



**HAL**  
open science

## Modelando Sistemas de Membranas en ntcc

Angela Villota, Jesus Aranda, Diaz Juan Francisco

► **To cite this version:**

Angela Villota, Jesus Aranda, Diaz Juan Francisco. Modelando Sistemas de Membranas en ntcc. XXXV Latin American Informatics Conference, Sep 2009, Pelotas, Brazil. inria-00430537

**HAL Id: inria-00430537**

**<https://inria.hal.science/inria-00430537v1>**

Submitted on 9 Nov 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modelando Sistemas de Membranas en *ntcc*<sup>\*</sup>

Ángela Villota<sup>1</sup>, Jesus Aranda<sup>2</sup>, and Juan Francisco Díaz<sup>1</sup>

<sup>1</sup> Universidad del Valle,  
Calle 13 No. 100 - 00. Cali, Colombia

<sup>2</sup> Universidad del Valle - Ecole Polytechnique  
21128 Palaiseau - Cedex, Francia

**Abstract** The *ntcc* calculus has been proposed as a formalism for modeling concurrent systems such as biological systems. In this paper we explore the expresiveness of the *ntcc* calculus for modeling, simulating and analyzing biological systems. A legitimate question is therefore: Is it possible to model membrane systems with *ntcc*? In this paper as our main contribution to the solution of this question, we propose a general mechanism for modeling membrane systems in *ntcc*. The application of that mechanism is illustrated with a model for the LDL cholesterol degradation pathway using membrane systems defined in *ntcc*.

**Key words:** *ntcc*, membrane systems, systems biology

## 1. Introducción

Los sistemas biológicos pueden ser vistos como sistemas en donde existen procesos que interactúan de forma concurrente y evolucionan en el tiempo. El área que aborda la biología desde esta perspectiva es llamada *biología de sistemas*[5,6]. En el contexto de la biología de sistemas se ha evidenciado la utilidad de las herramientas computacionales que permiten modelar y simular procesos biológicos conocidos. Entre estas herramientas se encuentran los cálculos de procesos que han mostrado ser útiles para modelar, simular y verificar propiedades de sistemas biológicos [1,2,7,8,9,10]. De entre las distintas posibilidades, este trabajo explora la expresividad de *ntcc*, un cálculo basado en el modelo CCP [4] (*Concurrent Constraint Programming*), en particular su adecuación para modelar sistemas de membranas. En [1,2,10] se muestra la utilización de *ntcc* para modelar procesos biológicos destacando los beneficios que trae la representación de información parcial, la independencia entre los constructores del cálculo y la demostración de propiedades a través de una lógica temporal (LTL).

Por otra parte, existen formalismos (Brane[9], BioAmbients[7], Membrane Systems[13]) que modelan el comportamiento de membranas biológicas y que

---

<sup>\*</sup> Research partially supported by the INRIA Equipe Associee FORCES. The work of Jesús Aranda has been supported by COLCIENCIAS (Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología “Francisco Jose de Caldas”), INRIA and EGIDE (Centre francais pour l’ accueil et les echanges internationaux).

son una alternativa diferente y sencilla para la descripción y simulación de sistemas biológicos. En este artículo se propone un método que permite describir sistemas biológicos en *ntcc* de forma similar a los formalismos que modelan membranas (Sistemas de Membranas). Inicialmente se realizó una comparación de la forma en que se modelan sistemas en Brane y BioAmbients, con respecto a *ntcc*, y en general con cualquier derivado de CCP. Se encontraron tres diferencias importantes:

- **El mecanismo de comunicación entre procesos:** en CCP la comunicación es asíncrona y se hace por medio del *store* que es visible para todos los procesos. Los estímulos o entradas están dados por medio de restricciones que los procesos pueden imponer o consultar. En BioAmbients/Brane, la comunicación es síncrona y requiere de la presencia de acciones y coacciones, o de canales de comunicación en los procesos.
- **La forma en que se construyen los sistemas:** en BioAmbients/Brane las membranas son contenedores de procesos que pueden tener otras membranas al interior (anidamiento), por tanto los sistemas en estos cálculos pueden ser vistos como jerarquías de membranas. En CCP los sistemas pueden ser especificados de forma modular, pero no existen jerarquías entre ellos.
- **Movilidad:** los cálculos de membranas proveen un conjunto de operaciones que permiten modelar el ingreso/salida de membranas; esto puede realizarse gracias a que existe movilidad como concepto del cálculo. En CCP no hay movilidad como concepto del cálculo y aunque es posible modelar la movilidad como está propuesto en [12], no es posible modelar el mismo comportamiento que en BioAmbients y Brane.

