A framework for modelling spatially dependent interactions of biological systems in CCP

Davide Chiarugi[1], Moreno Falaschi[2], Diana Hermith[2], and Carlos Olarte[3,4]

[1] Istituto di Scienza e Tecnologia dell'Informazione, CNR Pisa, Italy,
davide.chiarugi@gmail.com
[2] Dip. Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Italy
{moreno.falaschi,diana.hermith}@unisi.it
[3] Dep. de Electrónica y Ciencias de la Computación, Univ. Javeriana. Cali, Colombia
[4] Escola de Ciências e Tecnologia, Univ. Federal do Rio Grande do Norte. Natal, Brazil
carlosolarte@javerianacali.edu.co

Abstract. In this paper we show how to model biochemical reactions in Concurrent Constraint Programming (CCP) taking into account space and locality. In fact, in many cases, the distribution of the reactants in the cellular space is critical for the correct dynamics of the reactions. Thus, we propose a modelling framework which allows us to take into account space domains. Resorting on our approach, it is possible to describe accurately biochemical processes without abstracting away from features related to the spatial localisation. In order to describe locality in CCP, we add subexponentials, a concept coming from linear logic, to the constraint system and then, we can model declaratively when a certain reaction occurs within one location of the space domain. Clearly we can also express interactions between agents, taking place either in different spaces or in the same space. Metabolic pathways and, in particular, signalling pathways tend to be arranged in a physical space such that the product of one reaction is in the right place to become the reactant for the subsequent reaction in the pathway. Following this idea, we show through a simple case study how we can model a signalling cascade. Then, we exploit our framework to model a more complex signalling pathway, namely the TWEAK (TNF related Weak inducer of apoptosis), whose misfunctioning has implications in several important diseases.

1 Introduction

Systems Biology [17,30] aims, at a system-level understanding of biological phenomena, to establish a link between the properties of the "building blocks" and the set of complex behaviours exhibited by living organisms. In this context computational models of biological phenomena, accounting for the integrated functions of the components that constitute living systems are becoming increasingly necessary for accommodating the existing knowledge into a coherent framework [29].

At the present stage one of the main difficulties in building computational models arise from the characteristics of the available information. Indeed, even for the best-studied systems, the known data still fall short of describing exhaustively the properties of each molecular species; even less known are the details of spatial information and the timing of events.

Thus, desirable features of a computational modelling framework should regard the capability of dealing with information often being both incomplete and of non-uniform quality.

Our approach for specifying and studying biological system grounds on Constraint Programming. In a previous work [5], we used the `ntcc` calculus [22], a non-deterministic temporal extension of Concurrent Constraint Programming (CCP) [28], for representing reaction rules in biological systems. This language allowed us to model discrete time, and hence biological systems where reactions had a duration over time. In this paper we concentrate on other characteristics of biological systems. In particular we are interested in modelling spatial distributions of the biochemical reactions in the systems. This is quite important when dealing with cell membranes for instance, or more in general with the possibility to express the fact that a reaction can take place only when some reactants are in the same "*location*". Using CCP for modelling biological systems has some known advantages such as: constraints provide a compact representation of the *state of the system*; the execution of CCP processes can simulate the evolution of the modelled biological system; and CCP has a declarative semantics based on logic, thus allowing for the specification and verification properties of the modelled system.

Moreover, when compared to the other process algebra-based approaches, since CCP allows for an explicit notion of state (a kind of shared store) based on constraints, it can partially determine the value of a variable (consider, e.g., the constraint $x > 42$). Partial information is represented by a store which accumulates constraints, so the more information is obtained the more constraints are added to the store. Being able to deal with partial information is certainly useful in situations where either some components of the system are not fully specified or we do not have enough quantitative information about them.

More specifically, in this work we deal with the problem of representing spatial information in our CCP modelling framework, in order to model biological systems which needs a representation of space. To do this we rely on Linear Logic [15] with Subexponentials (SELL) [9]. In particular we enrich the constraint system of CCP to express constraints as formulas in a suitable fragment of SELL. This improvement allows us to model biochemical systems specifying explicitly spatial information. To show this feature we discuss first a case study represented by a simple model of a G-Protein Coupled Receptors based signalling pathway. Then we show how a much more complex signaling pathway called TWEAK (TNF related Weak inducer of apoptosis) can be modelled. This pathway is expressed in several different tissue types. It has implications with several diseases including autoimmune disorders, cancer and cardiovascular disorders. We believe that our proposal presents some advantages w.r.t. the already existing systems,

for the powerful combination of expressive power and possibility to equip the framework with logics for proving properties of the modelled biological systems.

The rest of the paper is structured as follows. In Section 2 we recall some fundamental concepts about linear logic, subexponential quantifiers and SELL. In Section 3 we report on the CCP calculi motivating their usefulness in modelling biochemical systems. Subsequently we show how embedding SELL into a CCP framework we obtain a SELL-Constraint system which can be used for specifying space and locality in biochemical systems. In Section 4 we discuss our case studies. Section 5 discusses the related work. Section 6 concludes the paper.

