FieldML: Data Structures for Modelling

Richard Christie, Poul Nielsen
r.christie@auckland.ac.nz
p.nielsen@auckland.ac.nz

Auckland Bioengineering Institute
Objective

- Expressive, efficient representation of fields based on minimal set of concepts.
- From this, create serialization format (ML) to interchange modelling data.

... Most packages will only be able to read or write a fraction of the format, translating to their internal structures.
What is a Field?

“"A set of values defined over some domain”

Domains:
- Discrete (list of items, e.g. points)
- Continuous (coordinate system, element/chart)
- Combinations (mesh/atlas, time signals)

Values:
- Real, integer, string, object...
- Multiple components or sub-fields (vectors, tensors)
Example Fields

- Spatially-varying coordinates, scalar.

- Data points. (Analogy with fields in database records.)

- Signals, interpolation in time.
What’s so special?

► “Everything is a field”; a necessary abstraction to deal with size and complexity:
  ▪ Each field representation only as detailed as problem requires.
  ▪ Arbitrary variation of any field with any other field.

► From minimal set of concepts, serialized format will be largely self-describing: e.g. basis functions defined with MathML expressions.

► C.f. typical FEM formats: fixed element types identified by magic numbers, special handling of coordinates, material properties etc. all leading to software limitations.
Building Block 1: Domains

► “Regions” introduce new namespaces for fields and sub-domains. Hierarchical; may encapsulate e.g. whole body or organ.

► Coordinate systems, lists (e.g. nodes), meshes (of particular interest).

... Are these just special types of field?
Building Block 2: Field Functions

- Field values = f(source field values)
- Functions include field value lookup, mathematical functions, even values returned by software constructs not part of FieldML (e.g. texture lookup).
- Particularly interested in finite element fields = dot product of basis functions with element field parameters.
Building Block 3: Field Parameters

► Degrees of Freedom contributing to field (not necessarily values of field).
► Parameter lists stored in objects, e.g. field itself, or domain objects (nodes, elements).

... Are these just a special type of field? Mappings of parameters to elements are then same as field functions.
Field Compositions in CMGUI

gfx define field Identity3 composite 1 0 0 1 0 0 0 1;
gfx define field F gradient coordinate undeformed_coordinates field deformed_coordinates;
gfx define field F_transpose transpose source_number_of_rows 3 field F;
gfx define field C matrix_multiply number_of_rows 3 fields F_transpose F;
gfx define field E2 add fields C Identity3 scale_factors 1 -1;
gfx define field E scale field E2 scale_factors 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5;
gfx define field principal_strains eigenvalues field E;
gfx define field principal_strain_vectors eigenvectors eigenvalues principal_strains;
Defining a finite element field

1. Define prototype element shape/chart field = $\xi$
2. Define basis function field in terms of $\xi$, e.g. MathML expressions
3. Define mesh field = set of elements with shape
4. Define discrete domain of nodes with nodal field parameters; alternatively global DOF vector.
5. Over each element:
   1. Define field function mapping nodal/global field parameters to element.
   2. Define continuous F.E. field as dot product of basis function and element parameters.
FieldML Challenges

► Handling Big Data. Need binary, compressed representations.
► Distributed memory multi-processing. Parallel I/O. Whole mesh connectivity needed to decompose mesh.
► Hierarchical meshes for Adaptive Mesh Refinement. Per-field refinement to minimise problem size.
► General embedded meshes.
FieldML Status

- All CMISS applications built with some level of FieldML concepts.
- We intend to develop open source CMGUI to follow full FieldML specification, but note migration issues and costs, plus some existing limitations:
  1. Incomplete implementation of Regions.
  2. Fixed nodal/element field parameter representations.
  3. Element parameter maps and basis functions are special/fixed functionality.
  4. Field composition functions are special; would like to add general mathematical formulas, e.g. MathML expressions.
Thank You!