Where membranes meet complexes.

Luca Cardelli, Sylvain Pradalier

June 15, 2005

Abstract

We introduce a calculus handling complexation of molecules and membranes. The approach is based on adding dynamic interfaces to processes, which induce bonds between molecules. The calculus is then extended with a notion of hierarchy to handle membranes.

Introduction In biochemistry, proteins and other molecules have a common way of interacting among themselves by forming "complexes". A complex is a combinations of two or more molecules attached together along complementary surfaces. *Complexation* is the formation of a complex and *decomplexation* is the breaking up of a complex, both of which may be actively triggered, e.g., by another complexation and decomplexation interactions.

When modeling even the simplest biological systems, the issue immediately comes up of how to model complexation. Shapiro and Regev [8, 4] proposed to use a fresh private channel in π-calculus to represent complexation, and a synchronization on that channel followed by discarding the channel represents decomplexation. This technique is flexible and allows many interesting variations, and is available within a well understood calculus. However, this technique uses some high-power features (private scopes, name passing) to represent what is essentially a combinatorial operation that, like parallel composition, ought to be expressible as a binary operator.

In this paper we explore using complexation as an operator, which turns out not to be so trivial to implement. We base our approach on attaching dynamic "interfaces" to processes to more easily track the "surfaces" along which processes interact to form complexes. Our notion of dynamic interfaces is similar to the mechanisms explored in beta-binders[6, 7]; however, we preserve binary complementary interactions as the fundamental mode of interactions between processes.

Membranes are another fundamental features in biology. They permit organisms to divide the space into compartments, thus enabling different living conditions for their contents. Brane Calculi[1] and Projective Brane Calculus[3] are languages specifically dealing with membranes. They provided a strong framework for describing or modeling membranes systems but they lack an elegant way of representing molecules and even more so, complexation. This paper proposes a first language that handles molecules *and* membranes. Because membranes can stick to each other like molecules, we represent membranes as molecules with a content. This is the same step that leads from π-calculus[5] to ambient-calculus[2].

Outline First we present the formalism for complexation. Then we show examples that illustrate the different operators. Next we extend the calculus with a notion of hierarchy that represents nesting membranes and we give an example of a biological case where both complexes and membranes are needed. We finally discuss possible enhancements to the syntax, and perspectives.

Acknowledgments A discussion with Tony Hoare provided the initial direction for this work. Vincent Danos pointed out some mistakes and helped to explain intuitions behind the calculus.

1 Complexation

In biochemistry complementary sites of molecules can be linked to form bonds. So the possibilities of interactions of a molecule are determined by its sites. The set of the sites of a molecule is called its interface.

This remark leads us to design processes with an interface, which controls the behavior of a process. We call our processes, molecules. We want an interface to be dynamic, thus processes are able to offer or retract a site on their interface.

As it is shown by examples (part 2), other operators are needed to model biological systems. Not surprisingly we obtain something close to π-calculus: for instance, we need synchronizations and restrictions on the scope of a site.

Before presenting formally the syntax we recall some basic notions about multisets.

1.1 Some notations for multisets

We write $\subseteq^+$, $\bigcap^+$, $+$ and $-$ for the inclusion, intersection, sum and difference of multisets. We consider multisets on $\Sigma^- \bigcup \Sigma^+$ and on those we define the multisets: $\overline{S} = \{a^+ | a^- \in S\}$ and $S^* = S \bigcup \overline{S}$. The "complementary intersection" is: $S \Box T = (S \bigcap^+ \overline{T})^*$. We also note $S \amalg T$ for $S + T - S \Box T$. As $(a^+ \in S \Box T) \Leftrightarrow (a^- \in S \Box T)$, we write $a \in S \Box T$ for $(a^+ \in S \Box T \wedge a^- \in S \Box T)$.

