Towards a Complex Automata Framework for Multi-Scale Modeling: Formalism and the Scale Separation Map

Alfons G. Hoekstra[1], Eric Lorenz[1], Jean-Luc Falcone[2], and Bastien Chopard[2]

[1] Section Computational Science, Faculty of Science, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{alfons,lorenz}@science.uva.nl, http://www.science.uva.nl/research/scs
[2] Department of Computer Science, University of Geneva, 24, Rue du Général
Dufour, 1211 Geneva 4, Switzerland
{Bastien.Chopard,Jean-Luc.Falcone}@cui.unige.ch, http://spc.unige.ch/

Abstract. Complex Automata were recently proposed as a paradigm to model multi-scale complex systems. The concept is formalized and the scale separation map is further investigated in relation with its capability to specify the components of Complex Automata. Five classes of scale separation are identified, each potentially giving rise to a specific multi-scale modeling paradigm. A number of canonical examples are briefly discussed.

Key words: Complex Automata, Cellular Automata, Multi-Scale Modeling, Scale Separation Map

1 Introduction

Complex Automata (CxA) were recently proposed as a paradigm to model multi-scale complex systems [1]. The key idea is that a multi-scale system can be decomposed into N single-scale Cellular Automata (CA) that mutually interact across the scales. The decomposition is achieved by building a scale map on which each system can be represented as an area according to its spatial and temporal scales. Processes having well separated scales are easily identified as the components of the multi-scale model.

In this contribution we further elaborate on the CxA framework by first providing a formal definition. Next, we investigate in some detail the concept of the scale separation map, by taking into account the temporal and spatial scales that are covered by a single CA. We thus identify 5 classes of scale separation, each potentially giving rise to specific multi-scale modeling paradigms.

To illustrate our framework we discuss a number of canonical examples, demonstrating how to formulate them as a CxA, how the decomposition into a number of single scale CAs can be obtained, and how these systems are characterized by their mutual positions on the scale separation map.

2 Complex Automata Modeling

2.1 Definition

A CxA is a collection of interacting CA. We refer the reader to standard textbooks for a complete definition of a CA [2]. Here we shall define a CA as a tuple

$$\mathcal{C} =< A(\Delta x, \Delta t, L, T), S, R, G, F > \tag{1}$$

where A is the spatial domain. It is made of cells of size Δx and it spans a region of size L. The quantity Δt is the time step and T is the number of iterations during which the CA will be run. Therefore, processes with time scales between Δt and T can be represented and spatial scales ranging from Δx to L can be resolved.

We call S the set of all possible states of each cell and R the evolution rule. G is the topology describing the neighborhood relation.

At the boundaries of A, additional information is needed (boundary conditions). We define F as the flux of information exchanged at each iteration between the system and its environment.

From definition 1 we can define a CxA as a graph $\mathcal{X} = (V, E)$ where V is the set of vertices and E the set of edges with the following properties

- Each vertex is a CA $\mathcal{C}_i =< A_i(\Delta x_i, \Delta t_i, L_i, T_i), S_i, R_i, G_i, F_i >$
- each edge E_{ij} is a coupling procedure describing the interaction between $\mathcal{C}_i$ and $\mathcal{C}_j$. In practice, E_{ij} will define how information is exchanged between the two subsystems.

During the initialization phase, this graph is built according to the modeler specifications.

2.2 Execution model

Each vertex (subsystem) knows its time step with respect to the global time as well as it spatial location within the whole computational domain. This knowledge is in part given by Δx_i and Δt_i

Edges mediate an asynchronous communication between the vertices they connect. The main idea is that data from one CA that need to be known by another is written in the link that connects them, as soon as it is available. Similarly, when the recipient CA needs new input information from its neighbors, it reads it from the link. If this information is not yet available, the CA has to wait. In this way one achieves a decentralized, asynchronous communication model, which is compatible with parallelization or distributed computing.

The format under which this information is exchanged depends on the type of coupling. Yet, clearly, a conceptual model is needed to find a generic way of exchanging information (with little programming from the users), or to have a simple model to program these links. Smart edges (i.e. edges with computing and communication capabilities) should result from this information exchange procedure.

