Topology-based Geometric Modelling for Biological Cellular Processes

Mathieu Poudret[1], Jean-Paul Comet[1], Pascale Le Gall[1],
Agnès Arnould[2], and Philippe Meseure[2]

[1] IBISC FRE 2873 CNRS, Université d'Évry, Génopole, 91000 Évry, France
{mathieu.poudret,jp.comet,pascale.legall}@ibisc.univ-evry.fr
[2] SIC, Université de Poitiers, 86962 Futuroscope, France
{arnould,meseure}@sic.univ-poitiers.fr

Abstract. In this paper, we present a framework to simulate biological cellular processes involving a strong topological and geometric structuration. Our framework is based on topology-based geometric modelling. The edge labels of a graph sub-class, the n-dimensional generalized maps, model the neighbouring relations between biological compartments. Thus membranes become topological objects of the n-dimensional generalized maps representation and can be handled just as compartments. The evolutions of the underlying topology during the simulation are expressed by using graph transformation rules. These rules are conditioned by geometric properties and molecular concentrations bound to the vertices of the graph. A simplified example, inspired by the phenomenon of gap junctions, illustrates how topological and geometric parameters are involved in the simulation of biological cellular processes controlled by the concentrations of molecules in the compartments.

Key words: topology-based geometric modelling, graph transformation, generalized map, simulation of biological processes

Introduction

The emergent field of systems biology aims at a system-level understanding of biological systems, taking into account molecular-level phenomena, space structuration of the cells, communication channels and exchanges with the outside space... Although systems are composed of molecules, the essence of biological systems lies in their dynamics and it cannot be described merely by enumerating components. The structures of the system and components both play indispensable roles for the biological functions of the systems. Spatial modelling is increasingly prominent in the biological sciences since scientists attempt to characterize spatial variability of processes. These processes can be spatially indexed either in a continuous manner with space or in a discrete manner following the cell structure into compartments. Thus it is an important challenge to understand the effects of spatial structure on the dynamics of the system, and reciprocally, the consequences of the dynamics on the spatial structure.

Modelling of molecular levels in Cell biology has already become a prolific field of modelling in biology. Different approaches have been used in different application fields: differential models for studying the evolutions of concentrations, Boolean or discrete modelling for genetic regulatory networks, convex analysis for metabolic pathways, rule-based modelling to simulate interacting molecular phenomena... Indeed, transformation rules are well-adapted to represent biochemical reactions like complexation or catalysis [CRFS04]. For such a model, formal methods like model checking have been fruitfully applied in order to verify that the model satisfies a known property of the biological system. Generally this approach considers that all molecules evolve in the same compartment and they can potentially interact even if they are separated *in vivo* by membranes. Recently such a model has been extended [CFS06] to take into account different compartments: these compartments are defined by the mean of molecule naming[1], rules are also needed to describe the exchanges through membranes. This abstraction is sufficient to model transport of molecules between subcellular compartments but does not capture dynamics of compartments themselves. Thus the cell structure (the subcellular compartments) which plays a crucial role in the biological phenomena is either forgotten or statically predefined. Exchanges between subcellular compartments which are related to the surfaces, as well as exocytoses and endocytoses which participate to regulation of concentrations are not well abstracted.

In this paper we present a modelling framework in which compartment rearrangements are taken into consideration and can be derived from constraints on concentrations in different compartments or on geometrical properties of compartments. Previous works have already proposed to manage the dynamics of structuration: Bioambients [RPS$^+$04], Brane calculi [Car05,DP05] or membrane computing [Pau02,Pau06] are suitable for representing various aspects of molecular localisation and compartmentalisation (exchange between compartments, compartment rearrangements, molecular interactions). In such models the evolution of the topological structure is not related to geometrical aspects (proximity of objects) nor related to molecular concentrations. In order to make the topological structure depend on the embedding (geometric positions and/or concentrations) and reciprocally, we base our work on a pure topological modelling which deals with the representation of the structure of objects (their decomposition into vertices, edges, faces and volumes) and with the neighbouring relations between topological objects. Among numerous topological models, we choose the n-dimensional generalised maps [Lie89] (or n-G-maps). The n-G-maps constitute a mathematically-defined representation on which many topological applications have been defined. In this paper we give a graph-oriented definition of the n-G-maps, and formalise transformations of n-G-maps as graph transformation rules, in order to take advantage of the whole corpus of graph transformation theory. Thus simulation of a biological system with topological rearrangements can be computed thanks to a set of basic transformation rules concerning topological rearrangements and embedding.

[1] For instance, $x :: A$ mean that the molecule x sits in a compartment A.

