Multiscale-Multiscience modeling: concepts and methodology

Bastien Chopard

University of Geneva, Switzerland

MAPPER Seasonal School, UCL, Feb. 1, 2012

Main Contributors

- Joris Borgdorff, UvA
- Jean-Luc Falcone, UNIGE
- Alfons Hoekstra, UvA

Multiscale, multiscience modeling framework

- Propose a modeling and simulation framework for multiscale, multisciences complex systems

Multiscale, multiscience modeling framework

- Propose a modeling and simulation framework for multiscale, multisciences complex systems
  - Theorectical concepts : Complex Automata approach (CxA)

Multiscale, multiscience modeling framework

- Propose a modeling and simulation framework for multiscale, multisciences complex systems
  - Theorectical concepts : Complex Automata approach (CxA)
  - A Multiscale Modeling Language : MML

Multiscale, multiscience modeling framework

- Propose a modeling and simulation framework for multiscale, multisciences complex systems
  - Theorectical concepts : Complex Automata approach (CxA)
  - A Multiscale Modeling Language : MML
  - Software environment : The MUSCLE coupling library

Motivations

- Very few methodological papers in the literature.

Motivations

- Very few methodological papers in the literature.
- Multiscale strategies are usually entangled with applications.

Motivations

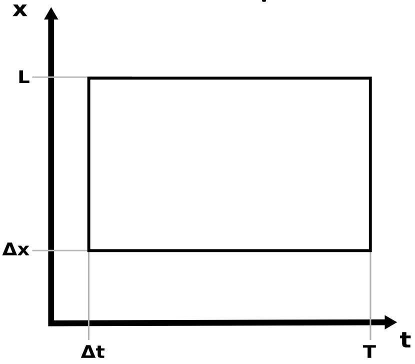
- Very few methodological papers in the literature.
- Multiscale strategies are usually entangled with applications.
- Can we develop a framework that help the design and deployment of complex multiscale-multiscience applications ?

From a multiscale systems to many single-scale systems :

Let us consider a system of size L evolving over a time T.
Computation with space and time discretization Δx and Δt

Resolved spatial scales : $\Delta x < \xi < L$ and
Resolved temporal scales : $\Delta t < \tau < T$

Scale Map

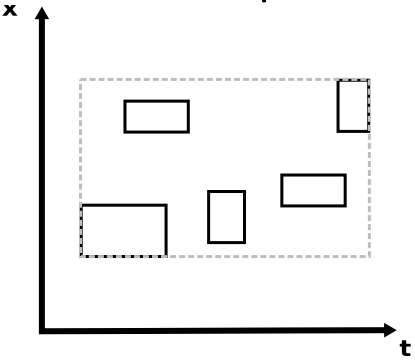


From a multiscale systems to many single-scale systems :

Let us consider a system of size L evolving over a time T.
Computation with space and time discretization Δx and Δt

Resolved spatial scales : $\Delta x < \xi < L$ and
Resolved temporal scales : $\Delta t < \tau < T$

Scale Map

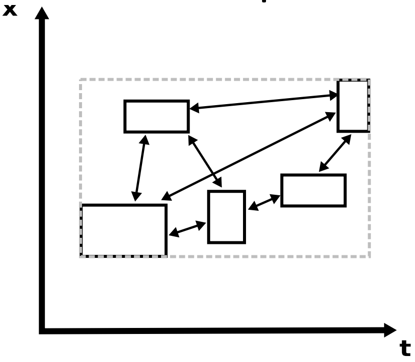


From a multiscale systems to many single-scale systems :

Let us consider a system of size L evolving over a time T.
Computation with space and time discretization Δx and Δt

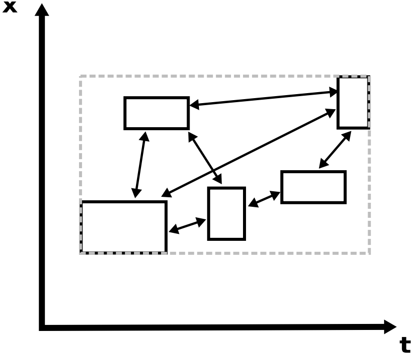
Resolved spatial scales : $\Delta x < \xi < L$ and
Resolved temporal scales : $\Delta t < \tau < T$

Scale Map



From a multiscale systems to many single-scale systems :

