A Generic Abstract Machine for Stochastic Process Calculi

Loïc Paulevé[1] Simon Youssef[2] Matthew Lakin[3] Andrew Phillips[3]

[1]IRCCyN - UMR CNRS 6597, Nantes (France)

[2]Ludwig-Maximilians-Universität, München (Germany)

[3]Microsoft Research, Cambridge (UK)

Computational Methods in Systems Biology, 2010

Overview

Motivation

- A range of process calculi for biology;
- Reduce overhead of creating a simulation tool.

Contribution

- Simple and generic framework for stochastic simulation of process calculi.
- Just-in-time compilation from process calculi terms to reactions.

Outline

1. The generic abstract machine: syntax and simulation methods;
2. instantiation to the stochastic π-calculus;
3. application to a model of plasmid co-transfection.

Generic Abstract Machine (1/2)

- Supports for unbounded numbers of species and reactions.
- Generic: n-ary reactions, general probability distributions, arbitrary stoichiometry, ...
- Deals with standard reactions ($I_1 + \cdots + I_n \xrightarrow{F} I'_1 + \cdots + I'_m$).

Main structures

$$
\begin{array}{rcl}
T ::= & (t, S, R) & \text{Term} \\
S ::= & \{I_1 \mapsto i_1, \ldots, I_N \mapsto i_N\} & \text{Populations} \\
R ::= & \{O_1 \mapsto A_1, \ldots, O_N \mapsto A_N\} & \text{Reactions} \\
O ::= & (J, F, J') & \text{Reaction} \\
J ::= & \{I_1, \ldots, I_N\} & \text{Species set}
\end{array}
$$

Populating the machine

$$P \oplus T \triangleq \mathit{species}(P) \oplus T$$

$\star \mathit{species}(P)$: converts a process P to a set of species J; depends on the language: defined later!

Generic Abstract Machine (2/2)

Adding a **new** species

$$
\begin{aligned}
I \oplus (t, S, R) \quad \triangleq \quad & (t, S', R \cup R') \textbf{ if } I \notin \mathrm{dom}(S) \\
& \textbf{and } S' = S\{I \mapsto 1\} \\
& \textbf{and } L = \mathit{reactions}(I, \mathrm{dom}(S)) \\
& \textbf{and } R' = \mathit{init}(L, (t, S', R))
\end{aligned}
$$

$\star \mathit{reactions}(I, \{I_1, \ldots, I_n\})$: list of all reactions between I and $\{I_1, \ldots, I_n\}$; depends on the language.
$\star \mathit{init}(\{O_1, \ldots, O_m\}, T)$: initialise reactions; depends on the simulation method.

Updating the population of a species

$$
\begin{aligned}
I \oplus (t, S, R) \quad \triangleq \quad & (t, S', R \cup R') \textbf{ if } S(I) = i \\
& \textbf{and } S' = S\{I \mapsto i + 1\} \\
& \textbf{and } R' = \mathit{updates}(I, (t, S', R))
\end{aligned}
$$

$\star \mathit{updates}(I, T)$: updates reactions according to the new population of I.

Simulation Methods

Simulation

$$\frac{(J, F, J'), t' = next(t, S, R)}{t, S, R \xrightarrow{F,(J,F,J')} J' \oplus ((t', S, R) \ominus J)}$$

⋆$next(T)$: next reaction occurring in the machine, with its timestamp.

Simulation Methods

Simulation

$$\frac{(J, F, J'), t' = \mathit{next}(t, S, R)}{t, S, R \xrightarrow{F,(J,F,J')} J' \oplus ((t', S, R) \ominus J)}$$

$\star \mathit{next}(T)$: next reaction occurring in the machine, with its timestamp.

Instantiating a simulation method

- Related functions: $\mathit{next}(T)$, $\mathit{init}(\{O_1, \ldots, O_n\}, T)$, $\mathit{update}(I, T)$.
- Store data associated to reactions: $R ::= \{O_1 \mapsto A_1, \ldots, O_N \mapsto A_N\}$.
- Pickup-one (or create another!): Direct Method, First Reaction Method, Next Reaction Method.