The paper is organised as follow. Section 1 presents graph transformation rules and n-dimensional generalised maps and then introduces the notion of graph transformation meta-rules for transcripting topological operations on n-G-maps in terms of graph transformation. In Section 2 we show how transformation rules can be conditioned by geometric properties and molecular concentrations bound to the vertices of the graph. A simplified example, inspired by the phenomenon of gap junctions, illustrates in Section 3 how topological and geometric parameters are involved in the simulation of biological cellular processes controlled by the concentrations of molecules in the compartments.

1 Generalized maps and graphs

1.1 Graph transformations

In this section, we recall some notions and notations concerning graph transformations (see [HMP01,EEPT06] for a detailed presentation).

Definition 1 (labelled graph). *A graph with labels in Σ_E is a couple (V, E) s. t. V is a set of vertices and $E \subset V \times \Sigma_E \times V$ is a set of non-oriented labelled edges.*

In the sequel, for a graph G, the set of vertices (resp. edges) will be generically denoted by V_G (resp. E_G). We introduce orbit graphs as particular subgraphs, those which are generated by a vertex and an identified subset of labels. Indeed, this will be useful to build cells (faces or volumes for example) when labels will represent neighbouring relations.

Definition 2 (orbit, orbit graph and orbit isomorphism). *Let us consider G a graph with labels in Σ_E, $\{l_1, ..., l_k\} \subset \Sigma_E$ ($k \geq 0$) a set of labels and two vertices $v, v' \in V_G$.*

We call orbit $< l_1, ..., l_k > (v)$, the smallest subset of V_G such that, $v \in < l_1, ..., l_k > (v)$ and such that for each $w \in < l_1, ..., l_k > (v)$ and each l_i ($i \in [1, k]$), if (w, l_i, w') is an edge of G, then $w' \in < l_1, ..., l_k > (v)$. The orbit $< l_1, ..., l_k > (v)$ is said to be incident to v.

We call orbit graph $<< l_1, ..., l_k >> (v)$, the full subgraph G' of G such that $V_{G'} = < l_1, ..., l_k > (v)$ and $E_{G'} \subset E$ is the set of all labels of the form (v, l, v') with $l \in \{l_1, ..., l_k\}$ and v and v' in $V_{G'}$.

The two orbits $< l_1, ..., l_k > (v)$ and $< l_1, ..., l_k > (v')$ of G are isomorphic iff it exists an isomorphism φ from $< l_1, ..., l_k > (v)$ to $< l_1, ..., l_k > (v')$ s. t. for each l_i ($1 \leq i \leq k$), if (w, l_i, w') is an edge of G, then $(\varphi(w), l_i, \varphi(w'))$ also.

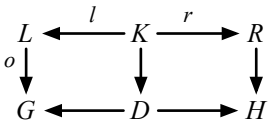
Definition 3 (graph morphism). *Let G, H be two graphs with labels in Σ_E. A graph morphism $f : G \to H$ consists of two functions $f_V : V_G \to V_H$ and $f_E : E_G \to E_H$ verifying that sources, targets and labels are preserved: $\forall (v, l, v') \in E, f_E(v, l, v') = (f_E(v), l, f_E(v'))$. Such a morphism is injective if both f_V and f_E are injective.*

In the sequel, for our purposes, we uniquely consider injective graph morphisms.

Definition 4 (transformation rule). *A graph transformation rule $p : L \leftarrow K \rightarrow R$ is a pair of graph morphisms $l : K \rightarrow L$ and $r : K \rightarrow R$. L is the left-hand side, R is the right-hand side and K is the interface of p.*

Intuitively, graph transformation rules allow one to modify a graph by identifying the subgraph to be removed, corresponding to the graph L with the subgraph K as interface, and the subgraph substituting the removed subgraph, corresponding to the graph R sharing with L the same interface K. This is formally defined by means of graph morphisms making explicit the inclusion between the whole graph G to be transformed and the subgraph L to be removed:

Definition 5 (direct derivation). *Let G be a graph and $p : L \leftarrow K \rightarrow R$ be a graph transformation rule. The rule p transforms G into a graph H, denoted by*[2] *$G \Rightarrow_{p,o} H$, if there are a graph morphism $o : L \rightarrow G$ and two square diagrams as in the following figure which are graph pushouts.*



From an operational point of view, such a derivation is applicable if the following condition holds:

Definition 6 (dangling condition). *Let G be a graph and $p : L \leftarrow K \rightarrow R$ be a transformation rule. A graph morphism $o : L \rightarrow G$ satisfies the dangling condition if no edge in $E_G - o_E(E_L)$ is incident to a vertex in $o_V(V_L) - o_V(V_K)$.*

Intuitively, when the dangling condition holds, the application of a rule p to a graph G along the graph morphism $o : L \rightarrow G$ consists in removing the left-hand side subgraph $L - K$ and adding the right-hand side subgraph $R - K$, up to the graph morphisms as indicated in Definition 5. The application of a transformation rule allows one to uniquely derive the graph H from the graph G. Finally, given a set S of transformation rules, a derivation from G to H is a sequence of applications of a rule in S from G and leading to the graph H.