Las anteriores diferencias explican las dificultades que se presentan para modelar las operaciones de membranas con los constructores de *ntcc*. Pero esto no quiere decir que no se puedan modelar sistemas biológicos teniendo en cuenta la localización de sustancias y procesos (al estilo de membranas). Para ello es necesario definir un conjunto de procesos y variables que hagan posible modelar el comportamiento de las operaciones de membranas y especificar sistemas en los cuales exista una jerarquía de membranas en *ntcc*.

Este documento está organizado así: en la sección 2 se encuentra una breve descripción de la sintaxis de *ntcc*. En la sección 3 se describen las técnicas que permiten modelar sistemas de membranas en *ntcc*. En la sección 4 se aplican las técnicas de la sección 3 para modelar la ruta endocítica de la LDL del colesterol. La sección 5 presenta las conclusiones y el trabajo futuro.

## 2. El Cálculo *ntcc*

El cálculo de programación concurrente por restricciones temporal, *ntcc* [3] es un formalismo para sistemas reactivos. La noción de tiempo es introducida por medio de una secuencia de unidades de tiempo, en donde cada unidad tiene su propio almacén de restricciones (*store*) y dura mientras los procesos realizan sus computaciones y añaden (*tell*) o infieren (*ask*) información del *store*.

Los procesos **ntcc** están parametrizados en un *sistema de restricciones*, el cual especifica qué tipo de restricciones puede manejar el *store*. Los procesos  $P, Q, \dots \in Proc$  son construidos como se muestra en el cuadro 1.

El proceso **tell**( $c$ ) adiciona la restricción  $c$  al *store*, dejando a  $c$  disponible para los otros procesos en la unidad de tiempo actual. La selección con guarda  $\sum_{i \in I}$  **when**  $c_i$  **do**  $P_i$ , donde  $I$  es un conjunto finito de índices, representa un proceso que, en la unidad de tiempo actual, escoge de manera no determinista un proceso  $P_j$  ( $j \in I$ ) cuya guarda  $c_j$  es deducida del *store* (las alternativas no escogidas son descartadas). El proceso  $P \parallel Q$  representa la composición paralela entre  $P$  y  $Q$  (en una unidad de tiempo  $P$  y  $Q$  operan concurrentemente, comunicándose por medio del *store*). El proceso (**local**  $x$ )  $P$  se comporta como  $P$  pero la información de la variable  $x$  es local a  $P$ , i.e.  $P$  no puede ver la información sobre una variable global  $x$  y los procesos que no hacen parte de  $P$  no pueden ver la información producida por  $P$  sobre  $x$ . El proceso **next**  $P$  ejecuta el proceso  $P$  en la siguiente unidad de tiempo. El proceso **unless**  $c$  **next**  $P$  representa la activación de  $P$  en la siguiente unidad de tiempo si y solo si  $c$  no puede ser deducida a partir del *store* en la unidad de tiempo actual. El proceso  $!P$  ejecuta  $P$  en todas las unidades de tiempo a partir de la actual (esta es la versión de retraso del operador replicación del cálculo  $\pi$ ). El proceso  $\star P$  representa un retraso sin límite pero finito de  $P$ , i.e.  $P$  finalmente será ejecutado en alguna unidad de tiempo en el futuro.

