

A uniform multiscale meta-model of BIOSHAPE

F. Buti, D. Cacciagrano, F. Corradini, E. Merelli, L. Tesei¹

*School of Science and Technology, Computer Science Division
University of Camerino
Via Madonna delle Carceri 9, 62032 Camerino, Italy*

Abstract

Existing approaches in multiscale (MS) science and engineering have evolved from a range of ideas and solutions that are reflective of their original problem domains. As a result, research in MS science has followed widely diverse and disjoint paths, which present a barrier to cross pollination of ideas and application of methods outside their application domains.

The status of the research environment calls for a methodological framework able to (i) provide a *common* language to modelling and simulating MS problems across a range of scientific and engineering disciplines and, consequently, (ii) characterize critical common issues arising in MS problems in an *uniform* setting.

In this paper, we contribute in this sense. Taking inspiration from the Complex Automata (CxA) MS approach, we formally define and enrich the meta-model of BIOSHAPE - put forward as a scale-independent MS simulation environment - and we exploit it to give a uniform treatment of generally defined coupling schemes, in particular the micro-macro one applied to the bone remodelling process. Similarly to CxA, also the BIOSHAPE meta-model enjoys two important features: namely, (i) a MS system can be decomposed in uniform single-scale models, each one described by a generic sequence of calls to well-defined operators, and (ii) the link between any two single-scale models can be expressed as a flow of data between a pair of these operators by well-defined coupling schemes. As a consequence, such features not only enforce and formally prove the scale-independence property of the BIOSHAPE simulator, but also makes the BIOSHAPE meta-model a common and uniform MS modelling paradigm across a range of heterogeneous application domains.

Keywords: Multiscale modelling, Multiscale simulation, Bone remodelling

1 Introduction

Although computational methodologies have improved vastly over the last ten years, it has become blatantly obvious that the most commonly employed techniques are not ideal for solving the challenging problems that exist at the interface of biology, chemistry, physics, and medicine.

¹ Email: {federico.but, diletta.cacciagrano, flavio.corradini, emanuela.merelli, luca.tesei}@unicam.it

Many of the most important events surrounding biomedical processes take place on different time and length scales, i.e., exhibit a multiscale (MS) behaviour that requires resolution of several scales and their concomitant coupled interactions.

1.1 State of the art in MS modelling

Conventional modelling techniques replace small scales by constitutive models and empirical closures that are incapable of capturing the complexity of coupled scale interactions. For these reasons, MS approaches to modelling complex systems are becoming increasingly important.

In the past, multiple (time and length) scales have been treated independently, except in the case of the coupling of quantum and classical (i.e., molecular mechanical) mechanics in a hybrid quantum mechanical/molecular mechanical (QM/MM) treatment, a scheme which was first devised by [17] with subsequent work by [16].

Efforts to date on the development of MS modelling technologies have been focused primarily on specific combinations of methods to demonstrate the capability of a specific scale-coupling technology. Among the most mature of these developments are the implementations of the quasicontinuum method [15].

Other scale-coupling procedures consider the construction of operators to link continuum PDE fields to discrete atomic fields. In several cases the scale-coupling operators consider the discretised PDE form (e.g., element mesh) when constructing these operators [4]. Others define the operators at the equation level.

The heterogeneous MS methods [9] define compression operators to relate discrete to continuum and reconstruction operators to relate continuum to discrete scales, while the equation-free MS method [14] links statistically averaged fine-scale realizations to the coarse scale.

1.2 Towards a common and uniform MS modelling approach

Most current MS modelling approaches either (i) are restricted to the coupling of two sub-systems, with a micro-macro scale relation, while there is a growing interest for complex problems requiring the coupling of many sub-models coming from different fields², or (ii) often involve ad hoc modelling assumptions, incomplete mathematical formulations and numerical implementations, that are inconsistent with both the mathematical and physical properties of the system.

² Biomedical systems, for instance, involve biological, chemical and physical processes evolving at different scales.

Furthermore, MS research efforts remain in general largely disjoint across disciplines and typically exploit (separately) two methods which consist of:

- including problems that allow the application of the same continuum model at all scales, with the primary barrier to a possible simulation being computing resources;
- encompassing applications where detailed physics at the atomistic-molecular level must be simulated to resolve the small scales, with the effect on and coupling to the continuum level being frequently unclear.

For example, in heterogeneous MS and some quasicontinuum methods small scales are subordinate to a conventional coarse-grain model and serve to replace empirical closures by more accurate estimation procedures. Likewise, ab initio molecular, atomistic and particle modelling approaches [10] focus on small scale interactions and less on their interface with coarse-grain models.