[Gillespie, 77; Gibson & Bruck, 00]

Simulation Methods

→ Next Reaction Method

time t			
Pop. S		Reactions R	
I_1	i_1	$J_1 \xrightarrow{F_1} J_1'$	a_1, t_1
...	...	...	...
I_n	i_n	$J_m \xrightarrow{F_m} J_m'$	a_m, t_m

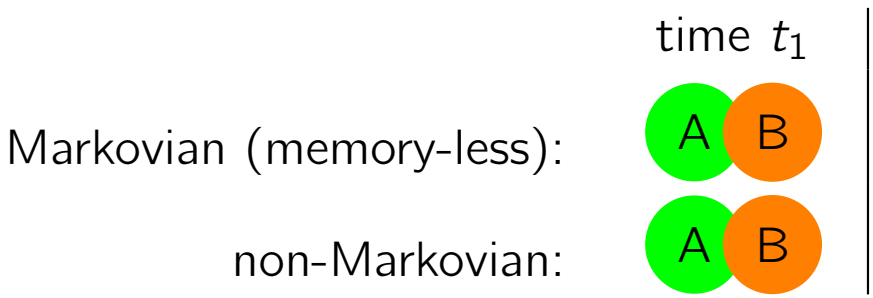
a_i: reaction propensity
t_i: reaction putative time
F_i: assumed to be exponential (non-Markovian case addressed by the next slide)

- Next reaction: least putative time.
- *init* reaction: $a_m = \mathit{propensity}(O, S)$; $t_m = t + \mathit{delay}(F_m, a_m)$
- *update* reaction: $a_m' = \mathit{propensity}(O, S)$; $t_m' = t + (a_m / a_m')(t_m - t)\}$
- Rescaling prevents spurious random number generation, but only for Markovian reactions.

Example of propensity: $\mathit{propensity}((\{I_1, I_2\}, F, J), S) \triangleq \mathit{rate}(F) \cdot i_1 \cdot i_2$ **if** $I_1 \neq I_2$
[Gibson & Bruck, 00]

Simulation Methods

→ Non-Markovian Next Reaction Method



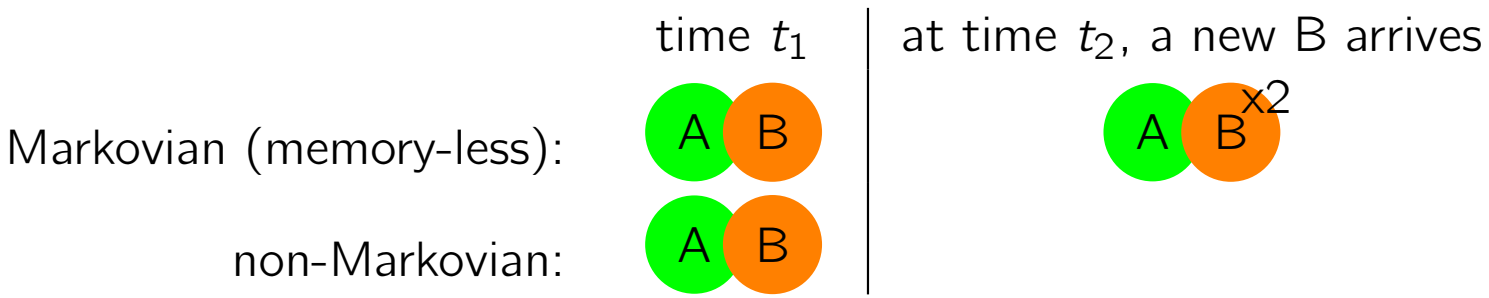
When F is non-Markovian:

- Every molecule is treated as a dedicated species.
- $rename(I) \triangleq \textbf{if } NM(I) \textbf{ then } I^{fresh(ctr)} \textbf{ else } I^0$
- $next'(t, S, R) \triangleq (J, F, rename(J')), t' \textbf{ if } (J, F, J'), t' = next(t, S, R)$
- Propensity a_m is not used.

Important note: this feature is independent from the instantiated calculus.