2.3 Genericity of CA's

The fact that each CA (vertex of the CxA) has a common instruction flow (because it is a CA) gives a way to implement generic coupling mechanisms and achieve the above proposed execution model.

A generic CA can be programmed using the propagation-collision paradigm instead of the gather-update paradigm. In this approach, the following steps implement all possible CA's

```
procedure executeCA
  for time t1 to t2, by step dt
     forall cells
         propagation
         computeBoundary
         collision
     end forall
  end for
end procedure
```

Here we assume that all submodels will be cast in this structure, which is possible if the model complies with the CA definition. Clearly, a lattice Boltzmann (LB)[3] model fulfills this requirement. In an agent based model, the same fundamental operations are also performed with `cells` replaced by `agents`. The propagation procedure sends the local states of each cell to the neighbors that need it. So, the propagation operator assumes an underlying topology of interconnection. In an agent based model, a special agent can be defined as a centralized information repository.

The data structure is a set of cells or agents which is traversed in any order because all the above operation are, in nature, parallel operations.

The `computeBoundary` procedure is needed to specify the values of the variable that are defined by the external environment. In the case of a LB fluid simulation, the missing density distributions at the wall that are computed in this procedure.

`Collision` is the procedure in which the evolution rule is executed for each cell, using the cell values obtained from `propagation` and `computeBoundary`.

2.4 Coupling mechanisms

In the literature, several ways of coupling have been identified. From reference [4], we have

- Sub-Domain coupling **(SDC)**, called domain decomposition in [4]
- Hierarchical-Model coupling **(HMC)**, called heterogenous modeling method (HMM) in [4].

In the SDC adjacent spatial domains are described by different models on space-time grids of possible different resolution. Note that the two adjacent subdomains can possibly overlap.

In HMC, some parameters or variables of a main model (e.g. the CA rule, collision operators) are first computed locally (i.e for some selected cells) on the fly by a finer scale model.

The above SDC and HMC coupling mechanisms can be easily incorporated in the flow structure of `executeCA`, in agreement with the concept of smart edges connecting the CA's together.

For instance, the `propagation` step produces information that is sent to neighboring spatial cells, possibly those belonging a another CA. In case of SDC (subdomain coupling) the outgoing information from the boundary of one system is exaclty what is needed by the adjacent subdomain. Thus `propagation` has to **`write`** the adequate information in the edge interconnecting the two sub-models. Similarly, the procedure `computeBoundary` will **`read`** the missing information from the corresponding edge.

The HMC (hierarchical model coupling) is typically affecting the `collision` procedure by giving the value of a parameter needed by the collision rule and which is computed by a subscale model. For instance, this parameter could be a threshold on the blood shear stress that will lead to blood clotting. Such a threshold would typically result from the own dynamics of the cells in the endothelium.

3 The Scale Separation Map

The Scale Separation Map (SSM) is defined as a two dimensional map with the horizontal axis coding for temporal scales and the vertical axis coding for spatial scales. Each subsystem occupies a certain area on this map. Fig. 1 shows an example of such SSM, in which three subsystems have been identified. Subsystem 1 operates on small spatial scales, and short time scales, process 2 at intermediate scales, and process three at large scales. This could e.g. be processes operating at the micro-, meso-, and macro scale.

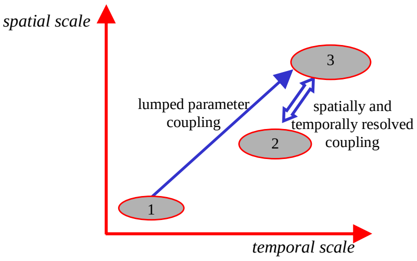
spatial scale
3
lumped parameter coupling
spatially and temporally resolved coupling
2
1
temporal scale

Fig. 1. A scale map, showing three subsystems and their mutual couplings.

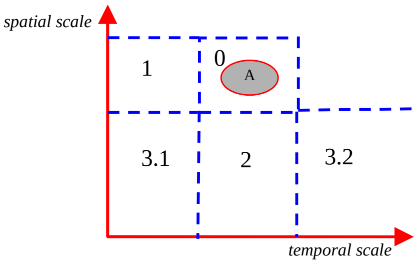

Fig. 2. Interaction regions on the scale map.

