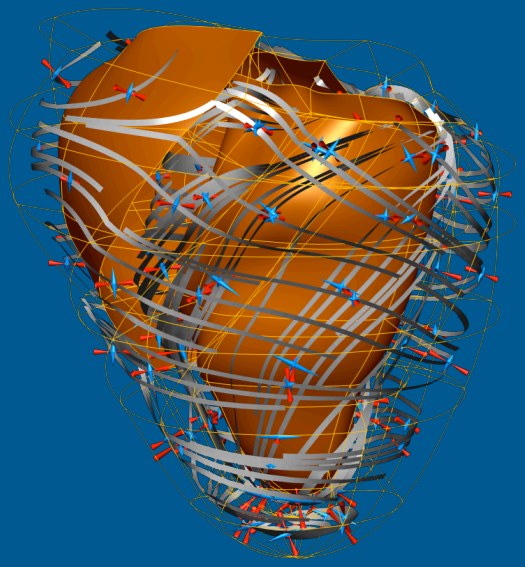
FieldML
version 0.2 status update

By Randall Britten



AUCKLAND
BIOENGINEERING INSTITUTE
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Pre-cursors

- Exnode/exelem file format
  - Not XML
  - Difficult to write parsers
  - Difficult to extend
  - Can’t annotate (e.g. RDF)

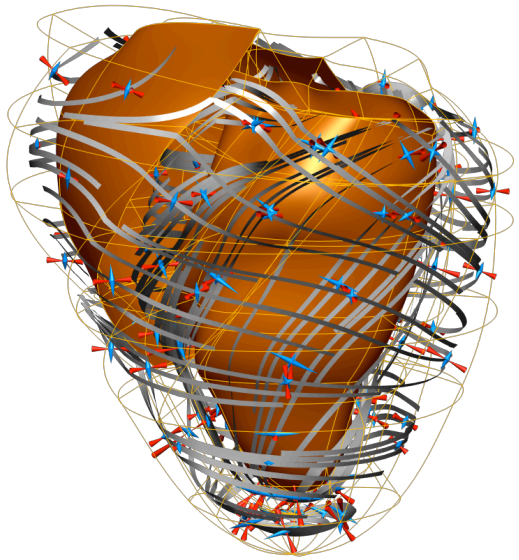
- FieldML 0.1
  - XML format
  - Supported only by cmgui, seldom/never used.
  - Limitations

Software with Fields

- CMISS-cm
- CMISS-cmgui (open source)
- OpenCMISS (open source)

FieldML requirements

- Serialisation
- Represent fields, especially using finite element approach.
- Geometry (1D, 2D, 3D … nD)
- General fields, e.g.:
  - Pressure, fluid velocity, strain tensor, cell model states
  - Material properties
  - Annotation
- Flexible dependencies: Variation with independent parameters: time, subject weight, joint angle