1.2 Generalized maps

The generalized maps have been introduced by P. Lienhardt [Lie89]. An n-dimensional generalized map (or n-G-map) defines the topology of an n-dimensional subdivision space. The n-G-maps allow the representation of the n-dimensional quasi-varieties, orientable or not. In order to represent space subdivisions, we can choose other topological representations like combinatorial maps [Tut84,BS85] or semi-simplicial sets [May67]. Nevertheless, we have advanced tools at our

[2] The index o is often left implicit.

disposal for the manipulation of n-G-maps which suffice to model biological compartments (Moka [VD03] is an example of such a tool). Moreover, n-G-maps have the advantage of providing a homogeneous formal definition for all dimensions.

Intuivively, the main idea is to decompose an object in basic elements, also called darts (graph vertices), which are connected between them (with graph edges). The decomposition of a 2D object is shown in Fig. 1. The 2D object is displayed on Fig. 1(a). In Fig. 1(b), the object is split in order to focus on the two faces (topological 2-cells) which composed it. On Fig. 1(c), the faces are decomposed into lower dimension elements: the 1-cells. In the end, see Fig. 1(d), the 1-cells are divided in basic units (graph vertices).

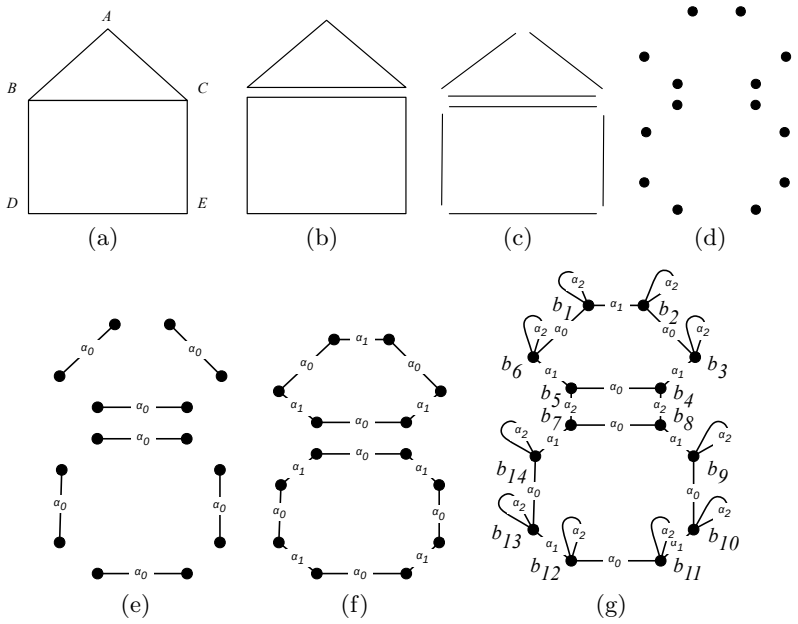


Fig. 1. Decomposition of a 2D object

To complete the decomposition of an object into a 2-G-map representation, we have to report the adjacency relation. Doing this, we recover the knowledge about topological cells constituting the initial object. On the figure, the adjacency relations are represented as labelled edges. Notice that in an n-G-map, there are $n+1$ kinds of labels: α_0 to α_n. Object edges are obtained by linking two vertices with an edge labelled by α_0 (Fig. 1(e)). α_1 labels allow one to get the faces (Fig. 1(f)). Finally, the adjacency relation between 2-cells is represented with the α_2 edges (see Fig. 1(g)).

Some vertices (for instance b_1) are incident to an α_2 loop, (for instance (b_1, α_2, b_1)). The definition of n-G-map requires that for all $0 \leq i \leq n$, all

vertices are both source and target of an edge labelled by α_i, the index i giving the dimension of the considered adjacency relation. For example, α_0 edges link two vertices to build 1-cells of the objects. n-dimensional generalised maps can be defined as particular cases of graphs (Def. 1):

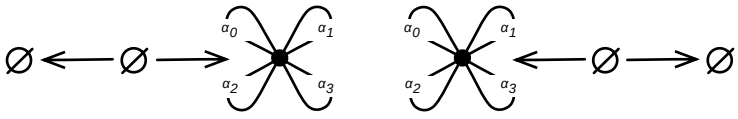
Definition 7 (n-G-map). *Let $n \geq 0$. An n-dimensional generalized map (or n-G-map) is a graph G with labels in $\Sigma_E = \{\alpha_0, \ldots, \alpha_n\}$, s. t.:*

- *for all $v \in V_G, l \in \Sigma_E$, there exists an unique $v' \in V$ s. t. $(v, l, v') \in E$;*
- *for each $v \in V_G$, for all $\alpha_i, \alpha_j \in \Sigma_E$ such that $0 \leq i < i+2 \leq j \leq n$, if $(v, \alpha_i, v_1), (v_1, \alpha_j, v_2), (v_2, \alpha_i, v_3), (v_3, \alpha_j, v_4)$ are edges of E_G, then $v_4 = v$.*