Consider two processes A and B with their own specific spatial - and temporal scale, denoted by ξ_i and τ_i respectively ($i \in \{A, B\}$). Assume that A has the

largest spatial scale. In case the spatial scales are the same, A has the largest temporal scale. In other words, $(\xi_B < \xi_A) \quad \text{OR} \quad (\xi_B = \xi_A \quad \text{AND} \quad \tau_B < \tau_A)$. We can now place A on the scale map and then investigate the different possibilities of placing B on the map relative to A. This will lead to a classification of types of multi-scale coupling, as in Fig. 2.
Depending on where B is, we find the following regions:

Region 0: A and B overlap, so we do not have a scale separation, we are dealing here with a single-scale multi-science model.

Region 1: Here $\xi_B = \xi_A \quad \text{AND} \quad \tau_B < \tau_A$, so we observe a separation of time scales at the same spatial scale.

Region 2: Here $\xi_B < \xi_A \quad \text{AND} \quad \tau_B = \tau_A$, so we observe a separation in spatial scales, like coarse and fine structures on the same temporal scale.

Region 3: Separation in time - and spatial scales. Region 3.1 is the well-known micro $\Leftrightarrow$ macro coupling, so fast processes on a small spatial scale coupled to slow processes on a large spatial scale. This type of multi-scale model has received most attention in the literature, and the SDC and HMC coupling paradigms explained earlier have mostly been applied in this region. In region 3.2 we have the reversed situation, a slow process on small spatial scales coupled to a fast process on large spatial scales. We believe that this region is very relevant in for instance coupling of biological with physical processes, where the biological process is e.g. the slow response of cells to a faster physical process on a larger scale (e.g. blood flow in arteries).

Note that we do not have to consider other regions of the scale map, because then the role of A and B just reverses, and we fall back to one of the five cases identified above.

Next we address the question of the area that process A and B occupy on the scale map, and from that, how to quantify the regions $0 - 3$ on the scale map. As discussed earlier, a single scale CA is characterized by a spatial discretization Δx and a system size L, where $\Delta x < \xi < L$. The number of CA cells in the full domain is then $N^{(x)} = L/\Delta x$. We introduce $\delta^{(x)}$ and $\eta^{(x)}$ so that the relevant spatial scale ξ is represented by $10^{\delta^{(x)}}$ cells (i.e. $\Delta x = \xi/10^{\delta^{(x)}}$) and the spatial extension of the CA is $10^{\eta^{(x)}}$ times the spatial scale, i.e. $L = \xi 10^{\eta^{(x)}}$, and therefore $N^{(x)} = 10^{\eta^{(x)}+\delta^{(x)}}$. Likewise for the temporal domain, i.e. a single scale CA has a time step Δt and the CA is simulated over a time span T, and we have $\Delta t < \tau < T$. The number of time steps $N^{(t)} = T/\Delta t$. The discretization has been chosen such that the temporal scale is represented by $10^{\delta^{(t)}}$ time steps (i.e. $\Delta t = \tau/10^{\delta^{(t)}}$) and that simulation time of the CA is $10^{\eta^{(t)}}$times the spatial scale, i.e. $T = \tau 10^{\eta^{(t)}}$ and $N^{(t)} = 10^{\eta^{(t)}+\delta^{(t)}}$.

A process' position on the scale map is now fully determined by the tuple $\{\xi, \delta^{(x)}, \eta^{(x)}; \tau, \delta^{(t)}, \eta^{(t)}\}$, and is drawn in Fig. 3, where the axes are now on a logarithmic scale. On such logarithmic SSM the process is rectangular with area $(\delta^{(t)} + \eta^{(t)}) \times (\delta^{(x)} + \eta^{(x)})$ asymmetrically centered around the point $(\log(\tau), \log(\xi))$.

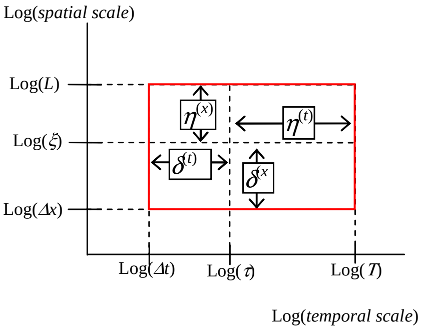

Fig. 3. Position of a process with parameters $\{\xi, \delta^{(x)}, \eta^{(x)}; \tau, \delta^{(t)}, \eta^{(t)}\}$ on the logarithmic scale map.