$P, Q, \dots ::= \mathbf{tell}(c)$	Tell
$\sum_{i \in I} \mathbf{when} \ c_i \ \mathbf{do} \ P_i$	Selección con guarda
$P \parallel Q$	Composición Paralela
( <b>local</b> $x$ ) $P$	Comportamiento Local
<b>next</b> $P$	Retraso de Unidad
<b>unless</b> $c$ <b>next</b> $P$	Vencimiento
$!P$	Replicación
$\star P$	Retraso acotado

Cuadro 1. Sintaxis de *ntcc*

### 3. Técnicas para la Modelación de Sistemas de Membranas en *ntcc*

A continuación se presentan dos técnicas que son claves para modelar sistemas de membranas: modelar membranas, su localización y jerarquía (ver sección 3.1), y modelar reacciones químicas, las cuales son el mecanismo de interacción entre los objetos situados en las membranas (ver sección 3.2).

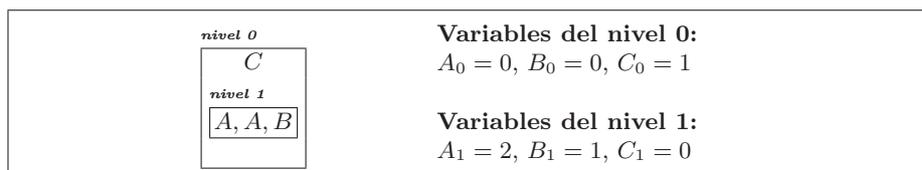
#### 3.1. Modelando Membranas, Localización y Jerarquías

Los procesos y definiciones propuestos a continuación están inspirados en la forma en que se modelan las membranas con Sistemas de Membranas (P Systems) que aparecen en [7].

Para la especificación de sistemas al estilo de membranas se utilizará un sistema de restricciones sobre dominios finitos. La ubicación de los elementos del sistema se realizará por medio de variables. La estructura del sistema es estática, los contenidos de las membranas son modificados debido al paso de sustancias a través de las membranas (entrada y salida de sustancias), pero no hay desplazamiento de membranas como tal. Una membrana en este enfoque, es una frontera que encierra una región llamada *nivel*. Los niveles que conforman un modelo pueden variar dependiendo de la descripción del proceso biológico que se desea modelar.

Al interior de cada nivel se encuentra ubicado un conjunto de objetos que representan las sustancias presentes en la descripción biológica del sistema. La ubicación de cada uno de estos objetos (y de los compuestos que conforman), está dada por una variable declarativa que puede tomar valores en los naturales, y denota la cantidad de cierto objeto (y compuesto) en un nivel dado.

En el cuadro 2 se muestra un ejemplo sencillo de un sistema con dos niveles, en donde se han ubicado tres tipos de objetos  $A$ ,  $B$  y  $C$ . Por cada tipo de objeto (v.gr.  $A$ ) se crean dos variables ( $A_0$  y  $A_1$ , en el caso de  $A$ ). Los valores de dichas variables representan la presencia y cantidad de los objetos de un tipo dado en cada nivel.



**Cuadro 2.** Ejemplo: Sistema con dos niveles y cuatro objetos.

### 3.2. Modelando Reacciones Químicas

Las reacciones químicas son procesos en los cuales una o más sustancias (reactivos) sufren transformaciones químicas para convertirse en otras (productos). Las reacciones químicas que se modelan a continuación son reacciones de *síntesis*, *descomposición* y *transporte*, o combinaciones de ellas. Una reacción de síntesis es aquella en que dos o más sustancias se unen para formar un compuesto. En una descomposición, una sustancia compleja (varias sustancias enlazadas) se divide en sus componentes. Las reacciones de transporte permiten mover una sustancia a través de las membranas del sistema.



De forma general, una reacción química de la forma  $R_1, \dots, R_n \rightarrow P_1, \dots, P_m$  es modelada en `ntcc` con el proceso `Reaction`. El proceso `Reaction` es genérico y permite modelar los tipos de reacciones mencionados (ver cuadro 3). Con el fin de dar continuidad a la evolución del proceso, los valores de las variables que están por fuera de una reacción son pasados a la siguiente unidad de tiempo sin

alteración; esto es representado por *Rest*. Para evitar inconsistencias es necesario que solo una reacción pueda ser activada en cada unidad de tiempo; esto puede lograrse usando selección no determinista (+) entre las reacciones y niveles.

$$Reaction \stackrel{def}{=} \mathbf{when} (R_1 \geq 1 \dots \wedge R_n \geq 1) \mathbf{do next} ( \mathbf{tell}( R_1 = R_1 - 1 \wedge \dots \wedge R_n = R_n - 1 \ P_1 = P_1 + 1 \wedge \dots \wedge P_m = P_m + 1 \wedge Rest) )$$

Nótese que el antecedente de una reacción es un conjunto de restricciones que pueden ser deducidas (o no) del *store*. En el caso en que estas restricciones puedan ser deducidas, se imponen otras en la siguiente unidad de tiempo (esto sucede a causa del constructor **next**).

