Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2010) 1–9
Procedia Computer Science

MML: towards a Multiscale Modeling Language

Jean-Luc Falcone[a], Bastien Chopard[a,*], Alfons Hoekstra[b]

[a] *University of Geneva, Switzerland*
[b] *University of Amsterdam, The Netherlands*

Abstract

Recent multiscale applications require more and more often the coupling of many sub-models, usually originating form different fields of science. Therefore, it is increasingly important to propose an effective description language that can help scientists with different background to co-develop a multiscale application. We propose a *Multiscale Modeling Language (MML)* i.e. a description language aiming at specifying the architecture of a multiscale simulation program. We will illustrate this approach by proposing a MML description of a computer model for restenosis in a stented vessel.

Keywords: Multiscale modeling, description language, cellular automata, lattice Boltzmann method, scale separation

1. Introduction

Most current multiscale applications are restricted to the coupling of two sub-systems, with a micro-macro scale relation. However, there is a growing interest for complex problems requiring the coupling of many sub-models, usually originating form different fields. Biomedical systems, for instance, involve biological, chemical and physical processes evolving at different scales. Following this approach, we recently developed a multiscale biomedical model for restenosis in stented coronary arteries [1]. This model involves three different processes, all acting at their own time scale: blood flow, smooth muscle cell proliferation and drug diffusion. Developing such a model in a consortium of computer scientists, clinicians and biologists demonstrated the importance of having a natural, intuitive and informative way to describe all the processes in interaction.

Since a few years, scientists are studying more and more complex applications by integrating together several new or existing single-science and single-scale models. Simultaneously, development teams are often composed of different specialists working on separated geographical

*Corresponding author
Email addresses: Jean-Luc.Falcone@unige.ch (Jean-Luc Falcone), Bastien.Chopard@unige.ch (Bastien Chopard), A.G.Hoekstra@uva.nl (Alfons Hoekstra)

locations. Therefore, it is increasingly important to propose an effective description language that can help to co-develop multiscale applications and collaborate efficiently to the architecture of the solver. Also, with such a detailed "blueprint" of an application, other groups can then easily extend the model by incrementally adding more features, without the need for a full re-engineering.

In this paper we propose a *Multiscale Modeling Language (MML)* i.e. a human-friendly description language based on a graphical representation and/or an XML language. MML aims at specifying the architecture of a multiscale simulation program. In particular, it allows the scientists to specify the list of sub-models, their coupling, (including the relation between the computational domains and scales), the type of coupling (coarse graining, scale splitting, amplification), input and output data, etc. We will illustrate this approach by proposing a MML description of a restenosis multiscale application.

Here we will assume that the sub-models are only based on the cellular automata or lattice Boltzmann methods and obey the general formalism proposed in the so-called CxA framework [2, 3, 4]. Within this framework, all the objects are well defined, which allows us to be quite specific. In addition, in that case, our MML description can be parsed automatically in order to produce the skeleton of the application and, soon, to generate the full coupling code of the multiscale simulation.

Although the current work is restricted to the CxA multiscale methodology, we are convinced that it can be extended to other numerical frameworks. The goal of this paper is mainly to stimulate the multiscale community to discuss the opportunity of MML and to open the door to its specification.

2. Multiscale coupling with CxA

2.1. General concepts

We have recently proposed in [3, 4] a methodology for multiscale modeling based on the coupling of any numbers of lattice Boltzmann [5] and cellular automata [6] sub-models, all running at their specific time and spatial scales. We coined this framework CxA for "Complex Automata". Together with the theoretical concepts, we have also developed a coupling software called MUSCLE [7, 8] which provides the glue to run simulations with these interacting sub-models.

In our framework, A distinguishing feature of a multiscale application is the relation that exists between the computational domains of each sub-model. We propose to consider two cases: (1) single domains where both sub-models share the same spatial domain and (2) multi-domain where the different sub-domains are adjacent or slightly overlapping.

However the main ideas of the CxA framework is that (1) cellular automata (CA) or Lattice Boltzmann (LB) models can be described by a generic sequence of calls to well defined operators (the *so-called main loop*) and that (2) the coupling between any two sub-models can be expressed as a flow a data between a pair of these operators. Figure 1 illustrate these features on the coupling of two models having for instance different temporal resolution and simulating the growth of coral resulting from the nutrient brought by water currents.