2 Linear Logic and Subexponential

We shall now review some basic proof theory of Girard's linear logic [15] with subexponentials [9]. Linear logic's connectives are $\otimes$ and $\parr$ and their units 1 and $\bot$ are *multiplicative*; the connectives $\oplus$ and & and their units 0 and $\top$ are *additive*; $\forall$ and $\exists$ are (first-order) quantifiers; and ! and ? are the exponentials.

Linear logic with subexponentials (*SELL*) shares with linear logic all connectives except the exponentials. The proof rules are the same as in standard linear logic [15]. However, instead of having a single pair of exponentials ! and ?, *SELL* may contain as many *labelled* exponentials, $?^l$ and $!^l$, as needed. These are called *subexponentials* [9]. The *subexponential signature* $\Sigma = \langle I, \preceq, U\rangle$ is built from a set of labels I, $U \subseteq I$ is a set specifying which subexponentials allow weakening and contraction, and $\preceq$ is the pre-order among the elements of I. We assume that U is closed wrt $\preceq$, *i.e.*, if $a \in U$ and $a \preceq b$, then $b \in U$.

We shall use an *intuitionistic* version of *SELL* [23] which is constructed by adding all the rules for the intuitionistic linear logic connectives as usual, except for the exponentials, whose introduction rules are as follows. For each $a \in I$, we add the introduction rules corresponding to dereliction and promotion, where we state explicitly the first-order signature $\mathcal{L}$ of the terms of the language:

$$\frac{\mathcal{L}; \Gamma, F \longrightarrow G}{\mathcal{L}; \Gamma, !^a F \longrightarrow G}\ !^a{}_L \quad \text{and} \quad \frac{\mathcal{L}; !^{x_1} F_1, \ldots !^{x_n} F_n \longrightarrow G}{\mathcal{L}; !^{x_1} F_1, \ldots !^{x_n} F_n \longrightarrow !^a G}\ !^a{}_R$$

The rules for $?^a$ are dual. Here, the rule $!^a{}_R$ (and $?^a{}_L$) have the side condition that $a \preceq x_i$ for all i. That is, one can only introduce a $!^a$ on the right (or a $?^a$ on the left) if all other formulas in the sequent are marked with indices that are greater or equal than a.

Observe that this means that provability is preserved *downwards*: if a formula $!^a P$ is provable from a set of hypothesis, so it is $!^b P$, for $b \preceq a$.

Furthermore, for all $a \in U$, we add the structural rules:

$$\frac{\mathcal{L}; \Gamma, !^a F, !^a F \longrightarrow G}{\mathcal{L}; \Gamma, !^a F \longrightarrow G}\ C \quad \text{and} \quad \frac{\mathcal{L}; \Gamma \longrightarrow G}{\mathcal{L}; \Gamma, !^a F \longrightarrow G}\ W$$

That is, we are also free to specify which indices are *unbounded* (those appearing in the set U), and which indices are *linear* or *bounded*.

The key difference to standard presentations of linear logic is that while linear logic has only seven logically distinct prefixes of bangs and question-marks, *SELL* allows for an unbounded number of such prefixes, *e.g.*, $!^i$, or $!^i?^j$. As showed in [23], by using different prefixes, it is possible to interpret subexponentials in more creative ways, such as temporal units or spatial and epistemic modalities in distributed systems. For instance, $!^i?^i P$ specifies that P is located at location i but its information is confined to the space i. In the forthcoming sections we shall use formulas of the shape $!^i?^i P$ to specify that the behaviour of P is attached to the location (domain of interaction) i. Before that, in the next section, we define a process calculi that is able to manipulate such formulas. This language will be used as the tool for the specification of biological phenomena.

3 CCP calculi

In this paper we shall use as modelling language Concurrent Constraint Programming (CCP) [28], a model for concurrency that combines the traditional operational view of process calculi with a *declarative* view based on logic (see a survey in [24]). This allows CCP to benefit from the large set of reasoning techniques of both process calculi and logic. Processes in CCP *interact* with each other by *telling* and *asking* constraints (pieces of information) in a common store of partial information. The type of constraints processes may act on is not fixed but parametric in a constraint system. Here we build on the ideas of specifying constraint systems as formulas in Girard's linear logic as in linear CCP (`lcc`) [13]. More precisely, we allow constraints to be formulas in a fragment of *SELL*. As we shall show later, this gives rise to a CCP language that is able to capture, declaratively, confinement of information.

Let us start by defining the fragment of *SELL* that will serve as the basis to the constraint system.