The second point of the n-G-map definition expresses some coherency constraints on the adjacency relations denoted by the labelled edges. Intuitively, in an n-G-map, if two i-dimensional topological units are stuck together then they are stuck along a $(i-1)$-dimensional unit. For instance, on Fig. 1(g), the 2-cell defined by $\{b_1, ..., b_6\}$ is stuck with the 2-cell defined by $\{b_7, ..., b_{14}\}$ along the 1-cell defined by the four vertices $\{b_5, b_4, b_8, b_7\}$. The coherency constraint requires that there is cycle $(\alpha_0, \alpha_2, \alpha_0, \alpha_2)$ starting from every vertex of $\{b_5, b_4, b_8, b_7\}$. For example, we have: $(b_5, \alpha_0, b_4)(b_4, \alpha_2, b_8)(b_8, \alpha_0, b_7)(b_7, \alpha_2, b_5)$. In order to represent a coherent topological object, such constraints should hold at any vertex for the graph to be a n-G-map. Let us now define i-cells of a n-G-map characterising topological subdivisions of an object.

Definition 8 (i-cell). *Let us consider $G = (V, E)$ an n-G-map, $v \in V$ a vertex and $i \in [1, n]$. The i-cell incident to v is the orbit graph (see definition 3) of G $<< \alpha_0, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_n >> (v)$. The i-cell at dart v is noted i-cell(v).*

1.3 Topological operations in terms of graph transformation



(a) dart addition rule (b) dart suppression rule

Fig. 2. dart addition and suppression rule

The evolution of the biological system structure is simulated using transformations of generalised maps. The set of atomic topological operations for n-G-maps has been defined [Lie89] and includes four different operations, namely dart addition, dart suppression i-sew and i-unsew. The first and second operations can be directly translated in transformation rules (see Fig. 2(a) and 2(b)) and consist in adding a vertex (dart) while ensuring coherency constraints

of n-G-maps. Nevertheless, the two last operations are generic and cannot be transformed directly in term of graph transformation rules because the i-sew operation depends, among others, on the orbits. To overcome this limitation, we introduce in this section the notion of graph transformation meta-rules which abstracts a set of graph transformation rules.

Transformation meta-rules define an infinite set of classical transformation rules. The idea is to propagate a local transformation pattern (expressed on a few vertices) along an orbit of the graph, independently of the form of this orbit. To specify which part(s) of the local pattern is attached to the elements of the orbit, we introduce an additional label, noted β.

Definition 9 (meta-rule definition). *Let us consider* $\{\alpha'_1, ..., \alpha'_k\} \subset \Sigma_E = \{\alpha_0, ..., \alpha_n\}$ *a set of labels and a label* $\beta \notin \Sigma_E$*. A graph transformation meta-rule on* $\{\alpha'_1, ..., \alpha'_k\}$*, noted* $L \leftarrow K \rightarrow R$*, is a graph transformation rule where graphs are labelled on edges with labels in* $\Sigma_E \cup \{\beta\}$ *and satisfying both following properties:*

- *for each edge in L (resp. R) of the form* (v, β, v')*, then* $v = v'$*;*
- *there exists at least in L an edge of the form* (v, β, v)*. Graphically* β*-edges are noted with dotted lines, labelled with* $< \alpha'_1, ..., \alpha'_k >$*.*

The β-edges specify that the connected pattern is repeated along the orbit defined by the labels $\{\alpha'_1, ..., \alpha'_k\}$. Let us remark that the existence of a β-edge in L differentiates the graph transformation meta-rules from the classical ones. Semantics of the previously defined meta-rules are translated in the following definition where a meta-rule is viewed as a set of graph transformation rules.

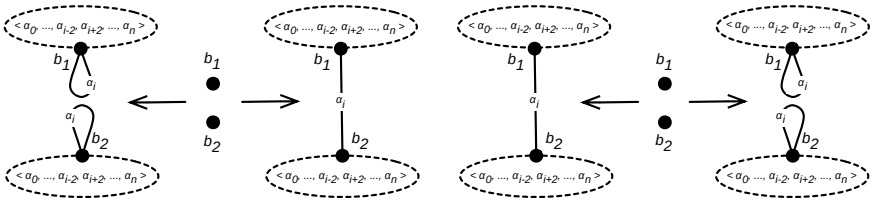
Definition 10 (meta-rule translation). *Let us consider three graphs L, K and R with labels on* Σ_E *and* $\{\alpha'_1, ..., \alpha'_k\} \subset \Sigma_E$*. The translation of a meta-rule* $p : L \leftarrow K \rightarrow R$ *on* $\{\alpha'_1, ..., \alpha'_k\}$ *is the set* $\mathcal{T}_p$ *of graph transformation rules* $L' \leftarrow K' \rightarrow R'$ *where*

