

A Spatial Calculus of Wrapped Compartments*

Livio Bioglio¹, Cristina Calcagno^{1,2}, Mario Coppo¹, Ferruccio Damiani¹,
Eva Sciacca¹, Salvatore Spinella¹, Angelo Troina¹

¹Dipartimento di Informatica, Università di Torino

²Dipartimento di Biologia Vegetale, Università di Torino

The Calculus of Wrapped Compartments (CWC) is a recently proposed modelling language for the representation and simulation of biological systems behaviour. Although CWC has no explicit structure modelling a spatial geometry, its compartment labelling feature can be exploited to model various examples of spatial interactions in a natural way. However, specifying large networks of compartments may require a long modelling phase. In this work we present a surface language for CWC that provides basic constructs for modelling spatial interactions. These constructs can be compiled away to obtain a standard CWC model, thus exploiting the existing CWC simulation tool. A case study concerning the modelling of Arbuscular Mycorrhizal fungi growth is discussed.

1 Introduction

Several complex biological phenomena include aspects in which space plays an essential role, key examples are the growth of tissues and organisms, embryogenesis and morphogenesis processes or cell proliferation. This has encouraged, in recent years, the development of formal models for the description of biological systems in which spatial properties can be taken into account [9, 3, 19], as required by the emerging field of *spatial systems biology* [26] which aims at integrating the analysis of biological systems with spatial properties.

The Calculus of Wrapped Compartments (CWC) [12, 11, 10] is a calculus for the description of biochemical systems which is based on the notion of a compartment which represents, in some sense, the abstraction of a region with specific properties (characterised by a *label*, a *wrap* and a *content*). Biochemical transformations are described via a set of stochastic reduction rules which characterise the behaviour of the represented system.

In a recent work [7] we have shown how CWC can be used to model spatial properties of biological systems. The idea is to exploit the notion of compartment to represent spatial regions (with a fixed, two-dimensional topology) in which the labels plays a key role in defining the spatial properties. In this framework, the movement and growth of system elements are described, via specific rules (involving adjacent compartments) and the functionalities of biological components are affected by the spatial constraints given by the sector in which they interact with other elements. CWC allows to model several spatial interactions in a very natural way. However, when the complexity of simulation scenarios increases, the specification of large networks of compartments each one having its own peculiar behaviour and initial state may require a long and error prone modelling phase.

In this paper we introduce a surface language for CWC that defines a framework in which the notion of space is included as an essential component of the system. The space is structured as a square grid, whose dimension must be declared as part of the system specification. The surface language provides

*This research is funded by the BioBITs Project (*Converging Technologies 2007*, area: Biotechnology-ICT), Regione Piemonte.

basic constructs for modelling spatial interactions on the grid. These constructs can be compiled away to obtain a standard CWC model, thus exploiting the existing CWC simulation tool.

A similar approach can be found in [19] where the topological structure of the components is expressed via explicit links which require ad-hoc rules to represent movements of biological entities and a logic-oriented language to flexibly specify complex simulation scenarios is provided.

In order to deal with larger biological systems, we are planning to extend the CWC to the spatial domain incrementally. At this early stage we neglected to consider problems related to the increase of the spatial rules with the increasing dimension of the grid¹. A partial solution to this problem is the use of appropriate data structures to represent entities scattered on a grid. A further step in this direction should be that of allowing the definition of different topological representations for spatial distributions of the biological entities, like in [7]. This requires, obviously, that also the surface language be enriched with primitives suitable to express different spatial topology and related concepts (like the notion of proximity of locations and that of movement in space). The right spatial topology could also help to minimise the number of spatial rules needed for modeling phenomena. A more ambitious goal will be that of providing a basis for computational geometry to our simulator, in order to identify spatially significant events for the simulation. This will require however a much bigger implementation effort.

Organisation of the Paper Section 2 recalls the CWC framework. Section 3 presents the surface language needed to describe spatial terms and rules. Section 4 presents a case study concerning some spatial aspects in the modelling of Arbuscular Mycorrhizal fungi. Section 5 concludes the paper by briefly discussing related work and possible directions for further work. The Appendix presents the software module implementing the surface language.

2 The Calculus of Wrapped Compartments

The Calculus of Wrapped Compartments (CWC) (see [12, 10, 11]) is based on a nested structure of ambients delimited by membranes with specific properties. Biological entities like cells, bacteria and their interactions can be easily described in CWC.

2.1 Term Syntax

Let \mathcal{A} be a set of *atomic elements* (*atoms* for short), ranged over by a, b, \dots , and \mathcal{L} a set of *compartment types* represented as *labels* ranged over by $\ell, \ell', \ell_1, \dots$. A *term* of CWC is a multiset \bar{t} of *simple terms* where a simple term is either an atom a or a compartment $(\bar{a} \mid \bar{t}')^\ell$ consisting of a *wrap* (represented by the multiset of atoms \bar{a}), a *content* (represented by the term \bar{t}') and a *type* (represented by the label ℓ).

As usual, the notation $n * t$ denotes n occurrences of the simple term t . We denote an empty term with \bullet . An example of CWC term is $2 * a \ b \ (c \ d \mid e \ f)^\ell$ representing a multiset (multisets are denoted by listing the elements separated by a space) consisting of two occurrences of a , one occurrence of b (e.g. three molecules) and an ℓ -type compartment $(c \ d \mid e \ f)^\ell$ which, in turn, consists of a wrap (a membrane) with two atoms c and d (e.g. two proteins) on its surface, and containing the atoms e (e.g. a molecule) and f (e.g. a DNA strand). See Figure 1 for some other examples with a simple graphical representation.