Definition 1 (SELL$^{\Subset}$-Constraint System). *A subexponential constraint system (`scs`) is a tuple $(\Sigma, \mathcal{C}, \vdash)$ where Σ is a subexponential signature containing a distinguished subexponential l_∞ representing the top element of the poset Σ. $\mathcal{C}$ is a set of formulas (constraints) built from a first-order signature and the grammar*

$F := 1 \mid A \mid F \otimes F \mid \exists \overline{x}.F \mid !^s?^s F$

where A is an atomic formula. We shall use c, c', d, d', etc, to denote elements of $\mathcal{C}$. We say that d entails *d', written as $d \vdash d'$, iff the sequent $d \longrightarrow d'$ is probable in SELL.*

Let us give some intuitions on the above fragment of *SELL*. The connective 1 corresponds to the empty store, i.e., the initial state of computation. The connective $\otimes$ allows processes to add more information to the store. The existential quantifier hides variables from constraints. The formula $!^l?^l c$ specifies that the constraint c is in the space-location l and this information is confined to that space. In what follows, we shall use both $[F]_l$ and $\bigtriangledown_l F$ to denote a formula of the shape $!^l?^l F$ for an aesthetic reason: the first notation will be used when the constraints are inside processes, while the second when they are in the store.

3.1 The language of Processes

In this section we propose `bccp`, a CCP-based language able to manipulate constraints built from a subexponential constraint system. In this language, it is possible to define hierarchies of spaces where information and processes can be confined. Unlike spatial CCP (`sccp` [18]), locations (or spaces) are linear in the sense that processes may consume constraints when evolving. This is particularly important in the context of biochemical systems since in a reaction, the left hand side components are consumed to produce the ones on the right hand side.

Definition 2 (Syntax). *Processes in `bccp` are built from constraints in the underlying subexponential constraint system as follows:*

$$P, Q := \mathbf{tell}(c) \mid (\mathbf{local}\,\overline{x})\,Q \mid \mathbf{ask}\ c\ \mathbf{then}\ Q \mid P \parallel Q \mid [P]_l \mid p(\overline{x})$$

where variables in $\overline{x}$ are pairwise distinct. We assume that for each process name, there is a unique process definition of the form $p(\overline{x}) \stackrel{\texttt{def}}{=} P$ where the set of free variables is a subset of $\overline{x}$.

Let us give some intuitions about the processes above. The process $\mathbf{tell}(c)$ adds c to the current store d producing the new store $d \otimes c$. The process $(\mathbf{local}\,\overline{x})\,Q$ creates a new set of variables $\overline{x}$ and declares them to be private to Q. We shall simply write $(\mathbf{local}\,x)\,Q$ instead of $(\mathbf{local}\,\{x\})\,Q$.

The process $\mathbf{ask}\ c\ \mathbf{then}\ Q$ evolves into Q if the store entails c. When this happens, the constraint c is consumed. The ask constructor can be then used as a synchronisation mechanism based on entailment of constraints.

The parallel composition of P and Q is denoted as $P \parallel Q$. The processes $[P]_l$ executes and confines the process P in the space l. Finally, given a process definition of the form $p(\overline{x}) \stackrel{\texttt{def}}{=} P$, the agent $p(\overline{y})$ executes the process $P[\overline{y}/\overline{x}]$.

3.2 Operational Semantics

The operational semantics of `bccp` is given by the transition relation $\gamma \longrightarrow \gamma'$ satisfying the rules on Figure 1. A *configuration* γ is a tuple of the form $\langle \overline{x}; \Gamma; c\rangle$ where c is a constraint specifying the current store, Γ is a multiset of processes, and $\overline{x}$ is the set of hidden (local) variables in c and Γ. The multiset $\Gamma = P_1, P_2, \ldots, P_n$ represents the process $P_1 \parallel P_2 ... \parallel P_n$. We shall indistinguishably use both notations to denote parallel composition of processes.

Processes are quotiented by a structural congruence relation $\cong$ satisfying: (1) renaming of bound variables; (2) $P \parallel Q \cong Q \parallel P$: and (3) $P \parallel (Q \parallel R) \cong (P \parallel Q) \parallel R$; (4) $(\mathbf{local}\,\emptyset; \emptyset)\,Q \equiv Q$.

Let $\longrightarrow^*$ be the reflexive and transitive closure of $\longrightarrow$. If $\langle \emptyset; \Gamma; 1\rangle \longrightarrow^* \langle \overline{x}'; \Gamma'; c'\rangle$ and the sequent $\exists \overline{x}'.c'_i \longrightarrow d$ is provable, we write $\Gamma \Downarrow_d$. Intuitively, the set $\{d \in \mathcal{C} \mid P \Downarrow_d\}$ captures the outputs of P under input 1.

