Artificial Biochemistry

Luca Cardelli
Microsoft Research

CMSB'06 Trento 2006-10-19

http://LucaCardelli.name
Reverse-Engineer This

Eukaryotic Cell

(10~100 trillion in human body)

Membranes everywhere

H. Lodish et al.
Molecular Cell Biology
fourth edition p.1
...and Model It

- Even if we understood it, how would we model it?
  - Millions of differential equations? Hmmm..

- And we will have to model it in order to understand it.

- What's different about modeling these systems?
Stochastic Collectives
Stochastic Collectives

“Collective“:
- A large set of interacting finite state automata:
  - Not quite language automata (“large set“)
  - Not quite cellular automata (“interacting” but not on a grid)
  - Not quite process algebra (“finite state” and “collective“)
  - Cf. “multi-agent systems” and “swarm intelligence”

“Stochastic“:
- Interactions have rates
  - Not quite discrete (hundreds or thousands of components)
  - Not quite continuous (non-trivial stochastic effects)
  - Not quite hybrid (no “switching” between regimes)

Very much like biochemistry
- Which is a large set of stochastically interacting molecules/proteins
- Are proteins finite state and subject to automata-like transitions?
  - Let’s say they are, at least because:
  - Much of the knowledge being accumulated in Systems Biology
    is described as state transition diagrams [Kitano].
State Transitions
Even More State Transitions

http://www.expasy.ch/cgi-bin/show_thumbnails.pl
Interacting Automata

Communicating automata: a graphical FSA-like notation for "finite state restriction-free π-calculus processes". Interacting automata do not even exchange values on communication.

The stochastic version has *rates* on communications, and delays.

"Finite state" means: no composition or restriction inside recursion.

Analyzable by standard Markovian techniques, by first computing the "product automaton" to obtain the underlying finite Markov transition system. [Buchholz]
Interacting Automata Transition Rules

Q: What kind of mass behavior can this produce?
(We need to understand that if want to understand biochemical systems.)
Groupies and Celebrities

**Groupie** (wants to be like somebody different)

- directive sample 0.1 200
- directive plot A(); B()
- new a@1.0:chan()
- new b@1.0:chan()
- let A() = do !a; A() or ?a; B()
- and B() = do !b; B() or ?b; A()
- run 100 of (A() | B())

**Celebrity** (does not want to be like somebody else)

- directive sample 0.1 200
- directive plot A(); B()
- new a@1.0:chan()
- new b@1.0:chan()
- let A() = do !a; A() or ?a; B()
- and B() = do !b; B() or ?b; A()
- run 100 of (A() | B())

A stochastic collective of celebrities:

Stable because as soon as a A finds itself in the majority, it is more likely to find somebody in the same state, and hence change, so the majority is weakened.

A stochastic collective of groupies:

Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B’s have plenty of A’s to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.
Both Together

A way to break the deadlocks: Groupies with just a few Celebrities

A tiny bit of “noise” can make a huge difference

---

directive sample 10.0 1000

directive plot Ga(); Gb(); Ca(); Cb()

new a@1.0:chan()
new b@1.0:chan()

let Ca() = do !a; Ca() or ?a; Cb()
and Cb() = do !b; Cb() or ?b; Ca()

let Ga() = do !a; Ga() or ?b; Gb()
and Gb() = do !b; Gb() or ?a; Ga()

run 1 of (Ca() | Cb())
run 100 of (Ga() | Gb())

---

Many Groupies

A few Celebrities

not deadlock


**Hysteric Groupies**

We can get more regular behavior from groupies if they “need more convincing”, or “hysteresis” (history-dependence), to switch states.

```plaintext
new a@1:0:chan()
new b@1:0:chan()

let Ga() = do !a; Ga() or ?b; ?b; Gb()
let Gb() = do !b; Gb() or ?a; ?a; Ga()
let Da() = !a; Da()
let Db() = !b; Db()

run 100 of (Ga() | Gb())
run 1 of (Da() | Db())
```