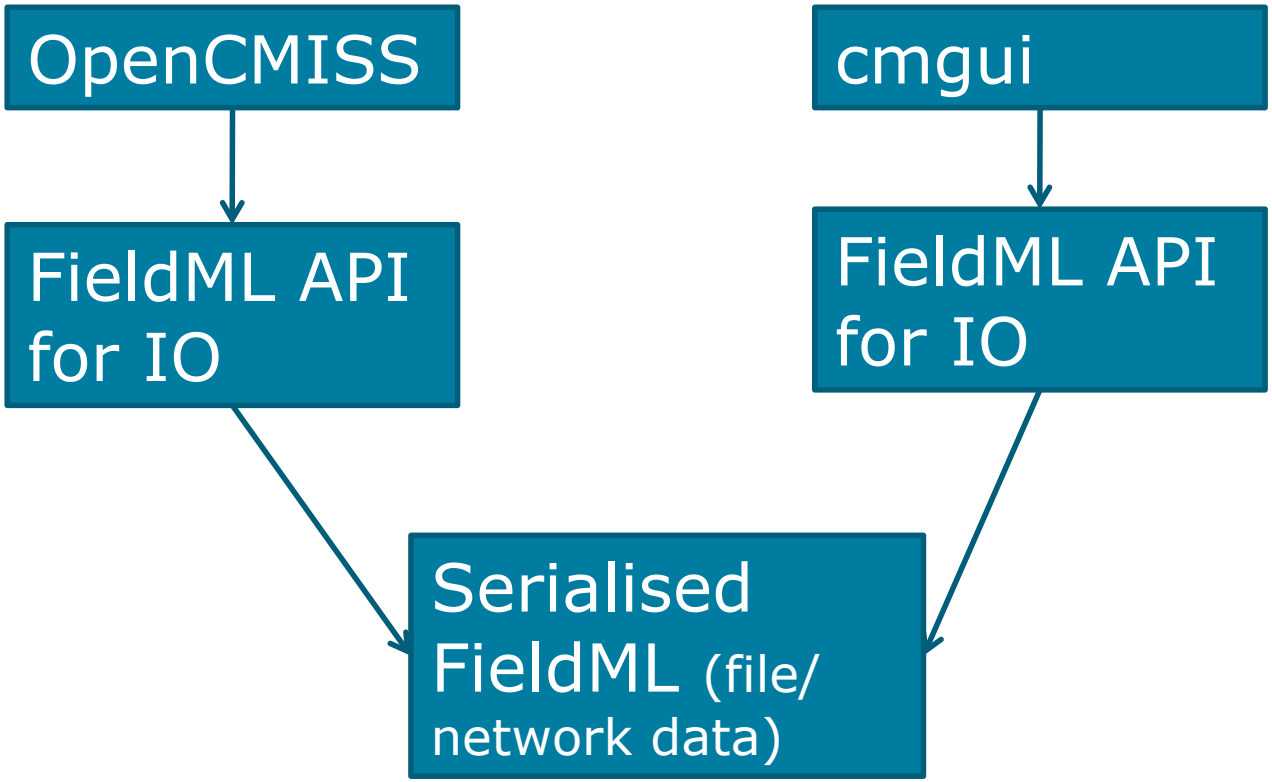
Technical goals

- Emphasis on parsimonious design
- XML Format
- FieldML API

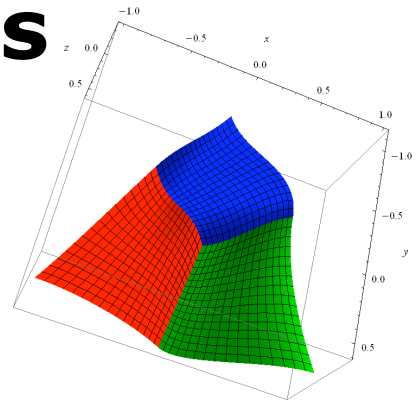
FieldML 0.2 Features
Scope limits

- Focus for this version on basic field data interchange
- Initial acceptance criteria:
  - Interchange between OpenCMISS and cmgui
- Focus on FEM fields
- Refer to basis functions in definition library
  - Future version will explicitly define basis functions e.g. via MathML
  - And in future, use MathML for general computed fields
- “Forwards compatible”