Now we give some intuitions about the operational rules. Rule $\mathrm{R_T}$ simply adds the constraint c to the store d. The Rule $\mathrm{R_L}$ augments the set of local variables to later execute P. Note that the premise of the rule guarantees that

$$\frac{}{\langle \overline{x}; \Gamma, \mathbf{tell}(c); d\rangle \longrightarrow \langle \overline{x}; \Gamma; d \otimes c\rangle}\ \mathrm{R_T} \qquad \frac{\overline{x}_i \cap \overline{y} = fv(d) \cap \overline{y} = fv(\Gamma) \cap \overline{y} = \emptyset}{\langle \overline{x}; \Gamma, (\mathbf{local}\,\overline{y})\,P; d\rangle \longrightarrow \langle \overline{x} \cup \overline{y}; \Gamma, P; d\rangle}\ \mathrm{R_L}$$

$$\frac{d \vdash c \otimes e}{\langle \overline{x}; \Gamma, \mathbf{ask}\ c\ \mathbf{then}\ P; d\rangle \longrightarrow \langle \overline{x}; \Gamma, P; e\rangle}\ \mathrm{R_A} \qquad \frac{\langle \overline{x}; \Gamma, Q; d\rangle \longrightarrow \langle \overline{x}'; \Gamma, Q'; d'\rangle}{\langle \overline{x}; \Gamma, P; d\rangle \longrightarrow \langle \overline{x}'; \Gamma, P'; d'\rangle}\ \mathrm{R_{Str}},\ \text{if P} \equiv \text{Q and P}' \equiv \text{Q}'$$

$$\frac{\langle \overline{x}; \Gamma; P; d\rangle \longrightarrow \langle \overline{x}'; \Gamma'; P'; d'\rangle}{\langle \overline{x}; \Gamma; [P]_l; \bigtriangledown_l(d) \otimes e\rangle \longrightarrow \langle \overline{x}'; \Gamma'; [P']_l; \bigtriangledown_l(d') \otimes e\rangle}\ \mathrm{R_S} \qquad \frac{p(\overline{x}) \stackrel{\mathbf{def}}{=} P}{\langle \overline{x}; \Gamma, p(\overline{y}); d\rangle \longrightarrow \langle \overline{x}; \Gamma, P[\overline{y}/\overline{x}]; d\rangle}\ \mathrm{R_C}$$

Fig. 1: Operational Semantics. $fv(\cdot)$ denotes the set of free variables. In $\mathrm{R_A}$, the constraint e is the most general constraint to avoid weakening the store (see [16])

no clash of variables occurs. If the current store d is able to entail c, then the agent **ask** c **then** P evolves into P and consumes the constraint c. Rule $\mathrm{R_{Str}}$ says that congruent processes have the same reductions.

In order to explain the rule $\mathrm{R_S}$, consider the process $[\mathbf{tell}(c)]_l$. What we observe from it is that the constraint $[c]_l$ is added to the store. This means that the output of $\mathbf{tell}(c)$ is confined to the space l. Now consider the process $[\mathbf{ask}\ c\ \mathbf{then}\ Q]_l$. In this case, to decide if Q must be executed, we need to infer whether c can be deduced from the information in location l. Hence, the premise of Rule $\mathrm{R_S}$ considers only the store $\bigtriangledown_l d$. Moreover, the new store in that location, i.e., d' is again placed at location l as shown in the conclusion of the rule.

Finally, Rule $\mathrm{R_C}$ simply unfolds the definition of the process name p.

Let us give an example how the local information can be confined into locations or spaces.

Example 1 (Local stores). Let a, b and $\mathtt{out}$ be subexponentials, $P = \mathbf{tell}(c)$, $Q = \mathbf{ask}\ c\ \mathbf{then}\ \mathbf{tell}(d)$ and $R = [P]_a \parallel [Q]_b$. Then, from the process $[R]_{\mathtt{out}}$ we observe the following: $\langle \emptyset; [R]_{\mathtt{out}}; 1\rangle \longrightarrow^* \langle \emptyset; [[Q]_b]_{\mathtt{out}}; \bigtriangledown_a c\rangle \not\longrightarrow$ Then, Q remains blocked since the information c is only available on the space of a. Note also that the sequent $!^{\mathtt{out}}?^{\mathtt{out}}!^a?^a c \longrightarrow !^{\mathtt{out}}?^{\mathtt{out}} c$ is not provable, i.e., information c is confined to the inner space a.

Now let $R = [P]_a \parallel [Q]_a$. Then, we observe as a final store the constraint: $\bigtriangledown_{\mathtt{out}} \bigtriangledown_a c \otimes \bigtriangledown_{\mathtt{out}} \bigtriangledown_a d$. This means that Q is able to entail the guard c in the space a to later add d to the store.

4 Modelling cellular domains in `bccp`

Signalling pathways allow cells to read environmental cues, translate them into intracellular commands, and react with an appropriate response. Cells are highly specialised and compartmentalised in order to control the temporal dynamics of biochemical networks, but also for developing mechanisms for precise spatial sensing of the relative localisation of signalling proteins and their interactions. These interactions take place in fixed locations, which gives order to cell signalling. Thus, the regulation of signalling within the cellular space is pivotal for self-organisation [10].