1.2 Syntax

- $a, b, c, \ldots \in \Sigma^- \bigcup \Sigma^+$ and $S, T, \ldots \in Multisets(\Sigma^- \bigcup \Sigma^+)$

- $P, Q, R, \ldots ::= \quad A_S \quad | \quad P\&P \quad | \quad (\nu a)P \quad | \quad X \quad | \quad rec_X.P$
- $A, B, C, \ldots ::= \quad 0 \quad | \quad A + B \quad | \quad A|B \quad | \quad \alpha.A$
- $\alpha ::= \quad a^{\pm}\langle S\rangle \quad | \quad a^{\pm}(S) \quad | \quad \mathit{offer}(a) \quad | \quad \mathit{retract}(a) \quad | \quad \mathit{fork}(P)$

$\Sigma^- \bigcup \Sigma^+$ is the set of sites. The second line defines syntax for molecules: it is either a sets of actions A with some interface S, or two molecules put in parallel with the parallel operator & (or complexation operator), or a restriction on the name of a site, or a recursion variable or a recursion on another molecule. Actions sets are defined in the third line: there is choice, parallel or shuffle, and prefix. Finally we have basic actions: polyadic output, polyadic input, offer of a site, retract of a site, fork of a molecule. Further explanation will be given with the operational semantics (1.5).

The main difference with π-calculus is the complexation operator & that behaves like a parallel except for interfaces. It is this operator that makes bonds between molecules. Other differences are restrictions that can only occur at the top-level (they cannot be prefixed, for instance), and the duality $\pm$ of sites that represents complementarity of sites.

Then we define the interface I(P) of a sets of molecules:

$$I(A_S) = S, \qquad I((\nu a)P) = I(P) - \{a^+, a^-\}, \qquad I(P\&Q) = I(P) \amalg I(Q)$$
$$I(X) = \emptyset, \qquad I(rec_X.P) = I(P)$$

For instance if $I(P) = \{a^+, a^+, b^-\}$ and $I(Q) = \{a^-, c^+\}$ then $I(P\&Q) = \{a^+, a^+, b^-\} + \{a^-, c^+\} - \{a^+, a^-\} = \{a^+, b^-, c^+\}$. The interesting case is $I(P\&Q) = I(P) \amalg I(Q)$. It is here that complexation occurs. Intuitively the $\amalg$ operator hides a a^+ on the left for each a^- on the right and vice versa. Each such hidden pair corresponds to a bond on the site "a": if there is a bond on a, then the two corresponding sites do not appear anymore in the interface. When we want to insist on hidden sites we index & with the names of theses sites.

We have to distinguish between a^+ and a^- that is the two end of a bond, or the sites of the two molecules, because bonds on molecules correspond to complementary shapes. Thus if molecule X can links with molecules Y and Z on a site a we do not want Y and Z to be able to link on a.

1.3 Free names

The definition of free names is quite obvious. We just detail cases that are not in the classical π-calculus:

- $fn(A_S) = fn(A) \bigcup Set(S)$ where Set(S) is the set-projection of the multiset S.

- $fn(a\langle S\rangle.A) = \{a\} \bigcup fn(A) \bigcup S$
- $fn(a(X).A) = \{a\} \bigcup (fn(A) \setminus X)$
- $fn(\mathit{offer}(a).A) = fn(\mathit{retract}(a).A) = fn(\mathit{fork}(P).A) = fn(A) \bigcup \{a\}$

1.4 Structural Congruence

The structural congruence is derived from the one of the π-calculus.

$$\frac{P \equiv_\alpha Q}{P \equiv Q}$$

$$\begin{array}{cl}
P\&Q \equiv Q\&P, & P\&(Q\&R) \equiv (P\&Q)\&R, \\
P\&0 \equiv P, & \dfrac{P \equiv P'}{P\&Q \equiv P'\&Q} \\
(\nu x).(\nu y).P \equiv (\nu y).(\nu x).P, & (\nu x).0 \equiv 0 \\
(\nu x).P\&Q \equiv (\nu x).(P\&Q) & \quad if\ x \notin fn(Q) \\
\frac{P \equiv Q}{rec_X.P \equiv rec_X.Q}, & \dfrac{A \equiv B}{A_S \equiv B_S} \\
A + B \equiv B + A, & A + (B + C) \equiv (A + B) + C, \\
A + 0 \equiv A, & \dfrac{A \equiv A'}{A + B \equiv A' + B} \\
A|B \equiv B|A, & A|(B|C) \equiv (A|B)|C, \\
A|0 \equiv A, & \dfrac{A \equiv A'}{A|B \equiv A'|B}
\end{array}$$