En adelante los procesos que modelen reacciones químicas y que sean derivados de *Reaction* se representarán así: *condiciones*  $\rightarrow$  *consecuencia*.

$$\mathbf{when} (A \geq 1 \wedge B \geq 1) \mathbf{do next} (A\&B = A\&B + 1 \wedge A = A - 1 \wedge B = B - 1) \quad (1)$$

$$\mathbf{when} (A\&B \geq 1) \mathbf{do next} (A = A + 1 \wedge B = B + 1 \wedge A\&B = A\&B - 1) \quad (2)$$

$$\mathbf{when} (A_i \geq 1) \mathbf{do next} (A_j = A_j + 1 \wedge A_i = A_i - 1) \quad (3)$$

**Cuadro 3.** Reacciones químicas en *ntcc*. (1): Síntesis, las sustancias *A*, *B* se unen para formar el compuesto *A&B*. (2): Descomposición, el compuesto *A&B* se separa en las sustancias *A*, *B*. (3): Transporte, la sustancia *A* se mueve de un nivel *i* a un nivel *j*

## 4. Modelo de la Ruta Endocítica de la LDL del Colesterol en el Cálculo *ntcc*

En la primera parte de esta sección se encuentra una descripción del sistema biológico que se desea modelar. En la segunda parte se presenta el modelo propuesto usando las técnicas presentadas en la sección anterior.

### 4.1. Ruta Endocítica de la LDL del Colesterol

El conjunto de pasos por medio de los cuales una célula lleva a su interior una partícula, para luego ser procesada por sus organelas, es conocido como *ruta endocítica*. Las células llevan a su interior partículas de colesterol siguiendo una ruta endocítica. El colesterol puede ser de dos clases: de alta densidad (high density) o de baja densidad (low density). El colesterol de baja densidad se transporta por el torrente sanguíneo al interior de una lipo-proteína conocida como la LDL (*low density lipoprotein*). En la Figura 1 (tomada de [11]) se muestra el proceso de la ruta endocítica de la LDL del colesterol.

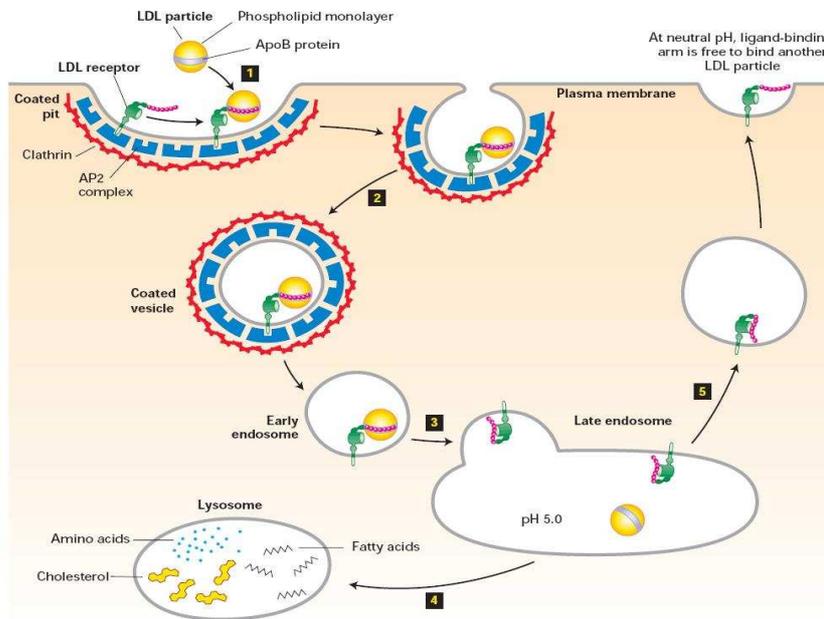
El primero de los pasos en la ruta endocítica se llama *endocitosis mediada por receptor*. El proceso se inicia cuando un receptor (ubicado en la membrana celular) enlaza a la partícula de LDL (ligando), a través de la proteína *apoB100* que está ubicada en la superficie de la partícula (paso 1 de la figura 1).