We have identified across our working examples the following set of cellular *locations* or *domains* depicted in Figure 2. Those interaction domains can can be helpful to formalise biochemical reactions in a pathway with respect to the type and place of an interaction. We will illustrate how our framework can be used for modelling a network of biochemical interactions by using two examples in the context of signaling pathways.

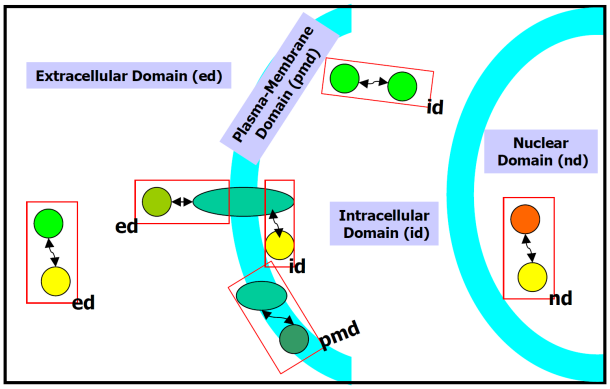

Fig. 2: Cellular locations for local interactions in a signaling pathway.

Example 2. **G-proteins-GPCRs interaction.** A large group of important signalling pathways shares a family of receptors called G-Protein Coupled Receptors (GPCRs) also known as seven-spanning-transmembrane receptors (7TMRs). The ligand-receptor binding on the extracellular side allows the receptor to interact on the intracellular side with a heterotrimeric GDP-binding protein (G-protein for short), a well characterised family of proteins involved in cell signalling. This event, in turn, triggers a cascade of reactions leading to the transduction of a stimulus in the corresponding response [19] (see, Table 1).

Now, consider our small network of reactions for the interaction of a molecule of type GPCR and a molecule of type G-protein. Then, we need to define a subexponential *ed* to represent the *extracellular domain* and the reaction can be modelled as: **ask** $([GPCR]_{id} \otimes [L]_{ed})$ **then tell**$([L - GPCR]_{ed})$.
This means that, if there is a copy of $GPCR$ and a copy of L in the space ed, then an interaction of the type ligand-receptor binding association takes places for producing a copy of the complex $L{-}GPCR$ in the extracellular domain. From the *intracellular domain* point of view, the presence of the complex $L - GPCR$ is required to unchain the second reaction, and thus, the third reaction takes place. Then, the intracellular domain can be modelled as:

$$\begin{array}{l}\textbf{ask } ([L - GPCR]_{ed} \otimes [GaGDP - bg]_{pmd} \otimes [GTP]_{id}) \textbf{ then} \\ \quad \textbf{tell}([GaGTP - GDP - bg]_{id}) \parallel \\ \quad\quad \textbf{ask } ([GaGTP]_{pmd} \otimes [GAP]_{id}) \textbf{ then tell}([(GaGDP - GAP]_{id})\end{array}$$

Interaction's locations	Biochemical reactions rules	Interaction's type
ed	$[L]_{ed} + [GPCR]_{pmd} \rightarrow [L - GPCR]_{ed}$	LRBA
id	$[GaGDP - bg]_{pmd} + [GTP]_{id} \rightarrow [GaGTP - GDP - bg]_{id}$	PPA
id	$[GaGTP]_{pmd} + [GAP]_{id} \rightarrow [GaGDP - GAP]_{id}$	PPA

(a)

ID	Name	Type	Function
L	Ligand	Ligand	Receptor binding
$GPCR$	G-protein coupled receptor	Cell surface receptor	Receptor activity
$GaGDP - bg$	Heterotrimeric G-protein complex	Membrane- anchored protein	Signal transducer activity
$GaGTP$	G-protein GTP-bound alpha-subunit	Membrane- anchored protein	Signal transducer activity
GTP	Guanosine triphosphate	Purine nucleoside triphosphate	Second-messenger activity
GAP	GTPase-activating protein	GTPase	GTPase activity

(b)

Table 1: 1a. A reactive model for the interaction between GPCRs and G-proteins. Abbreviations: LRBA (Ligand-receptor binding association), PPA (Protein-protein association). 1b. Reacting species for the interaction GPCR-G-protein

This example shows that we can model the interactions between proteins that are situated in different "domains" and the store is updated with the location where the interaction between the proteins takes place.