Simulation Methods

→ Non-Markovian Next Reaction Method



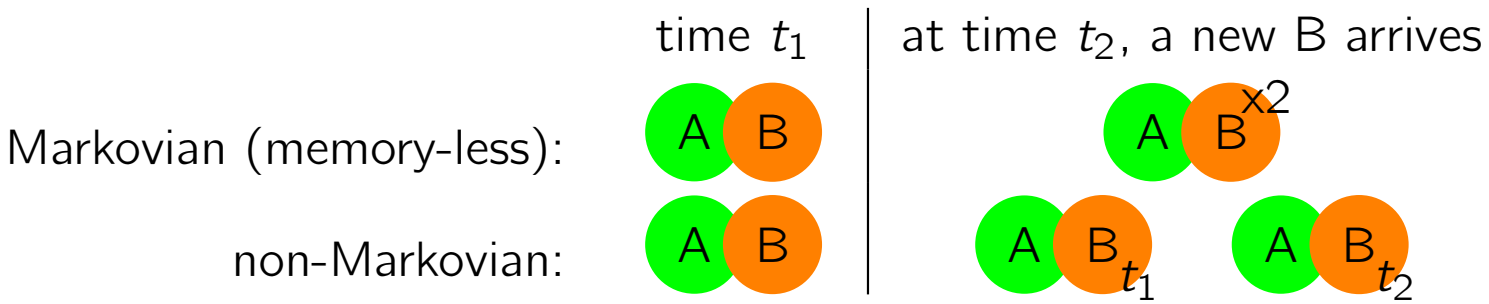
When F is non-Markovian:

- Every molecule is treated as a dedicated species.
- $rename(I) \triangleq \textbf{if } NM(I) \textbf{ then } I^{fresh(ctr)} \textbf{ else } I^0$
- $next'(t, S, R) \triangleq (J, F, rename(J')), t' \textbf{ if } (J, F, J'), t' = next(t, S, R)$
- Propensity a_m is not used.

Important note: this feature is independent from the instantiated calculus.

Simulation Methods

→ Non-Markovian Next Reaction Method



When F is non-Markovian:

- Every molecule is treated as a dedicated species.
- $rename(I) \triangleq$ **if** $NM(I)$ **then** $I^{fresh(ctr)}$ **else** I^0
- $next'(t, S, R) \triangleq (J, F, rename(J')), t'$ **if** $(J, F, J'), t' = next(t, S, R)$
- Propensity a_m is not used.

Important note: this feature is independent from the instantiated calculus.

From Stochastic π-Calculus to Reactions

$\rightarrow$ Stochastic π-calculus definition

- Normal form for processes: a species is either an instance or a complex.

$$
\begin{array}{rcll}
P ::= & I_1 \mid \ldots \mid I_N & & \text{Species} \\
I ::= & X(\tilde{n}) & & \text{Instance} \\
\mid & \nu\tilde{z}\,(X_1(\tilde{n}_1) \mid \ldots \mid X_M(\tilde{n}_M)) & & \text{Complex} \\
C ::= & \pi_1.P_1 + \ldots + \pi_N.P_N & & \text{Choice} \\
E ::= & X_1(\tilde{m}_1) \mapsto C_1, \ldots, X_N(\tilde{m}_N) \mapsto C_N & & \text{Environment}
\end{array}
$$

$$
\begin{array}{rcll}
\pi ::= & \tau_r & & \text{Delay} \\
\mid & !x(\tilde{n}) & & \text{Send} \\
\mid & !x(\nu\tilde{m}) & & \text{Bind} \\
\mid & ?x(\tilde{m}) & & \text{Receive}
\end{array}
$$

- Efficient handling of complexes.

From Stochastic π-Calculus to Reactions

$\rightarrow$ Extracting actions from species

$$actions(I) = \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is an **instance**

$$X(\tilde{n}) \triangleq \pi_1.P_1 + \cdots + \pi_N.P_N$$

From Stochastic π-Calculus to Reactions

$\rightarrow$ Extracting actions from species

$$actions(I) = \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is an **instance**

$$X(\tilde{n}) \triangleq \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is a **complex**

$$\nu\tilde{z}\,(!x.P \mid ?x.Q)$$

From Stochastic π-Calculus to Reactions

$\rightarrow$ Extracting actions from species

$$actions(I) = \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is an **instance**

$$X(\tilde{n}) \triangleq \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is a **complex**

$$\nu\tilde{z}\ (!x.P \mid ?x.Q)$$

$$\Longrightarrow \text{expand actions}$$

$$\nu\tilde{z}\ (!x.(P \mid ?x.Q) + ?x.(!x.P \mid Q) + \tau_x.(P|Q))$$

From Stochastic π-Calculus to Reactions

→ Extracting actions from species

$$actions(I) = \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is an **instance**

$$X(\tilde{n}) \triangleq \pi_1.P_1 + \cdots + \pi_N.P_N$$

If I is a **complex**

$$\nu\tilde{z}\,(!x.P \mid ?x.Q)$$

$$\Longrightarrow \text{expand actions}$$

$$\nu\tilde{z}\,(!x.(P \mid ?x.Q) + ?x.(!x.P \mid Q) + \tau_x.(P|Q))$$

$$\Longrightarrow \text{forward restrictions}$$

$$?x.\nu\tilde{z}\,(P \mid ?x.Q) + ?x.\nu\tilde{z}\,(!x.P \mid Q) + \tau_x.\nu\tilde{z}(P|Q)$$