¹note that in a 2D model the space-related rules grow according to the square of the grid dimension

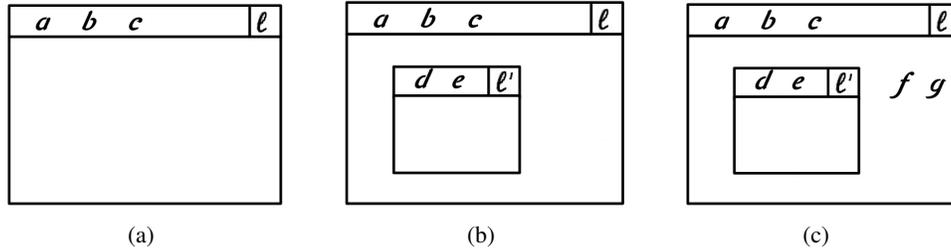


Figure 1: **(a)** represents $(a b c \mid \bullet)^\ell$; **(b)** represents $(a b c \mid (d e \mid \bullet)^{\ell'})^\ell$; **(c)** represents $(a b c \mid (d e \mid \bullet)^{\ell'} f g)^\ell$

2.2 Rewriting Rules

System transformations are defined by rewriting rules, defined by resorting to CWC terms that may contain variables. We call *pattern* the l.h.s. component \bar{p} of a rewrite rule and *open term* the r.h.s. component \bar{o} of a rewrite rule, defined as multiset of *simple patterns* p and *simple open terms* o given by the following syntax:

$$\begin{array}{lcl} p & ::= & a \mid (\bar{a} x \mid \bar{p} X)^\ell \\ o & ::= & a \mid (\bar{q} \mid \bar{o})^\ell \mid X \\ q & ::= & a \mid x \end{array}$$

where \bar{a} is a multiset of atoms, \bar{p} is a pattern (a, possibly empty, multiset of simple patterns), x is a *wrap variable* (can be instantiated by a multiset of atoms), X is a *content variable* (can be instantiated by a CWC term), \bar{q} is a multiset of atoms and wrap variables and \bar{o} is an open term (a, possibly empty, multiset of simple open terms). Patterns are intended to match, via substitution of variables with ground terms (containing no variables), with compartments occurring as subterms of the term representing the whole system. Note that we force *exactly* one variable to occur in each compartment content and wrap of our patterns and simple patterns. This prevents ambiguities in the instantiations needed to match a given compartment.²

A *rewrite rule* is a triple (ℓ, \bar{p}, \bar{o}) , denoted by $\ell : \bar{p} \mapsto \bar{o}$, where \bar{p} and \bar{o} are such that the variables occurring in \bar{o} are a subset of the variables occurring in \bar{p} . The application of a rule $\ell : \bar{p} \mapsto \bar{o}$ to a term \bar{t} is performed in the following way: 1) Find in \bar{t} (if it exists) a compartment of type ℓ with content u and a substitution σ of variables by ground terms such that $u = \sigma(\bar{p} X)$ ³ and 2) Replace in \bar{t} the subterm u with $\sigma(\bar{o} X)$. We write $\bar{t} \mapsto \bar{t}'$ if \bar{t}' is obtained by applying a rewrite rule to \bar{t} . The rewrite rule $\ell : \bar{p} \mapsto \bar{o}$ can be applied to any compartment of type ℓ with \bar{p} in its content (that will be rewritten with \bar{o}).

For instance, the rewrite rule $\ell : a b \mapsto c$ means that in all compartments of type ℓ an occurrence of $a b$ can be replaced by c

While the rule does not change the label ℓ of the compartment where the rule is applied, it may change all the labels of the compartments occurring in its content. For instance, the rewrite rule $\ell : (a x \mid X)^{\ell_1} \mapsto (a x \mid X)^{\ell_2}$ means that, if contained in a compartment of type ℓ , all compartments of type ℓ_1 and containing an a in their wrap can change their type to ℓ_2 .

For uniformity reasons we assume that the whole system is always represented by a term consisting of a single compartment with distinguished label \top and empty wrap, i.e., any system is represented by a

² The linearity condition, in biological terms, corresponds to excluding that a transformation can depend on the presence of two (or more) identical (and generic) components in different compartments (see also [20]).

³The implicit (distinguished) variable X matches with all the remaining part of the compartment content.

term of the shape $(\bullet \mid \bar{t})^\top$, which will be also written as \bar{t} , for simplicity.

2.3 Stochastic Simulation

A stochastic simulation model for biological systems can be defined along the lines of the one presented by Gillespie in [14], which is, *de facto*, the standard way to model quantitative aspects of biological systems. The basic idea of Gillespie's algorithm is that a rate function is associated with each considered chemical reaction which is used as the parameter of an exponential distribution modelling the probability that the reaction takes place. In the standard approach this reaction rate is obtained by multiplying the kinetic constant of the reaction by the number of possible combinations of reactants that may occur in the region in which the reaction takes place, thus modelling the law of mass action. For rules defining spatial movement the kinetic constant can be interpreted as the speed of the movement. In [12], the reaction rate is defined in a more general way by associating to each reduction rule a function which can also define rates based on different principles as, for instance, the Michaelis-Menten nonlinear kinetics.

For simplicity, in this paper, we will follow the standard approach in defining reaction rates. Each reduction rule is then enriched by the kinetic constant k of the reaction that it represents (notation $\ell : \bar{p} \xrightarrow{k} \bar{d}$). For instance in evaluating the application rate of the stochastic rewrite rule $R = \ell : a b \xrightarrow{k} c$ (written in the simplified form) to the term $\bar{t} = a a b b$ in a compartment of type ℓ we must consider the number of the possible combinations of reactants of the form $a b$ in \bar{t} . Since each occurrence of a can react with each occurrence of b , this number is 4. So the application rate of R is $k \cdot 4$.

2.4 The CWC simulator

The CWC simulator [1] is a tool under development at the Computer Science Department of the Turin University, based on Gillespie's direct method algorithm [14]. It treats CWC models with different rating semantics (law of mass action, Michaelis-Menten kinetics, Hill equation) and it can run independent stochastic simulations over CWC models, featuring deep parallel optimizations for multi-core platforms on the top of FastFlow [2]. It also performs online analysis by a modular statistical framework.

3 A Surface Language

In this section we embed CWC into a surface language able to express, in a synthetic form, both spatial (in a two-dimensional grid) and biochemical CWC transformations. The semantics of a surface language model is defined by translation into a standard CWC model.

We distinguish between two kind of compartments:

1. *Standard* compartments (corresponding to the usual CWC compartments), used to represent entities (like bacteria or cells) that can move through space.
2. *Spatial* compartments, used to represent portions of space. Each spatial compartment defines a location in a two dimensional grid through a special atom, called *coordinate*, that occurs on its wrap. A coordinate is denoted by `row.column`, where `row` and `column` are integers. Spatial compartments have distinguished labels, called *spatial labels*, that can be used to provide a specific characterisation of a portion of space.