Example 3. **TWEAK (TNF related Weak inducer of apoptosis) signaling pathway.**
This pathway [2] is expressed in several different tissue types and has implications with several diseases including autoimmune disorders, cancer and cardiovascular disorders. The binding on the extracellular domain between the TWEAK receptor (Fn14 or TNFRSF12A) and the trimeric form of TWEAK (TNFSF12-TNFSF12-TNFSF12) activates a downstream signalling cascade within the cell. The following is a set of biochemical interactions for complexes formation in the TWEAK-Fn14 signalling pathway (see Table 2). The next abbreviations are used: LRBA (Ligand-receptor binding association), HCA (Heterotrimeric complex association), PPA (Protein-protein association).

ed: $[TNFSF12 - TNFSF12 - TNFSF12]_{ed} + [TNFRSF12A]_{pmd} \rightarrow [TNFSF12 - TNFSF12 - TNFSF12 - TNFRSF12A]_{ed}$:LRBA
id: $[TNFRSF12A]_{pmd} + [TRAF1]_{id} \rightarrow [TNFRSF12A - TRAF1]_{id}$:PPA
id: $[TNFRSF12A]_{pmd} + [TRAF3]_{id} \rightarrow [TNFRSF12A - TRAF3]_{id}$:PPA
id: $[TNFRSF12A]_{pmd} + [TRAF5]_{id} \rightarrow [TNFRSF12A - TRAF5]_{id}$:PPA
id: $[TNFRSF12A]_{pmd} + [RAC1]_{id} \rightarrow [TNFRSF12A - RAC1]_{id}$:PPA
id: $[HDAC1]_{id} + [RELA]_{id} \rightarrow [HDAC1 - RELA]_{id}$:PPA
id: $[TNFRSF12A]_{pmd} + [TRAF2]_{id} + [BIRC2]_{id} \rightarrow [TNFRSF12A - TRAF2 - BIRC2]_{id}$:HCA
id: $[FADD]_{id} + [RIPK1]_{id} + [CASP8]_{id} \rightarrow [FADD - CASP8 - RIPK1]_{id}$:HCA
nd: $[RELB]_{nd} + [NFKB2]_{nd} \rightarrow [RELB - NFKB2]_{nd}$:PPA

This biological scenario is more complex than the one in the first example. The biochemical reaction rules involve more species placed in several "locations". However, our approach keeps the attention on four basic environments (ed, pmd, id, nd) that basically command the reactive computation within a cell at the level of local interactions. Therefore, we propose below a set of `bccp` processes occurring throughout this signalling cascade. P1 and P2 stand for proteins of different type placed in a particular cellular location. The previous set of molecular interactions combined with Table 2, offers the substrate for the computational process described by using `bccp`.

`ed-Rule 1.`	**ask** $([P1]_{ed} \otimes [P2]_{pmd})$ **then tell**$([P1 - P2]_{ed})$
`id-Rule 1a.`	**ask** $([P1 \otimes P2]_{id})$ **then tell**$([P1 - P2]_{id})$
`id-Rule 1b.`	**ask** $([P1]_{pmd} \otimes [P2]_{id})$ **then tell**$([P1 - P2]_{id})$
`id-Rule 2a.`	**ask** $([P1]_{id} \otimes [P2]_{id} \otimes [P3]_{pmd})$ **then tell**$([P1 - P2 - P3]_{id})$
`id-Rule 2b.`	**ask** $([P1 \otimes P2 \otimes P3]_{id})$ **then tell**$([P1 - P2 - P3]_{id})$
`nd-Rule 1.`	**ask** $([P1 \otimes P2]_{nd})$ **then tell**$([P1 - P2]_{nd})$

Our examples show how to formalise in our framework the behaviour of a system in which the spatial location matters. In fact, we have expressed spatial locations for protein-protein interactions via SELL. As future work we aim to exploit our model for proving properties of the modeled systems, by means of the underlying proof system based on linear logic.

ID	Name	Type	Function	Process
$[TNFSF12 - TNFSF12 - TNFSF12]_{ed}$	Tumor necrosis factor ligand superfamily member 12	Ligand	Receptor binding	CC-ST
$[Fn14]_{pmd}$	Fibroblast growth factor inducible immediate early response protein 14	Cell surface receptor	Receptor activity	CC-ST
$[TRAF1]_{id}$	TNF receptor associated factor 1	Cell surface receptor	Receptor activity	CC-ST
$[TRAF2]_{id}$	TNF receptor associated factor 2	Adapter molecule	Receptor signalling complex scaffold activity	CC-ST
$[TRAF3]_{id}$	TNF receptor associated factor 3	Adapter molecule	Receptor signalling complex scaffold activity	CC-ST
$[FADD]_{id}$	FAS associating protein with death domain, MORT1	Adapter molecule	Receptor signaling complex scaffold activity	CC-ST
$[BIRC2]_{id}$	Baculoviral IAP repeat containing protein 2	Enzyme: Ligase	Ligase activity	CC-ST
$[RAC1]_{id}$	Ras related C3 botulinum toxin substrate 1	GTPase	GTPase activity	ST
$[TRAF5]_{id}$	TNF receptor-associated factor 5	Adapter molecule	Signal transducer activity	ST
$[RIPK1]_{id}$	Receptor TNFRSF-interacting serine-threonine kinase 1	Serine/threonine kinase enzyme	Pro. serine/threonine kinase activity	CC
$[RELB]_{nd}$	V REL avian reticuloendotheliosis viral oncogene homolog B	Transcription factor	Transcription factor activity	NNAM
$[NFKB2]_{nd}$	Nuc. factor kappa B subunit 2	Transcription factor	Transcription factor activity	NNAM
$[RELA]_{id}$	NF of kappa light chain gene enhancer in B cells 3	Transcription factor	Transcription factor activity	NNAM
$[HDAC1]_{id}$	Histone deacetylase 1	Trans. reg. prot.	Trans. regulator activity	NNAM
$[CASP8]_{id}$	Caspase 8	Cysteine protease enzyme	Cysteine-type peptidase activity	A