Multiformity among MS model components (i.e., single-scale models) and, consequently, coupling procedures based on approximation techniques have strong effects both in modelling and simulation phase. In fact:

(Modelling phase) A MS model is not necessarily more expressive³ than a single-scale one only because it is MS. It is well-known that a MS model can be more or less expressive according to what models are taken into account (for each scale) and how they are *homogenized* (e.g., coupled). Homogenization is in fact a very delicate and complex task - when single-scale models are heterogeneous, as well as when systems to model admit different homogenization techniques - which can lead to loss of information between scales.

(Simulation phase) The development of tools to support MS simulations must explicitly take into account the variability of the models at different scales and the methods of communicating information between them accounting for scale coupling.

1.3 *Contribution of the paper: a meta-model as a common and uniform MS modelling framework*

Indeed, mathematical and physical issues arising in MS problems - like scale representation, scale separation and inter-scale communication - are common and occur across a wide range of scientific disciplines. Consequently, the need of describing MS problems across multiple scientific disciplines and realizing MS simulations requires a common modelling approach being able to (i) unify these common principles, (ii) support the effective combination of various single-scale models through well-defined scale coupling procedures, and con-

³ In the sense of accurate, precise, faithful to the real system which one aims at describing.

sequently (iii) help to model and to simulate MS systems in a rigorous and systematic way.

Despite this widely acknowledged need, there is a scarcity of methodological papers toward this direction. The most notable exception is [13], where the Complex Automata (CxA) paradigm is introduced to solve, in a uniform, flexible and intuitive way, problems in which several different physical processes at different spatial and temporal scales interact.

Independently, BIOSHAPE⁴ [6,5] has been proposed as a scale-independent MS simulation environment based on a uniform MS model - as in the CxA case - even if no formal proof is given by the authors in [6,5] (see Section 2). Scale-independence can be viewed as a consequence of the BIOSHAPE ability to treat biological entities of any size in a uniform way, e.g., as geometric *shapes* equipped with perception, interaction and movement capabilities.

In this paper, we formally define and enrich the meta-model of BIOSHAPE (see Section 4) and we exploit it to give a uniform treatment of generally defined coupling schemes, in particular the micro-macro one applied to the *bone remodelling* [11,5], a biological process which can be considered multiscale, since macroscopic behaviour and microstructure strongly influence each other (see Subsection 5.2). We explicitly take inspiration from the CxA paradigm (see Section 3), a similarity which has been already foreseen in [7] where bone remodelling has been modeled by a 2-scale CxA (see Subsection 5.1) and simulated running such a model in BIOSHAPE.

Similarly to CxA, also the BIOSHAPE meta-model enjoys two important features: namely, (i) a MS system can be decomposed in uniform single-scale models, each one described by a generic sequence of calls to well-defined operators, and (ii) the link between any two single-scale models can be expressed as a flow of data between a pair of these operators by well-defined coupling schemes.

As a consequence, such features not only enforce and formally prove the scale-independence property of BIOSHAPE simulator, but also makes the BIOSHAPE meta-model a common and uniform MS modelling paradigm across a range of heterogeneous application domains.

2 A brief overview of BIOSHAPE

BIOSHAPE is a spatial 3D simulator which has been engineered in the perspective to be a *uniform, particle-based, space- and geometry-oriented* multiscale modelling and simulation environment. BIOSHAPE is *scale-independent* since it treats biological entities of any size simply as geometric 3D *shapes*. A shape can be either a basic one (a polyhedron, a sphere, a cone or a cylinder) or a

⁴ BIOSHAPE Project: <http://cosy.cs.unicam.it/bioshape/>.

correctly composed one (an aggregation of shapes glued on common surfaces of contact). Every element involved in the simulated process is a *3D process*, i.e., an independent entity that owns a particular 3D shape and a particular behaviour.

Every entity has associated its physical motion law. The behaviour of every entity, i.e., the way it interacts with other entities and with the environment, is defined partially through a process algebra approach, namely, the *Shape Calculus* [3,2,1], and partially through Java programming. The Shape Calculus is a formal language defining the structure of 3D processes with channels through which they can bind with other compatible entities. The binding can happen only if the two involved processes collide on a certain common surface in which both have active compatible channels (this is called a collision-dependent behaviour). The result is the creation of a new 3D process whose shape is the composition of the shapes of the original processes and the behaviour is a combination of the original ones. A two-phase collision detection algorithm is defined in order to establish whether shapes collide and when the first of such events (called *first time of contact*) happens within a timestep duration.

The Shape Calculus allows to express also the *split* of 3D processes, an event that can represent the division of a previously established bond or a “reaction” producing new entities with completely different behaviour. The split process determines also the split of the composed shape of the original process into smaller component shapes. The occurrence of a split is an internal decision of the process, and thus it is considered a non-collision-driven behaviour. It can be determined if any process is going to split and when, making it possible to calculate the first time of split within a timestep duration.