Scale Separation Map



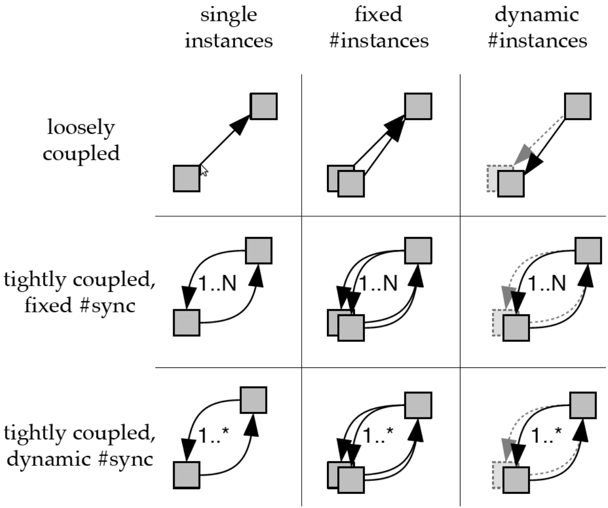
- Submodels
- Smart Conduits

The CxA approach and beyond

- A multiscale problem is a graph of coupled (single-scale) submodels
- The submodels may implement many different numerical methods
- but they are decribed with the same generic **execution loop**
- Submodels should not know about the rest of the system : they are autonomous components
- Only the **smart conduits** know about the properties of the submodels they connect.

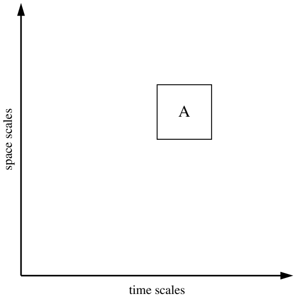
A. G. Hoekstra, A. Caiazzo, E. Lorenz, J.-L. Falcone, and B. Chopard. *Complex Automata : multi-scale Modeling with coupled Cellular Automata*, in **Modelling Complex Systems by Cellular Automata**, chapter 3, Springer Verlag, 2010.

Coupling topologies (workflow)



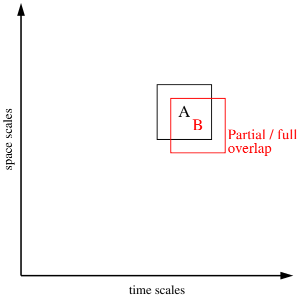
I. Relation between the scales

- The Scale Separation Map (SSM) specifies the relation between the sub-models in five regions :.



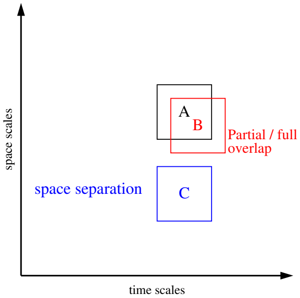
I. Relation between the scales

- The Scale Separation Map (SSM) specifies the relation between the sub-models in five regions :.



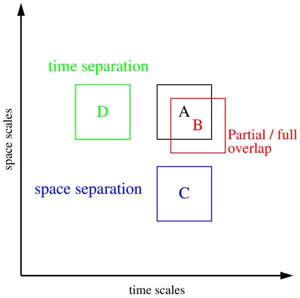
I. Relation between the scales

- The Scale Separation Map (SSM) specifies the relation between the sub-models in five regions :.



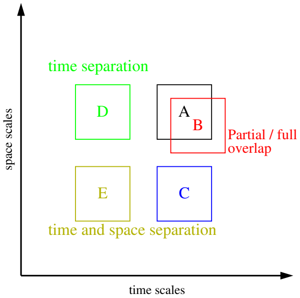
I. Relation between the scales

- The Scale Separation Map (SSM) specifies the relation between the sub-models in five regions :.



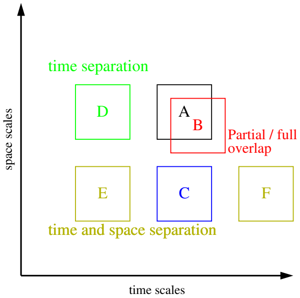
I. Relation between the scales

- The Scale Separation Map (SSM) specifies the relation between the sub-models in five regions :.