Table 2: TWEAK-Fn14 cell signalling pathway. Abbreviations: CC-ST (Cell communication-signal transduction), ST (Signal transduction), CC (Cell communication), NNAM (Nucleotide and nucleic acid metabolism), A (Apoptosis)

5 Related work

A variety of formalisms such as Petri nets, process algebras and temporal logic to cite only a few, are being fruitfully used for modelling biological systems. In spite of their peculiar advantages and weak points, these methods share some useful features. Their compositionality allows to manage efficiently the flow of biological data either by easily integrating new knowledge into existing models [5] or to specify larger systems simply by composing smaller sub-systems. Moreover these formalisms permit to describe a system at different levels of abstractions within the same framework. Our languages is described by a process calculus, and thus has relations with several other languages having a similar characteristic. We can mention the work of Regev and Shapiro which proposed the "cell-as-computation" abstraction modelled by a process calculus: a system of interacting molecules is rendered as a system of concurrent computational entities [27]. Then Bio-PEPA [7], BlenX [11] and Brane calculi [4] to cite only a few, have been extensively used for modelling biological systems and building toolkits.

Several formalisms and strategies have been proposed to address specifically the modelling of spatial information in biological interactions. In BIOCHAM [14] different locations for molecular compounds can be defined by giving them a name. These symbolic locations may represent cell compartments (e.g. nucleus, cytoplasm, etc.) or different cells (e.g. c11, c12, c21, c22). P-systems allow a parallel and nondeterministic description of membranes by means of trees or graphs [1,25,21]. Brane Calculi [4] and its variants, Bioambients [26], Beta binders [12] and the K-calculus [20] are based on or get inspiration from the $\pi-$calculus, enriched with primitives for compartmentalised computation. In the π@-calculus [8] the modelling of compartments remains closer to the original π-calculus. Even though spatial information can be effectively encoded also by the formalisms mentioned above, we aim at gaining this possibility also for a CCP based framework in a general and flexible way. This framework allows to exploit an already running platform base for the implementation [6]. Thus we can provide a modelling toolkit that both enjoys all the advantages of CCP and express spatial information. Such a framework can be easily extended to deal with biologically relevant issues, e.g. time delays or non deterministic/stochastic computations [5] and has a direct connection with theorem proving [23].

Even if in `bccp` we are not dealing directly with quantitative information, our framework can be extended for modelling reactive computations [22]. The attractiveness of our `bccp` framework relies on its strong adequacy with respect to SELL, as shown in [23]. This means that proof search corresponds to computational steps and vice versa. Thus, we can express properties of the system via SELL and prove if a given specification (process) satisfies the property. From a biological point of view, this methodology can give rise to a better understanding of cell signalling pathways by means of the formalisation and proofs of properties for protein-protein interactions (PPIs) in biochemical networks.

6 Conclusions and future research

In this paper we report on a formal method that can be used for specifying space in models of biochemical systems described in CCP. Essentially this method consists in embedding Linear Logic with Subexponentials in a CCP framework by improving the constraint system to express constraints as formulas in a suitable fragment of SELL. We then discussed two case studies represented by a model of a G-Protein Coupled Receptors based signalling pathway, and by a more complex signaling pathway called TWEAK (TNF related Weak inducer of apoptosis), which is involved in several diseases. In this context we showed how our technique allows to describe a set of biochemical interactions taking into account locality and spatial constraints.

As discussed in this paper and in our previous works, the characteristics of the CCP formalism turn out to be particularly suitable for modelling biological phenomena. The work presented here enhances the expressiveness of CCP allowing to describe explicitly important features of living systems in the context of a well-assessed framework. We are currently developing this proposal in several directions. On the one hand we are improving our simulation tool BioWayS for modelling (non spatial) biological pathways allowing biochemical reactions to occur in a SELL-constraint system and, thus, considering space as a possible constraint. On the other hand we are studying how SELL based specifications of living systems can be used to prove properties which can be interesting from a biological point of view, e.g. following [3].