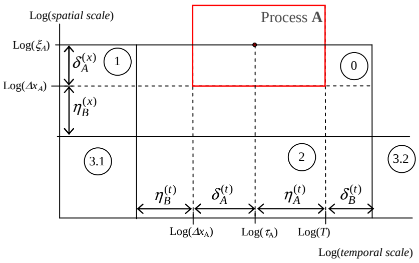

Fig. 4. Interaction regions on the logarithmic scale map, in more detail.

In the special case that $\delta^{(x)} = \eta^{(x)} = \delta^{(t)} = \eta^{(t)} = 1$ (a reasonable first order assumption) we see that the process is symmetrically centered around $(\log(\tau), \log(\xi))$ and that the size of the box extends 2 decades in each dimension.

In Fig. 4 we show the extension of Fig. 2, where regions $1 - 3$ now have well defined positions and size. Depending on the location of process B, that is the point $(\log(\tau_B), \log(\xi_B))$ on the SMM, and with all information on the spatial and temporal extensions of process A and B, we can unambiguously find in which region of the scale map they are located with respect to each other.

Consider once more region 3. In region 3.1 we find that $L_B < \Delta x_A$ and $T_B < \Delta t_A$. As said earlier, this is the classical micro ⇔ macro coupling, and in our language this means the full spatio-temporal extend $T_B \times L_B$ of process B is smaller than one single spatio-temporal step $\Delta t_A \times \Delta x_A$ of process A.

Region 3.2 also exhibits separation of time and length scales, but now the situation is quite different. We find that, just like in region 3.1, $L_B < \Delta x_A$. So, the spatial extend of process B is smaller than the grid spacing of process A. However, now we find that $T_A < \Delta t_B$. In other words, the full time scale of process A is smaller then the time step in process B. This will result in other modeling and simulation paradigms than in region 3.1. Typically, the coupling between A and B will involve time averages of the dynamics of the fast process A.

Let us now turn our attention to the regions where there is overlap on the temporal - or spatial scales, or both (regions 0, 1, and 2, in Fig. 4). In all these cases we can argue that we have partial or full overlap of the scales, giving rise to different types of (multi-scale) modeling and simulation. We say that the scales fully overlap if the point $(\log(\tau_B), \log(\xi_B))$ falls within (one of) the scales spanned by process A. On the other hand, there is partial overlap if $(\log(\tau_B), \log(\xi_B))$ falls *outside* (one of) the scales spanned by process A, but the rectangular area of process B still overlaps with (one of) the scales spanned by process A. The region of partial scale overlap can also be considered as a region of gradual scale separation, a boundary region between the scale separated regions 1, 2 and 3 and region 0. Simulations of this kind of multi-scale system would typically involve CxA's with local grid refinements, or multiple time stepping approaches, or a combination of both. We save this more detailed discussion for a future publication.

4 Examples

In this section we briefly indicate how some problems can be expressed as a CxA, with the benefit of code reusability, modeling flexibility and design efficiency.

4.1 Grid refinement

The problem of grid refinement in LB fluid simulations is important in order to resolve small spatial scales, for instance around boundaries. The two subdomains C and F are the regions where a coarse and fine grid are defined, respectively. Such a situation is described in more detail in [5]. The two regions are updated by applying a LB (i.e a CA) dynamics such as `executeCA`. In addition, they interact because C and F have a common spatial interface, at the boundary between the two regions. Thus, the coupling is of SCD type. The CxA edge connecting the two submodels consists of transfering the information for the cells at the interface, from one grid to the other. In grid refinement the grid spacing is typically refined by some small number (say 2 or 4), and the timestep is scaled accordingly. So, in the language of the SSM, processes C and F will have partially overlapping spatial and temporal scales.

Typically, the edge will collect the data from grid C, rescale it, then perform a space and time interpolation in order to provide grid F with a data set at scales Δx_F and Δt_F. Similarly, the same edge will also translate the data produced by F to the requirement of grid C.