- $L' = (\{v_{a,i} \mid a \in V_\beta, i \in V_L\}, \{(v_{a,i}, l, v_{b,i}) \mid l \in \{\alpha'_1, ..., \alpha'_k\}, (a, l, b) \in E_\beta, (i, \beta, i) \in E_L\} \cup \{(v_{a,i}, l, v_{a,j}) \mid l \in \Sigma_E, a \in V_\beta, (i, l, j) \in E_L\})$
- $R' = (\{v_{a,i} \mid a \in V_\beta, i \in V_R\}, \{(v_{a,i}, l, v_{b,i}) \mid l \in \{\alpha'_1, ..., \alpha'_k\}, (a, l, b) \in E_\beta, (i, \beta, i) \in E_R\} \cup \{(v_{a,i}, l, v_{a,j}) \mid l \in \Sigma_E, a \in V_\beta, (i, l, j) \in E_R\})$
- $K' = (\{v_{a,i} \mid a \in V_\beta, i \in V_K\}, \{(v_{a,i}, l, v_{b,i}) \mid l \in \{\alpha'_1, ..., \alpha'_k\}, (a, l, b) \in E_\beta, (i, \beta, i) \in E_K\} \cup \{(v_{a,i}, l, v_{a,j}) \mid l \in \Sigma_E, a \in V_\beta, (i, l, j) \in E_K\})$

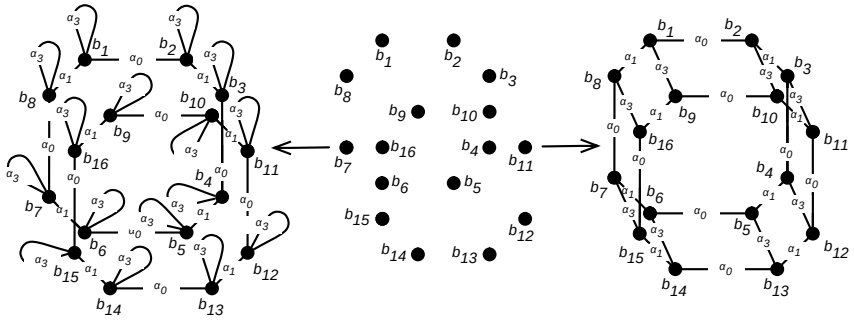
where $G_\beta = (V_\beta, E_\beta)$ *is any graph with edge labels in* $\{\alpha'_1, ..., \alpha'_k\}$*.*

By definition of the meta-rule, if there exist several β-edges in L (and R), the corresponding classical rules replace all β-edges by isomorphic sub-graphs (each of them is isomorphic to the graph G_β). Thus a meta-rule with several β-edges is applicable only on isomorphic orbits.

Definition 11 (meta-rule application). *The application of a meta-rule* $p : L \leftarrow K \rightarrow R$ *is the application of a rule of* $\mathcal{T}_r$ *for which the graph* G_β *is isomorphic to a particular orbit* $< \alpha'_1, ..., \alpha'_k >$ *of G.*



(a) i-sew meta-rule (b) i-unsew meta-rule



(c) 3-sew example

Fig. 3. i-sew, i-unsew meta-rule and 3-sew example

The graph obtained by an application of a meta-rule on an initial graph is then a graph (the application of the classical rules is possible only if those satisfy the dangling condition). Moreover a same rule can be applied with different "anchorings" for β-edges (the vertices of G which are associated to the vertices of L with a β-edge) : when there is a unique β-edge in L, all anchorings lead to the same graph. In case of several β-edges, different anchorings can lead to different graphs corresponding to different manners to sew two isomorphic faces.

It is now possible to transcribe in terms of graph transformation meta-rules the third atomic transformation named i-sew (in a similar way, it is possible to transcribe also the i-unsew).

Definition 12 (i-sew). *Let us consider G an n-G-map, i an integer such that $0 \leq i \leq n$, $v, v' \in V_G$ such that $< \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v)$ and $< \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v')$ are isomorphic and φ the unique isomorphism of $< \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v)$ on $< \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v')$ such that $\varphi(v) = v'$. G' the result of the i-sew of v and v' in G is define by:*

$$\begin{aligned}
&- V_{G'} = V_G; \\
&- E_{G'} = E_G - \{ \ (w, \alpha_i, w) \mid w \in < \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v) \ \} \\
&\qquad\quad - \{ \ (w, \alpha_i, w) \mid w \in < \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v') \ \}. \\
&\qquad\quad \cup \{ \ (w, \alpha_i, \varphi(w)) \mid w \in < \alpha_0, ..., \alpha_{i-2}, \alpha_{i+2}, ..., \alpha_n > (v) \ \}
\end{aligned}$$

Naturally, it can be proved that Definition 12 defines the same operation that the i-sew meta-rule (see Fig. 3(a)). Indeed, in both the definition and the meta-

rule, we remove the α_i loops and we replace it by α_i edges between the two orbits. More generally, the i-sew and i-unsew rules produce n-G-maps when applied on n-G-maps. The graphical representations of these canonical operations are given in Fig. 3(a) and 3(b) whereas Fig. 3(c) shows a classical graph transformation rule deduced from the i-sew meta-rules.

