Multi-scale Modeling with Cellular Automata: The Complex Automata Approach

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

[1] Section Computational Science, Faculty of Science, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{alfons,acaiazzo}@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. Cellular Automata are commonly used to describe complex natural phenomena. In many cases it is required to capture the multi-scale nature of these phenomena. A single Cellular Automata model may not be able to efficiently simulate a wide range of spatial and temporal scales. It is our goal to establish a Cellular Automata modeling paradigm for multi-scale processes. Here we will demonstrate that Complex Automata, a paradigm that we recently introduced, are capable to facilitate such modeling.

Keywords: Multi-Scale Modeling, Complex Automata.

1 Introduction

Cellular Automata (CA) are generally acknowledged to be a powerful way to describe and model natural phenomena. [1,2,3] There are even tempting claims that Nature itself is one big (quantum) information processing system, e.g. [4], and that CA may actually be nature's way to do this processing [5,6,7]. We will not embark on this philosophical road, but ask ourselves a more mundane question. Can we use CA to model the inherently multi-scale processes in nature *and* use these models for efficient simulations on digital computers?

The ever increasing availability of experimental data on every scale, from 'atom to material' or from 'gene to health', in combination with the likewise ever increasing computational power [8,9], facilitates modeling and simulation of natural phenomena taking into account all required spatial and temporal scales (see e.g. [10]). Multi-scale modeling and simulation, as a paradigm in Computational Science, is becoming more and more important, as witnessed by e.g. dedicated special issues [11] and thematic journals [12,13].

When using CA to model a natural process, the underlying lattice and one iteration of the CA have a clear physical meaning in terms of lattice spacing

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 192–199, 2008.

and time step. We denote by $A(\Delta x, \Delta t, L, T)$ the spatio-temporal domain of a CA, whose spatial domain is made of cells of size Δx and it spans a region of size L, while the quantity Δt is the time step and T is the end of the simulated time interval. Therefore, processes with time scales between Δt and T can be represented and spatial scales ranging from Δx to L can be resolved. When executing such CA on a digital computer we note that the execution time T_{ex} scales as

$$T_{ex} \sim \frac{T}{\Delta t}\left(\frac{L}{\Delta x}\right)^{D}, \qquad (1)$$

where D is the spatial dimension of the simulated domain. Trying to model a multi-scale system with a single CA would require to choose Δx and Δt in such a way that the smallest microscopic details and fastest dynamical response of the system are captured, yet the overall system size (L) and slowest dynamical time scale (T) need to be covered. For instance, in modeling human physiology the relevant range of spatial scales is from nanometer to meter (i.e. a factor 10^9) whereas temporal scale is from microseconds to human lifetime (i.e a factor 10^{15}). Such numbers, in combination with Eq. 1 immediately show that one will never be able to simulate multi-scale systems with a single CA spanning a wide range of scales.

In this paper we propose Complex Automata (CxA), a set of single scale CA representing processes operating on different spatio-temporal scales, supplied with adequate coupling templates between the scales, as a CA-based methodology to model and simulate multi-scale phenomena. These CA are typically based on Lattice Boltzmann Models (LBM) [1,3] or other generalized CA. Besides generalized CA to represent single scale models, the CxA approach also includes Agent Based Models (ABM).

2 Multi-scale Cellular Automata

The literature on using Cellular Automata to model multi-scale phenomena is relatively small, maybe with the exception of using CA to model land usage and geographical systems (e.g. [14]). Furthermore, many papers exist that use CA in multi-scale modeling, but there CA is typically coupled to other types of models (e.g. [15]).

The bulk of CA multi-scale attempts are grid refinement methods, also termed multi-blocks. The idea is to adapt the local grid size to the local process scale, i.e. using a fine grid in regions where small scale processes occur and a coarse grid where larger scales are sufficient. A common approach is to couple grids of different scales with an overlap region [16]. To exchange information from the coarse grid to the fine grid, an interpolation scheme must be used. However, as both space and time scale are different, interpolation must be spatial and temporal. To allow a smooth coupling, a re-scaling of the variables is is also required (e.g. [17]). Another interesting way to couple a coarse and fine grid for a CA was proposed by Weimar [18]. When the two grids are not overlapping, each grid must provide boundary conditions for the other one. The information provided by the fine grid to the coarse grid is simply the averaging of the micro cell adjacent to a macro cell. However, to reconstruct the lacking information at microscopic boundaries, the author proposes an iterative statistical scheme to ensure the information flowing in each boundary micro cell exhibit the same correlations than the information flowing between micro cells.