For simplicity we assume that the wraps of each spatial compartment contains only the coordinate. Therefore, spatial compartment differentiations can be expressed only in terms of labels.⁴

For example, the spatial compartment $(1.2 \rfloor 2 * b)^{soil}$ represents the cell of the grid located in the first row and the second column, and has type *soil*, the spatial compartment $(2.3 \rfloor 3 * b \ c)^{water}$ represents a *water*-type spatial compartment in position 2.3. In our grid we assume that molecules can float only through neighbour cells: all the rules of interaction between spatial compartments must obviously contain the indexes of their location. For example, the rule $\top : (1.2 \ x \rfloor a \ X)^{water} (2.2 \ y \rfloor Y)^{soil} \xrightarrow{k} (1.2 \ x \rfloor X)^{water} (2.2 \ y \rfloor a \ Y)^{soil}$ moves the molecule *a* from the *water* compartment in position 1.2 to the *soil* compartment in position 2.2 with a rate *k* representing in this case, the speed of the movement of *a* in downwards direction from a cell of *water*-type to a cell of *soil*-type.

Let *R* and *C* denote the dimensions of our *R* × *C* grid defined by *R* rows and *C* columns. To increase the expressivity of the language we define a few structures to denote portions (i.e. sets of cells) of the grid. With Θ we denote a set of coordinates of the grid and we use the notion $r.c \in \Theta$ when the coordinate *r.c* is contained in the set Θ . We define rectangles by $rect[r.c, r'.c']$ where *r.c*, *r'.c'* represent the edges of the rectangle. We project rows and columns of our grid with the constructions $row[i]$ and $col[j]$ respectively.

Example 3.1 *The set $\Theta = \{6.6\} \cup rect[1.1, 3.2] \cup col[5]$ represents the set of coordinates*

$$\Theta = \{6.6\} \cup \{1.1, 2.1, 3.1, 1.2, 2.2, 3.2\} \cup \{i.5 \mid \forall i \in [1, R]\}.$$

Note that $row[i]$ is just a shorthand for $rect[i.1, i.C]$. Similarly for columns. We use $[*]$ as shorthand to indicate the whole grid (i.e. $rect[1.1, R.C]$).

We also define four *direction* operators, *N*, *W*, *S*, *E* that applied to a range of cells shift them, respectively, up, left, down and right. For instance $E(1.1) = 1.2$. In the intuitive way, we also define the four diagonal movements (namely, *NW*, *SW*, *NE*, *SE*). With Δ we denote a set of directions and we use the special symbol \diamond to denote the set containing all eight possible directions.

We convene that when a coordinate, for effect of a shift, goes out of the range of the grid the corresponding point is eliminated from the set.

3.1 Surface Terms

We define the initial state of the system under analysis as a set of compartments modelling the two-dimensional grid containing the biological entities of interest.

Let Θ denote a set of coordinates and ℓ_s a spatial label. We use the notation:

$$\Theta, \ell_s \boxplus \bar{t}$$

to define a set of cells of the grid. Namely $\Theta, \ell_s \boxplus \bar{t}$ denotes the top level CWC term:

$$(\bullet \rfloor (r_1.c_1 \rfloor \bar{t})^{\ell_s} \dots (r_n.c_n \rfloor \bar{t})^{\ell_s})^\top$$

where $r_i.c_j$ range over all elements of Θ .

A spatial CWC term is thus defined by the set of grid cells covering the entire grid.

⁴Allowing the wrap of spatial compartments to contain other atoms, thus providing an additional mean to express spatial compartment differentiations, should not pose particular technical problems (extend the rules of the surface language to deal with a general wrap content also for spatial compartments should be straightforward).

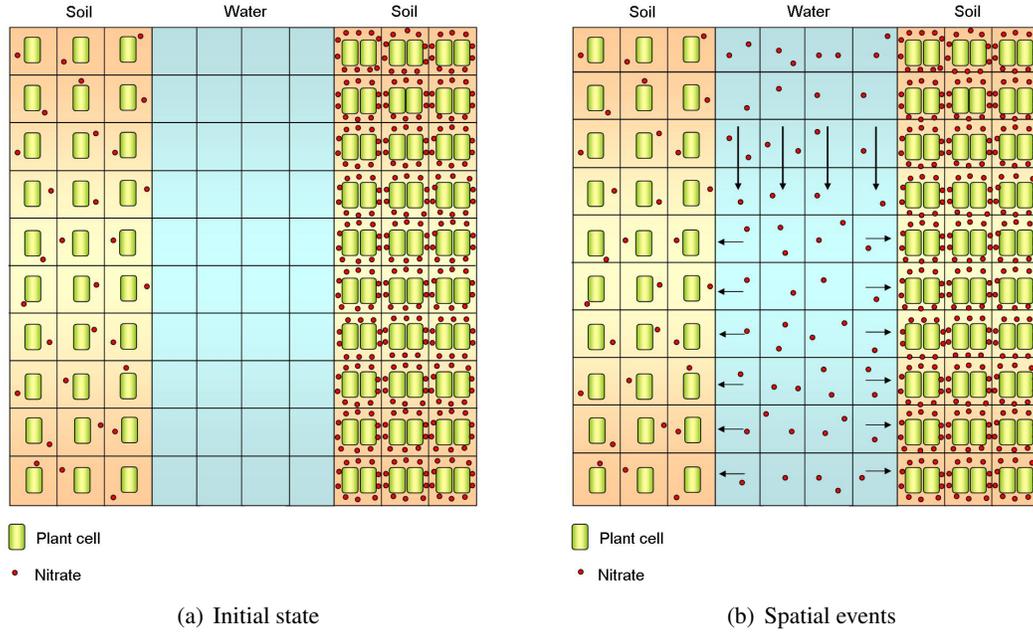


Figure 2: Graphical representations of the grid described in the Example 3.2.

Example 3.2 *The CWC term obtained by the three grid cell constructions:*

$\text{rect}[1.1, 10.3], \text{soil} \boxplus \text{nitr} (\text{receptors}] \text{cytoplasm nucleus})^{\text{PlantCell}}$

$\text{rect}[1.4, 10.7], \text{water} \boxplus \bullet$

$\text{rect}[1.8, 10.10], \text{soil} \boxplus 10 * \text{nitr} 2 * (\text{receptors}] \text{cytoplasm nucleus})^{\text{PlantCell}}$

builds a 10×10 grid composed by two portions of soil (the right-most one reached by nitrates and plant cells) divided by a river of water (see Figure 2(a)).