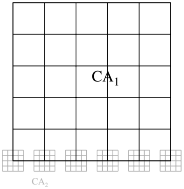
I. Relation between the scales

- The Scale Separation Map (SSM) specifies the relation between the sub-models in five regions :.
- There is more than the standard micro-macro relation and more than than the "bi-scale" modeling



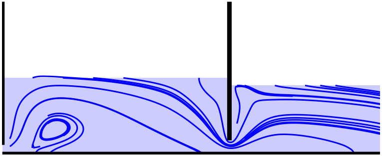
II. Relation between computational domains

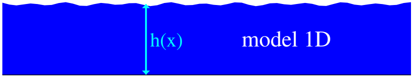
single-Domain (sD)



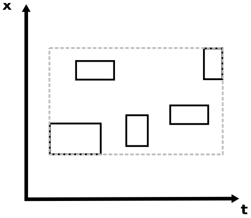
(Example : advection-diffusion, suspension flows)

multi-Domain (mD)





III Generic "Submodel Execution Loop"

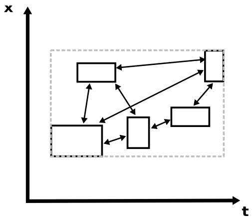


- f_{init} is for initialization
- S is for one iteration of the Solver
- B is to specify the Boundaries
- O_i is for Intermediate Observation
- O_f is for Final Observation

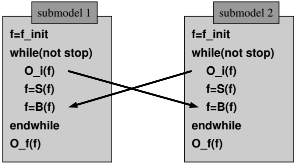
submodel

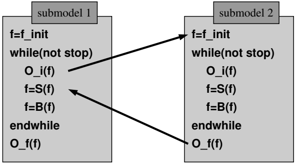
```
f=f_init
while(not stop)
    O_i(f)
    f=S(f)
    f=B(f)
endwhile
O_f(f)
```

IV. Coupling Templates



- One has several operators in the submodel execution loop
- O_i, O_f as origin
- f_{init}, B and S as possible destinations





Example : Coral growth

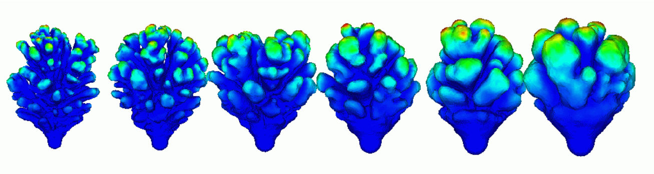
FLOW

```
D := D_init
f := f_init
while not EndCondition
  D := U(D)
  f := B(f)
  f := C(f)
  f := P(f)
  O_i[f]
end
O_f[f]
```

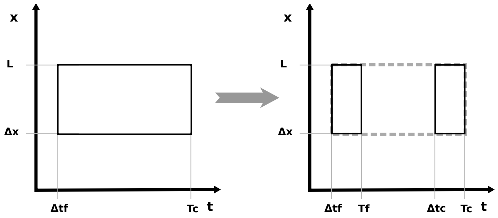
CORAL

```
D := D_init
f := f_init
while not EndCondition
  D := U(D)
  f := B(f)
  f := C(f)
  f := P(f)
  O_i[f]
end
O_f[f]
```

Coral grows due to nutrient brought by water flow



Coupling Speedup : Coral growth



Fluid is computed for

$$\frac{T_f}{\Delta t_f}\frac{T_c}{\Delta t_c} \quad \text{iterations instead of} \quad \frac{T_c}{\Delta t_f}$$

Speedup :

$$S = \frac{\Delta t_c}{T_f} >> 1$$

Classification of problems

- relation in the Scale Separation Map
- single-Domain (sD) or multi-Domain (mD) relation
- coupling templates

SPACE \ TIME	overlap: single domain	overlap: multi domain	separation: single domain	separation: multi domain
overlap	snow transport advection-diffusion ... O_i to S	Fluid-Structure Grid transition ... O_i to B	Forest-Savannah-Fire ... O_i to f_init O_f to S	Coral Growth ... O_i to f_init O_f to B
separation	Algae-Water ... O_i to S	Wave propagation ... O_i to B	Suspension O_i to f_init O_f to S	Bio-Physics Tissue-Fluid O_i to f_init O_f to B

Relation between the scales separation and the coupling templates

We consider two submodels, X and Y with **single-domain** (sD) relation

name	coupling	temporal scale relation
interact	$O_i^X \to S^Y$	overlap
call	$O_i^X \to f_{init}^Y$	X larger than Y
realease	$O_f^Y \to S^X$	Y smaller than X
relay	$O_f^X \to f_{init}^Y$	any

When the relation between computational domains is **multi-domain**, change $S \to B$

Thus, the relation in the SSM determines the workflow

Mathematical formulation of couplings

SEL operators can be used to express coupling strategies and estimate errors

- Time splitting
- Coarse graining
- Amplification
- ...