Other ways of coupling multi-scale CA come from two theoretical frameworks. The first one is based on higher-order CA [19]. In this framework, the CA rules are not only able to change the cell state, but also the rules themselves, the neighborhood and the topology. Moreover, these models are also able to take in account hierarchical CA where higher level cells are connected to one or more lower level cells. The second one results from the work of Israeli and Goldenfeld [20] who have shown that it is possible to coarse-grain 1D nearest-neighbor CA, by defining a macroscopic CA whose behavior is similar to a microscopic CA. That is an important result because the authors have achieved the coarse-graining of CA known to be irreducible.

3 Complex Automata

Complex Automata (CxA) were recently introduced as a modeling paradigm for multi-scale systems using CA, LBM and Agent Based Models (ABM) as the building blocks [21,22]. The key idea behind CxA is that a multi-scale system can be decomposed into N single-scale Cellular Automata that mutually interact across the scales. The decomposition is achieved by building a Scale Separation Map (SSM) 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.

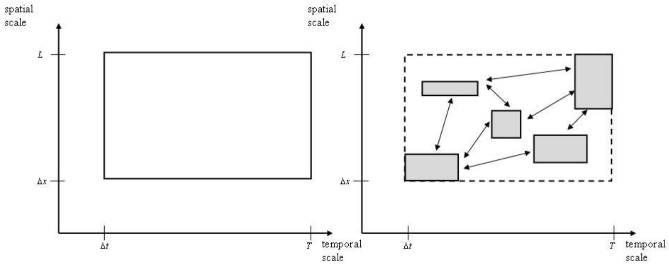


Fig. 1. The Scale Separation Map with left a single CA and right a hypothetical CxA modeling the same process

Fig. 1 shows a SSM, where the horizontal axis represents the temporal scales and the vertical axis the spatial scales. On the left a CA with spatio-temporal domain $A(\Delta x, \Delta t, L, T)$ is represented on the SSM. Assuming that the process to be simulated is really multi-scale in the sense that it contains relevant subprocesses on a wide range of scales, simulations based on the finest discretizations are not really feasible (Eq. 1), the approach we propose in CxA modeling is to try to split the original CA into a number of single-scale CA and let these CA exchange information in such a way that the dynamical behavior of the multi-scale process is mimicked as accurate as possible. This is shown schematically in right part in Fig. 1. In [22] we explain in much more detail how this can be achieved and provide a large number of examples. Here we will continue to introduce a more formal definition of a CxA, providing argument why we believe that the CxA approach *does* represent a possible way to construct CA-based models of multi-scale systems.

3.1 Preliminary Definitions

Formally, a Cellular Automaton can be defined as

$$\mathrm{CA} = \{(\varDelta x, L, \varDelta t, T), \mathbb{F}, \varPhi, f_{init} \in \mathbb{F}, \mathbf{u}\} \tag{2}$$

specifying the spatial-temporal domain with discretization parameters $(\varDelta x, \varDelta t)$, the space of states $(\mathbb{F})$, the initial condition (f_{init}) and the update rule $(\varPhi)$. In Eq. (2) we include in the definition a field $\mathbf{u}$, collecting the external data the CA depends on. The state of the system is described by a $\hat{f}^{t_n} \in \mathbb{F}$, denoting the numerical solution at the n-th time step:

$$\begin{aligned} &\hat{f}^0 = f_{init}[u_0], \text{ initial condition} \\ &\hat{f}^{t_n+\varDelta t} = \varPhi[\mathbf{u}; \hat{f}^n], \quad n > 0 \end{aligned} \tag{3}$$

with u_0 an external field connected to the initial condition.

Additionally, restricting to CA, LBM or ABM, we constrain the update rule $\varPhi$ to the form

$$\varPhi[\mathbf{u}; f] = \left(B[u_B] \circ P \circ C[u_C]\right)[f], \tag{4}$$

i.e. written as a composition of three operators: *collision* $C[u_C]$, depending on external parameters u_C, *propagation* P, depending on the topology of the domain, and *boundary condition* $B[u_B]$, depending on external parameters denoted by u_B. This form of a CA update rule is inspired by Lattice Gas Automata (see e.g. [1,3]), and applied in defining coupling templates for CxA [22]. Note that Chopard et al. discuss this in more detail in another manuscript submitted to ACRI08.

In what follows, we let the definition of CA (2) depend on a (small) parameter h, related to spatial and temporal discretizations (for example $\varDelta x_h = h$, $\varDelta t_h = \alpha h$). Accordingly, the evolution space and the update rule depend on h as well: $\varPhi_h : \mathbb{F}_h \to \mathbb{F}_h$. Shortly, we will call $\hat{f}_h$ the numerical outcome of the CA_h.