From Stochastic π-Calculus to Reactions

$\rightarrow$ Computation of reactions from species

$$\star reactions(I, \{I_1, \ldots, I_N\}) \triangleq unary(I) \cup binary(I, \{I_1, \ldots, I_N\})$$

$$\overset{I}{!x(a).P_1 + ?x(b).P_2 + \tau_y.P_3}$$

Reactions

$$I \xrightarrow{F_y} \text{species}\,(P_3)$$

From Stochastic π-Calculus to Reactions

$\rightarrow$ Computation of reactions from species

$$\star reactions(l, \{l_1, \ldots, l_N\}) \triangleq unary(l) \cup binary(l, \{l_1, \ldots, l_N\})$$

$$\underset{!x(a).P_1 + ?x(b).P_2 + \tau_y.P_3}{l} \qquad \underset{!x(\nu u).Q}{l_1}$$

Reactions

$$l \xrightarrow{F_y} \text{species}\,(P_3)$$

$$l + l \xrightarrow{F_X} \text{species}\left(P_1 \mid P_{2\{b:=a\}}\right)$$

From Stochastic π-Calculus to Reactions

$\rightarrow$ Computation of reactions from species

$$\star reactions(I, \{I_1, \ldots, I_N\}) \triangleq unary(I) \cup binary(I, \{I_1, \ldots, I_N\})$$

$$\underset{!x(a).P_1 + ?x(b).P_2 + \tau_y.P_3}{I} \qquad \underset{!x(\nu u).Q}{I_1}$$

Reactions

$$I \xrightarrow{F_y} \text{species}\,(P_3)$$

$$I + I \xrightarrow{F_X} \text{species}\left(P_1 \mid P_{2\{b:=a\}}\right)$$

$$I + I_1 \xrightarrow{F_x} \text{species}\left(\nu u\,(P_{2\{b:=u\}} \mid Q)\right)$$

Simple Example

$$A = !x(\nu u).AB(u) \qquad AB(u) = !u.A$$
$$B = ?x(u).BA(u) \qquad BA(u) = ?u.B$$

time t			
Populations S		Reactions R	
A	100	$A + B \xrightarrow{F_x} \nu u\,(AB(u) \mid BA(u))$	$100^2 \cdot rate(F_x), t_1$
B	100		

1. $(100 \cdot A \mid 100 \cdot B) \oplus (0, \emptyset, \emptyset)$

Simple Example

$$A = !x(\nu u).AB(u) \qquad AB(u) = !u.A$$
$$B = ?x(u).BA(u) \qquad BA(u) = ?u.B$$

time t_1			
Populations S		Reactions R	
A	99	$A + B \xrightarrow{F_x} \nu u\,(AB(u) \mid BA(u))$	$100^2 \cdot rate(F_x), t_1$
B	99		

1. $(100 \cdot A \mid 100 \cdot B) \oplus (0, \emptyset, \emptyset)$
2. $\nu u\,(AB(u) \mid BA(u)) \oplus ((t_1, S, R) \ominus \{A, B\})$

Simple Example

$$A = !x(\nu u).AB(u) \qquad AB(u) = !u.A$$
$$B = ?x(u).BA(u) \qquad BA(u) = ?u.B$$

time t_1			
Populations S		Reactions R	
A	99	$A + B \xrightarrow{F_x} \nu u\,(AB(u) \mid BA(u))$	$99^2 \cdot rate(F_x), t_2$
B	99		

1. $(100 \cdot A \mid 100 \cdot B) \oplus (0, \emptyset, \emptyset)$
2. $\nu u\,(AB(u) \mid BA(u)) \oplus ((t_1, S, R) \ominus \{A, B\})$

Simple Example

$$A = !x(\nu u).AB(u) \qquad AB(u) = !u.A$$
$$B = ?x(u).BA(u) \qquad BA(u) = ?u.B$$

time t_1			
Populations S		Reactions R	
A	99	$A + B \xrightarrow{F_x} \nu u(AB(u) \mid BA(u))$	$99^2 \cdot rate(F_x), t_2$
B	99		
$\nu u(AB(u) \mid BA(u))$	1		