Finally, BIOSHAPE permits the definition of services, software entities that can be communication services - within 3D processes or with system services -, information repositories - such as force fields in space, chemical gradients or perception services - or can represent diffused species, i.e. entities that the modeller chooses to represent not physically, with shapes, but simply as concentrations. This is because, for instance, they are well-diffused in the space or because their magnitude is too small w.r.t. the scale chosen for the model. All these services are programmed software entities. Any interaction of 3D processes with them is considered a non-collision-driven behaviour.

The BIOSHAPE software architecture has been engineered from the perspective of supporting *cluster* and *distributed* computational approaches, to satisfy the great computational power demanded for simulation.

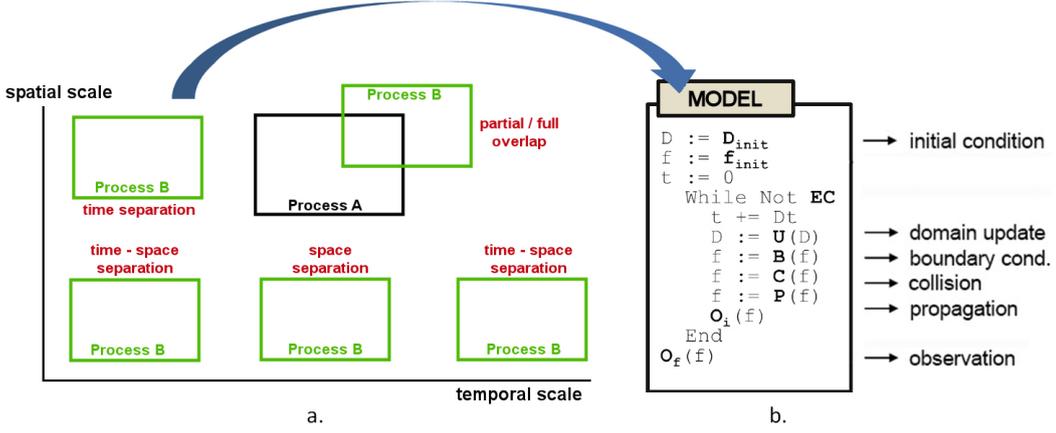


Fig. 1. a. Scale Separation Map representing temporal and spatial relations between two processes, A and B; b. Main loop of an arbitrary process.

3 CxA: from a uniform MS modelling approach to a uniform MS simulation environment

In the following, we recall the basic notions of the paradigm and we also describe the associated simulation environment, referring to [12] for further details.

The main ideas of the CxA paradigm is that (i) *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*, see Fig. 1 (b)) and that (ii) the link between any two sub-models can be expressed as a flow of data between a pair of these operators by well-defined coupling schemes (see Fig. 2).

Being the main loop of any CA-LB uniformly defined, such coupling schemes only depend on the CA-LB spatio-temporal “positions” in a *Scale Separation Map* (SSM), where each CA-LB is represented as an area according to its spatial and temporal scales (see Fig. 1 (a)). Formally:

Definition 3.1 [Complex Automata (CxA)] *A CxA \mathcal{A} is a graph (V, E) , where*

$$V(\text{vertex set}) = \{C_k \stackrel{\text{def}}{\langle} D_k(\Delta x_k, \Delta t_k, L_k, T_k), \mathbb{F}_k, \Phi_k, f_{\text{init}}^k, u_k, O_k \rangle | C_k \text{ is a CA}\},$$

$$E(\text{edge set}) = \{E_{hk} | E_{hk} \text{ is a coupling scheme between } C_h \text{ and } C_k, \text{ as in Fig. 2}\}.$$

D_k is the domain, made of spatial cells of size Δx_k and spanning a region of size L_k , while the quantity Δt_k is the time step and $T_k/\Delta t_k$ is the number of iterations during which C_k will be run. \mathbb{F}_k denote the set of states, $f_{\text{init}}^k \in \mathbb{F}_k$ is the initial state and any state evolves according to the update rule Φ_k , constrained to be in the form of $\Phi_k = P \circ C \circ B$. *Boundary condition* B and *Collision* C depend, possibly, on the field u_k collecting the external data ex-

	Time Overlap		Time Separation	
Space Overlap	Single Domain  Coupling through collision operator. <i>Snow transport, diffusion/advection, ...</i>	Multi Domain  Coupling through boundary condition. <i>Fluid structure, grid refinement, ...</i>	Single Domain  Coupling through collision operator. <i>Forest-Savannah-Fire interactions</i>	Multi Domain  Coupling through boundary, initial conditions. <i>Coral Growth, ...</i>
	Single Domain  Coupling through collision operator. <i>Algae-Water ecological model, ...</i>	Multi Domain  Coupling through boundary condition. <i>Wave propagation in two media, ...</i>	Hierarchical Coupling  Coupling through collision operator and initialization. <i>Suspension Fluid, ...</i>	"Physics-Biology Coupling"  Coupling through boundary conditions and initialization. <i>Oscillating blood flow and endothelial cells, .</i>

Fig. 2. SSM and Coupling schemes.

changed at each iteration between C_k and its environment, while *Propagation* P depends on the topology of the domain. The functional O_k , the observable, specifies the quantity we are interested in.