$$\frac{A \equiv A'}{\alpha.A \equiv \alpha.A'}$$

1.5 Operational Semantics

Operational semantics of the π-calculus is often presented in a labeled version. We do not need it here and thus prefer the simpler unlabeled version.

- $\frac{P \equiv P' \ ,\ P' \to Q' \ ,\ Q' \equiv Q}{P \to Q}$(Cong)
- $(a^{\pm}\langle R\rangle.A)_S \quad \& \quad (a^{\mp}(X).B)_T \quad \to \quad A_S \quad \& \quad B\{R/X\}_T$ if $a \in S \Box T$: (Com)
- $\frac{P \quad \to \quad P'}{P\&Q \quad \to \quad P'\&Q}$(&) , $\frac{P \quad \to \quad P'}{(\nu a)P \quad \to \quad (\nu a)P'}(\nu)$ and $\frac{P\{rec_x.P/X\} \quad \to \quad Q}{rec_X.P \quad \to \quad Q}$(Rec)
- $\frac{A_S\&P \quad \to \quad A'_{S'}\&P'}{(A+B)_S\&P \quad \to \quad A'_{S'}\&P'}$(Sum): if $A \neq A'$ and $\frac{A_S\&P \quad \to \quad A'_{S'}\&P'}{(A|B)_S\&P \quad \to \quad (A'|B)_{S'}\&P'}$(Par)
- $(\mathit{offer}(a).A)_S \quad \to \quad A_{S+a}$: (Offer)
- $(\mathit{retract}(a).A)_S \quad \to \quad A_{S-a}$ if $a \in S$: (Retract)
- $(\mathit{fork}(P).A)_S \quad \to \quad A_S \quad \& \quad P$: (Fork)

Rules Cong, &, ν and Rec are classical rules. Sum and Par are structural rules to handle choice and parallel operator on actions. Note that & and Par rules are similar : both operators are parallel operators. The difference is that & permits communication and is at the level of molecules while | is in a molecule and just permits shuffle of actions.

Offer and Retract rules are basic modifications of interfaces. Fork is the synthesis of a new molecule.

The rules Com is remarkable. We ask that $a \in S \Box T$ because we want to restrict communication to linked molecules. Only these molecules should be able to exchange information. This implies that communication occurs between an a^+ and a^-. So a second duality is added to input/output.

One can think to fuse the two dualities into one but then one has to symmetrize the communication with bounded output, for instance. But such output aren't as well understood as simple ones, thus we prefer to keep simple outputs.

2 Use of different connectors and examples

2.1 The graph structure of the complex isn't forgotten

Here is a basic example illustrating how communications permit to synchronize molecules, and showing that we do not loose the graph-structure of a complex. Let:

- $P = (\mathit{offer}(1^+).1^+\langle\rangle.\mathit{offer}(2^+).2^+\langle\rangle.2^+\langle\rangle.\mathit{retract}(1^+))\{\}$
- $Q = (\mathit{offer}(1^-).1^-\langle\rangle.\mathit{offer}(3^+).3^+\langle\rangle)\{\}$
- $R = ((\mathit{offer}(2^-)|\mathit{offer}(3^-)).(2^-\langle\rangle|3^-\langle\rangle).2^-\langle\rangle.\mathit{ret}3^-)\{\}$

(P|Q).(P'|Q') is an abbreviation for P.Q.(P'|Q') + Q.P.(P'|Q')).
First P and Q offer and link on 1, then they synchronized so that they know they can try to link R. Then R offers 2 and 3 and synchronized with P and Q. Now all the three molecules know that they are linked. P and R synchronize once again to agree on the release of Q.