3.2 Complex Automata Formalism

To begin with, as in the left diagram in Fig. 1, we consider a multi-scale system represented as a single CA_h defined as in (2). Building a CxA, instead of describing the system with a single $\hat{f}_h$, we lower the dimension of the problem and the computational complexity, introducing a set of discretization parameters $H = (h_1, \ldots, h_M)$ and building a corresponding Complex Automaton $\mathrm{CxA}_H = (\mathrm{CA}_{h_1}, \ldots, \mathrm{CA}_{h_M})$, where each CA_{h_m} is an object as in (2). Similarly as before, we denote with $\hat{f}_H$ the numerical outcome of the complex automata simulation and $\hat{f}_{h_m}$ the state variable of the single CAs. Note that for the evolution space of a CxA it holds $\mathbb{F}_H^{CxA} \subset \mathbb{F}_{h_1} \times \ldots \times \mathbb{F}_{h_M}$. In fact, part of the single scale evolution spaces could be shared by several Automata, in case of space overlap and single domain coupling [22].

3.3 Coupled Evolution of a CxA

For the sake of simplicity, we describe the formalism restricting to the evolution of two coupled single scale models. From equations (3)-(4), we have the following general representation

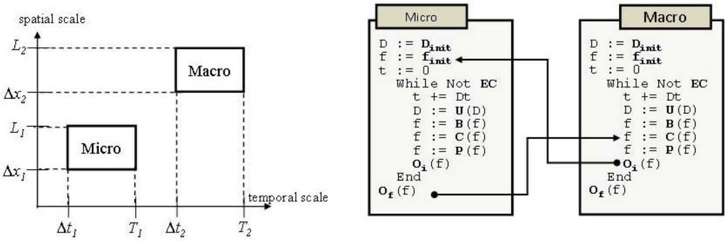
$$
\begin{aligned}
&\hat{f}_{1,h_1}^{t_{1,0}} = f_{init,1}[\hat{f}_2] \\
&\hat{f}_{1,h_1}^{t_{1,n_1}+\Delta t_{h_1}} = \Big(B_{h_1}[\hat{f}_2] \circ P_{h_1} \circ C_{h_1}[\hat{f}_2]\Big)\,[\hat{f}_{h_1}^{t_{1,n_1}}], \quad n_1 > 0 \\
&\\
&\hat{f}_{2,h_2}^{t_{2,0}} = f_{init,2}[\hat{f}_1] \\
&\hat{f}_{2,h_2}^{t_{2,n_2}+\Delta t_{h_2}} = \Big(B_{h_2}[\hat{f}_1] \circ P_{h_2} \circ C_{h_2}[\hat{f}_1]\Big)\,[\hat{f}_{h_2}^{t_{2,n_2}}], \quad n_2 > 0
\end{aligned}
\tag{5}
$$

where two CAs are fully coupled in all the components. In detail,

- $f_{init,1}[\hat{f}_2]$ denotes a coupling through initial conditions (i.e. the initial condition of 1 depends on the results of 2)
- $B_{H_1}[\hat{f}_2]$ expresses coupling through boundary conditions,
- $C_{H_1}[\hat{f}_2]$ expresses the coupling through collision operator.

In general, for different situations (multidomain/singledomain, time/space separation/overlap) we can restrict the set of possible couplings to a well-specified coupling template. [22] Consider the example of a microscopic fast process coupled to a macroscopic slow process (micro-macro coupling). The macroscopic process takes input from explicit simulations of microscopic processes at each time step and on each lattice site of the macroscopic process. The microscopic processes run to completion, assuming that they are much faster than the macroscopic process and therefore are in quasi-equilibrium on the macroscopic time scales (this approach is known in the literature as the Heterogeneous Multi-scale Method, see [23]). The macroscopic process could e.g. be a fluid flow with takes its viscosity from an underlying microscopic process (e.g. explicit suspension model).

In Fig. 2 we show for this example of micro-macro coupling the SMM (left) and the coupling template (right). The later is defined in [22] and shows how the operators as defined in (4) are coupled to each other. A close inspection of this coupling template shows indeed that upon each iteration of the macroscopic process the microscopic process executes a complete simulation, taking input



```
Micro
D := D_init
f := f_init
t := 0
  While Not EC
    t += Dt
    D := U(D)
    f := B(f)
    f := C(f)
    f := P(f)
    O_i(f)
  End
O_f(f)
```

```
Macro
D := D_init
f := f_init
t := 0
  While Not EC
    t += Dt
    D := U(D)
    f := B(f)
    f := C(f)
    f := P(f)
    O_i(f)
  End
O_f(f)
```

Fig. 2. Micro-macro coupling example. Left: SSM. Right: coupling template.

from the macroscopic process. In turn, the output from the microscopic process is fed into the Collision operator of the macroscopic process.