CXALITE⁵ and MUSCLE⁶ are two significant tools implementing the CxA paradigm. CXALITE takes care of the necessary operation to generate a CxA whereas the programmer must only define the necessary (single-scale) simulation kernels (simple Java classes) to be coupled. CXALITE code-base is very small and extremely portable. On the other hand, it does not provide facilities for computational distribution, kernel interfaces do not provide support for unit of measurement and no graphical user interface is available to monitor the evolution of the simulation.

MUSCLE is based on JADE⁷ framework (*Java Agent DEvelopment Framework*), a middleware for the deployment of multi-agent systems. Thanks to JADE, MUSCLE can smoothly run on several machines on which kernels are executed as displaced agents. On the other hand, the tool inherits JADE complexity and dependencies hence, it is more difficult to set up and also lesser portable than CXALITE.

4 BIOSHAPE: from a uniform MS simulation environment to a uniform MS meta-model

Inspired by the CxA approach, we enrich the meta-model of BIOSHAPE in order to capture within a general scheme a lot of coupling mechanisms. Firstly, we formally describe the components needed to run a model and the main loop of the simulation. Then, we show how two models, specified at different

⁵ CXALite: <http://github.com/paradigmatic/CxALite/>.

⁶ MUSCLE: <http://developer.berlios.de/projects/muscle>.

⁷ JADE: <http://jade.tilab.com/>.

scales, can be connected and made run together implementing a given *coupling scheme*. Finally, we introduce a general graph of interconnected models running a MS simulation.

Definition 4.1 [BIOSHAPE Model Template] *A BIOSHAPE model template is a tuple $\mathcal{M} = \langle \sigma, \tau, \mathcal{W}, \mathcal{P}, \mathcal{S}, \mathcal{O}_i, \mathcal{O}_f \rangle$ where σ and τ are the spatial and temporal scales (e.g. micrometers and milliseconds) of the model. \mathcal{W} is a world space template, i.e. a geometrical shape giving the form and the magnitude of the physical world of the model. Inside this bounded space, all the 3D processes will live and interact. \mathcal{P} is a set of 3D processes templates, i.e., the “species” of 3D processes that will populate the world. \mathcal{S} is a set of services that can be activated in the world. Finally, \mathcal{O}_i and \mathcal{O}_f are, respectively, a functional extrapolating observables from the configuration of the world at each step of the main simulation loop, and a functional giving observables at the final stage of the simulation.*

An instance W of a given world space template \mathcal{W} is determined by fixing a global 3D coordinate system in which the shape is placed, giving the possibility of measuring distances with the given space scale, and making it possible to create 3D processes inside the shape in specified positions. Moreover, an instance P of a 3D process template in \mathcal{P} is constructed by creating a copy of the shape and of the behaviour, by giving it a position in the fixed coordinate system of W and by specifying its initial instant velocity. A collection of instances \mathbf{N} is a 3D network. A 3D network must always be *well-formed*, i.e. there not exist two processes whose shapes interpenetrate or are located outside the boundary of W .

Definition 4.2 [BIOSHAPE Model] *A BIOSHAPE model is a tuple $M = \langle \mathcal{M}, \Delta t, W, \mathbf{N}, \mathcal{S}, EC \rangle$ where \mathcal{M} is a model template, Δt is the maximal duration of a time step in the simulation main loop, W is an instance of \mathcal{W} (of \mathcal{M}), \mathbf{N} is a network of 3D processes obtained instantiating processes from \mathcal{P} (of \mathcal{M}), \mathcal{S} is a set of services in \mathcal{S} (of \mathcal{M}) and EC is the equilibrium condition that, when satisfied, makes the main simulation loop stop. Let \mathbb{M} denote the set of all BIOSHAPE models.*

Δt can be different from the time scale τ of the model because it is constrained by the assumptions of the collision detection algorithm used in the simulation. Moreover, the lower is Δt the better is the approximation of continuum motion of shapes in space, making the model more accurate. Note also that we want to impose that termination is always guaranteed, thus EC must always be a disjunction of a model-dependent condition with a condition like $t \geq T_{\max}$, where t is the simulated time and T_{\max} is the maximum simulated time allowed.