Una vez ocurrido el enlace, el complejo receptor-partícula ingresa a la célula en una vesícula de transporte formada por endocitosis; ésta vesícula está rodeada

por clatrina (proteína que recubre la membrana). En el momento en que la clatrina se desprende de la vesícula, ésta se convierte en un endosoma temprano (pasos 2 y 3 de la figura 1).

A continuación, el endosoma temprano se fusiona con un endosoma tardío (una de las organelas de la célula). El ambiente al interior del endosoma tardío es ácido en comparación con el exterior (ph entre 5 y 6), lo que causa que se rompa el enlace entre el receptor y el ligando. Después de la separación, el receptor debe ser devuelto a la superficie de la membrana celular con el fin de servir como mecanismo para la captura de otras partículas de LDL al exterior de la célula. El receptor es enviado en el interior de una vesícula que se desprende del endosoma tardío (paso 5 de la figura 1).

El endosoma tardío que contiene únicamente la partícula de LDL se fusiona con un lisosoma. El lisosoma es una organela que contiene las enzimas necesarias para degradar la cubierta de la partícula y extraer el colesterol de su interior para luego ser aprovechado por la célula (paso 6 de la figura 1).



**Figura 1.** Ruta endocítica de la lipoproteína de baja densidad del colesterol, tomada de [11]

#### 4.2. Modelo en ntcc

Con base en la descripción biológica de [11] y descrita anteriormente, se proponen cuatro niveles como estructura del sistema. El conjunto de objetos que van a localizarse en los niveles definidos anteriormente es el siguiente:

$$S = \{LDLpart, LDLRcpt, Clth, LateEndo\}$$

en donde *LDLpart*, representa la lipoproteína, *LDLRcpt* representa el receptor, *Clth* representa a la proteína que cubre la vesícula que ingresa a la célula (clatrina) y *LateEndo* es el endosoma tardío. Los objetos anteriormente mencionados pueden formar compuestos. Estos compuestos se representan por la unión de los nombres de los objetos con un "&" intermedio; por ejemplo *LDLpart&LDLRcpt* representa al compuesto que resulta al unirse un ligando con el receptor. Cada uno de los niveles modela una región de la célula en donde hay sustancias e interacción escrita como procesos *ntcc*, que agrupan las reacciones que pueden ocurrir en cada uno de los niveles.

**Nivel 0.** Modela el exterior de la célula, en este nivel está localizada la lipoproteína de LDL. No hay reacciones en este nivel.

**Nivel 1.** Modela la membrana celular, no como una división, sino como el espacio en donde está localizado el receptor. En esta región se localiza una reacción que permite la unión entre el ligando (*LDLpart*) y el receptor (*LDLRcpt*), así como también la localización del compuesto formado en el nivel 1 (membrana de la célula).

**Nivel 2.** Modela el interior de la célula, aquí están localizados el endosoma tardío y el lisosoma. Comprende seis reacciones que modelan el comportamiento al interior de la célula. El proceso  $R_{2,1}$  modela el paso 2 de la figura 1, en donde una vesícula envuelta en clatrina que contiene el receptor y el ligando unidos ingresa a la célula. Por medio de los procesos  $R_{2,2}$  y  $R_{2,3}$  se modela el momento en que la clatrina se desprende de la vesícula, convirtiéndose en endosoma temprano y la fusión de éste con el endosoma tardío. En  $R_{2,4}$  se modela la ruptura del enlace entre el receptor y el ligando, posteriormente en  $R_{2,5}$  se modela el paso del receptor del nivel 2, al nivel 1 para que sea reutilizado, como se puede ver en el paso 5 de la figura 1. Por medio del proceso  $R_{2,6}$  se modela la fusión entre el endosoma tardío y el lisosoma.