2 Embedding

In Section 1, the definitions and transformation rules only take topology into account, i.e. only the structure of objects. We have not considered geometrical nor biochemical properties of the objects. Most of the time, such additional information is needed. It is associated to the topological structure by an embedding of the n-G-map topological cells. For example, the geometrical embedding associates points to 0-cells, curves to 1-cells, surfaces to 2-cells and volumes to 3-cells. We can consider various kinds of embedding depending on the application field. For instance, in visualization we have to bind reflectance properties to 2-cells [FMH05]. In the case of biological cellular process simulation, we may want to associate biochemical data (for example molecules concentrations) to the compartments (3-cells) or membranes (2-cells).

According to this point, a well-founded simulation model must consider both topological transformations and embedding transformations. In our case, an embedding transformation is needed to model the transport of the molecules from one compartment to another one. In this section, we give some elements on how to consider embedding in our rules.

The embedding of G-maps is realised by adding labels to graph vertices. Since each vertex belongs to one topological cell of each dimension (see Section 1.2 and Definition 8), each vertex may be embedded for each dimension. Moreover, all vertices of a given i-cell have the same embedding for the dimension i. This constraint is easily kept by modifying simultaneously the i-embedding of all vertices of one i-cell.

Definition 13 (embedding). *An embedding for a graph G is a family Σ_V of applications $\eta_{i,\sigma}$ on vertices of G, where i is dimension of edge label α_i, and σ an embedding name.*

Thus, the nature of the biological compartment denoted by the 3-cell incident to a vertex v (noted 3-cell(v)) may be given by the embedding $\eta_{3,type}(v)$ and the geometric embedding of the i-cell(v) is given by $\eta_{i,geo}(v)$. We use this embedding to enrich the transformation rules. The left-hand side of enriched rules may match embedding and the right-hand side may modify it. The embedding expressions may be composed of operations, predicates and logic connectors in order to build rule preconditions. For instance,

$$distance(\eta_{3,geo}(b_1), \eta_{3,geo}(b_2)) \leq \epsilon \wedge \eta_{3,type}(b_1) = A \wedge \eta_{3,type}(b_2) = B$$

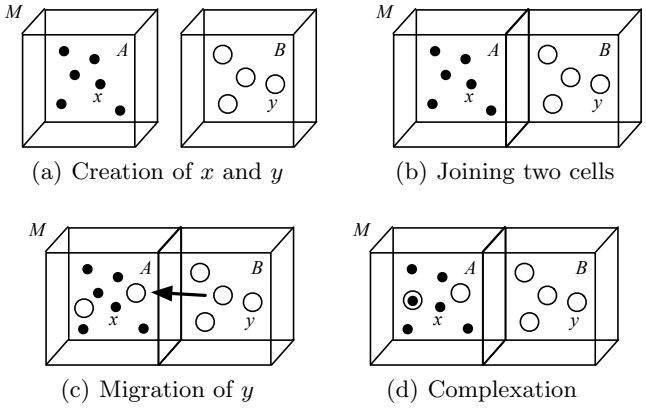
explains that the average distance between the two compartments $3 - cell(b_1)$ and $3 - cell(b_2)$ are less or equal than ϵ and that the two compartments have

respectively the biological type A and B. This embedding property may, for example, precondition the sew of the two compartments (see Fig. 5(d)). Moreover, the embedding may be composed of the affectation (noted :=) and the sequence (noted ;) operations to build the rules postconditions. For example,

$$\eta_{3,conc(x)}(b_1) := decr(\eta_{3,conc(x)}(b_1));$$
$$\eta_{3,conc(y)}(b_1) := decr(\eta_{3,conc(y)}(b_1));$$
$$\eta_{3,conc(xy)}(b_1) := incr(\eta_{3,conc(xy)}(b_1))$$

introduced the concentration modifications inside the compartment $3 - cell(b_1)$ during the complexation of molecules x and y. When we use an expression such as a rule post-condition (see fig. 5(f)), classically on the left-hand side of the :=, η denotes the embedding of the graph before the rule application and on the right-hand side, it denotes the one of the produced graph after the rule application. Is it easy to define such preconditions and postconditions as a formal language, and extend previous definitions to embedding graphs, but this is out of the scope of this article.