FieldML for data interchange

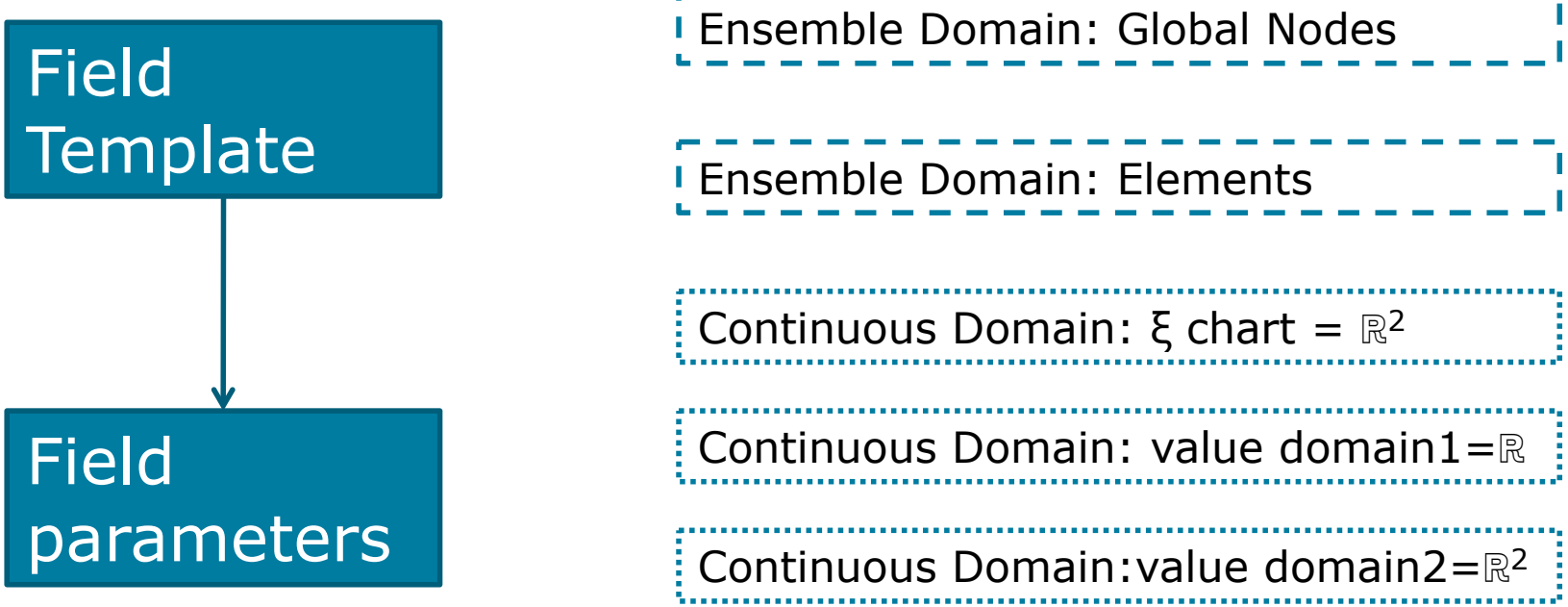


FieldML 0.2 Feature goals



- Generalised Linear mapping
- Bulk numerical arrays in HDF5
- A field on a mesh can use
  - Different shapes per element
  - Different interpolation per element
    - Or, indeed, some other way of obtaining a value for the field, e.g. constant (without handling as $\varphi(\xi)=1$)
- Regions: very basic namespaces
- Library:
  - Fundamental domains
  - FEM interpolation types (cubic Hermite etc)