1. $(100 \cdot A \mid 100 \cdot B) \oplus (0, \emptyset, \emptyset)$
2. $\nu u(AB(u) \mid BA(u)) \oplus ((t_1, S, R) \ominus \{A, B\})$
3. actions$(\nu u(!u.A \mid ?u.B)) = \tau_u.(A \mid B)$

Simple Example

$$A = !x(\nu u).AB(u) \qquad AB(u) = !u.A$$
$$B = ?x(u).BA(u) \qquad BA(u) = ?u.B$$

time t_1			
Populations S		Reactions R	
A	99	$A + B \xrightarrow{F_x} \nu u(AB(u) \mid BA(u))$	$99^2 \cdot rate(F_x), t_2$
B	99	$\nu u(AB(u) \mid BA(u)) \xrightarrow{F_u} A + B$	$1 \cdot rate(F_u), t_3$
$\nu u(AB(u) \mid BA(u))$	1		

1. $(100 \cdot A \mid 100 \cdot B) \oplus (0, \emptyset, \emptyset)$
2. $\nu u(AB(u) \mid BA(u)) \oplus ((t_1, S, R) \ominus \{A, B\})$
3. $\text{actions}(\nu u(!u.A \mid ?u.B)) = \tau_u.(A \mid B)$

Simple Example

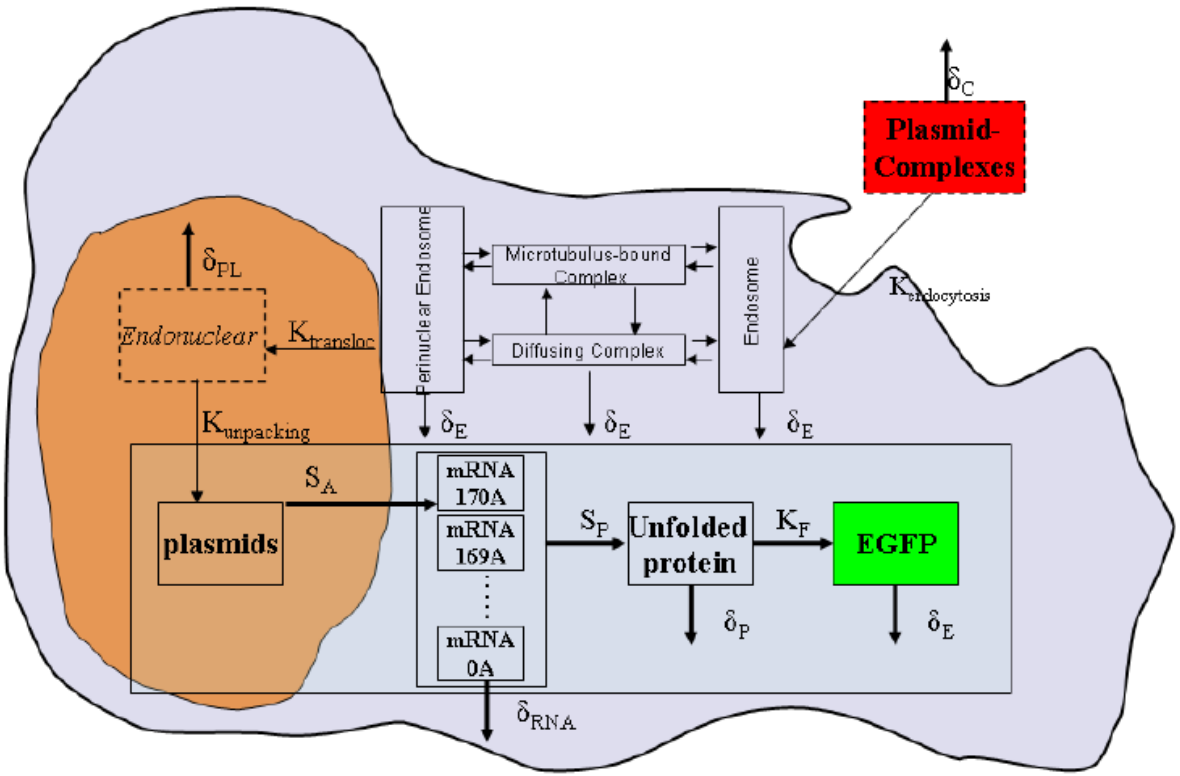
$$A = !x(\nu u).AB(u) \qquad AB(u) = !u.A$$
$$B = ?x(u).BA(u) \qquad BA(u) = ?u.B$$