N.B.: It will not oscillate without doping (noise)
Hysteric 3-Way Groupies

N.B.: It will not oscillate without doping (noise)

directive sample 3.0 1000
directive plot A(); B(); C()

new a@1.0:chan()
new b@1.0:chan()
new c@1.0:chan()

let A() = do !a; A() or ?c; ?c; C()
and B() = do !b; B() or ?a; ?a; A()
and C() = do !c; C() or ?b; ?b; B()

let Da() = !a; Da()
and Db() = !b; Db()
and Dc() = !c; Dc()

run 100 of (A() | B() | C())
run 1 of (Da() | Db() | Dc())
Semantics of Collective Behavior
“Micromodels”: Continuous Time Markov Chains

- The underlying semantics of stochastic π-calculus (and stochastic interacting automata). Well established in many ways.
  - Automata with rates on transitions.

- “The” correct semantics for chemistry, executable.
  - Gillespie stochastic simulation algorithm

- Lots of advantages
  - Compositional, compact, mechanistic, etc.

- But do not give a good sense of “collective” properties.
  - Yes one can do simulation.
  - Yes one can do program analysis.
  - Yes one can do modelchecking.
  - But somewhat lacking in “analytical properties” and “predictive power”.
“Macromodels”: Ordinary Differential Equations

- They always ask:
  - “Yes, but how does your automata model relate to the 75 ODE models in the literature?”

- Going from processes/automata to ODEs directly:
  - In principle: just write down the Rate Equation: [Calder, Hillston]
    - Determine the set of all possible states $S$ of each process.
    - Determine the rates of the transitions between such states.
    - Let $[S]$ be the “number of processes in state $S$” as a function of time.
    - Define for each state $S$:
      $$[S]^* = \text{(rate of change of the number of processes in state } S)$$
      - Cumulative rate of transitions from any state $S'$ to state $S$, times $[S']$,
      - minus cumulative rate of transitions from $S$ to any state $S''$, times $[S]$.
  - Intuitive (rate = inflow minus outflow), but often clumsy to write down precisely.

- But why go to the trouble?
  - If we first convert processes to chemical reactions, then we can convert to ODEs by standard means!
From Chemistry to ODEs
Chemical Reactions

$A \rightarrow^r B_1 + ... + B_n$  Degradation  $[A]^* = -r[A]$  Exponential Decay

$A_1 + A_2 \rightarrow^r B_1 + ... + B_n$  Asymmetric Collision  $[A_i]^* = -r[A_1][A_2]$  Mass Action Law

$A + A \rightarrow^r B_1 + ... + B_n$  Symmetric Collision  $[A]^* = -r[A][A]-1$  Mass Action Law  (assuming $A\neq B_i, A_j$ for all $i,j$)

No other reactions!

Trimolecular reactions:

$A + B + C \rightarrow^r D$

the measured “$r$” is an (imperfect) aggregate of e.g.:

$A + B \leftrightarrow AB$

$AB + C \rightarrow D$

Enzymatic reactions:

$S \overset{E}{\rightarrow^r} P$

the “$r$” is given by Michaelis-Menten (approximated steady-state) laws:

$E + S \leftrightarrow ES$

$ES \rightarrow P + E$
From Reactions to ODEs

\[ v_1: A + B \rightarrow k_1 C + C \]
\[ v_2: A + C \rightarrow k_2 D \]
\[ v_3: C \rightarrow k_3 E + F \]
\[ v_4: F + F \rightarrow k_4 B \]

Write the coefficients by columns

\[ \begin{array}{c|cccc}
\text{Species} & X & A & B & C & D & E & F \\
\hline
\text{Quantity changes} & v_1 & v_2 & v_3 & v_4 \\
\hline
A & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
B & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
C & 2 & -1 & -1 & 0 & 0 & 0 & 0 \\
D & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
F & 0 & 0 & 0 & 0 & 0 & 1 & -2 \\
\end{array} \]