2.2 Associativity and NaCl

Let:

- $Na = (\mathit{offer}(e^+).Na')\{\}$
- $Cl = (\mathit{offer}(e^-).Cl')\{\}$
- $S = Na_1 \quad \& \quad Na_2 \quad \& \quad Cl_1 \quad \& \quad Cl_2$

We indexed molecules because we want to see the variations permitted by the associativity of the &.
S can evolve by linking Na and Cl molecules. This models the ionization: $Na + Cl \rightarrow Na^+ + Cl^-$:

$$\begin{array}{rlllll} S & \rightarrow^* & [\ (Na_1)_{e^+}|(Na'_2)_{e^+}\ ] & \&_{e^{+2},e^{-2}} & [\ (Cl'_1)_{e^-}|(Cl'_2)_{e^-}\ ] \\ & \equiv & [\ (Na'_1)_{e^+}\&_{e,e^-}(Cl'_1)_{e^-}\ ] & | & [\ (Na'_2)_{e^+}\&_{e,e^-}(Cl'_2)_{e^-}\ ] \\ & \equiv & [\ (Na'_1)_{e^+}\&_{e,e^-}(Cl'_2)_{e^-}\ ] & | & [\ (Na'_2)_{e^+}\&_{e,e^-}(Cl'_1)_{e^-}\ ] \end{array}$$

The difference between the last two lines is the pairs of linked molecules. In the last one we have (Na_1, Cl_2) and (Na_2, Cl_1) while in the other one we have (Na_1, Cl_1) and (Na_2, Cl_2). This is permitted by the rule of associativity in the structural congruence. In the first line we do not detail which Na^+ is linked with which Cl^-. This is convenient for such links as $Na^+ - Cl^-$ that are always shifting from a molecule to another one. The intuition is that all different configurations coexist. The system is always shifting from one to the other.

2.3 Communication and strong links: Use of νa

Now we want to model stable bonds. When such a link is made, no rearrangement with other molecule is possible. The only way to remove such a link is to retract one of the linked site. This corresponds to covalent bonds for instance. Let:

- $P = (\nu c)(\quad offer(p^-).p^-\langle c^+\rangle.retract(p^-).offer(c^-).$
  $Work.retract(c^-).offer(p^-)\quad)_{\{\}}$
- $Q = (\quad offer(p^+).p^+(x).retract(p^+).offer(x).$
  $Work'.retract(x).offer(p^+)\quad)_{\{\}}$
- $S = P\&Q$

Work and Work' are some actions to do when the strong link is established. They begin and end by synchronization on the site c for P and x for Q (which are the same actually).

$$\begin{array}{rcl}
S & = & (\nu c)[\quad (\mathit{offer}(p^-).p^-\langle c^+\rangle.\mathit{retract}(p^-).\mathit{offer}(c^-). \\
& & \quad Work.\mathit{retract}(c^-).\mathit{offer}(p^-))_{\{\}} \quad] \\
& \& & (\mathit{offer}(p^+).p^+(x).\mathit{retract}(p^+).\mathit{offer}(x). \\
& & \quad Work'.\mathit{retract}(x).\mathit{offer}(p^+))_{\{\}} \\
\xrightarrow{\tau,\tau} & & (\nu c)[\quad (p^-\langle c^+\rangle.\mathit{retract}(p^-).\mathit{offer}(c^-). \\
& & \quad Work.\mathit{retract}(c^-).\mathit{offer}(p^-))_{\{p^-\}} \quad] \\
& \&_{p^+,p^-} & (p^+(x).\mathit{retract}(p^+).\mathit{offer}(x). \\
& & \quad Work'.\mathit{retract}(x).\mathit{offer}(p^+))_{\{p^+\}} \\
\xrightarrow{\tau} & & (\nu c)[\quad (\mathit{retract}(p^-).\mathit{offer}(c^-). \\
& & \quad Work.\mathit{retract}(c^-).\mathit{offer}(p^-))_{\{p^-\}} \\
& \&_{p^+,p^-} & (\mathit{retract}(p^+).\mathit{offer}(x). \\
& & \quad Work'.\mathit{retract}(x).\mathit{offer}(p^+))_{\{p^+\}} \quad] \\
\xrightarrow{\tau,\dots} & & (\nu c)[\quad (Work.\mathit{retract}(c^-).\mathit{offer}(p^-))_{\{c^-\}} \\
& \&_{c^+,c^-} & (Work'.\mathit{retract}(x).\mathit{offer}(p^+))_{\{c^+\}} \quad]
\end{array}$$

