaboration.

Relationships between models needs to be provided and machine interpretable

Annotate fully - enforce a minimum set of metadata

- - metadata specifications
        - CellML Metadata 1.0
        - Simulation Metadata Specification

          CellML models don't always deal with the general case. e.g. PCEnv

          It would be extremely useful if authors could provide information in a machine readable format, not just about their model, but also about the particular simulation they have run. In this way, it becomes possible to reproduce any simulation results obtained by the author simply by loading the information into a simulation tool and asking it to re-run the simulation.
        - CellML Graph Metadata Specification

          CellML provides a mechanism to describe mathematical models. While CellML can describe the mathematics used by many models published in scientific journals, it does not describe all of the information to create graphs of those models. One step towards this is reproducing experimental results, and this is described in the simulation metadata specification. However, it is also useful to be able to produce the exact graphs from a paper, and this specification allows for metadata which provides the information needed to do this.

◊ Custom Subset Metadata

CellML provides a mechanism to describe mathematical models. CellML allows models of arbitrary complexity to be created, and models can be combined using the import functionality in CellML 1.1 to create even more complex models. One consequence of this is that CellML models can potentially have massive numbers of variables and even components. While backend tools can process such large numbers of variables on modern computers, this creates a significant problem when designing user interfaces. One principle of good user interface design is that you should present the user with the minimum number of choices possible. This makes it faster for the user to locate the functionality they desire, and so improves user productivity. In order to cut down on the choices of variables, components, or other parts of the model available, however, it is necessary to determine which parts of the model the user is likely to edit, in advance. As CellML has been designed to be domain independent, there is no generic way to do this automatically. Moreover, even if we could do this, different users may put the model to different uses, and so have different preferences as to what they will edit.

It is clear that there needs to be some way to specify which parts of the model users will change the most often, so restricted views of the models can be shown to the user (with an option to show the full model if desired). CellML provides a grouping facility, through the use of the cellml:group element. However, this element is inappropriate for this particular application, because:

Grouping only works for components. However, it would be useful to define sets of variables, mathematical equations, and other elements in the model, in addition to components. This could be overcome by making a revised version of the specification, but this would likely introduce many incompatibilities with existing models.

CellML has been designed to separate data, which is essential to the evaluation of the model, from metadata. Because the information being conveyed here relates to how the model should be displayed, rather than the mathematical interpretation of the model, it is clearly metadata. Therefore, it fits better into the existing CellML metadata (cmeta) framework, as RDF, than it does as a part of the core CellML specification.

Some specific proposals:

- Proposal: Best current practice for the top-level mathematics operator
- Proposal: Best current practice for including external code in CellML models

Please please please make use of the cellml-duscussion mailing list

Example: I want to turn my model into a CellML 1.1 model, can you help?

The CellML.org website

- It's about being a community, not a University of Auckland Bioengineering portal.
- Don't have to use a physiome workspace for your daily activity. But it may be useful to at least **create a project level space** that pulls in related data in the repository that you have contributed. Given the annotation of your project space, this will be pulled in automatically.
- You may want to use the software centers (see below)
- If you provide the metadata and enter it into the curation process, it will be published.
- Intended to be used by journal pubishers

We need to

- Have a more formal representation of particular software projects (try the plone software center)
- Promotion of ideas on discussion list into formal proposals
- by example documentation of best practices
- *smart* groupings of models and navigation interface
- conference calls?