Time splitting

Assume we have a sD problem with the following SEL

$$P_{\Delta t} C_{\Delta t} = P_{\Delta t} C^{(1)}_{\Delta t} C^{(2)}_{\Delta t}$$

Then if $C^{(1)}_{\Delta t}$ acts at a longer time scale than $C^{(2)}_{\Delta t}$ we may want to approximate

$$[P_{\Delta t} C_{\Delta t}]^M \approx P_{M\Delta t} C^{(1)}_{M\Delta t} [C^{(2)}_{\Delta t}]^M$$

Coarse graining

This strategy consists in expressing a sD problem as

$$[P_{\Delta x} C_{\Delta x}]^n \approx \Gamma^{-1} [P_{2\Delta x} C_{2\Delta x}]^{n/2} \Gamma$$

where Γ is a projection operator (implemented in the smart conduit)

Amplification

We consider a process acting at low intensity but for a long time, in a time periodic environment. For instance a growth process in a pulsatile flow.
We have two coupled (mD) processes which are iterated $n >> 1$ times

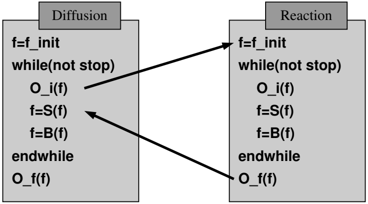
$$[P^{(1)} C^{(1)}]^n \qquad \text{and} \qquad [P^{(2)} C^{(2)}(k)]^n$$

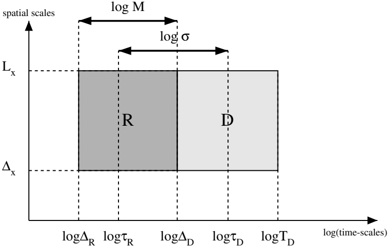
where k expresses the intensity of process $C^{(2)}$.
If the period of process $C^{(1)}$ is $m << n$, we can approximate the above evolution as

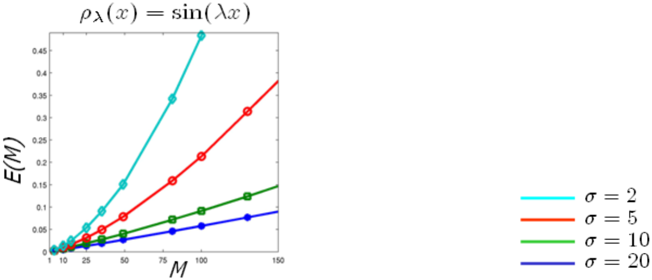
$$[P^{(1)} C^{(1)}]^m \qquad \text{and} \qquad [P^{(2)} C^{(2)}(k')]^m$$

with $k' = (n/m)k$, for a linear process.

Reaction-Diffusion with time splitting

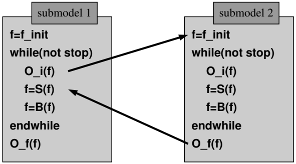






A. Caiazzo, J-L. Falcone, B. Chopard and A. G. Hoekstra, *Asymptotic analysis of Complex Automata models for reaction-diffusion systems*, Applied Numerical Mathematics 59 pp. 2023–2034 (2009)

CxA Execution Model



- Submodels are autonomous processes
- Asynchronous communication through the conduits :
  - Data is written to the conduit as soon as ready.
  - Submodels read the data they need from the conduits (wait if needed).
- Only local interactions are necessary : parallelization is possible and natural
- Propagation of the termination condition

Send-Receive through the conduits

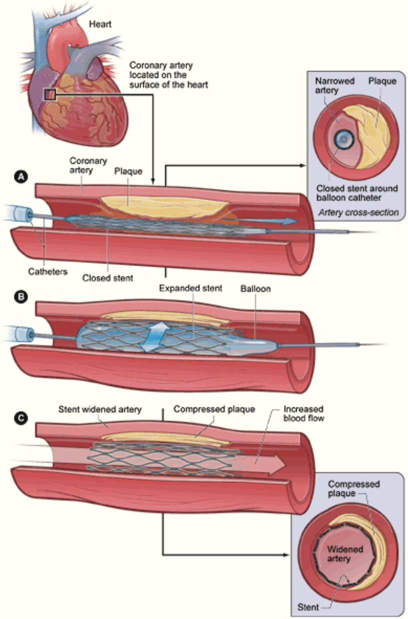
Example of the Coral submodel :

```
  while not EndConditions
     DomainConduit.send(D)
     f := B(f)
     velocityMap := VelocityConduit.receive()
     f :=  S(f,velocityMap)
  end
DomainConduit.stop()
myStop()
```