First step is achieved by basics offer of p^+ and p^-. The bond on site p is automatically made. Second one is due to a communication on p. The molecules exchange the new site c. Then they retract sites p^+ and p^- and offers c^+ and c^-. Even if there is some associativity rearrangement between the second and the third step, that is some P or Q link on site p with these molecules, the privacy of c permit that these two molecules will eventually link. Indeed they will eventually retract p^+ and p^- and offer c^+ and c^-. As c is private no other molecule can make a bond on c. Thus this bond on site c is strong and won't be broken unless one of them retracts.

2.4 Synthesis of Transcription factors: $fork(P)$ and $rec_X.P$

A very basic model of transcription factors synthesis shows how $fork(P)$ models synthesis of a new molecule and how $rec_X.P$ models infinite behavior of molecule. Let:

- $dna_1 = rec_X[\quad (\quad tf_2^+().fork(TF_1\&X)\quad)_{tf_2^+}\quad]$
- $TF_i = rec_Y[\quad (\quad tf_i^-\langle\rangle.Y + 0\quad)_{tf_i^-}\quad]$ for i = 1,2.
- and $S = dna_1\&TF_2$

After dna_1 has synchronized with TF_2 on the site tf_2, the fork action releases a TF_1 molecule which is product of the reaction and another dna_1 molecule as dna isn't destroyed in the process.

When adding interfaces to processes to support complexation, we end up with processes "inside" interfaces. How can then an inner process trigger the creation of a new entity at the outer level ? This cannot be done by simple parallel composition, as usual, because now we are at the wrong level. Therefore we use a fork primitive to allow an "inner" process to create an "outer" entity.

3 Complexation of Membranes

We want now to represent membranes. This can be easily done by adding a content to molecules. We have to design an oriented version of the calculus as in [3] because we do not want the shifting effects of the associativity congruence rule (see part2.2) to occur between two sides of a membrane. Such a case would correspond to a receptor that keeps changing sides of its membrane and this is not relevant for biology. Moreover [3] has shown how oriented actions are better from both biological and mathematical points of view.

3.1 Syntax

We take the notation of [3] for action of membranes: f comes from fuse, w from wrap and b from bubble.

- $a, b, c, \ldots \in \Sigma^- \bigcup \Sigma^+$ and $S, T, \ldots \in Multisets(\Sigma^- \bigcup \Sigma^+)$
- $P, Q, R, \ldots ::= \quad A[\![P]\!]_S \quad | \quad P\&P \quad | \quad (\nu a)P \quad | \quad X \quad | \quad rec_X.P$
- $A, B, C, \ldots ::= \quad 0 \quad | \quad A + B \quad | \quad A|B \quad | \quad \alpha^{\uparrow}.A \quad | \quad \alpha^{\downarrow}.A$
- $\alpha ::= \quad a\langle S\rangle \quad | \quad a(S) \quad | \quad \mathit{offer}(a) \quad | \quad \mathit{retract}(a) \quad | \quad [S]$
  $| \quad f \quad | \quad f^{\perp} \quad | \quad w \quad | \quad w(A[\![]\!]_{S,T}) \quad | \quad b(A[\![]\!]_{S,T})$

Arrows on actions give orientation to actions. $[S]$ is the action corresponding to Bind&Release of [1]: it permits molecules to cross membranes. Here we do not match on names of molecules but on interfaces. Actions *f, w and b* are for membranes reactions. They are precisely described in [1, 3]. We do not need anymore the fork action because of the pino action that plays its role. Interfaces are now defined by:

$$I(A[\![P]\!]_{S,T}) = S, \qquad I((\nu a)P) = I(P) - a, \qquad I(P\&Q) = I(P) \amalg I(Q)$$
$$I(X) = \emptyset, \qquad I(rec_X.P) = I(P)$$

One may want to define the notion of inner and outer interface as we are speaking of two sets of interfaces for a molecule. Actually the inner interface notion will occur if we allow holes in our molecule. When we speak of interface we are concerned with what the molecule is showing to its environment and not with the sites inside.