time t_2			
Populations S		Reactions R	
A	98	$A + B \xrightarrow{F_x} \nu u(AB(u) \mid BA(u))$	$98^2 \cdot rate(F_x), t_4$
B	98	$\nu u(AB(u) \mid BA(u)) \xrightarrow{F_u} A + B$	$2 \cdot rate(F_u), t_3'$
$\nu u(AB(u) \mid BA(u))$	2		

1. $(100 \cdot A \mid 100 \cdot B) \oplus (0, \emptyset, \emptyset)$
2. $\nu u(AB(u) \mid BA(u)) \oplus ((t_1, S, R) \ominus \{A, B\})$
3. $\text{actions}(\nu u(!u.A \mid ?u.B)) = \tau_u.(A \mid B)$

Application to Plasmid Co-Transfection

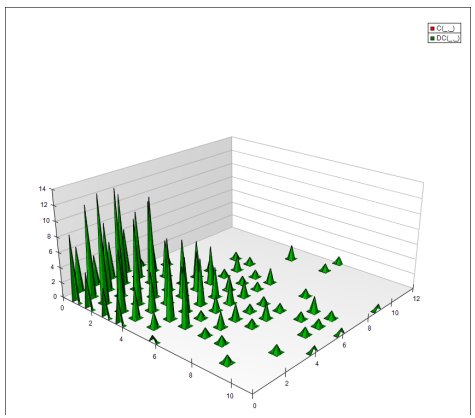
- Process $C(g, r)$ a complex of g green plasmids and r red plasmids
- Complexes can grow in size: new species created on-the-fly
- mRNA degradation is non-Markovian: use the Erlang distribution

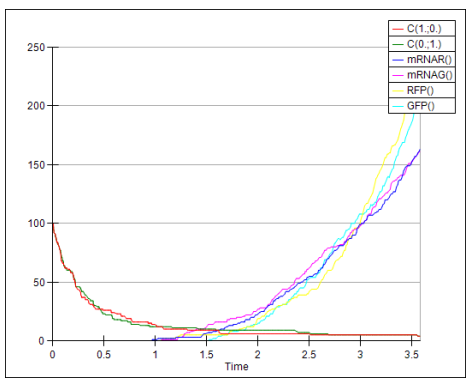


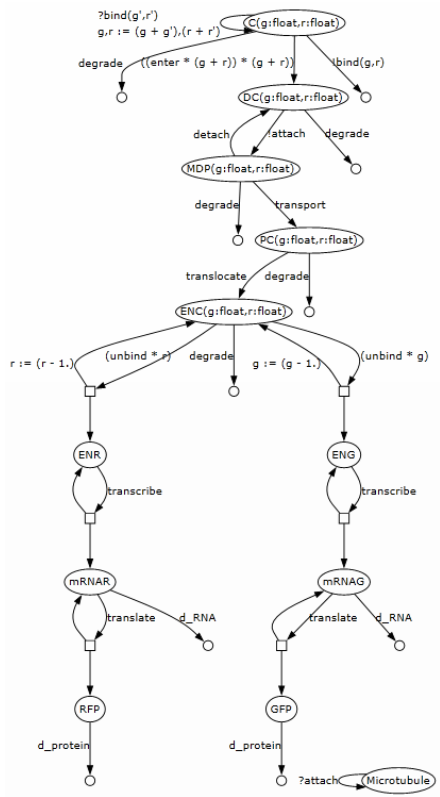
- Choose language and simulation algorithm to suit the system.

Application to Plasmid Co-Transfection

- Process $C(g, r)$ a complex of g green plasmids and r red plasmids
- Complexes can grow in size: new species created on-the-fly
- mRNA degradation is non-Markovian: use the Erlang distribution







- Choose language and simulation algorithm to suit the system.

Conclusion

- Simple formal framework for stochastic process calculi simulations;
- Support for non-Markovian reactions;
- Generic framework: applies to a broad range of process calculi;
- Already done: stochastic π-calculus, DSD, Bioambient, ...

Outlook

- Benchmarks.
- Simulate models written in mixed frameworks.

Implementation

- Available at `http://research.microsoft.com/spim`.