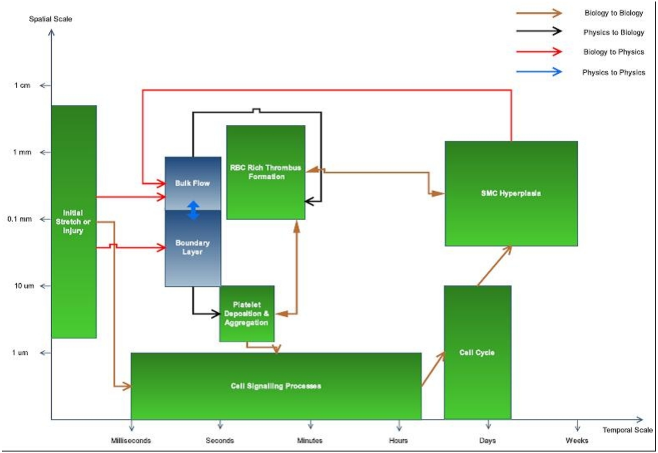
An implementation : the MUSCLE environment (Jan Hegwald, TUB)

- Jade (Java Agent based lightweight middleware) as a platform to build the coupling software.
- More general than the CxA methodology
- Allows us to couple submodels (and legacy codes in C, Fortran).
- A “Jade coordinator” is used to setup the system then goes away,
- Needs a configuration file (CxA file)

Biomedical application : in-stent restenosis



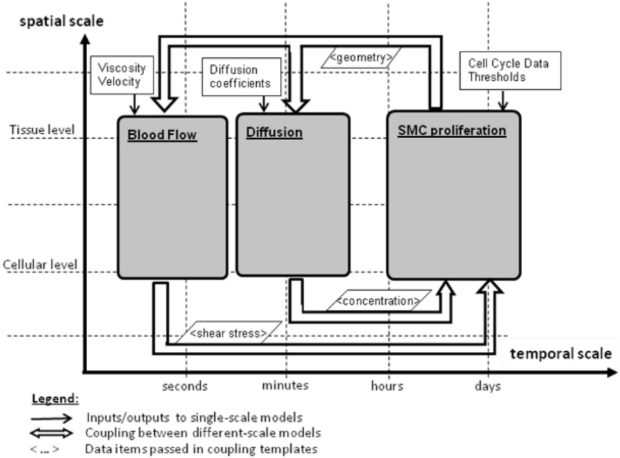
Restenosis : the full Scale Separation Map



Restenosis : Scale Separation Map

- A 3-submodel simplification (time separation is achieved)

D Evans, PV Lawford, J Gunn, D Walker, DR Hose, RH 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. R. Soc. A** 366, pp. 3343–3360, 2008

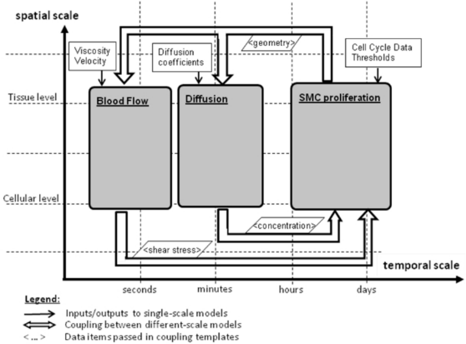


MML : a Multiscale Modeling Language

- the SSM turned out to be very powerful to design applications
- Formalize the methodology into a language : high level representation of a complex multiscale application
- Allows scientists with different backgrounds and geographical locations to better co-develop a multiscale application
- Provide blueprints of a complex multiscale application that can be further augmented by other groups
- Standard for publication
- Automatic execution on a computing resources

MML : a descriptive language

- SSM
- gMML
- xMML



J-L Falcone, B. Chopard and A. Hoekstra, MML : towards a Multiscale Modeling Language, Procedia Computer Science 1 :11, 819-826, 2010

Main ingredients of MML

- Sub-models
- Spatial and temporal scales
- Computational domain relation
- Coupling templates
- Smart conduits

Smart conduits

The coupling between submodels is achieved with three **computational elements**

- **conduit :** one way, point to point
- **filter :** state-full conduit, performing data transformation
- **mapper :** multi-port, data-transformation device.
- Smart conduits can be parametrized and stored in a repository to be reused