A BIOSHAPE model M is directly runnable. A run correspond to a simula-

init ($\mathcal{M}, \Delta t, W, \mathbf{N}, \mathbf{S}, EC$)	Initialization of the model
$t \leftarrow 0$	Simulated time counter
while $\neg EC$ do	Main Loop
$\mathbf{N}, \mathbf{S}, ftos \leftarrow \mathbf{ncdInteract}(\mathbf{N}, \mathbf{S})$	Non-collision-driven int., first time of split
$\mathbf{N}, \mathbf{S}, ftoc \leftarrow \mathbf{cdInteract}(\mathbf{N}, \mathbf{S})$	Collision-driven int., first time of contact
$step \leftarrow \mathbf{min}\{ftos, ftoc, \Delta t\}$	Determines the actual duration of this step
$t \leftarrow t + step$	Advances simulated time
$\mathbf{N}, \mathbf{S} \leftarrow \mathbf{advance}(\mathbf{N}, \mathbf{S})$	Advances and resolves all splits/collisions
$\mathcal{O}_i(\mathbf{N}, \mathbf{S})$	Output of current step
endwhile	
$\mathcal{O}_f(\mathbf{N}, \mathbf{S})$	Final output of the simulation

Fig. 3. The BIOSHAPE simulation main loop.

tion of the model in which all processes are animated and interact in possibly different ways. Non-deterministic behaviours of processes are resolved either by collisions (collision-driven behaviours) or by internal decision of the process (non-collision-driven behaviours). In each case, a certain amount of randomness may be part of the process. The main loop of the simulation of *any* BIOSHAPE model is shown in Fig. 3. \mathbf{N} and \mathbf{S} can be considered the variables of the model, as they are initialized and also changed throughout the simulation loop. All the other components can also be affected at the initialization step, but not during the simulation. At each iteration, non-collision-driven behaviours allow to determine the first time of split (possibly infinite if no split operation is foreseen within Δt), while collision-driven behaviours allow to determine the first time of contact of at least two entities within Δt (also in this case possibly infinite). Entities in collision may just bounce or decide to bind and form a new 3D process with possibly different behaviours (this depends on the channels they are exposing on their surfaces and where the collision happened). After the update of simulated time of the determined duration, all collisions and/or split operations happening exactly at that moment are resolved changing the current network of 3D processes and possibly the current services.

Definition 4.3 [Coupling Scheme] *Let M_h and M_k be two BIOSHAPE models. A coupling scheme between them is a tuple $E_{hk} = \langle \mathcal{W}_{hk}, W_{hk}, \mathcal{C}_{hk}, \mathcal{C}_{kh} \rangle$ where \mathcal{W}_{hk} is a world space template, W_{hk} is an instance of \mathcal{W}_{hk} such that $W_h \cup W_k \subseteq W_{hk}$, $\mathcal{C}_{hk} = \langle \mathbf{send}_h, \mathbf{op}_h, \mathbf{receive}_k, \mathbf{op}_k \rangle$ and $\mathcal{C}_{kh} = \langle \mathbf{send}_k, \mathbf{op}_k, \mathbf{receive}_h, \mathbf{op}_h \rangle$ are connectors. Connector \mathcal{C}_{hk} is a connection arrow from an operation \mathbf{op}_h of the main loop of model M_h to an operation \mathbf{op}_k of the main loop of model M_k together with two transformation functions $\mathbf{send}_h: \mathbb{M} \rightarrow \mathcal{D}$, $\mathbf{receive}_k: \mathcal{D} \times \mathbb{M} \rightarrow \mathbb{M}$. \mathcal{D} is a generic domain for data exchanging. Connector \mathcal{C}_{kh} has the same structure provided that the roles of h and k are exchanged.*

Two coupled models run together: the 3D coordinate system of W_{hk} be-

comes a global system in which W_h and W_k are placed (internally they continue to use their local coordinate systems). Moreover, they start their main loops at the same instant and run it in parallel with points of synchronization given by the two connectors.

The send is always *non-blocking*, so the control flow continues with the next operation in the main loop. In detail, whenever in a model M_i ($i \in \{h, k\}$) an operation op_i - associated with a starting arrow labelled send_i of a connector - is considered for execution, it is executed in M_i , the variables of the model change obtaining some M'_i , and some data $d = \text{send}_i(M'_i)$ are extracted and sent through the connector to the corresponding receive operation.

On the other side, the receive is always *blocking*. In detail, whenever in a model M_i ($i \in \{h, k\}$) an operation op_i - associated with an ending arrow labelled receive_i of a connector - is considered for execution, it is suspended until the corresponding send operation is executed by the main loop of the connected model. The data d , obtained through the connector upon the send is executed, are used by the receive to adapt the values of the variables of the current model: $M'_i = \text{receive}_i(d, M_i)$. Then the operation op_i is executed in M'_i yielding M''_i . Finally, the control flow continues with the next operation in the main loop.