3.2 Surface Rewrite Rules

We consider rules for modelling three kind of events.

Non-Spatial Events: are described by standard CWC rules, i.e. by rules of the shape:

$$\ell : \bar{p} \xrightarrow{k} \bar{o}$$

Non-spatial rules can be applied to any compartment of type ℓ occurring in any portion of the grid and do not depend on a particular location.

Example 3.3 *A plant cell might perform its activity in any location of the grid. The following rules, describing some usual activities within a cell, might happen in any spatial compartment containing the plant cells under considerations:*

$\text{PlantCell} : \text{nucleus} \xrightarrow{k_1} \text{nucleus mRNA}$

$\text{PlantCell} : \text{mRNA cytoplasm} \xrightarrow{k_2} \text{mRNA cytoplasm protein}.$

Spatial Events: are described by rules that can be applied to specific spatial compartments. These rules allow to change the spatial label of the considered compartment. Spatial events are described by rules of

the following shape:

$$\Theta \triangleright \ell_s : \bar{p} \xrightarrow{k} \ell'_s : \bar{o}$$

Spatial rules can be applied only within the spatial compartments with coordinates contained in the set Θ and with the spatial label ℓ_s . The application of the rule may also change the label of the spatial compartments ℓ_s to ℓ'_s . This rule is translated into the CWC set of rules:

$$\top : (\mathbf{r}_i.c_i \ x \mid \bar{p} \ X)^{\ell_s} \xrightarrow{k} (\mathbf{r}_i.c_i \ x \mid \bar{o} \ X)^{\ell'_s} \quad \forall \mathbf{r}_i.c_i \in \Theta.$$

Note that spatial rules are analogous to non spatial ones. The only difference is the explicit indication of the set Θ which allows to write a single rule instead of a set of rule (one for each element of Θ).

Example 3.4 *If we suppose that the river of water in the middle of the grid defined in Example 3.2 has a downward streaming, we might consider the initial part of the river (framed by the first row $\text{rect}[1.4, 1.7]$) to be a source of nitrates (as they are coming from a region which is not modelled in the actual considered grid). The spatial rule:*

$$\text{rect}[1.4, 1.7] \triangleright \text{water} : \bullet \xrightarrow{k_3} \text{water} : \text{nitr}$$

models the arrival of nitrates at the first modeled portion of the river (in this case it does not change the label of the spatial compartment involved by the rule).

Spatial Movement Events: are described by rules considering the content of two adjacent spatial compartments and are described by rules of the following shape:

$$\Theta \triangleleft \Delta \triangleright \ell_{s_1}, \ell_{s_2} : \bar{p}_1, \bar{p}_2 \xrightarrow{k} \ell'_{s_1}, \ell'_{s_2} : \bar{o}_1, \bar{o}_2$$

This rule changes the content of two adjacent (according to the possible directions contained in Δ) spatial compartments and thus allows to define the movement of objects. The pattern matching is performed by checking the content of a spatial compartment of type ℓ_{s_1} located in a portion of the grid defined by Θ and the content of the adjacent spatial compartment of type ℓ_{s_2} . Such a rule could also change the labels of the spatial compartments. This rule is translated into the CWC set of rules:

$$\top : (\mathbf{r}_i.c_i \ x \mid \bar{p}_1 \ X)^{\ell_{s_1}} (\text{dir}(\mathbf{r}_i.c_i) \ y \mid \bar{p}_2 \ Y)^{\ell_{s_2}} \xrightarrow{k} (\mathbf{r}_i.c_i \ x \mid \bar{o}_1 \ X)^{\ell'_{s_1}} (\text{dir}(\mathbf{r}_i.c_i) \ y \mid \bar{o}_2 \ Y)^{\ell'_{s_2}}$$

for all $\mathbf{r}_i.c_i \in \Theta$ and for all $\text{dir} \in \Delta$.

Example 3.5 *We assume that the flux of the river moves the nitrates in the water according to a downward direction in our grid and with a constant speed in any portion of the river with the following rule:*

$$\text{rect}[1.4, 9.7] \triangleleft \{S\} \triangleright \text{water}, \text{water} : \text{nitr}, \bullet \xrightarrow{k_4} \text{water}, \text{water} : \bullet, \text{nitr}$$

when nitrates reach the down-most row in our grid they just disappear (non moving event):

$$\text{rect}[10.4, 10.7] \triangleright \text{water} : \text{nitr} \xrightarrow{k_4} \text{water} : \bullet.$$

Moreover, nitrates streaming in the river may be absorbed by the soil on the riverside with the rule:

$$\text{rect}[1.4, 10.7] \triangleleft \{W, E\} \triangleright \text{water}, \text{soil} : \text{nitr}, \bullet \xrightarrow{k_5} \text{water}, \text{soil} : \bullet, \text{nitr}.$$

A graphical representation of these events is shown in Figure 2(b). Other rules can be defined to move the nitrates within the soil etc.

4 Case Study: A Growth Model for AM Fungi

In this section we illustrate a case study concerning the modelling of Arbuscular Mycorrhizal fungi growth.

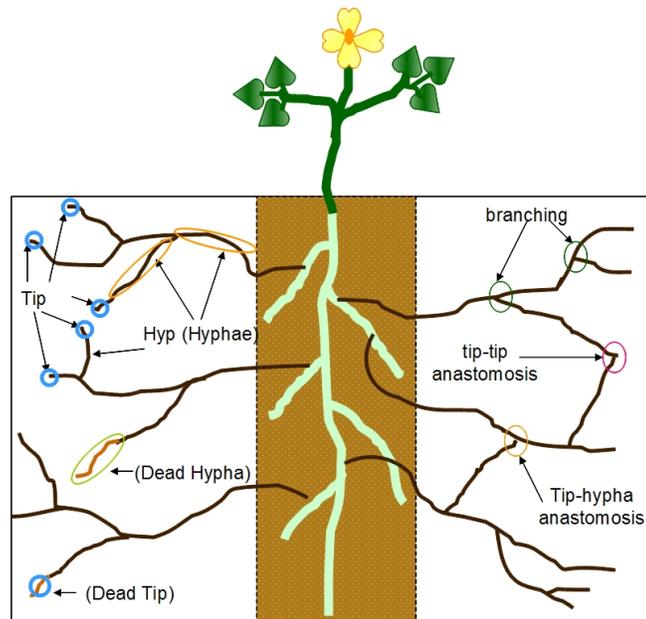


Figure 3: Extraradical mycelia of an arbuscular mycorrhizal fungus.