Overview of design



Overview of design

Field Template

Overview of design

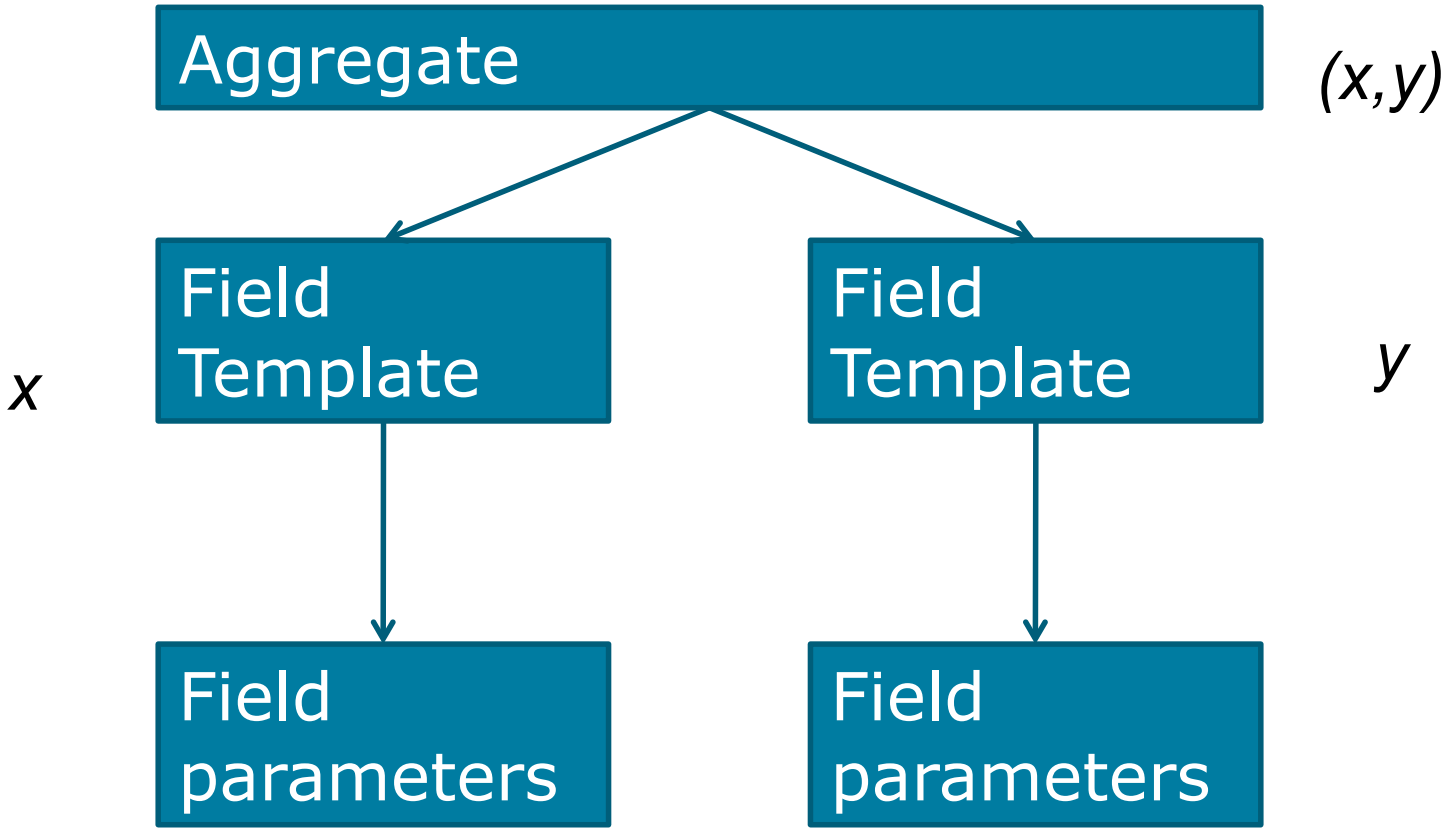
Field Template

Map: element local nodes → global nodes
Map: element → evaluator (i.e. interpolation)
Map: (global node, parameter id) →parameter value

Overview of design

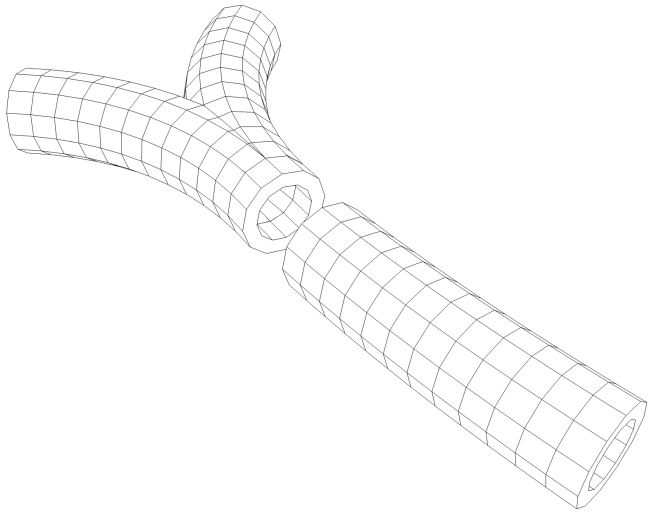
Field Template

Overview of design



Deferred to Future versions of FieldML

- General computed fields
- Coordinate system transformations
- Bulk numerical arrays in formats other than HDF5, e.g. DICOM
- Assembling meshes from sub-parts
- Physical units



Working design representation: FieldML-Java

- Need for rigour
- Java OO design
- Focuses on information structure
- Automatically creates XML mock-ups
- Available online: code.google.com/p/fieldml-java2

FieldML repository

- Based on PMR2 (Physiome model repository 2) (currently used for CellML repository)
- Combined repository
- Possibly combined workspaces