References

1. R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, G. Pardini, and L. Tesei. Spatial p systems. *Natural Computing*, 10(1):3–16, 2011.
2. M. Bhattacharjee et al. A bioinformatics resource for tweak-fn14 signaling pathway. *Journal of Signal Transduction*, 2012:10 pages, 2012.
3. C. Bodei, A. Bracciali, D. Chiarugi, and R. Gori. A taxonomy of causality-based biological properties. In *Proceedings of FBTC 2010*, pages 116–133, 2009.
4. L. Cardelli. Brane calculi. In Vincent Danos and Vincent Schächter, editors, *In Proc. of CMSB*, volume 3082 of *LNCS*. Springer, 2005.
5. D. Chiarugi, M. Falaschi, C. Olarte, and C. Palamidessi. Compositional modelling of signalling pathways in timed concurrent constraint programming. In *Proc. of ACM, BCB'10*, pages 414–417, New York, NY, USA, 2010. ACM.
6. D. Chiarugi, D. Hermith, M. Falaschi, C. Olarte, and C. Palamidessi. Bioways: A ccp-based tool for simulating biological systems. Available at `http://www.dianahermith.com/bioways/`.
7. F. Ciocchetta, A. Degasperi, J. Hillston, and M. Calder. Some investigations concerning the ctmc and the ode model derived from bio-pepa. *Electron. Notes Theor. Comput. Sci.*, 229:145–163, February 2009.
8. C.Versari and R. Gorrieri. π@: A π-based process calculus for the implementation of compartmentalised bio-inspired calculi. In G. Zavattaro M. Bernardo, P. Degano, editor, *Formal Methods for Computational Systems Biology*, pages 449–506, 2008.

9. V. Danos, J.B. Joinet, and H. Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In *Kurt Gödel Colloquium*, volume 713 of *Lecture Notes in Computer Science*, pages 159–171. Springer, 1993.
10. L. Dehmelt and Philippe I. H. Bastiaens. Spatial organization of intracellular communication: insights from imaging. *Nature Reviews Molecular Cell Biology*, 11:440–452, 2010.
11. L. Dematte, R. Larcher, A. Palmisano, C. Priami, and A. Romanel. Programming biology in blenx. *In Systems Biology for Signaling Networks*, 1(1):777–820, 2010.
12. L. Dematte, C. Priami, and A. Romanel. The beta workbench: a computational tool to study the dynamics of biological systems. *Briefings in Bioinformatics*, 9(5):437–449, 2008.
13. F. Fages, P. Ruet, and S. Soliman. Linear concurrent constraint programming: Operational and phase semantics. *Information and Comput.*, 165(1):14–41, 2001.
14. F. Fages and S. Soliman. Formal cell biology in biocham. *In Proc. of SFM, LNCS*, 5016, 2008.
15. J.Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
16. Rémy Haemmerlé. Observational equivalences for linear logic concurrent constraint languages. *TPLP*, 11(4-5):469–485, 2011.
17. H. Kitano. *Foundations of Systems Biology*. The Massachusetts Institute of Technology Press, 2001.
18. S. Knight, C. Palamidessi, P. Panangaden, and F. D. Valencia. Spatial and epistemic modalities in constraint-based process calculi. In *CONCUR*, volume 7454 of *LNCS*, pages 317–332, 2012.
19. Prof. Dr. G. Krauss. *Biochemistry of Signal Transduction and Regulation, Third Edition*. Wiley-VCH Verlag GmbH and Co. KGaA, 2003.
20. C. Laneve and F Tarissan. A simple calculus for proteins and cells. *Electr. Notes Theor. Comput. Sci.*, 171(2):139–154, 2007.
21. V. Manca and L. Bianco. Biological networks in metabolic p systems. *Biosystems*, 372:165–182, 2007.
22. M. Nielsen, C. Palamidessi, and F. Valencia. Temporal concurrent constraint programming: Denotation, logic and applications. *Nordic Journal of Computing*, 9(1):145–188, 2002.
23. V. Nigam, C. Olarte, and E. Pimentel. A general proof system for modalities in concurrent constraint programming. In *CONCUR*, pages 410–424, 2013.
24. Carlos Olarte, Camilo Rueda, and Frank D. Valencia. Models and emerging trends of concurrent constraint programming. *Constraints*, pages 1–44, 2013.
25. G. Păun and G. Rozenberg. A guide to membrane computing. *Theoretical Computer Science*, 287(1):73–100, 2002.
26. A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E.Y Shapiro. Bioambients: an abstraction for biological compartments. *Theor. Comp. Sci.*, 325(1):1–40, 2004.
27. A. Regev and E. Shapiro. Cellular abstractions: Cells as computation. *Nature*, 419(343), 2002.
28. V. Saraswat, M. Rinard, and P. Panangaden. Semantic foundations of concurrent constraint programming. In *POPL'91*, pages 333–352. ACM, 1991.
29. B. Di Ventura, C. Lemerle, K. Michalodimitrakis, and L. Serrano. From in vivo to in silico biology and back. *Nature.*, (5127):527–533, 2006.
30. O. Wolkenhauer, H. Kitano, and Cho Kwang-Hyun. Systems biology: Looking at opportunities and challenges in applying systems theory to molecular and cell biology. *IEEE Control Systems Magazine*, 23(4), 2003.