In our generic description, the state of the CA or LB model over the computational domain D is denoted by the symbol f. The operators are U for updating the computational domain, B for imposing the boundary conditions, C for performing the so-called collision step and P for the propagation of the f on the computational domain. Note that these operators are specific to the

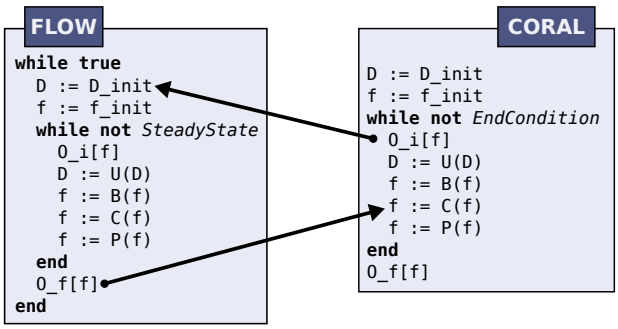

Figure 1: Example of the generic loops and coupling template describing two coupled submodels.

LB and CA models and we refer the reader to [3, 4] for a full description. Finally the operator O_i and O_f denotes any observation that needs to be made on the system.

We have observed that not all coupling patterns are present in the rather large set of multiscale applications we have considered in [4]. Only a few the pairs of operators are actually needed to express the multiscale coupling between two sub-models. Among these possible coupling, some reflect a single-domain relation and other a multi-domain relation. Also the nature of coupling (coarse graining, scale splitting or amplification [4]) further specifies the pair of operators that can be involved.

This rather small set of possible couplings suggests that a graphical language can be devised to express the existing interaction between sub-models in our CxA framework. This is the very purpose of this paper. But before proposing a set of rule to implement this MML language, we need to specify a few more elements of the framework.

First, as discussed above, a coupling template corresponds to a flow of data between two operators. However, the data may need some transformation between the two ends of the link. For this reason a *filter* can be inserted along the data path to performed the required transformation, such as a change of scale, an interpolation or decimation. The filter is the component that knows the scale requirements of both subsystems, whereas the sub-model are thought of as being stand-alone and ready to be used in any different context.

Second, when more than two models are coupled a simple link is not enough and another component called *mapper* is needed. A mapper acts as a central element which receives information from all connected sub-models (through their operators O), combines this information as required by the nature of the coupling, and sends it to the adequate operators of the recipient sub-models.

2.2. ISR application

To illustrate better the MML language we shall define below for describing the implementation of a multiscale application, we first briefly present the in-stent restenosis (ISR) model [1] that was developed in the COAST project as a demonstrator of the CxA framework.

The ISR application aims at developing a computer model of the proliferation of the smooth muscle cells (SMC) following the deployment of a stent of a blood vessel [1]. This undesired phenomena is called hyperplasia. SMC proliferation inside the lumen is the result of an initial injury of the vessel wall by the stent. The blood flow affects hyperplasia through a coupling between the shear stress and the SMC growth rate. In turn, hyperplasia changes the geometry

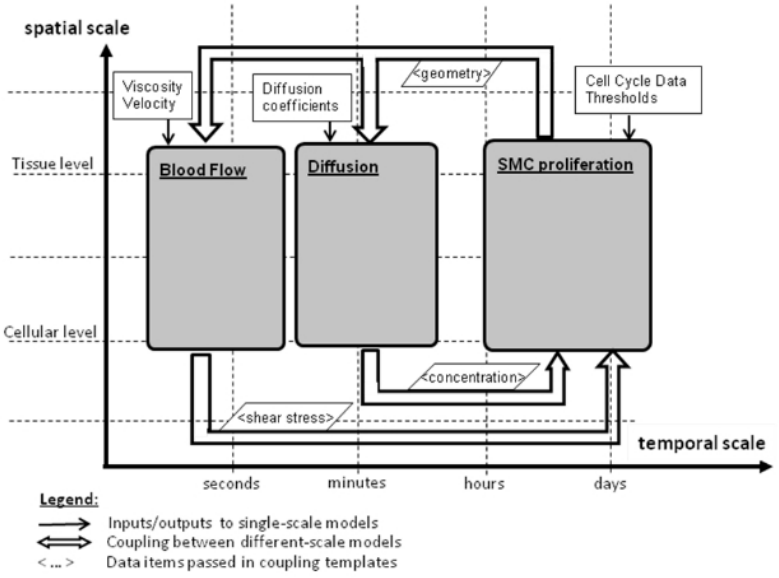

Figure 2: Schematic representation of the in-stent restenosis application on a scale separation map.

of the vessel, thus creating a feedback on the blood flow. A way to reduce SMC proliferation is to use a drug eluted stent from which some chemical diffuses in the wall vessel to prevent hyperplasia.