**Nivel 3.** Modela el interior del lisosoma; este es el nivel a donde debe llegar la partícula de LDL para ser degradada por la célula. Este nivel alberga tres reacciones, los procesos  $R_{3,1}$  y  $R_{3,3}$  modelan el procesamiento de la LDL del colesterol y el proceso  $R_{3,2}$  permite retornar el objeto *LateEndo* al nivel 2, para que pueda ser reutilizado en un nuevo proceso.

**Procesos adicionales.** Los anteriores niveles componen el proceso *CellActivity* por medio de + (selección no determinista) para evitar que dos o más reacciones de distintos niveles, sean activadas y causen inconsistencias. La característica principal de este proceso es la persistencia. Por medio del operador ! se garantiza que durante la evolución del sistema, los procesos que simulan la célula están disponibles en todas las unidades de tiempo.

*Init* es un proceso que asigna valores iniciales a las variables otorgando una configuración inicial para la simulación; en este caso en particular se ubican

un lisosoma, y un endosoma temprano en el nivel 2 (interior de la célula) y un receptor sobre la membrana (nivel 1). El proceso principal es *System*, el cual contiene la actividad de la célula en paralelo con las condiciones iniciales y el proceso  $\mathbf{tell}(LDLpart_0 = n_0)$  que inicia la evolución del proceso, con  $n_0$  partículas de LDL del colesterol al exterior de la célula. Todos los procesos aquí descritos se encuentran en el cuadro 4.

$Level1$	$\stackrel{def}{=} LDLpart_0, LDLRcpt_1 \rightarrow LDLpart \& LDLRcpt_1$
$Level2$	$\stackrel{def}{=} R_{2,1} + R_{2,2} + R_{2,3} + R_{2,4} + R_{2,5} + R_{2,6}$
$R_{2,1}$	$= Clth_2, LDLpart \& LDLRcpt_1 \rightarrow Clth \& LDLpart \& LDLRcpt_2$
$R_{2,2}$	$= Clth \& LDLpart \& LDLRcpt_2 \rightarrow Clth_2, LDLpart \& LDLRcpt_2$
$R_{2,3}$	$= LateEndo_2, LDLpart \& LDLRcpt_2 \rightarrow LateEndo \& LDLpart \& LDLRcpt_2$
$R_{2,4}$	$= LateEndo \& LDLpart \& LDLRcpt_2 \rightarrow LateEndo \& LDLpart_2, LDLRcpt_2$
$R_{2,5}$	$= LDLRcpt_2 \rightarrow LDLRcpt_1$
$R_{2,6}$	$= LateEndo \& LDLpart_2 \rightarrow LateEndo \& LDLpart_3$
$Level3$	$\stackrel{def}{=} R_{3,1} + R_{3,1} + R_{3,1}$
$R_{3,1}$	$= LateEndo \& LDLpart_3 \rightarrow LateEndo_3, LDLpart_3$
$R_{3,2}$	$= LateEndo_3 \rightarrow LateEndo_2$
$R_{3,3}$	$= LDLpart_3 \rightarrow Chol_3$
$CellActivity$	$\stackrel{def}{=} !(Level\ 1 + Level\ 2 + Level\ 3)$
$Init$	$\stackrel{def}{=} \mathbf{tell}(LDLRcpt_1 = 1 \wedge Clth_2 = 1 \wedge LateEndo_2 = 1)$
$System$	$\stackrel{def}{=} \mathbf{tell}(LDLpart_0 = n_0) \parallel Init \parallel CellActivity$