Example 4.4 Figure 4 shows, graphically, two coupled models and the corresponding connectors. This example is the well-known micro-macro coupling scheme: time and space scales of the macro model (M_{mac}) are bigger than those of the micro model (M_{mic}). Moreover, $W_{\text{mic}} \subset W_{\text{mac}}$ and the shape that constitutes W_{mic} can simply be a portion of the space W_{mac} or can even be mapped on one of the shapes in \mathbf{N}_{mac} ⁸. The connector from M_{mac} to M_{mic} starts from the observables of each step, i.e., at the end of each iteration the send_{mac} function is called to extrapolate significant data from the macro model. This information is then processed by the $\text{receive}_{\text{mic}}$ function in order to properly initialize the micro model. This is then run completely until the equilibrium condition is reached (the simulated duration of the whole run should be comparable with the duration of the macro step). The observables at the end of the simulation are then coded back to the macro model by the function send_{mic} of the other connector. This information is received at the beginning of the next step of the macro model (that was waiting for it due to the blocking receive) in order to adapt its state using the data calculated by the micro model (usually qualitative and quantitative parameters needed for the macro model).

Definition 4.5 [MS Simulation Graph] *A MS simulation graph is a tuple $G =$*

⁸ In this case, the connectors may implement a sort of friction effect on the 3D processes of \mathbf{N}_{mic} .

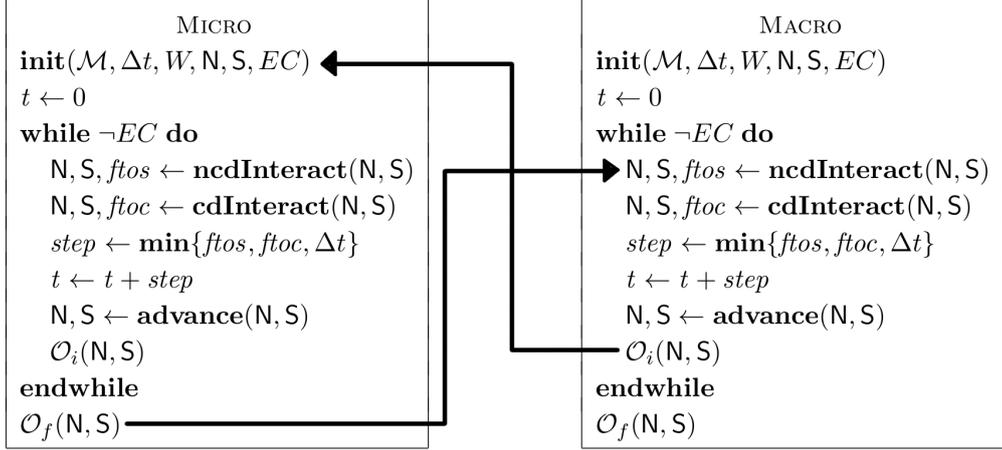


Fig. 4. The micro-macro coupling scheme in BIOSHAPE.

$\langle V, E, \mathcal{W}_G, W_G \rangle$ where V is a finite set of nodes each containing a BIOSHAPE model M_i , E is a finite set of edges that are coupling schemes, $E = \{E_{hk} \mid M_h, M_k \in V\}$, connecting nodes, \mathcal{W}_G is a world space template, and W_G is an instance of \mathcal{W}_G . The latter is such that it encompasses all the world spaces of the templates in E .

Every model in G starts at the same instant and runs its main loop. Then, the synchronization points given by the connectors defined in E will determine the actual control flow of execution. When there is no synchronization any interleaving execution is admitted. The only aspect that still needs to be specified is the case in which a node in the graph is connected with more than one edge and there is a connector starting (or ending) in the same instruction op_i of the main loop of the model. In this case, the execution of the multiple send (or receive) operations associated to op_i is made in sequence, and any interleaving is accepted as valid. In other words, it is requested that these operations are independent w.r.t. each other: the merging of all their effects on the variables should never create a conflict. In case both send and receive operations are associate to op_i at the same time, the send ones must be executed first in order to respect their non-blocking nature.

5 A case study: the bone remodelling process

As a case study, we take into account the bone remodelling. In the following, we formulate a MS model of the process respectively in terms of CxA (Subsection 5.1) and BIOSHAPE (Subsection 5.2), exploiting, in both cases, the corresponding micro-macro coupling scheme.