We have proposed [2] to express graphically the ISR application using the scale separation map (SSM), as illustrated in Fig. 2. This diagram presents the three sub-models (BF for blood flow, DD for drug diffusion and SMC for smooth muscle cell evolution) as a function of their characteristic spatio-temporal scales. This drawing further shows the coupling between the sub-models. This representation turned out to be very useful when discussing with biologists or clinicians since it helped people to extract the important ingredients of a multiscale problem and to identify the potential for simulation speedup offered by the separation of scales.

The SSM is therefore the first instance of a MML graphical language. It is well adapted to describe the coupling between sub-models having well separated scales. But it soon will be unreadable when the scales of several models overlap. For this reason we proposed in the next section additionnal diagrams of a multiscale application which augment the SSM view by providing extra information.

3. Coupling Diagram

MML can be represented as diagrams. Those diagrams help to design and describe a complex coupling by showing which are the components that constitutes the model, the flow of information between components, the space and time scale of components and exchanged data and the execution timeline. Such representation has several usages: first it can be easily hand-drawn to facilitate scientific and programming discussion, then it illustrates coupling specifications, finally it provides a standard way of describing a model for publication.

This visual aspect of MML is similar to *Unified Modelling Language* (UML) diagrams well known in software engineering [9]. Indeed, MML diagrams complement UML models by providing higher level information related to scientific coupling of subsystems (notably scale information and coupling approach). Both systems can then be used side by side. For example, the components of an MML diagram can be specified further by providing UML class and activity diagrams giving implementation detais.

All information shown in diagrams can be inferred from the XML representation (see below), so given a particular MML XML file, diagrams can be automatically generated.

Our MML proposition includes two diagrams. The first is the scale separation map shown in Fig. 2. The other one is the coupling diagram detailed below.

In the coupling (or dataflow) diagram components are represented by boxes of different shapes. Their size and their position in the diagram do not carry any information. Their relative position in the diagram must be chosen in order to increase readbility. The shape of components are:

- *Submodels* are represented by rectangular boxes. The title should lay in the upper part such as to leave the bottom part open for optional annotations. The center must stay empty because, it may be used to attach coupling arrows.
- *Filters* are shown as rectangular boxes with rounded corners. When hand-drawn, oval shapes can be substituted. Their orientation is not relevant.
- *Mappers* Mapper are represented by hexagonal boxes. Their orientation is not relevant.

In addition to components, the coupling diagram represents the dataflow by an edge (or "arrow") linking two components. The symbols at extremities and the point where they attach to the components carry information about the coupling.

In principle each arrow represents should be labelled to describe the kind of data that is exchanged (i.e. velocity, pressure, number of cells). However, to increase readability, several arrows linking the same components at the same points with the same extremities can be showned as one. In this case, the resulting arrow is labeled with a list of labels, joined by commas.

Arrow ends. The arrow extremity markers show the *main loop operators* involved in the information exchange. The extremity attached to the *sending submodel* can have to markers:

- When the information is sent during the submodel computation (operator O_i) the arrow extremity comes with a black circle.
- When the information is sent after computation (operator O_f), the symbol is a black diammond.

When the data originates from a filter or a mapper, the marker is ommited. This simplification can also be made for submodels which do not follow the main loop described in Sec. 2.1 or for preliminary drafts. The different markers are summarized in Fig. 3.

For the marker extremity which is attached to the *sending submodel* we have four possibilities:

- When the information is used in the *initial condition*, the arrow finishes with a white diammond.

Figure 3: Coupling arrows marker palette.