4.1 Biological Model

The arbuscular mycorrhizal (AM) symbiosis is an example of association with high compatibility formed between fungi belonging to the Glomeromycota phylum and the roots of most land plants [15]. AM fungi are obligate symbionts, in the absence of a host plant, spores of AM fungi germinate and produce a limited amount of mycelium. The recognition between the two symbionts is driven by the perception of diffusible signals and once reached the root surface the AM fungus enters in the root, overcomes the epidermal layer and it grows inter- and intracellularly all along the root in order to spread fungal structures. Once inside the inner layers of the cortical cells the differentiation of specialized, highly branched intracellular hyphae called arbuscules occur. Arbuscules are considered the major site for nutrients exchange between the two organisms. The fungus supply the host with essential nutrients such as phosphate, nitrate and other minerals from the soil. In return, AM fungi receive carbohydrates derived from photosynthesis in the host.

Simultaneously to intraradical colonization, the fungus develops an extensive network of hyphae which explores and exploits soil microhabitats for nutrient acquisition. AM fungi have different hyphal growth patterns, anastomosis and branching frequencies which result in the occupation of different niche in the soil and probably reflect a functional diversity [18] (see Figure 3). The mycelial network that develops outside the roots is considered as the most functionally diverse component of this symbiosis. Extraradical mycelia (ERM) not only provide extensive pathways for nutrient fluxes through the soil, but also have strong influences upon biogeochemical cycling and agro-ecosystem functioning [22]. The mechanisms by which fungal networks extend and function remain poorly characterized. The functioning of ERM presumably relies on the existence of a complex regulation of fungal gene expression with regard

to nutrient sensing and acquisition. The fungal life cycle is then completed by the formation, from the external mycelium, of a new generation of spores able to survive under unfavourable conditions.

Investigations on carbon (C) metabolism in AM fungi have proved useful to offer some explanation for their obligate biotrophism. As mentioned above, an AM fungus relies almost entirely on the host plant for its carbon supply. Intraradical fungal structures (presumably the arbuscules) are known to take up photosynthetically fixed plant C as hexoses. Unfortunately, no fungal hexose transporter-coding gene has been characterized yet in AM fungi.

In order to quantify the contribution of arbuscular mycorrhizal (AM) fungi to plant nutrition, the development and extent of the external fungal mycelium and its nutrient uptake capacity are of particular importance. Shnepf and colleagues [25] developed and analysed a model of the extraradical growth of AM fungi associated with plant roots considering the growth of fungal hyphae from a cylindrical root in radial polar coordinates.

Measurements of fungal growth can only be made in the presence of plant. Due to this practical difficulty experimental data for calibrating the spatial and temporal explicit models are scarce. Jakobsen and colleagues [16] presented measurements of hyphal length densities of three AM fungi: *Scutellospora calospora* (Nicol.& Gerd.) Walker & Sanders; *Glomus* sp. associated with clover (*Trifolium subterraneum* L.); these data appeared suitable for comparison with modelled hyphal length densities.

The model in [25] describes, by means of a system of Partial Differential Equations (PDE), the development and distribution of the fungal mycelium in soil in terms of the creation and death of hyphae, tip-tip and tip-hypha anastomosis, and the nature of the root-fungus interface. It is calibrated and corroborated using published experimental data for hyphal length densities at different distances away from root surfaces. A good agreement between measured and simulated values was found for the three fungal species with different morphologies associated with *Trifolium subterraneum* L. The model and findings are expected to contribute to the quantification of the role of AM fungi in plant mineral nutrition and the interpretation of different foraging strategies among fungal species.

4.2 Surface CWC Model

In this Section we describe how to model the growth of arbuscular mycorrhizal fungi using the surface spatial CWC. We model the growth of AM fungal hyphae in a soil environment partitioned into 13 different layers (spatial compartments with label *soil*) to account for the distance in centimetres between the plant root and the fungal hyphae where the soil layer at the interface with the plant root is at position 1.1. We describe the mycelium by two atoms: the hyphae (atom *Hyp*) related to the length densities (number of hyphae in a given compartment) and the hyphal tips (atom *Tip*). The plant root (atom *Root*) is contained in the *soil* compartment at position 1.1.

The tips and hyphae at the root-fungus interface proliferate according to the following spatial events:

$$\{1.1\} \triangleright \text{soil} : \text{Root} \xrightarrow{\tilde{a}} \text{soil} : \text{Root Hyp}$$

$$\{1.1\} \triangleright \text{soil} : \text{Root} \xrightarrow{a} \text{soil} : \text{Root Tip}$$

where \tilde{a} and a is the root proliferation factor for the hyphae and tips respectively.

Hyphal tips are important, because growth occurs due to the elongation of the region just behind the tips. Therefore, the spatial movement event describing the hyphal segment created during a tip shift to a nearby compartment is:

$$[*] \triangleleft \{E, W\} \triangleright soil, soil : Tip, \bullet \xrightarrow{v} soil, soil : Hyp, Tip$$

where v is the rate of tip movement. The hyphal length is related to tips movement, i.e. an hyphal trail is left behind as tips move through the compartments. We consider hyphal death to be linearly proportional to the hyphal density, so that the rule describing this spatial event is:

$$[*] \triangleright soil : Hyp \xrightarrow{d_H} soil : \bullet$$

where d_H is the rate of hyphal death.

Mycorrhizal fungi are known to branch mainly apically where one tip splits into two. In the simplest case, branching and tip death are linearly proportional to the existing tips in that location modelled with the following spatial events:

$$[*] \triangleright soil : Tip \xrightarrow{b_T} soil : 2 * Tip$$

$$[*] \triangleright soil : Tip \xrightarrow{d_T} soil : \bullet$$

where b_T is the tip branching rate and d_T is the tip death rate.