3.2 Free names

We have to define free names for the new operators:

- $fn(A[\![P]\!]_{S,T}) = fn(A) \bigcup fn(P) \bigcup Set(S) \bigcup Set(S)$
- $fn(f^{\updownarrow}.A) = fn(f^{\perp\updownarrow}.A) = fn(w^{\updownarrow}.A) = fn(A)$
- $fn(w^{\updownarrow}(B[\![]\!]_S).A) = fn(b^{\updownarrow}(B[\![]\!]_S).A) = fn(A) \bigcup fn(B) \bigcup Set(S)$
- $fn([S\rangle^{\updownarrow}.A) = fn(A) \bigcup Set(S)$

3.3 Structural Congruence

Some rules need to be added to or changed in the structural congruence:

- $\frac{A \equiv B \quad P \equiv Q}{A[\![P]\!]_S \equiv B[\![Q]\!]_S}$
- $\frac{A \equiv B \quad \alpha \equiv \beta}{\alpha^{\updownarrow}.A \equiv \beta^{\updownarrow}.B}$
- $\alpha \equiv \alpha$, $\frac{A \equiv B}{w(A[\![]\!]_S) \equiv w(B[\![]\!]_S)}$ and $\frac{A \equiv B}{b(A[\![]\!]_S) \equiv b(B[\![]\!]_S)}$

3.4 Operational Semantics

The list is a bit long. This is because orientation doubles reactions.

- $(a^{\pm\uparrow}[\![R]\!].A)[\![P]\!]_{S,S'} \quad \& \quad (a^{\mp\uparrow}(X).B)[\![Q]\!]_{T,T'} \quad \rightarrow \quad A[\![P]\!]_{S,S'} \quad \& \quad B\{R/X\}[\![Q]\!]_{T,T'}$ if $a \in S \Box T$
- $(a^{\pm\downarrow}\langle R\rangle.A)[\![\quad (a^{\mp\uparrow}(U).B)[\![Q]\!]_{T,T'}, Q' \quad]\!]_{S,S'} \quad \rightarrow \quad A[\![\quad B\{R/U\}[\![Q]\!]_{T,T'}, Q' \quad]\!]_{S,S'}$ if $a \in S' \Box T$

These are the two rules for communication. The first one is just the extension to the oriented case of the rule (Com). The second one is the case of communication between nested membranes. Of course for the second rule there is also the version where input and output are exchanged.

- $(\mathit{offer}(a)^{\uparrow}.A)[\![P]\!]_{S,T} \quad \rightarrow \quad A_{S+a,T}$
- $(\mathit{offer}(a)^{\downarrow}.A)[\![P]\!]_{S,T} \quad \rightarrow \quad A_{S,T+a}$
- $(\mathit{retract}(a)^{\uparrow}.A)[\![P]\!]_{S,T} \quad \rightarrow \quad A_{S-a,T}$ if $a \in S$
- $(\mathit{retract}(a)^{\downarrow}.A)[\![P]\!]_{S,T} \quad \rightarrow \quad A_{S,T-a}$ if $a \in T$

Theses rules are just extensions to the oriented case of Offer and Retract rules.