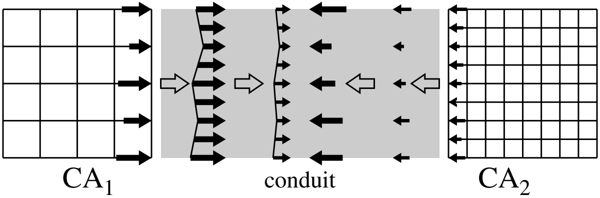
Filter

1. Transfer information between subsystems:

Data elements are sent with timestamps

Filter

2. Write data to conduit, interpolate and rescale:



Data elements are sent with timestamps

Filter

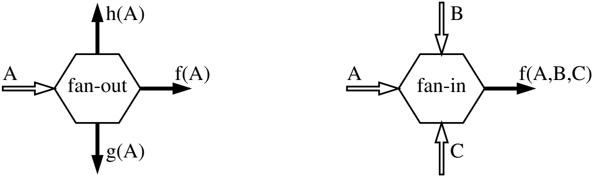
3. Read from conduit the boundary values:

Data elements are sent with timestamps

Mappers

- In principle they are not submodels
- Useful to optimize a coupling (do not repeat twice the same calculation, build complex coupling)

We propose two types of mappers : fan-in and fan-out. The output is produced when all inputs are present

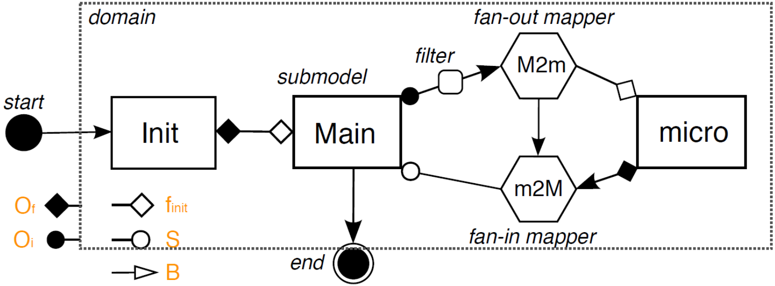


gMML

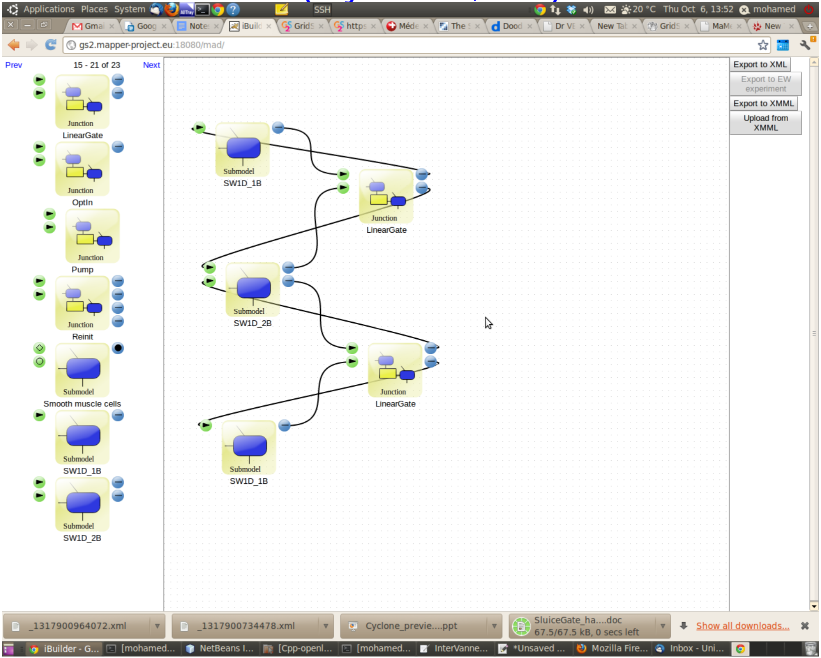
Graphical representation (UML-like)

- Submodel are shown as *rectangles*
- Conduits are shown as *lines* with their *extremities* showing the coupling template (operators in the SEL).
- Filters are shown as *round square* across a conduit
- Mappers are shown as *hexagons*
- Submodel sharing the same computational domain (sD) can be surrounded by a dashed line.

gMML : an example



MAD tool (Cyfronet, PL)



xMML

- XML-like language
- Full description language
- Can be generated from gMML (MAD tool) and vice-versa
- From application description to “glue-code” production and scheduling

xMML example

```
<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>
```

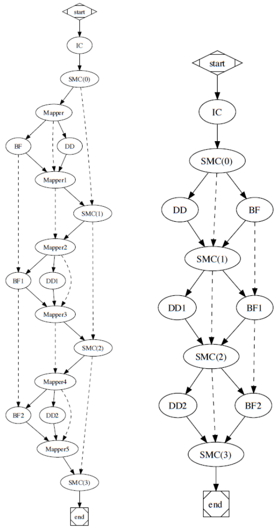
xMML example continued

```
  <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>
</model>
```