3 An example



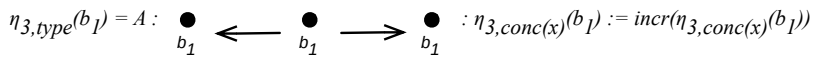
(a) Creation of x and y (b) Joining two cells

(c) Migration of y (d) Complexation

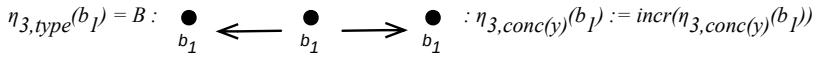
Fig. 4. Biological cellular process example

In Section 2, we have defined conditioned rules that transform the biological compartments. In this section, we give an example inspired by the gap junctions [Alb02] in order to illustrate how to describe a biological phenomenon using our framework. A gap junction is a junction between two cells that allows different molecules to pass from one cell to the other. A graphical representation of the phenomenon is given in Fig. 4. Here, we choose to distinguish two kinds of cells, depending on the type of molecules they generate. In the middle M, the

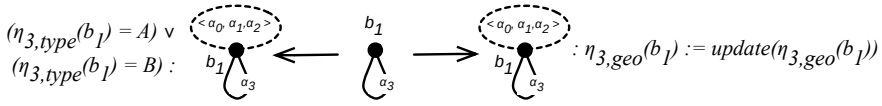
cells of type A generate molecules x and the cells B generate molecules y (see Fig. 4(a)). The cells A and B move and can join, forming a gap junction, when they are sufficiently near (see Fig. 4(b)). When two cells are stuck, molecules y can cross the gap junction. Finally, we decide that molecules x and y can be complexed into a new molecule xy. This very simple biological-inspired example poorly represents the expressiveness of our framework, based on the topology-base geometric modelling. Nevertheless, it is representative enough to apply a large part of the tools we have introduced in a real modelling case.



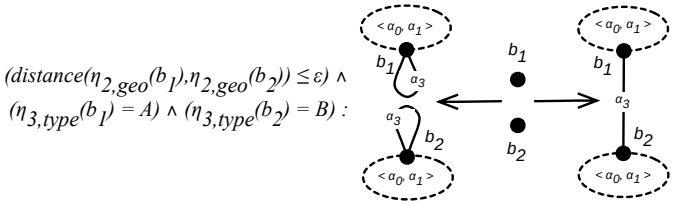
(a) Creation of x



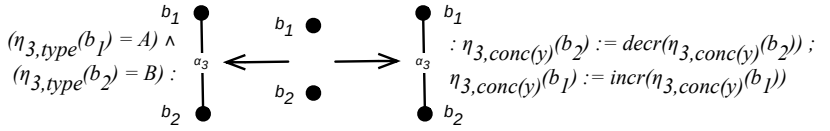
(b) Creation of y



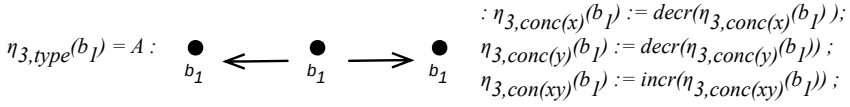
(c) Updating the cells position



(d) Joining two cells



(e) Migration of molecules y



(f) Complexation of x and y

Fig. 5. Rules

The transformation rules that describe the previous process are given in Fig. 5. We do not consider in this paper the initialisation of the system (creation of cells, default positions and concentrations). We assume that the following rules transform an initialized process. The first rule (resp. second rule), introduced in Fig. 5(a) (resp. Fig. 5(b)), matches a compartment (here a biological cell) of type A (resp. type B) and increases the concentration of molecules x (resp. molecules y) in this compartment. The third rule (see Fig. 5(c)) symbolizes the displacement of cells in space, it matches a free compartment (a compartment which is not stuck with another) in order to update its position. The following rule (see Fig. 5(d)) matches two different-typed near cells and sticks them according to the sew transformation rule that is introduced in Section 1.3. The next one (see Fig. 5(e)) matches two stuck cells and transport some y-typed molecules from the B-typed cell to the A-typed one (decreasing and increasing the corresponding concentrations). The last rule (see Fig. 5(f)) matches a cell of type A, decreases its concentration of x and y in order to increase the concentration of molecules xy, according to the complexation of molecules x and y in a new molecule xy.

Perspectives

Many technical improvements can extend our framework introduced in this paper. First, in Section 1.3, we assume that only one kind of orbit appears in a meta-rule. Nevertheless, some complex topological operations, for instance hole suppression [VD03], need to consider two different kinds of orbit simultaneously. Moreover, in our meta-rules, all vertices are duplicated along the orbits. Because of this, we cannot write a rule that links the vertices of an orbit to a unique vertex (to build a pyramid, for instance). We have to update our syntax in order to consider such operations.