**Cuadro 4.** Modelo de la ruta endocítica de la LDL del colesterol con el cálculo *ntcc*

La selección no determinista entre las reacciones y los niveles es útil porque permite garantizar que solo una reacción pueda ser activada en cada unidad de tiempo; esto es necesario ya que las variables declarativas no cambian su valor. No obstante el modelo no garantiza la ejecución de cada proceso (activación de una reacción) en un momento dado, e.g con el proceso  $!(r_1 + r_2)$  no se puede garantizar que en alguna unidad de tiempo  $r_1$  sea ejecutado. A continuación se muestra un segundo modelo que incluye el operador  $\parallel$  entre las reacciones en cada nivel y los niveles en *CellActivity*. La composición paralela de reacciones causa inconsistencias, porque permite la activación de varias de ellas en la misma unidad de tiempo; por tanto también es necesario modificar el proceso *Reaction* e incluir el proceso *Control<sub>i</sub>* que evita que los valores de las variables que hacen parte de la reacción sean modificados por otros procesos. En *CellActivity* se incluye el proceso *Control<sub>gen</sub>* que se encarga de actualizar los valores de las variables que no hacen parte de una reacción y preservar la continuidad de la evolución del sistema. El proceso *Control<sub>gen</sub>* se compone de varios procesos en paralelo, uno por cada variable. Dichos procesos son de la forma:

**unless** *Control<sub>var</sub>* = 1 **do next** *var*  $\forall var \in Vars,$

donde *Vars* es el conjunto de variables del sistema. Para el funcionamiento de los procesos de control se declara una variable de control por cada una de las variables del sistema. El nombre de cada variable de control está dado por el prefijo *Control\_* y el nombre de la variable con la que está relacionada. Por ejemplo para la variable *LDLpart<sub>0</sub>* la variable de control correspondiente tiene el nombre *Control\_LDLpart<sub>0</sub>*. En el cuadro 5 se encuentran los procesos que cambian para el segundo modelo.

$  \begin{aligned}  \text{Reaction} &\stackrel{def}{=} \text{when } (R_1 \geq 1 \dots \wedge R_n \geq 1) \text{ do next } ( \text{tell}( R_1 = R_1 - 1 \wedge \dots \\  &\quad \wedge R_n = R_n - 1 \ P_1 = P_1 + 1 \wedge \dots \wedge P_m = P_m + 1 ) ) \parallel \text{Control}_i \\  \text{Control}_i &\stackrel{def}{=} \text{Control\_Ri} = 1, \ \text{Control\_Pj} = 1 \text{ para } 1 \leq i \leq n \text{ y } 1 \leq j \leq m \\  \text{Level1} &\stackrel{def}{=} \text{LDLpart}_0, \text{LDLRcpt}_1 \rightarrow \text{LDLpart}\&\text{LDLRcpt}_1 \\  \text{Level2} &\stackrel{def}{=} R_{2,1} \parallel R_{2,2} \parallel R_{2,3} \parallel R_{2,4} \parallel R_{2,5} \parallel R_{2,6} \\  \text{Level3} &\stackrel{def}{=} R_{3,1} \parallel R_{3,1} \parallel R_{3,1} \\  \text{CellActivity} &\stackrel{def}{=} !( \text{Level } 1 \parallel \text{Level } 2 \parallel \text{Level } 3 \parallel \text{Control}_{gen} ) \\  \text{Control}_{gen} &\stackrel{def}{=} \Pi \text{ unless } \text{Control\_var} = 1 \text{ do next } var, \forall var \in Vars  \end{aligned}  $
---

Cuadro 5. Modelo con paralelismo y el proceso *Control<sub>gen</sub>*

## 5. Conclusiones y Trabajo Futuro

Con este trabajo se muestra que es posible modelar sistemas de membranas en *ntcc*. Las técnicas propuestas pueden ser aplicadas para modelar procesos biológicos que consideren la interacción de membranas. Sin embargo, existen aún inconvenientes con la selección no determinista y la composición paralela entre las reacciones y los niveles.