From MML to execution

coupling consitency, deadlock detection, automatic scheduling (execution graph)



Multiscale APPlications on European e-infRastructures

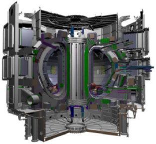
From applications → MML → computing infrastructure

- Running tightly coupled **Distributed Multiscale Applications** using several supercomputing platforms
- Deploy middleware implementing the CxA-MML-MUSCLE approach on the e-Infrastructure (EGI, PRACE, DEISA)

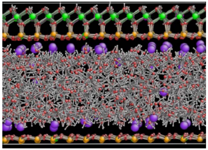
http ://www.mapper-project.eu

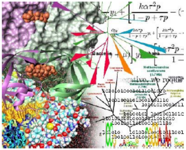
Application portfolio


virtual physiological human


fusion


hydrology

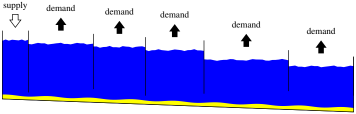

nano material science

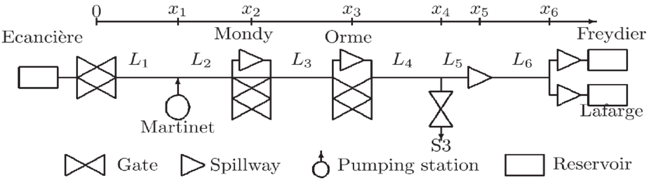

computational biology

- Participants : UvA NL, UCL UK, UU UK, PSNC PL, CYFRONET PL, LMU DE, UNIGE CH, CHALMERS SE, MPG DE

Simulation of irrigation canals

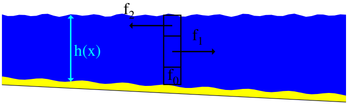
Develop a simulation tools for the optimal management of irrigation canals