We can formulate the CxA dynamics as follows (based on Eq. (5))

$$\begin{aligned}
\hat{f}_{1,h_1}^{t_{1,0}} &= f_{init,1}[\hat{f}_2] \\
\hat{f}_{1,h_1}^{t_{1,n_1}+\Delta t_{h_1}} &= (B_{h_1} \circ P_{h_1} \circ C_{h_1})\,[\hat{f}_{1,h_1}^{t_{1,n_1}}], \quad n_1 > 0 \\
\\
\hat{f}_{2,h_2}^{t_{2,0}} &= f_{init,2} \\
\hat{f}_{2,h_2}^{t_{2,n_2}+\Delta t_{h_2}} &= \left(B_{h_2} \circ P_{h_2} \circ C_{h_2}[\hat{f}_1]\right)[\hat{f}_{h_2}^{t_{2,n_2}}], \quad n_2 > 0
\end{aligned} \tag{6}$$

where 1 refers to the micro-scale and 2 to the macro-scale. As drawn in Fig. 1, the micro-scale model 1 is run until completion (i.e. until the final time T_1), then a single time step Δt_{h_2} is performed for the macro-scale model.

We can now compare an estimation of the execution time of the CxA model of Fig. 2 with that of using a single CA for the same system, as in the left part of Fig. 1. For the single CA the execution time would be $T_{CA} = k_{CA}\frac{T_2}{\Delta t_1}\left(\frac{L_2}{\Delta x_1}\right)^D$, which is Eq. (1) using the subscripts as introduced in Fig. 2. For the CxA the execution time becomes

$$T_{CxA} = k_2 \frac{T_2}{\Delta t_2}\left(\frac{L_2}{\Delta x_2}\right)^D \left(1 + k_1 \frac{T_1}{\Delta t_1}\left(\frac{L_1}{\Delta x_1}\right)^D\right). \tag{7}$$

Note that k_{CA}, k_1 and k_2 are constant numbers. The execution time is determined by the total number of iterations of the macro scale process, and the time per iteration is in turn determined by on the one hand the macro scale operators and on the other hand the time needed for the micro scale CA simulations. Next one can compute a speedup, comparing the single scale CA formulation and the CxA formulation as $S = T_{CA}/T_{CxA}$. After some algebra we find

$$S = \left(\frac{k_{CA}}{k_2}\frac{\Delta t_2}{\Delta t_1}\left(\frac{\Delta x_2}{\Delta x_1}\right)^D\right) \Big/ \left(1 + k_1\frac{T_1}{\Delta t_1}\left(\frac{L_1}{\Delta x_1}\right)^D\right). \tag{8}$$

Under the reasonable assumption that the execution time for a full micro scale simulation needs much more time than a single iteration of the macro scale model, i.e. when $k_1 \frac{T_1}{\Delta t_1} \left(\frac{L_1}{\Delta x_1} \right)^D >> 1$, Eq. (8) reduces to $S = \frac{k_{CA}}{k_1 k_2} \frac{\Delta t_2}{T_1} \left(\frac{\Delta x_2}{L_1} \right)^D$. Note that $\frac{\Delta t_2}{T_1} > 1$ and $\frac{\Delta x_2}{L_1} > 1$, and can be interpreted as the distance on the SSM (Fig. 2). So, if the scale separation is large enough, the obtained speedups can be huge, principally rendering a CxA simulation feasible.

3.4 Error Analysis

The previous arguments demonstrate the improvements in computational efficiency offered by the CxA formulation. On the other hand, replacing the original multi-scale model with many coupled single-scale algorithms, we face a partial loss of precision.

A possible measure of this lowering in accuracy can be obtained considering the difference in the numerical results of he original CA_h and the Complex Automaton CxA_H, which we call *scale-splitting error* [24].

In general, a detailed and rigorous investigation of the scale-splitting error requires a good base knowledge of the single scale Automaton and of the full multiscale algorithm. An example of error investigation using the formalism introduced in section 3.2 for a simple CxA model can be found in [24] and another contribution submitted to ACRI 2008. [25]

4 Conclusions

In this manuscript we have introduced a methodology for CA based multi-scale modeling and simulation. Our approach, Complex Automata, relies on the possibility to split a multi-scale model into a set of mutually couple single scale CAs. This manuscript introduces a formalism for CxA modeling and presents a few partial results for the case of micro-macro coupling. Development of CxA methodology is an ongoing project, and in the near future we intend to publish a more concise formalism, containing a larger set of examples.