- $A[\![]\!]_{U,U'} \& ([S\rangle^{\uparrow}.B)[\![R]\!]_{T,T'} \quad \rightarrow \quad (B)[\![A[\![]\!]_{U,U'} \& R]\!]_{T,T'} \qquad$ if $S = U$

- $([S\rangle^{\downarrow}.B)[\![A[\![]\!]_{U,U'} \& R]\!]_{T,T'} \quad \rightarrow \quad A[\![]\!]_{U,U'} \& (B)[\![R]\!]_{T,T'} \qquad \text{if } S = U$

These rules just look for a molecule with the proper interface and make it cross the membrane. The content of A is empty because we want it to be a molecule.

- $(b^{\uparrow}(A[\![]\!]_{S,S'}).B)[\![P]\!]_{T,T'} \quad \rightarrow \quad B[\![P]\!]_{T,T'} \quad \& \quad A[\![]\!]_{S,S'}$
- $(b^{\downarrow}(A[\![]\!]_{S,S'}).B)[\![P]\!]_{T,T'} \quad \rightarrow \quad B[\![P \quad \& \quad A[\![]\!]_{S,S'}]\!]_{T,T'}$

Emission of a new membrane inside or outside. These are Drip and Pino reactions.

- $(w^{\uparrow}.A)[\![P]\!]_{S,S'} \quad \& \quad (w^{\uparrow}(C[\![]\!]_{T,T'}).B)[\![Q]\!]_{U,U'} \quad \rightarrow \quad B[\![C[\![A[\![P]\!]_{S,S'}]\!]_{T,T'} \quad \& \quad Q]\!]_{U,U'}$
- $(w^{\downarrow}(C[\![]\!]_{T,T'}).B)[\![(w^{\uparrow}.A)[\![P]\!]_{S,S'} \quad \& \quad Q]\!]_{U,U'} \quad \rightarrow \quad B[\![Q]\!]_{U,U'} \quad \& \quad C[\![A[\![P]\!]_{S,S'}]\!]_{T,T'}$
- $(w^{\downarrow}.A)[\![R \quad \& \quad (w^{\uparrow}(C[\![]\!]_{U,U'}).B)[\![Q]\!]_{T,T'}]\!]_{S,S'} \quad \rightarrow \quad A[\![C[\![B[\![R]\!]_{T,T'} \quad \& \quad Q]\!]_{U,U'}]\!]_{S,S'}$

Reactions Phago, Bud and Swap. In each of them a membrane engulfs itself in another membrane, wrapping itself in a piece of new membrane. Swap reaction reaction could seem very strange. Actually Swap is Phago or Bud reaction seen from the point of view of P. See [3] for more details.

- $(f^{\perp\downarrow}.A)[\![(f^{\uparrow}.B)[\![P]\!]_{T,T'} \& Q]\!]_{S,S'} \quad \rightarrow \quad P \quad \& \quad (A|\overline{B})[\![Q]\!]_{S+T',S'+T}$ where $\overline{B}$ just reverses orientations of actions in B.
- $(f^{\perp\uparrow}.A)[\![P]\!]_{S,S'} \& (f^{\uparrow}.B)[\![Q]\!]_{T,T'} \quad \rightarrow \quad (A|B)[\![P \& Q]\!]_{S+T,S'+T'}$

These are Exo and Mate reactions. They are both fusions of membranes either vertical (Exo) or horizontal (Mate). Note that Exo reverses orientations of actions in B and add T' to S and T to S'.

The structural rules Sum and Par extend to:

- $\dfrac{A[\![P]\!]_{S,S'} \ \& \ R \quad \rightarrow \quad A'[\![P']\!]_{T,T'} \ \& \ R'}{(A|C)[\![P]\!]_{S,S'} \ \& \ R \quad \rightarrow \quad (A'|C)[\![P']\!]_{T,T'} \ \& \ R'}$
- $\dfrac{A[\![P]\!]_{S,S'} \ \& \ R \quad \rightarrow \quad A'[\![P']\!]_{T,T'} \ \& \ R'}{(A+C)[\![P]\!]_{S,S'} \ \& \ R \quad \rightarrow \quad A'[\![P']\!]_{T,T'} \ \& \ R'}$ if $A \neq A'$

3.5 Modeling of a molecule repaired in a Goldi apparatus

We next present a small example to illustrate interactions between membranes and molecules. Let:

- $M = (\mathit{damaged}^{+\uparrow}\langle\rangle.\mathit{retract}(\mathit{damaged}^{+}).\mathit{offer}(\mathit{undamaged}^{+}))[\![]\!]_{\mathit{damaged}^{+},\_}$
- $G = (\mathit{golgi}|[\mathit{damaged}^{+}]^{\uparrow}|[\mathit{undamaged}^{+}]^{\downarrow})[\![\quad (\mathit{damaged}^{-}()^{\uparrow})[\![]\!]_{\mathit{damaged}^{-},\_} \quad]\!]_{\_,\_}$
- $S = M \& G$

M is a damaged molecule. G is a golgi apparatus containing a molecule, which can repair M. So M has to go in G to be repaired. Since M has proper interface: $damaged^+$, G can use its action $[damaged]^\uparrow$ to capture M and put it into its content. The content of G is then:

$(damaged^{+\uparrow}\langle\rangle.retract(damaged^+).offer(undamaged^+))[\![]\!]_{damaged^+,\_}$

& $(damaged^-()^\uparrow)[\![]\!]_{damaged^-,\_}$

The synchronization on the site *damaged* followed by the retracting of $damaged^+$ by M and the offering of $undamaged^+$ represent the repairing of M. The interface of M is now $undamaged^+$ and it can be released outside of G.

4 Possible extensions and Perspectives

4.1 Possible extensions

Different extensions are possible to enhance the syntax. A first interesting feature consists of test actions $?(a)$ and $\neg?(a)$ with theses rules:

- $(?(a).A)[\![P]\!]_{S,S'}\&Q \quad \rightarrow \quad (A)[\![P]\!]_{S,S'}\&Q \quad : \quad \text{if } a \in S\Box I(Q)$
- $(\neg?(a).A)[\![P]\!]_{S,S'}\&Q \quad \rightarrow \quad (A)[\![P]\!]_{S,S'}\&Q \quad : \quad \text{if } a \notin S\Box I(Q)$

Tests can be useful because synchronization does not permit to check if some site is not connected. It can also be a simpler way to check if a site is connected without requiring to “program” both sides of the link.

Another extension would be to the possibility to have molecules on membranes. Here a molecule can only be between membranes. So we loose the possibility of representing transmembranal molecules. A simple solution is to permit to have terms of the first calculus between membranes and on membranes. To compute the interface of a membrane we then have to add all interfaces of molecules sitting on this membrane.

4.2 Perspectives

An interesting perspective is to design a projective version of the calculus as in [3]; it means to build a “projective” equivalence relation that identifies molecules representing the same system under different points of view. This equivalence should be compatible with reduction. This implies that each reaction has a “symmetric” one with respect to the projective equivalence. This is almost achieved in the calculus we present. For instance the two communications rules are “symmetric” : they correspond to the same system seen from two different points of view. The only problem is the ν operator. Taking a point of view inside a ν requires to have some symmetric operator to ν. If one think of ν as beginning of a scope, its symmetric would end a scope.

Another perspective is to reduce the calculus to just what we need. For instance, examples do not use the full power of name passing communications. Our conjecture is that bounded outputs are sufficient to express the biological properties.

Finally a stochastic semantics is needed as a basis for simulations.

References

[1] Luca Cardelli. Brane calculi. April 2004.

[2] Luca Cardelli and Andrew Gordon. Mobile ambients. *Theoritical Computer Science*, 240/1:177–213, 2000.

[3] Vincent Danos and Sylvain Pradalier. Projective Brane-calculus. 3082:134–148, April 2004.

[4] A.Regev W.Silverman E.Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. 2001.

[5] Robin Milner. *Communicating and Mobile Systems.*

[6] Corrado Priami and Paola Quaglia. Beta binders for biological interactions. April 2004.

[7] Corrado Priami and Paola Quaglia. Operational patterns in beta binders. 2005.

[8] C.Priami A.Regev E.Shapiro W.Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. 2001.