The secretory pathway of the biological cell is the place of such a complex topological dynamics. Thus, in the Golgi Apparatus, membrane-bounded compartments can merge, be stuck, divided or perforated. Several Golgi Apparatus abstract models exist and are distinguished by their consideration of the topology [KRSJ04]. For instance, in the maturation model, the Golgi is divided in disconnected saccules and the proteins move from a saccule to another using transport vesicles. In the continuous model, proteins diffuse through a connected Golgi. This case study will help us to find out how to improve our framework in order to include not only simulation tools but also verification tools that can be used by the biologist to validate (or invalidate) models.

Conclusion

In this paper, we present a framework to simulate biological cellular processes that involve a strong topological and geometric structure. Our framework is based on topology-based geometric modelling. We use a graph sub-class, the generalised maps, in order to model the biological compartments. The topological cells: 3-cells (that represent the compartments), 2-cells (that represent the

membranes) are represented with a sub-graph of the generalised map and the neighbouring relations between those cells are modelled by the means of a constraint labelling of the edges. The evolution of the underlying topology during the simulation are expressed by using graph transformation rules. We have extended the notion of graph transformation rules to the notion of meta-rules that not only transform one pattern but a family of patterns present in a graph. The application of a meta-rule is the construction and then the application of the classical graph transformation rules that are abstracted by the meta-rule. The rules we introduce are conditioned by geometric properties and molecular concentration bound to the vertices of the generalised-map. Those conditions allow us to manipulate the topological operations as transformation step that we can automatically iterate. Finally, a simplified example inspired by the phenomenon of gap junctions [Alb02], illustrates how topological and geometric parameters are involved in the simulation of biological cellular processes controlled by the concentrations of molecules in the compartments.

Acknowledgements The biological questions around the Golgi Apparatus are discussed with the biologists within GolgiTop, a workgroup supported by Génopole (Évry, France). We especially thank François Képès, Jean-Marc Verbavatz, Janine Guespin and Alain Rambourg, members of GolgiTop, for the numerous discussions about recent hypotheses of the biological mechanism associated with the Golgi Apparatus.

References

[Alb02] B. Alberts. *Molecular biology of the cell, 4th edition.* Garland Science, New York, 2002.

[BS85] R. Bryant and D. Singerman. Foundations of the theory of maps on surfaces with boundaries. *Quart. J. Math. Oxford Ser. (2)*, 36(141):17–41, 1985.

[Car05] L. Cardelli. Brane calculi. interactions of biological membranes. In *Proc. Computational Methods in Systems Biology*, volume 3082 of *Lecture Notes in BioInformatics*, pages 257–280. Springer, 2005.

[CFS06] L. Calzone, F. Fages, and S. Soliman. Biocham: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[CRFS04] N. Chabrier-Rivier, F. Fages, and S. Soliman. The biochemical abstract machine BIOCHAM. In *CMSB'04*, volume 3082 of *Lecture Notes in BioInformatics*, pages 172–191. Springer, 2004.

[DP05] V. Danos and S. Pradalier. Projective brane calculus. In *CMSB'04)*, volume 3082 of *Lecture Notes in BioInformatics*, pages 134–148. Springer, 2005.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation (Monographs in Theoretical Computer Science. An EATCS Series).* Springer, Secaucus, NJ, USA, 2006.

[FMH05] D. Fradin, D. Meneveaux, and S. Horna. Out-of-core photon-mapping for large buildings. In *EGSR 2005*, 2005.

[HMP01] A. Habel, J. Muller, and D. Plump. Double-pushout graph transformation revisited. *Mathematical Structures in Computer Science*, 11(5):637–688, 2001.

[KRSJ04] F. Képès, A. Rambourg, and B. Satiat-Jeunemaître. Morphodynamics of the secretory pathway. *International review of cytology*, 242:55–120, 2004.

[Lie89] P. Lienhardt. Subdivision of n-dimensional spaces and n-dimensional generalized maps. In *SCG'89*, pages 228–236, Saarbruchen, Germany, 1989. ACM Press.

[May67] J.-P. May. *Simplicial Objects in Algebric Topology*, volume 11 of *Van Nostrand Mathematical Studies*. Van Nostrand, first edition, 1967.

[Pau02] Gheorghe Paun. *Membrane Computing: An Introduction*. Springer, Secaucus, NJ, USA, 2002.

[Pau06] Gh. Paun. Research topics in membrane computing. In *Proceedings of the Fourth Brainstorming Week on Membrane Computing, Vol. II*, pages 235–251. Fenix Editora, 2006.

[RPS+04] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: an abstraction for biological compartments. *Theor. Comput. Sci.*, 325(1):141–167, 2004.

[Tut84] W. Tutte. *Graph Theory*, volume 21 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1984.

[VD03] F. Vidil and G. Damiand. Moka. www.sic.sp2mi.univ-poitiers.fr/moka/, 2003.