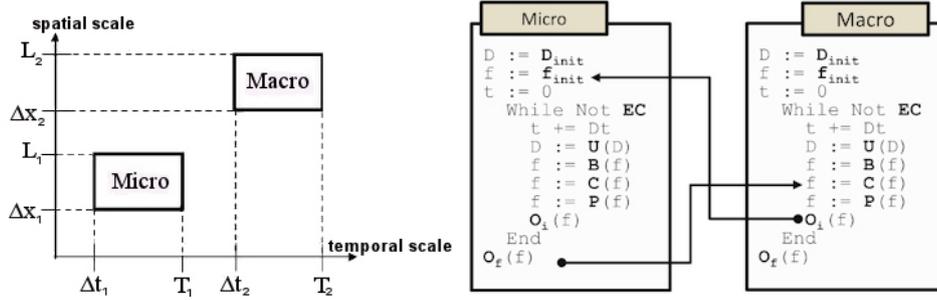


Fig. 5. Micro-macro coupling scheme in CxA

5.1 MS model of trabecular bone remodelling in CxA

The description of bone remodelling in term of CxA in [7] (where the reader can find the complete description) consists of a CA, whose cells are in turn CAs: the macro CA C_1 models a portion of trabecular bone as a lattice of *Basic Multicellular Units* (BMU), while each micro CA $C_{(i,2)}$ (where i corresponds to the cell i in C_1) models a single BMU as a lattice of osteocytes and their surrounding mineralized tissue.

Each $C_{(i,2)}$ is linked to C_1 by the micro-macro coupling scheme (Fig. 5): upon each C_1 iteration each $C_{(i,2)}$ executes a complete simulation, taking input from C_1 . In turn, each $C_{(i,2)}$ output is fed into the C_1 collision operator.

In any micro CA, each cell j modifies its mass according to an error signal between the mechanical stimulus and the internal equilibrium state; a local collision formula is calculated to restore the equilibrium condition when it does not hold. Consequently, the change in mass modifies the stress field in the bone and, therefore, the stimulus operating on j .

In the macro CA, a global *Meshless Cell Method* (MCM) [8] analysis evaluates the stress field and defines the loading conditions operating on each i . Since i modifies the microstructure by processes of formation and adaptation of trabeculae (after a complete execution of $C_{(i,2)}$), the MCM analysis is performed until there is no change in the relative densities and there is no change in the stress field.

5.2 MS model of trabecular bone remodelling in BIOSHAPE

Let us show how the same model can be expressed in BIOSHAPE with the new MS features introduced in Section 4. We will use the micro-macro coupling scheme depicted in Fig. 4. The macro model is at the tissue spatial (millimeter) and temporal (month) scale and represents a certain 3D portion of trabecular bone tissue as a lattice of cubes. Each lattice cube can contain (i) a full/void cube representing resp. a fully mineralised/fluid tissue part, and one of five basic shapes (surface polyhedrons), able to “discretize” the trabecular surface (3D version of 2D shapes in Fig.2 (C) of [5]).

To each lattice cube in the macro model we associate a micro model, representing a BMU. The micro spatial scale is micrometer, while the time scale is a day. The world space of the micro model is exactly shaped as a lattice cube in the macro model. The shapes that we use in the micro model are exactly those described in the micro model of [5], but here the coupling mechanism comes directly and automatically from the coupling scheme.

The information d that is exchanged through the connectors are also the same: the mineralization density of the bone portion, which determines whether the micro model has to be activated or not. If the cube belongs to a bone marrow region then bone remodelling does not occur: in that case, the micro model will be initialized by `sendmac` in such a way that it exits immediately.

We iterate the micro-macro coupling scheme for each pair $(M_{\text{mac}}, M_{\text{mic}}^i)$, where $i \in \{1, \dots, n\}$ and n is the number of lattice cubes in which the world space of M_{mac} is divided. Thus, we obtain a MS simulation graph composed of 1 node for M_{mac} plus n nodes for M_{mic}^i , being each one associated to a different lattice cube. For each macro iteration, n micro complete simulations will be run, determining a new mineralization density in the associated lattice cube and, consequently, the replacement of the contained shape by another shape (that one associated to the computed mineralization density).

Note that the condition of independence at the multiple `sendmac` and `receivemac` calls (one for each of the n connectors starting from them) is respected: since every M_{mic}^i is associated to a disjoint region of the space, there will be no interference in updating the data at M_{mac} .

6 Conclusion and further work

Taking inspiration from the CxA paradigm, we have formally defined and enriched the meta-model of BIOSHAPE - a scale-independent MS simulation environment - and we have exploited it to give a uniform treatment of generally defined coupling schemes, in particular the micro-macro one applied to the bone remodelling. Due to the lack of space, we have only explained how the micro-macro scheme can be implemented in BIOSHAPE, but we claim that all schemes that have been identified in the CxA framework can be treated also in the BIOSHAPE meta-model.