Related efforts

- CellML
- ModML

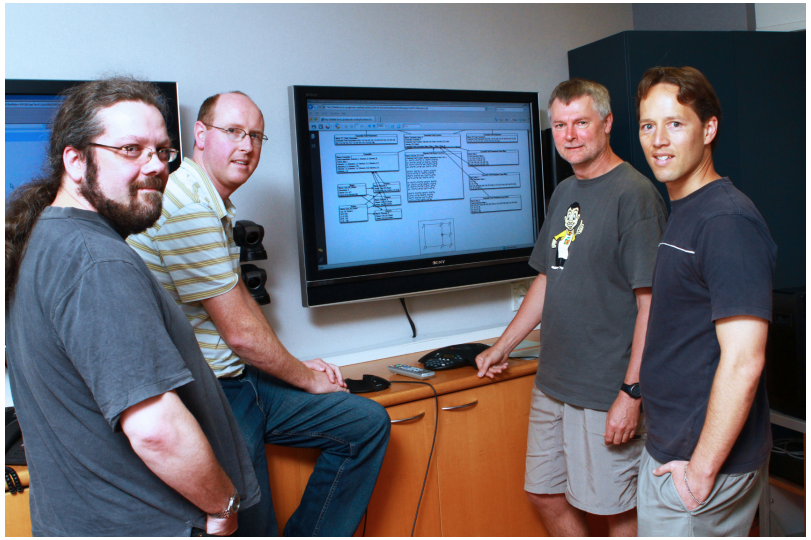
Participation encouraged

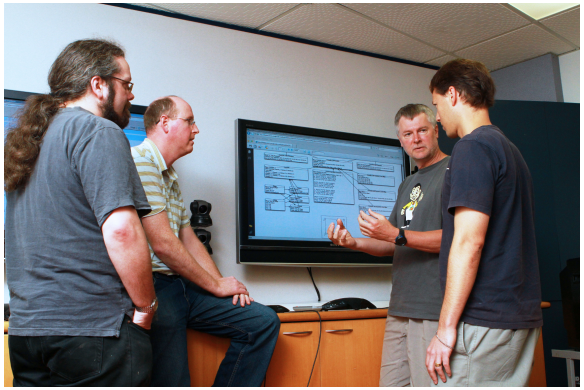
- Website: www.fieldml.org
- Tracker: tracker.physiomeproject.org