From the point of view of grids C and F the coupling is achieved simply by writing and reading the specific buffers managed by the edge, thus fully decoupling the model part from its interaction with other system components.

4.2 Time splitting in Reaction-Diffusion

Time splitting has been introduced in LB models for reaction-diffusion [6] in order to deal with a wide range of reaction and diffusion constants in the same simulation. The idea is to consider diffusion as a first, larger scale model. Reaction acts between the diffusion steps, at each spatial cell, with a finer time resolution than the diffusion process. The reaction process is modeled as an implicit solver for a local differential equation. So we have here a HMC type of coupling. In this case, the subsystem coupling edge simply consists in reading the input chemical concentrations given by the diffusion process and pass them to the reaction solver. The new concentrations are then written back to the cell for the next diffusion step. Depending on the characteristic time scales of the diffusion (specified by the diffusion constants) and the reactions (specified by the rate constants), the CxA is has fully separated time scales (region 1 on the SMM), or the time scales are partially overlapping.

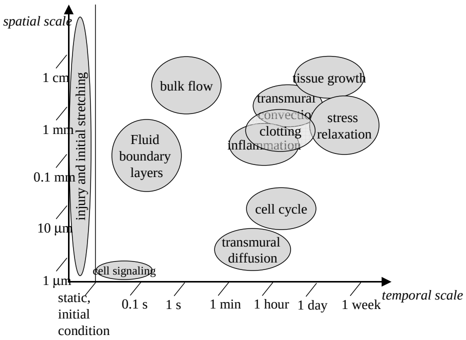

Fig. 5. A SSM for in stent re-stenosis.

4.3 In-Stent Restenosis

Within the EU funded COAST project [7] we have chosen the treatment and progression of coronary artery disease as a prototypical multi-scale multi-science complex system to be modeled within the CxA framework. We will address the adverse vessel wall remodelling (re-stenosis), which occurs in some patients after placement of a metal frame (stent) within the artery lumen to expand and support the vessel at the site of a stenosis. This restenosis is due to the growth of scar tissue in between the strut of the stent, which tends to block the lumen. A full description of this application is beyond the scope of this manuscript. Here, it serves as an illustrative example of how we plan to apply CxA modeling. In-Stent Restenosis involves a large number of biological and physical processes on many spatial and temporal scales. Fig. 5 shows a simplified SMM for this process, showing relevant physical processes (such as bulk flow, transmural diffusion) and biological processes (such as cell signaling, inflammation and clotting) on their (estimated) characteristic scales. We are currently in the process of defining single scale CA or agent-based models for all processes, and once this is available, we will cast this SMM into the form of Fig. 3 and 4.

5 Discussion and Conclusions

The simulation of multiscale, multiscience complex systems are a central challenge in computationa science. Cellular Automata are a powerful framework to model spatially extended dynamical systems. In order to cope with multiscale and multiphysics, we have introduced Complex Automata as a set of interacting CA's. Together with a description of system components through a scale separation map, CxA offer a flexible and intuitive framework to solve problems in which several different physical processes at different spatial and temporal scales interact.

Acknowledgments. This research is supported by the European Commission, through the COAST project [7] (EU-FP6-IST-FET Contract 033664)

References

1. Hoekstra, A., Chopard, B., Lawford., P., Hose, R., Krafczyk, M., and Bernsdorf, J.: Introducing Complex Automata for Modelling Multi-Scale Complex Systems. In Proceedings of the European Conference on Complex Systems ECCS'06, ISBN 0-9554123-0-7 (2006).
2. Chopard, B. and Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press (1998).
3. Succi, S.: The Lattice Boltzmann Equation, for Fluid Dynamics and Beyond. Oxford University Press (2001).
4. E, W., Engquist, B., Li, X, Ren, W., and Vanden-Eijnden, E.: Heterogeneous Multiscale Methods, A Review. Commun. Comput. Phys., 2, 367-450 (2007).
5. Dupuis, A. and Chopard, B.: Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. Phys. Rev. E, 67, 066707 (2003).
6. Alemani, D., Chopard, B., Buffle, J. and Galceran, J.: LBGK method coupled to time splitting technique for solving reaction-diffusion processes in complex systems. Physical Chemistry and Chemical Physics 7, 3331–3341 (2005).
7. http://www.complex-automata.org/