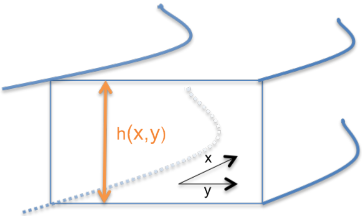


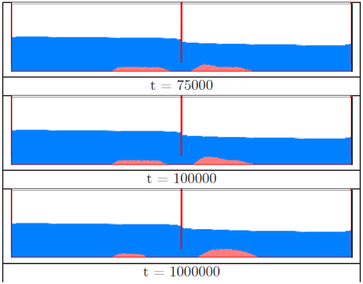


L. Lefèvre, E. Mendes et al. (ESISAR Valence, INP-Grenoble)

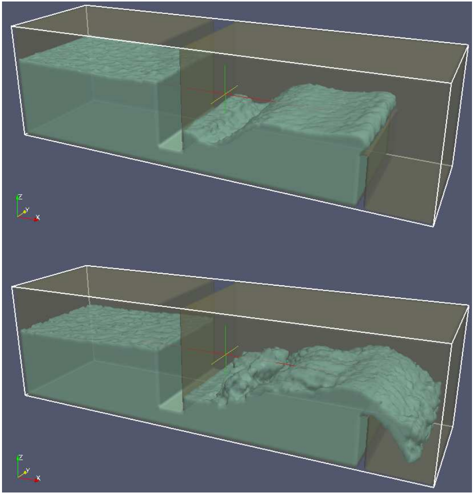
Submodels



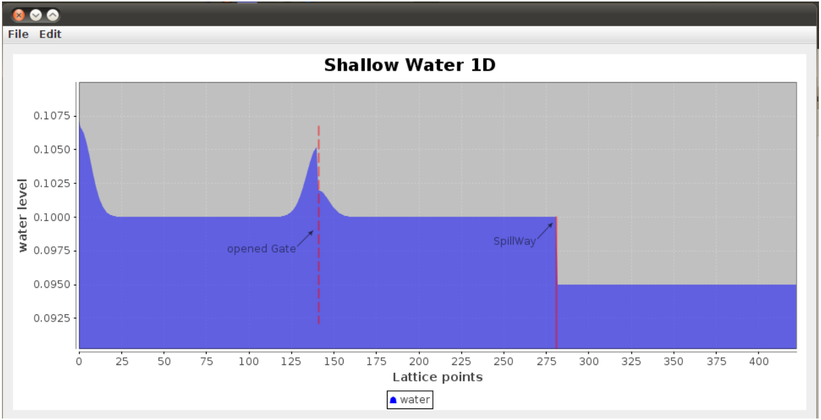




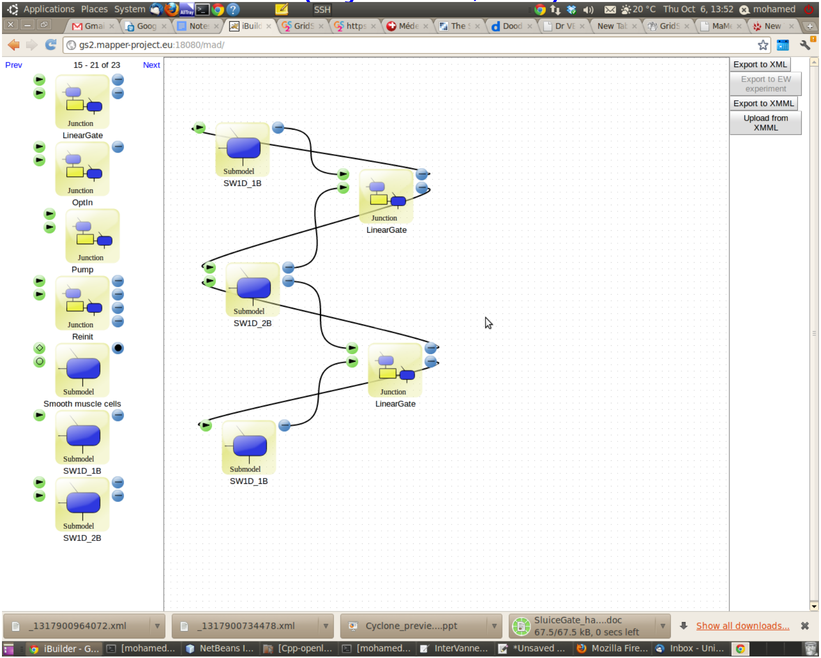
3D, free surface



Coupling 1D SW models (Mohamed Ben Belgacem)



MAD tool (Cyfronet, PL)



Submodel (kernel) and interface to MUSCLE

```
SW1D can1;= new SW1D(L, dx, dt, width, 0.03d);

for (int j = 0; j < nbriteration; j++) {
    can1.collision();
    can1.propagation();

  //Observation: Collects data to send to the Gate
    info = new HashMap<String, Double>();
    info.put("f1", can1.getf1(nx));
    info.put("h", can1.getH()[nx]);
    f_out.send(info);//send the Data to the Gate

  // Boundary: receive the data from the Gate
    double fin = f_in.receive();
    can1.setf2(nx, fin);// update the distribution function

    can1.bounceBack();  // boundary at the left end
      }
```

MUSCLE Coupling script

```
# declare kernels
cxa.add_kernel('SW1D1', 'com.unige.irigcan.kernel.d1.SW1D_1B_kernel')
cxa.add_kernel('SW1D2', 'com.unige.irigcan.kernel.d1.SW1D_2B_kernel')
cxa.add_kernel('SW1D3', 'com.unige.irigcan.kernel.d1.SW1D_1B_kernel')
cxa.add_kernel('Gate', 'com.unige.irigcan.junction.Gate_kernel')
cxa.add_kernel('Spill', 'com.unige.irigcan.junction.Spill_kernel')
```

MUSCLE Coupling script (continued)

```
# configure connection scheme
cs = cxa.cs

cs.attach('SW1D1' => 'Gate') { tie('f_out', 'f1_in')}
cs.attach('SW1D2' => 'Gate') { tie('f_out', 'f2_in')}
cs.attach('Gate' => 'SW1D1') { tie('f1_out', 'f_in')}
cs.attach('Gate' => 'SW1D2') { tie('f2_out', 'f_in')}
#
cs.attach('SW1D2' => 'Spill') { tie('f_out_X', 'f1_in')}
cs.attach('SW1D3' => 'Spill') { tie('f_out', 'f2_in')}
#
cs.attach('Spill' => 'SW1D2') { tie('f1_out', 'f_in_X')}
cs.attach('Spill' => 'SW1D3') { tie('f2_out', 'f_in')}
```

See simulation...

Thank you for your attention