Alternatively, if we assume that branching decreases with increasing tip density and ceases at a given maximal tip density, we employ the spatial event:

$$[*] \triangleright soil : 2 * Tip \xrightarrow{c_T} soil : \bullet$$

where $c_T = \frac{b_T}{T_{max}}$. From a biological point of view, this behaviour take into account the volume saturation when the tip density achieves the maximal number of tips T_{max} .

The fusion of two hyphal tips or a tip with a hypha can create interconnected networks by means of anastomosis:

$$[*] \triangleright soil : 2 * Tip \xrightarrow{a_1} soil : Tip$$

$$[*] \triangleright soil : Tip Hyp \xrightarrow{a_2} soil : Tip$$

where a_1 and a_2 are the tip-tip and tip-hypha anastomosis rate constants, respectively.

The initial state of the system is given by the following grid cell definition:

$$\{1.1\}, soil \boxplus Root \ T_0 * Tip \ H_0 * Hyp \\ \text{rect}[1.2, 1.13], soil \boxplus \bullet$$

where T_0 and H_0 are the initial number of tips and hyphae respectively at the interface with the plant root.

4.3 Results

We run 60 simulations on the model for the fungal species *Scutellospora calospora* and *Glomus* sp. Figure 4 show the mean values of hyphae (atoms *Hyp*) of the resulting stochastic simulations in function of the elapsed time in days and of the distance from the root surface. The rate parameters of the model are taken from [25].

The results for *S. calospora* are in accordance with the linear PDE model of [25] which is characterized by linear branching with a relatively small net branching rate and both kinds of anastomosis are

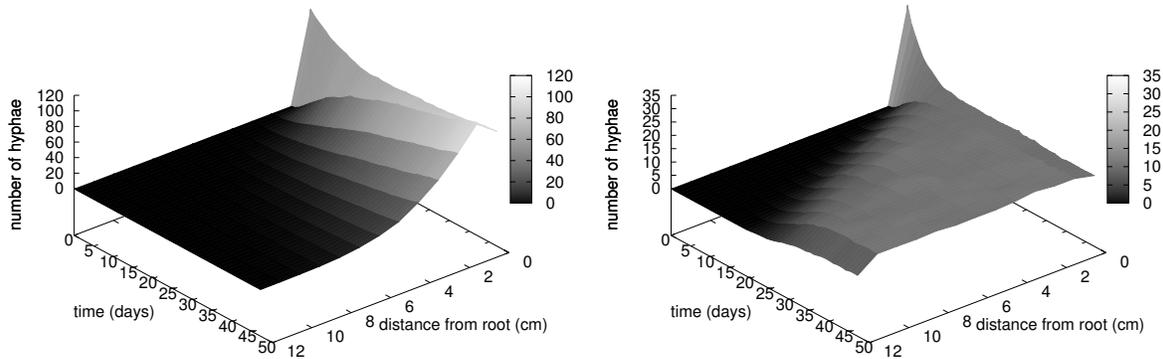


Figure 4: Mean values of 60 stochastic simulations of hyphal growth (atoms *Hyp*) results for *S. calospora* and *Glomus sp. fungi*.

negligible when compared with the other species. This model imply that the fungus is mainly growing and allocating resources for getting a wider catchment area rather than local exploitation of mineral resources via hyphal branching.

The model for *Glomus sp.* considers the effect of nonlinear branching due to the competition between tips for space. The results obtained for *Glomus sp.* are in accordance with the non-linear PDE model of [25] which imply that local exploitation for resources via hyphal branching is important for this fungus as long as the hyphal tip density is small. Reaching near the maximum tip density, branching decreases. Symbioses between a given host plant and different AM fungi have been shown to differ functionally [23].

5 Conclusions and Related Works

For the well-mixed chemical systems (even divided into nested compartments) often found in cellular biology, interaction and distribution analysis are sufficient to study the system's behaviour. However, there are many other situations, like in cell growth and developmental biology where dynamic spatial arrangements of cells determines fundamental functionalities, where a spatial analysis becomes essential. Thus, a realistic modelling of biological processes requires space to be taken into account [17].

This has brought to the extension of many formalisms developed for the analysis of biological systems with (even continuous) spatial features.

In [9], Cardelli and Gardner develop a calculus of processes located in a three-dimensional geometric space. The calculus, introduces a single new geometric construct, called *frame shift*, which applies a three-dimensional space transformation to an evolving process. In such a work, standard notions of process equivalence give rise to geometric invariants.

In [5], a variant of P-systems embodying concepts of space and position inside a membrane is presented. The objects inside a membrane are associated with a specific position. Rules can alter the position of the objects. The authors also define *exclusive* objects (only one exclusive object can be contained inside a membrane). In [4], an spatial extension of CLS is given in a 2D/3D space. The spatial terms of the calculus may move autonomously during the passage of time, and may interact when the constraints on their positions are satisfied. The authors consider a *hard-sphere* based notion of space: two objects, represented as spheres, cannot occupy the same space, thus conflicts may arise by moving objects. Such

conflicts are resolved by specific algorithms considering the forces involved and appropriate pushing among the objects.

BioShape [6] is a spatial, particle-based and multi-scale 3D simulator. It treats biological entities of different size as geometric 3D *shapes*. A shape is either basic (polyhedron, sphere, cone or cylinder) or composed (aggregation of shapes glued on common surfaces of contact). Every element involved in the simulation is a 3D process and has associated its physical motion law.

Adding too many features to the model (e.g., coordinates, position, extension, motion direction and speed, rotation, collision and overlap detection, communication range, etc.) could heavily rise the complexity of the analysis. To overcome this risk, a detailed study of the possible subsets of these features, chosen to meet the requirements of particular classes of biological phenomena, might be considered.

In this paper we pursued this direction by extending CWC with a surface language providing a framework for incorporating basic spatial features (namely, coordinates, position and movement). In future work we plan to extend the surface language to deal with three dimensional spaces and to investigate the possibility to incorporate other spatial features to the CWC simulation framework.

Notably, the framework presented in this paper could also be applied to other calculi which are able to express compartmentalisation (see, e.g., BioAmbients [24], Brane Calculi [8], Beta-Binders [13], etc.).