El problema que causa la selección no determinista interfiere con la automatización de la simulación del proceso biológico, ya que no se puede garantizar que la máquina alcance algunos estados del proceso. Este problema es resuelto con la composición paralela, pero requiere de un cuidadoso manejo de las variables declarativas. Una variable declarativa no puede cambiar su valor, por tanto no se puede permitir que dos reacciones que involucran una misma sustancia sean activadas en la misma unidad de tiempo. Por ejemplo, en el caso en que existan más de una partícula de LDL al exterior de la célula y más de un receptor en la membrana, se genera una inconsistencia cuando se imponen las restricciones  $\text{LDL\_part}\&\text{LDL\_Rcpt} = \text{LDL\_part}\&\text{LDL\_Rcpt} + 1$ , y  $\text{LDL\_part}\&\text{LDL\_Rcpt} = \text{LDL\_part}\&\text{LDL\_Rcpt} - 1$  en la misma unidad de tiempo. La inconsistencia no se presenta cuando hay más de una partícula de LDL al exterior y un solo receptor, así que en este caso la competencia por un recurso no es un problema. La utilización de composición paralela hace que el modelo sea más cercano al proceso biológico.

Las técnicas propuestas en este trabajo modelan membranas como fronteras de los niveles. También es posible modelar interacción al interior de una membrana, como ocurre en la ruta endocítica de la LDL del colesterol. Para lograr

esto, se incluyó en el modelo un nivel (nivel 1) y se ubicó en dicho nivel los objetos y reacciones que representan el comportamiento de la membrana celular en dicho proceso biológico.

Todo apunta a que la mejor alternativa es conservar la composición paralela entre las reacciones y los niveles, pues así el modelo es más fiel al sistema biológico, y es posible automatizar la simulación y verificar propiedades. Las inconsistencias son un problema inherente al tipo de variable y los dominios. Una propuesta interesante es examinar la utilización de otros sistemas de restricciones, como por ejemplo flujos, en los cuales se puedan actualizar los valores de las variables; así desaparecerían las inconsistencias. Como trabajo futuro se propone explorar sistemas de restricciones distintos a los dominios finitos, con el fin de dar solución al problema de las inconsistencias. Adicionalmente, se espera aplicar la lógica temporal (LTL) para verificar propiedades en el modelo.

## Referencias

1. Gutierrez J., Perez J.A., Rueda C.: Modelamiento de Sistemas Biologicos usando Calculos de Procesos Concurrentes. *Epíclon*, 4(1) pp. 79–101 (2005)
2. Gutierrez J., Perez J.A., Rueda C., Valencia F.D.: Timed Concurrent Constraint Programming for Analysing Biological Systems. *Electronic Notes in Theoretical Computer Science (ENTCS) Volume 171(2)*, pp. 117–137 (2007)
3. Valencia F.: Temporal Concurrent Constraint Programming. Tesis doctoral, Universidad de Aarhus (2002)
4. Saraswat V., Rinard M.C., Panangaden P.: The Semantic Foundation of Concurrent constraint Programming. *POPL 1990*, pp. 232–245 (1990)
5. Kitano H.: Systems Biology: A Brief Overview. *Science* Vol 295 (2002)
6. Kitano H.: Systems Biology: Toward System-level Understanding of Biological Systems. *Foundations of Systems Biology*, pp. 1–29. MIT Press, Cambridge/MA (2001)
7. Busi N., Zandron C.: Modeling and Analysis of Biological Processes by Mem(Brane) Calculi and Systems. En: *Simulation Conference, 2006. WSC 06. Volumen de invierno diciembre 3-6*, pp. 1646–1655 (2006)
8. Regev A., Panina E.M., Silverman W., Cardelli L., Shapiro E.: BioAmbients: an abstraction for biological compartments. *Theoretical Computer Science Volume 325 (1)*, pp. 141–167 (2004)
9. Cardelli L.: Brane Calculi Interactions of Biological Membranes. En: *Computational Methods in Systems Biology, International Conference CMSB (2004)*
10. Arbelaez A., Gutierrez J., Olarte C., Rueda C.: A Generic Framework to Model, Simulate and Verify Genetic Regulatory Networks (Poster). En: *CLEI 2006 (32nd Latinamerican Conference on Informatics)* (2006)
11. Lodish H., Berk A., Scott M.: *Molecular cell biology*. - 5 ed. W. H. Freeman (2003)
12. Laneve C., Montanari U.: Mobility in the cc-paradigm. *Mathematical foundations of computer science. LNCS 629*, pp. 336–345 (1992)
13. Paun G.: Computing with Membranes. *Journal of Computer and System Sciences*, 61(1) pp. 108–143 (2000)