- View latest version of design
- Meeting notes
- Mailing list info

FieldML 0.2 workgroup

- Caton Little
- Chris Bradley
- Poul Nielsen
- Randall Britten
- Poul Nielsen



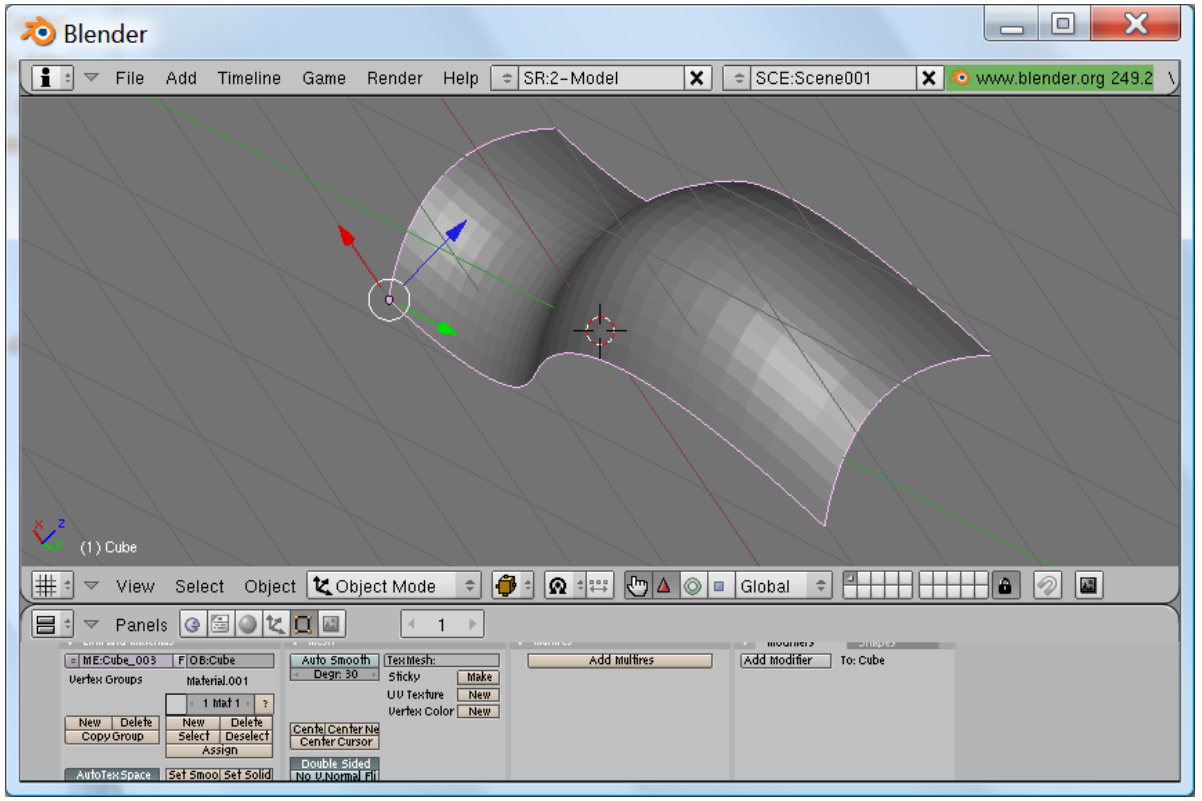


Acknowledgments

- IUPS Physiome project
- Wellcome Trust
- NIH
- FRST
- European Union
  - VPH (Framework 7)
  - euHeart