References

- [1] M. Aldinucci, M. Coppo, F. Damiani, M. Drocco, E. Giovannetti, E. Grassi, E. Sciacca, S. Spinella & A. Troina (2010): *CWC Simulator*. Dipartimento di Informatica, Università di Torino. <http://cwcsimulator.sourceforge.net/>.
- [2] Marco Aldinucci & Massimo Torquati (2009): *FastFlow website*. FastFlow. <http://mc-fastflow.sourceforge.net/>.
- [3] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo & G. Pardini (2011): *Spatial Calculus of Looping Sequences*. *Theoretical Computer Science*.
- [4] Roberto Barbuti, Andrea Maggiolo-Schettini, Paolo Milazzo & Giovanni Pardini (2009): *Spatial Calculus of Looping Sequences*. *Electr. Notes Theor. Comput. Sci.* 229(1), pp. 21–39. Available at <http://dx.doi.org/10.1016/j.entcs.2009.02.003>.
- [5] Roberto Barbuti, Andrea Maggiolo-Schettini, Paolo Milazzo, Giovanni Pardini & Luca Tesei (2011): *Spatial P systems*. *Natural Computing* 10(1), pp. 3–16. Available at <http://dx.doi.org/10.1007/s11047-010-9187-z>.
- [6] Federico Buti, Diletta Cacciagrano, Flavio Corradini, Emanuela Merelli & Luca Tesei (2010): *BioShape: a spatial shape-based scale-independent simulation environment for biological systems*. *Procedia CS* 1(1), pp. 827–835. Available at <http://dx.doi.org/10.1016/j.procs.2010.04.090>.
- [7] C. Calcagno, M. Coppo, F. Damiani, M. Drocco, E. Sciacca, S. Spinella & A. Troina (To Appear): *Modelling Spatial Interactions in the AM Symbiosis using CWC*. In: *CompMod 2011*.
- [8] L. Cardelli (2004): *Brane Calculi*. In: *Proc. of CMSB'04, LNCS 3082*, Springer, pp. 257–278.
- [9] Luca Cardelli & Philippa Gardner (2010): *Processes in Space*. In: *Proc. of the 6th international conference on Computability in Europe, CiE'10*, Springer-Verlag, pp. 78–87.
- [10] M. Coppo, F. Damiani, M. Drocco, E. Grassi, M. Guether & A. Troina (2011): *Modelling Ammonium Transporters in Arbuscular Mycorrhiza Symbiosis*. *Transactions on Computational Systems Biology XIII*, pp. 85–109.
- [11] M. Coppo, F. Damiani, M. Drocco, E. Grassi, E. Sciacca, S. Spinella & A. Troina (2010): *Hybrid Calculus of Wrapped Compartments*. In: *Proceedings Compendium of the Fourth Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC'10)*, 40, EPTCS, pp. 103–121.

- [12] Mario Coppo, Ferruccio Damiani, Maurizio Drocco, Elena Grassi & Angelo Troina (2010): *Stochastic Calculus of Wrapped Compartments*. In: *8th Workshop on Quantitative Aspects of Programming Languages (QAPL'10)*, 28, EPTCS, pp. 82–98.
- [13] P. Degano, D. Prandi, C. Priami & P. Quaglia (2006): *Beta-binders for Biological Quantitative Experiments*. *Electr. Notes Theor. Comput. Sci.* 164(3), pp. 101–117. Available at <http://dx.doi.org/10.1016/j.entcs.2006.07.014>.
- [14] D. Gillespie (1977): *Exact stochastic simulation of coupled chemical reactions*. *J. Phys. Chem.* 81, pp. 2340–2361.
- [15] M.J. Harrison (2005): *Signaling in the arbuscular mycorrhizal symbiosis*. *Annu. Rev. Microbiol.* 59, pp. 19–42.
- [16] I. Jakobsen, LK Abbott & AD Robson (1992): *External hyphae of vesicular-arbuscular mycorrhizal fungi associated with Trifolium subterraneum L. 1. Spread of hyphae and phosphorus inflow into roots*. *New Phytologist* 120(3), pp. 371–379.
- [17] B. Kholodenko (2006): *Cell-signalling dynamics in time and space*. *Nature Reviews Molecular Cell Biology* 7, pp. 165–176.
- [18] H. Maherali & J.N. Klironomos (2007): *Influence of phylogeny on fungal community assembly and ecosystem functioning*. *Science* 316(5832), p. 1746.
- [19] Sara Montagna & Mirko Viroli (2010): *A Framework for Modelling and Simulating Networks of Cells*. *Electr. Notes Theor. Comput. Sci.* 268, pp. 115–129. Available at <http://dx.doi.org/10.1016/j.entcs.2010.12.009>.
- [20] Nicolas Oury & Gordon Plotkin (2011): *Multi-Level Modelling via Stochastic Multi-Level Multiset Rewriting*. Draft submitted to MSCS.
- [21] T. Parr et al.: *ANTLR website*. <http://www.antlr.org/>.
- [22] S. Purin & M.C. Rillig (2008): *Parasitism of arbuscular mycorrhizal fungi: reviewing the evidence*. *FEMS Microbiology Letters* 279(1), pp. 8–14.
- [23] S. Ravnskov & I. Jakobsen (1995): *Functional compatibility in arbuscular mycorrhizas measured as hyphal P transport to the plant*. *New Phytologist* 129(4), pp. 611–618.
- [24] A. Regev, E. M. Panina, W. Silverman, L. Cardelli & E. Y. Shapiro (2004): *BioAmbients: an abstraction for biological compartments*. *Theor. Comput. Sci.* 325(1), pp. 141–167.
- [25] A. Schnepf, T. Roose & P. Schweiger (2008): *Growth model for arbuscular mycorrhizal fungi*. *Journal of The Royal Society Interface* 5(24), p. 773.
- [26] Antoine Spicher, Olivier Michel & Jean-Louis Giavitto (2011): *Interaction-Based Simulations for Integrative Spatial Systems Biology*. In: *Understanding the Dynamics of Biological Systems*, Springer New York, pp. 195–231.

Appendix: Implementation of the Surface Language

This Section presents a software module implementing the translation of a surface language model into the corresponding standard CWC model that can be executed by the CWC simulator (cf. Sec. 2.4). The module is written in Java by means of the ANTLR parser generator [21]. The input syntax of the software is defined as following.

Patterns, Terms and Open Terms: pattern, terms and open terms follow the syntax of CWC. In the definition of a compartment, its label is written in braces, as the first element in the round brackets. The symbol \lfloor is translated into $|$, and the empty sequence \bullet into $\backslash e$. If a pattern, term or open term is repeated several times, we write the number of repetitions before it.