Acknowledgments. This research is supported by the European Commission, through the COAST project (EU-FP6-IST-FET Contract 033664, www.complex-automata.org).

References

1. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press, Cambridge (1998)
2. Deutsch, A., Dormann, S.: Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis. Birkhäuser, Basel (2005)
3. Sloot, P.M.A., Hoekstra, A.G.: Modeling Dynamic Systems with Cellular Automata. In: Fishwick, P. (ed.) Handbook of Dynamic System Modeling, ch. 21, Chapman & Hall/CRC (2007)
4. Lloyd, S.: Computational Capacity of the Universe. Phys. Rev. Lett. 23, 237901 (2002)

5. Zuse, K.: The Computing Universe. Int. J. Theor. Phys. 21, 580–600 (1982)
6. Zuse, K.: Rechnender Raum, `http://www.idsia.ch/~juergen/digitalphysics.html`
7. Wolfram, S.: A New Kind of Science. Wolfram Media, Inc. (2002)
8. Bader, D.A. (ed.): Petascale Computing: Algorithms and Applications. Chapman & Hall/CRC (2008)
9. Hoekstra, A.G., Portegies Zwart, S., Bubak, M., Sloot, P.M.A.: Towards Distributed Petascale Computing. In: Bader, D.A. (ed.) Petascale Computing: Algorithms and Applications, ch. 8, Chapman & Hall/CRC (2008)
10. Sloot, P.M.A., Frenkel, D., van der Vorst, H., et al.: White paper on computational e-science, studying complex systems in silico, a national research invitiative (2007), `http://www.science.uva.nl/research/pscs/papers/archive/Sloot2007a.pdf`
11. Special issue on Multiphysics modeling. IEEE Comput. Sci Eng. 7, 14–53 (2005)
12. SIAM Multiscale Modeling and Simulation, `http://epubs.siam.org/sam-bin/dbq/toclist/MMS`
13. International Journal for Multiscale Computational Engineering, `http://www.edata-center.com/journals/61fd1b191cf7e96f.html`
14. White, R.: Modeling Multi-scale Processes in a Cellular Automata Framework. In: Portugali, J. (ed.) Complex Artificial Environments, Simulation, Cognition and VR in the Study and Planning of Cities, pp. 165–177. Springer, Heidelberg (2006)
15. Ribba, B., Alarcón, T., Marron, K., Maini, P.K., Agur, Z.: The Use of Hybrid Cellular Automata Models for Improving Cancer Therapy. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 444–453. Springer, Heidelberg (2004)
16. Lin, C., Lai, Y.G.: Lattice Boltzmann method on composite grids. Phys. Rev. E 62, 2219–2225 (2000)
17. Dupuis, A., Chopard, B.: Theory and Applications of an Alternative Lattice Boltzmann Grid Refinement Algorithm. Phys. Rev. E 67, 066707 (2003)
18. Weimar, J.R.: Coupling Microscopic and Macroscopic Cellular Automata. Int. J. of Parallel Computing 27, 601–611 (2001)
19. Baas, N.A., Helvik, T.: Higher Order Cellular Automata. Advances in Complex Systems 8, 169–192 (2005)
20. Israeli, N., Goldenfeld, N.: Computational Irreducibility and the Predictability of Complex Physical Systems. Phys. Rev. Let. 92, 74105 (2004)
21. Hoekstra, A.G., Lorenz, E., Falcone, J.-L., Chopard, B.: Towards a Complex Automata Framework for Multi-scale Modeling: Formalism and the Scale Separation Map. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4487, pp. 922–930. Springer, Heidelberg (2007)
22. Hoekstra, A.G., Lorenz, E., Falcone, J.-L., Chopard, B.: Toward a Complex Automata Formalism for MultiScale Modeling. Int. J. Multiscale Comput. Eng. 5, 491–502 (2007)
23. W., E., Engquist, B., Li, X., Ren, W., Vanden-Eijnden, E.: Heterogeneous Multiscale Methods: A Review. Commun. Comput. Phys. 2, 367–450 (2007)
24. Caiazzo, A., Falcone, J.L., Chopard, B., Hoekstra, A.G.: Scale-splitting error in Complex Automata models for Reaction-Diffusion systems. In: Bubak, M., et al. (eds.) ICCS 2008, part II. LNCS, vol. 5102, pp. 291–300. Springer, Heidelberg (2008)
25. Caiazzo, A., Falcone, J.L., Chopard, B., Hoekstra, A.G.: Error Investigations in Complex Automata models for Reaction-Diffusion systems. In: Umeo, H., et al. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 260–267. Springer, Heidelberg (2008)