Discussion

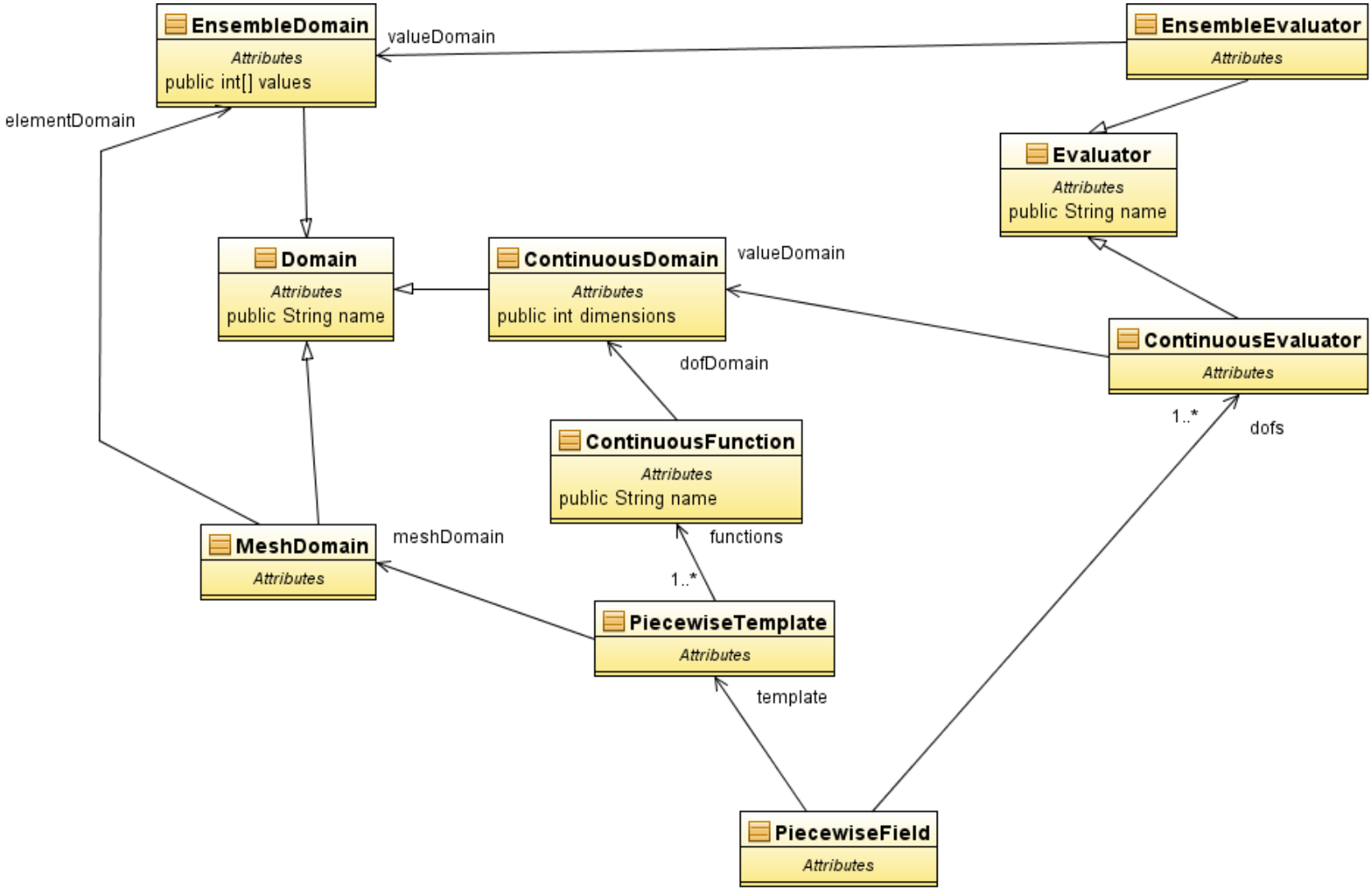
Collada “make-shift” export



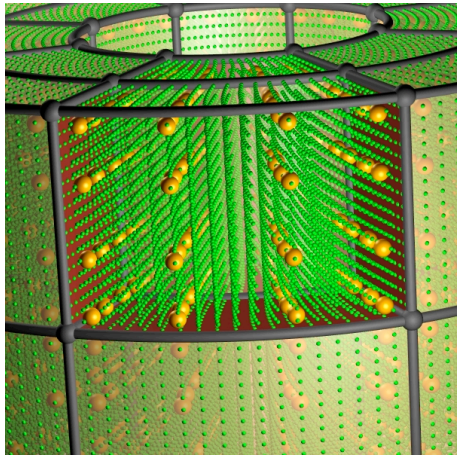
XML mockup

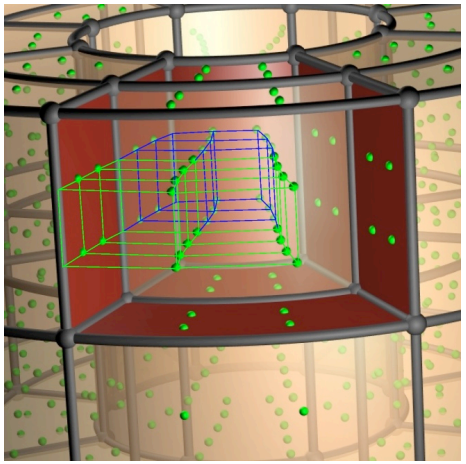
```
<fieldml>

 <ContinuousDomain dimensions="1" name="test_mesh.co-ordinates.dz/ds" />


 <ContinuousDomain dimensions="1" name="test_mesh.co-ordinates.d2z/ds1ds2" />
 …

 <EnsembleDomain name="test_mesh.edge_direction">
   <values>1  2  </values>
 </EnsembleDomain>


 <EnsembleDomain name="test_mesh.elements">
   <values>1  2  </values>
 </EnsembleDomain>
…
 <ContinuousParameters name="test_mesh.node.x" valueDomain="library.co-ordinates.rc.1d">
   <parameterDomains>"test_mesh.nodes"  </parameterDomains>
   <entries>
     <MapEntry><keys>1  </keys><values>0.0  </values></MapEntry>
     <MapEntry><keys>2  </keys><values>1.0  </values></MapEntry>
…
 </ContinuousParameters>
…
 <PiecewiseTemplate name="test_mesh.coordinates.L2" meshDomain="test_mesh.domain">
   <elementMaps>
     <TemplateMap element="1" evaluator="test_mesh.element.bilinear_lagrange" />
     <TemplateMap element="2" evaluator="test_mesh.element.bilinear_lagrange" />
   </elementMaps>
 </PiecewiseTemplate>
…
</fieldml>
```