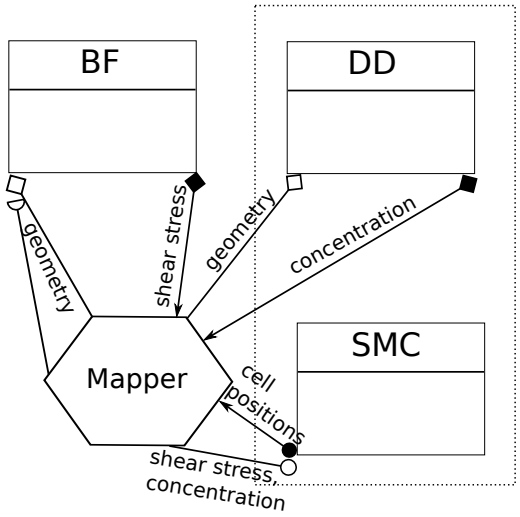

Figure 4: Coupling diagram for the ISR model presented in Sec. 2.2. *BF:* blood flow; *DD:* Drug diffusion; *SMC:* smooth muscle cells.

- When the information is used to update the domain (operator U), the arrow finishes with a white half circle.
- When the information is used in collision operator (operator C), the arrow end is a white circle.
- When the information is used to update boundaries (operator B), the arrow end is a white triangle.

When the coupling recipient is a filter or a mapper the end marker is a simple arrow end. Again, this "default" marker can also be used for submodels which do not follow the main loop or for preliminary drafts.

Attachement points. The exact places in submodels where coupling arrows are attached inform about the space and time scale of the exchanged data. As in scale maps, the x axis refers to the time frequency and the y axis to the space frequency. In both directions we have three possibilities:

- **Time axis**: data at the same time frequency than the connected submodel (to the left), data at a frequency equals to the duration length of the connected model (to the right) and data sent at an intermediate frequency (centered horizontaly).
- **Space axis**: data at the same space frequency than the connected submodel (to the bottom), data at a frequency equals to the total size of the connected model (to the right) and data sent at an intermediate frequency (centered vertically).

Since both scales can be addressed separately, we have a total of nine possible attachment points.

For instance in Fig. 4 the shear stress data flows from the blood flow model to the mapper. The source extremity is attached to the bottom-right corner to the BF submodel. It means that the data is sent at a fine spatial scale but at a time resolution equaling the BF total duration (i.e. the time needed to reach the flow steady-state).

The coupling diagram can be completed with additional features. Here we only mention an indication for the *domain relation* (multi-domain or single domain). When two or several models share the same physical domain (for instance one plant growth model and an animal growth model occurring in the same patch of land), we can draw a dotted rectangle around the submodels. For example, in the ISR coupling diagram (Fig. 4) the smooth muscle cell and the drug diffusion submodels share the same domain (artery wall).

4. XML representation

In addition to the visual models presented above, MML includes an XML [10] representation called xMML. Using this representation, users can generate text files describing formally coupled multi-scale systems. xMML grammar is designed to allow easy interact with both human users and software. Therefore, they can be written either with a plain text editor or a dedicated program with GUI. For example, it is possible to write a graphical editor for the diagrams seen above which can produce the corresponding xMML files.

Usage of such files includes the following cases:

- Generating automatically MML diagrams.
- Producing a specification skeleton for a given coupled model, in HTML, LaTeX, etc.
- Generating coupling "glue-code" for middlewares using ad-hoc generators.

The XML format was chosen because for three main reasons. First XML is now very common and parsing libraries exists for almost every language used in scientifc computing (XML parsing libraries are avalaible for Fortran too). Then format based on XML can be automatically validated if a grammar file is defined (DTD, schemas, etc.). This simplify the writing of robust parsers because most checks are done automatically before semantic analysis. Finally, the Extensible Stylesheet Language family (XSL) allows user to describe complex transformation from XML to other formats.

The main drawback of XML is its verbosity which can make the file size bigger than expected and can hinder human readability. However, this verbosity can be strongly reduced by limiting hierarchical levels and using attributes instead of inner tags every time it is possible.