A final note is about the implementation of the BIOSHAPE meta-model in the simulator. BIOSHAPE is engineered to natively run a given model on a certain number of available computational platforms. Thus, this feature can be immediately used to run n models in n nodes, each of which is run on several available platforms. Each model is, in the current version, controlled by a *global coordinator* software agent that is responsible of the balanced distribution of the calculus on the assigned platforms and is the controller that

gives the pace for the steps of the main loop.

Thus, to realize the new version, it is only needed to implement an infrastructure of communication among general coordinators that implements coupling schemes with the described synchronization mechanisms. In this way, an executor for the general MS simulation graph is obtained.

References

- [1] Bartocci, E., D. Cacciagrano, M. R. Di Berardini, E. Merelli and L. Tesei, *Shape Calculus: Timed Operational Semantics and Well-Formedness*, Technical report, <http://arxiv.org/abs/1011.2488> (2010).
- [2] Bartocci, E., D. Cacciagrano, M. R. Di Berardini, E. Merelli and L. Tesei, *Timed Operational Semantics and Well-formedness of Shape Calculus*, Scientific Annals of Computer Science **20** (2010), pp. 33–52.
- [3] Bartocci, E., F. Corradini, M. R. Di Berardini, E. Merelli and L. Tesei, *Shape Calculus. A Spatial Mobile Calculus for 3D Shapes*, Scientific Annals of Computer Science **20** (2010), pp. 1–31.
- [4] Belytschko, T. and S. Xiao, *Coupling methods for continuum model with molecular model.*, Int. J. Multiscale Comput. Eng. **1** (2003), pp. 227–249.
- [5] Buti, F., D. Cacciagrano, F. Corradini, E. Merelli, L. Tesei and M. Pani, *Bone Remodelling in BioShape*, Electronic Notes in Theoretical Computer Science **268** (2010), pp. 17–29, *In Proc. of CS2Bio 2010*.
- [6] Buti, F., D. R. Cacciagrano, F. Corradini, E. Merelli and L. Tesei, *BioShape: a spatial shape-based scale-independent simulation environment for biological systems*, Procedia Computer Science **1** (2010), pp. 827–835, *In Proc. of 7th Int. Workshop on Multiphysics Multiscale Systems (ICCS 2010)*.
- [7] Cacciagrano, D., F. Corradini and E. Merelli, *Bone Remodelling: A Complex Automata-Based Model Running in BioShape*, in: *ACRI 2010*, LNCS **6350**, 2010, pp. 116–127.
- [8] Cristofolini, L., F. Taddei, M. Baleani, F. Baruffaldi, S. Stea and M. Viceconti, *Multiscale investigation of the functional properties of the human femur*, Philos Transact A Math Phys Eng Sci **366** (2008), pp. 3319–3341.
- [9] E, W. and B. Engquist, *The heterogeneous multi-scale methods*, Comm. Math. Sci **1** (2002), pp. 87–132.
- [10] Fineberg, J. and M. Marder, *Instability in dynamic fracture*, PHYS. REP. **67** (1999), pp. 141–144.
- [11] Frost, H., *Skeletal structural adaptations to mechanical usage (SATMU): 2. Redefining Wolff’s Law: the remodeling problem*, Anat. Rec. **226** (1990), pp. 414–422.
- [12] Hoekstra, A. G., A. Caiazzo, E. Lorenz, J. L. Falcone and B. Chopard, *Complex Automata: multi-scale Modeling with coupled Cellular Automata*, in: A. G. Hoekstra, J. Krok and P. M. A. Sloot, editors, *Simulating Complex Systems by Cellular Automata*, Springer, 2010, 1st edition pp. 29–57.
- [13] Hoekstra, A. G., J.-L. Falcone, A. Caiazzo and B. Chopard, *Multi-scale Modeling with Cellular Automata: The Complex Automata Approach*, in: *ACRI 2008*, LNCS **5191**, 2008, pp. 192–199.
- [14] Kevrekidis, I. G., C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg and C. Theodoropoulos, *Equation-Free Multiscale Computation: enabling microscopic simulators to perform system-level tasks*, Communications in Mathematical Sciences **1** (2003), pp. 715–762.
- [15] Shenoy, V. B. and Others, *Quasicontinuum models of interfacial structure and deformation*, Phys. Rev. Lett. **80** (1998), pp. 742–745.

- [16] Singh, U. C. and P. A. Kollman, *A combined ab initio quantum mechanical and molecular mechanical method for carrying out simulations on complex molecular systems: Applications to the $CH_3CI + CI^-$ exchange reaction and gas phase protonation of polyethers*, J. of Comp. Chem. **7** (1986), pp. 718-730.
- [17] Warshel, A. and M. Levitt, *Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme.*, J. of Mol. Bio. **103** (1976), pp. 227-249.