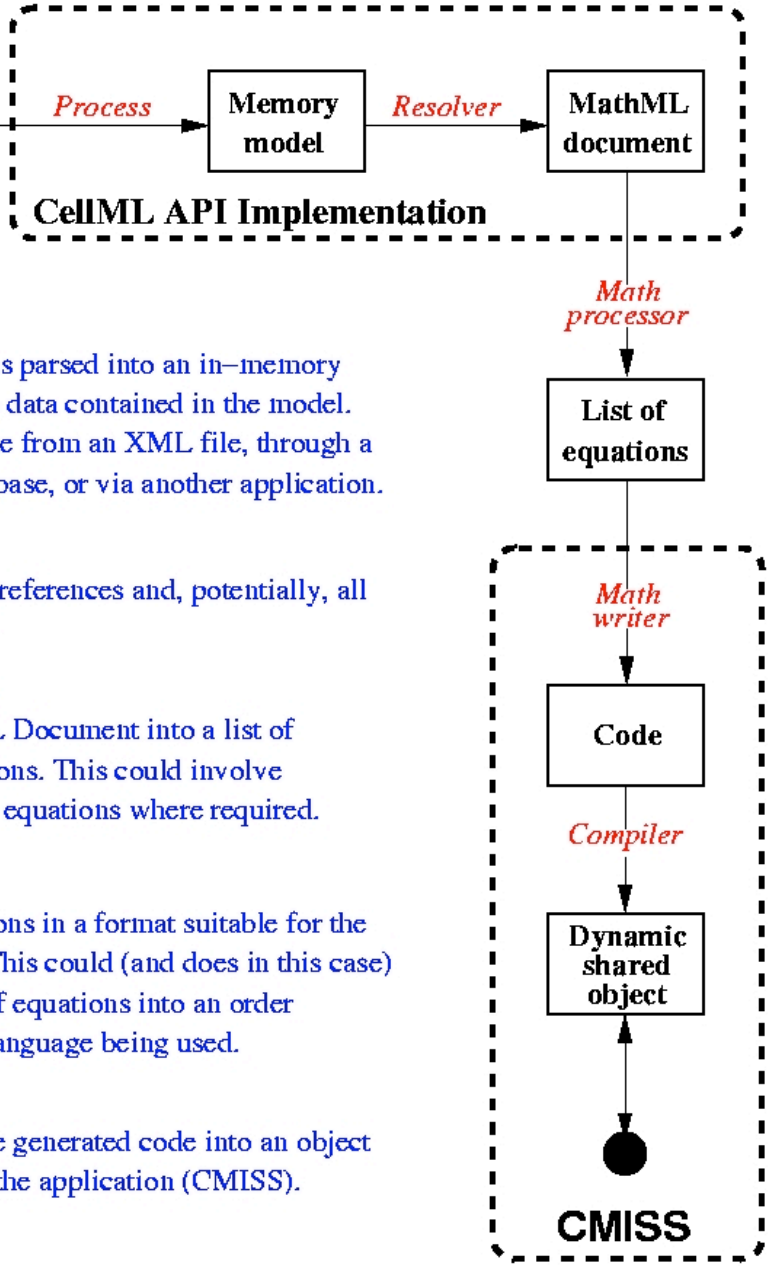
Class diagram



Incorporating CellML into continuum models







Process
The CellML model is parsed into an in–memory representation of the data contained in the model. The model may come from an XML file, through a connection to a database, or via another application.

Resolver
Resolve all variable references and, potentially, all unit inconsistencies.

Math processor
Process the MathML Document into a list of mathematical equations. This could involve simplification of the equations where required.

Math writer
Write out the equations in a format suitable for the language required. This could (and does in this case) require the sorting of equations into an order appropriate for the language being used.

Compiler
Compile and link the generated code into an object ready to be used by the application (CMISS).