The feasability of coupling code generation was demonstrated by Armstrong et al. [11]. They have realized a coupling framework called BFG2 (Bespoke Framework Generator 2) used in

```
<model id="suspensionFlow">
  <description>
    Flow with a suspension of particles. The conentration
    of particles affect locally the flow viscosity and the
    particles are advected by the flow.
  </description>
  <submodel id="flow">
    <spacescale dimension="2" dx="1 mm" lx="10 cm" ly="30 cm" />
    <spacescale dt="1 ms" t="1 min" />
    <ports>
      <in id="concentration" operator="C" dt="1 ms" dx="1 mm" />
      <out id="velocity" operator="Oi" dt="10 ms" dx="1 mm" />
    </ports>
  </submodel>
  <submodel id="advectionDiffusion">
    <spacescale dimension="2" dx="1 mm" lx="10 cm" ly="30 cm" />
    <spacescale dt="10 ms" t="1 min" />
    <ports>
      <in id="velocity" operator="C" dt="10 ms" dx="1 mm" />
      <out id="concentration" operator="Oi" dt="10 ms" dx="1 mm" />
    </ports>
  </submodel>
  <coupling from="flow.velocity" to="advectionDiffusion.velocity" />
  <coupling from="advectionDiffusion.concentration" to="flow.concentration">
    <filter kind="timeInterpolation" />
  </coupling>b
</model>
```

Figure 5: xMML file example.

earth system modelling. BFG2 uses XML metadata to produce coupling code for submodels written in Fortran. Although some ideas are similar, xMML contains are higher level description with emphasis on multiscale coupling. Note that small coupling details can be added in xMML documents inside extra section to allow code generation for a given framework.

Here, due to lack of space, we do not present the full proposed grammar specifications of xMML. Instead we present in Fig. 5 a typical example, which we expect to be self-explanatory. This example illustrates a multiscale problem describing the advection-diffusion of suspensions due to a fluid flow. In turn, the suspensions locally modify the fluid viscosity according to a given phenomenological law. The full model is composed of two sub-models which are not further specified here.

5. Conclusion

The Multiscale Modeling Language we have presented here aims at giving scientists with different backgroud and geographical locations a better way to co-develop a complex multiscale

application with many coupled sub-models. The current version of MML is a proposition, open to discussion in the community and which can be easily further specified or modified if needed.

This work is supported by the European Commission (COAST project EU-FP6-IST-FET Contract 033664)

[1] D. Evans, P.-V. Lawford, J. Gunn, D. Walker, D.-R. Hose, R. Smallwood, B. Chopard, M. Krafczyk, J. Bernsdorf, A. Hoekstra, The application of multi-scale modelling to the process of development and prevention of stenosis in a stented coronary artery, Phil. Trans. Roy. Soc.In press.

[2] A. Hoekstra, E. Lorenz, J.-L. Falcone, B. Chopard, Towards a complex automata formalism for multuscale modeling, Int. J. Multiscale Computational Engineering 5 (6) (2008) 491–502.

[3] A. Hoekstra, F. J.-L, A. Caiazzo, B. Chopard, Multi-scale modeling with cellular automata: The complex automata approach, in: H. U. et al. (Ed.), ACRI 2008, Vol. LNCS 5191, Springer-Verlag Berlin Heidelberg 2008, 2008, pp. 192–199.

[4] A. G. Hoekstra, A. Caiazzo, E. Lorenz, J.-L. Falcone, B. Chopard, Complex automata: multi-scale modeling with coupled cellular automata, in: A. Hoekstra, J. Kroc, P. Sloot (Eds.), Modeling complex systems with cellular automata, Vol. chapter 3, Springer Verlag, 2010.

[5] S. Succi, The Lattice Boltzmann Equation, For Fluid Dynamics and Beyond, Oxford University Press, 2001.

[6] B. Chopard, M. Droz, Cellular Automata Modeling of Physical Systems, Cambridge University Press, 1998.

[7] J. Hegewald, M. Krafczyk, J. Tölke, A. Hoekstra, B. Chopard, An agent-based coupling platform for complex automata, in: M. B. et al. (Ed.), ICCS 2008, Vol. LNCS 5102, Springer-Verlag Berlin Heidelberg 2008, 2008, pp. 291–300.

[8] http://developer.berlios.de/projects/muscle/.

[9] Object Management Group, UML 2.0, `http://www.omg.org/spec/UML/2.0/` (2005).

[10] World Wide Web Consortium, Extensible markup language 1.0, `http://www.w3.org/TR/xml/` (2008).

[11] C. W. Armstrong, R. W. Ford, G. D. Riley, Coupling integrated earth system model components with bfg2, Concurr. Comput. : Pract. Exper. 21 (6) (2009) 767–791. doi:http://dx.doi.org/10.1002/cpe.v21:6.