Grid Coordinates: the row and the column of a grid coordinate are divided by a comma. All the constructions of the surface language are implemented. The components of a set of coordinates are divided

by a blank space.

Directions: for the directions we use the same keywords of the surface language, plus the special identifiers +, x, * to identify all the orthogonal directions (N,S,W,E), all the diagonal directions (NW,SW,NE,SE) and all the directions, respectively.

Model Name: the name of the model is defined following the syntax

$$\text{model } string ;$$

where *string* is the name of the model.

Grid Dimensions: the dimensions of the grid are expressed with the syntax

$$\text{grid } r, c ;$$

where *r* and *c* are the number of rows and columns of the grid, respectively.

Grid Cell Construction: the notation of a grid cell $\Theta, \ell_s \boxplus \bar{i}$ is translated into the code line

$$\text{cell } \langle \Theta \rangle \{ \ell_s \} \bar{i} ;$$

the module writes as many CWC compartments as the number of coordinates in Θ : each of these copies has the same label ℓ_s and the same content \bar{i} , but a different coordinate in the wrap.

Non Spatial Events: the notation of a non spatial event $\ell : \bar{p} \xrightarrow{k} \bar{o}$ is translated into the code line

$$\text{nse } \{ \ell \} \bar{p} [k] \bar{o} ;$$

the module translates this line in a unique CWC rule.

Spatial Events: the notation of a spatial event $\Theta \triangleright \ell_s : \bar{p} \xrightarrow{k} \ell'_s : \bar{o}$ is translated into the code line

$$\text{se } \langle \Theta \rangle \{ \ell_s \} \bar{p} [k] \{ \ell'_s \} \bar{o} ;$$

As shortcuts, the absence of $\langle \Theta \rangle$ indicates the whole grid, and the absence of $\{ \ell'_s \}$ indicates that the label of the spatial compartment does not change. The module writes a CWC rule for each coordinate in Θ : a rule differs from the others only in the coordinate written in its wrap.

Spatial Movement Events: the notation of a spatial movement event $\Theta \triangleleft \Delta \triangleright \ell_{s_1}, \ell_{s_2} : \bar{p}_1, \bar{p}_2 \xrightarrow{k} \ell'_{s_1}, \ell'_{s_2} : \bar{o}_1, \bar{o}_2$ is translated into the code line

$$\text{sme } \langle \Theta \rangle [\Delta] \{ \ell_{s_1} \} \bar{p}_1 \{ \ell_{s_2} \} \bar{p}_2 [k] \{ \ell'_{s_1} \} \bar{o}_1 \{ \ell'_{s_2} \} \bar{o}_2 ;$$

As shortcuts, the absence of $\langle \Theta \rangle$ indicates the whole grid, and the absence of $\{ \ell'_{s_1} \}$ or $\{ \ell'_{s_2} \}$ indicates that the label of the spatial compartment does not change. In case of absence of $\{ \ell'_{s_2} \}$, an underscore is used to separate \bar{o}_1 and \bar{o}_2 . For each coordinate in Θ , the module writes as many CWC rules as the number of directions in Δ ; in case of a coordinate on the edge of the grid, the module writes a CWC rule for a direction only if this one identifies an adjacent spatial compartment on the grid. The number of CWC rules is therefore less or equal to $|\Theta| \times |\Delta|$.

Monitors: a monitor permits to expose what pattern we need to monitor: at the end of simulation, all the states of this pattern are written in a log file. The syntax to design a monitor is the following:

$$\text{monitor } string \langle \Theta \rangle \{ \ell_s \} \bar{p} ;$$

where *string* is a string describing the monitor and \bar{p} is the pattern, contained into a spatial compartment labelled by ℓ_s and the coordinates Θ , to monitor. As shortcut, the absence of $\langle \Theta \rangle$ indicates the average of the monitors in the whole grid, and the absence of $\{ \ell_s \}$ indicates to write a monitor for each label defined in the model. The module writes a monitor for each coordinate in Θ ; in case of the absence of $\{ \ell_s \}$, the module writes a monitor for each combination of coordinates in Θ and spatial labels defined in the model.

The construction of a model follows the order used to describe the translator syntax: first the model name and the grid dimensions, then the rules of the model. After the rules, we define the grid cell, and finally the monitors.

Listing 1: Input file for the Surface Language parser to model the *S. calospora* fungus growth

```
model "AM Fungi Growth Model S. Calospora";

grid 1,13;

// Tips and hyphae proliferation at the root-fungus interface
se <0,0> {grid} Root [2.52] Root Tip;
se <0,0> {grid} Root [3.5] Root Hyp;

// Tips branching and death
se {grid} Tip [0.02] 2 Tip;
se {grid} Tip [0.0052] \e;

// Hyphae death
se {grid} Hyp [0.18] \e;

// Hyphal creation during a tip shift to a nearby compartment
sme [E W] {grid} Tip {grid} \e [0.125] Hyp _ Tip;

// Initial state
cell <0,0> {grid} Root 97 Tip 115 Hyp;
cell <rect[0,1 0,12] > {grid} \e;

monitor "Hyp" <rect[0,0 0,12] > {grid} Hyp;
```

Listings 1 and 2 show the input file for the CWC Surface Language software to model the *S. calospora* and *Glomus* sp. fungi growth.

Listing 2: Input file for the Surface Language parser to model the *Glomus* sp. fungus growth

```
model "AM Fungi Growth Model Glomus sp.";

grid 1,13;

// Tips and hyphae proliferation at the root-fungus interface
se <0,0> {grid} Root [2.24] Root Tip;
se <0,0> {grid} Root [1.04] Root Hyp;

// Tips branching and death
se {grid} Tip [1.91] 2 Tip;
se {grid} Tip [0.15] \e;

// Tips saturation
se {grid} 2 Tip [0.15] \e;

// Hyphae death
se {grid} Hyp [0.28] \e;

// Hyphal creation during a tip shift to a nearby compartment
sme [E W] {grid} Tip {grid} \e [0.065] Hyp _ Tip;

// Initial state
cell <0,0> {grid} Root 84 Tip 35 Hyp;
cell <rect[0,1 0,12] > {grid} \e;

monitor "Hyp" <rect[0,0 0,12] > {grid} Hyp;
```