Read the concentration changes from the rows

Set a rate law for each reaction (Degradation/Asymmetric/Symmetric)

\[ [A]^* = -l_1 - l_2 \]
\[ [B]^* = -l_1 + l_4 \]
\[ [C]^* = 2l_1 - l_2 - l_3 \]
\[ [D]^* = l_2 \]
\[ [E]^* = l_3 \]
\[ [F]^* = l_3 - 2l_4 \]

Caveat: A deterministic approximation of a stochastic system (i.e. possibly misleading)
From Processes to Chemistry
Chemical Ground Form (CGF)

\[
E ::= X_1 = M_1, \ldots, X_n = M_n
\]

Definitions \((n \geq 0)\)

\[
M ::= \pi_1; P_1 \oplus \ldots \oplus \pi_n; P_n
\]

Molecules \((n \geq 0)\)

\[
P ::= X_1 | \ldots | X_n
\]

Solutions \((n \geq 0)\)

\[
\pi ::= \tau_r ? n(r) ! n(r)
\]

Interactions (delay, input, output)

\[
\text{CGF} ::= E, P
\]

Definitions with Initial Conditions

\[
\begin{align*}
A &= !a; A \oplus ?b; B \\
B &= !b; B \oplus ?a; A \\
A|A|B|B
\end{align*}
\]

Ex: interacting automata

(which are CGFs using “\(\mid\)" only in initial conditions):

\(\oplus\) is stochastic choice (vs. + for chemical reactions)

0 is the null solution \((P|0 = 0|P = P)\)

and null molecule \((M \oplus 0 = 0 \oplus M = M)\) \((\tau_0; P = 0)\)

\(X_i\) are distinct in \(E\)

Each name \(n\) is assigned a fixed rate \(r: n(r)\)

(To translate chemistry back to processes we need a bit more than simple automata: we may have “\(+\)” on the right of \(\rightarrow\), that is we may need “\(\mid\)” after \(\pi_i\).)
CGF to Chemistry

Unary reactions.

\[ E: \quad X = \tau_r(X \mid X) \]
\[ C(E): \quad X \rightarrow r \, X + X \]

Unbounded state, but only 1 species. No problem!

Binary reactions.

\[ E: \quad A = \oplus n;B \quad C = \oplus !n;D \]
\[ C(E): \quad A + C \rightarrow \rho^{(n)} B + D \]
\[ A + C \rightarrow \rho^{(n)} B + D \]

That is:

\[ A + C \rightarrow 2\rho^{(n)} B + D \]

The same interaction can occur multiple times and must be taken into account:

Symmetric reactions:

\[ E: \quad X = \oplus !a;0 \quad Y = \oplus \oplus ?a;Y \]
\[ C(E): \quad X + X \rightarrow 2\rho(a) Y \]

The rate of \( a \) is doubled because two reactions are possible.
Automata to Chemistry

A + B → B + B
B + A → A + A
A + C → C + C
C + B → B + B
B + A → A + A
Process Rate Semantics
Same Chemistry

\[ B \rightarrow^s A \]
\[ A + B \rightarrow^r A + A \]
\[ A + A \rightarrow^{2r} A + B \]

Same chemistry, hence equivalent automata

\begin{verbatim}
  directive sample 0.002 10000
  directive plot A(); B()
  
  new a@1.0:chan()
  new b@1.0:chan()

  let A() = do !a; A() or !b; A() or ?b; B()
  and B() = do delay@1.0; A() or ?a; A()

  run 10000 of B()
\end{verbatim}
Same ODEs

\[ \tau: B \rightarrow A \]
a: \( A+B \rightarrow A+A \)
b: \( A+A \rightarrow 2A+B \)


\[ \tau: B \rightarrow A \]
a: \( A+B \rightarrow A+A \)
b: \( A+A \rightarrow B+B \)

Semantic Relationships

Processes \rightleftharpoons \text{Chemistry}

Effective Rate 
\text{OpSem} \approx \text{ODEs} \approx \text{ODEs}

Rate Equation

Law of Mass Action

Hermanns \rightarrow \text{CTMCs} \rightarrow \text{Gillespie}

Developing the Use of Process Algebra in the Derivation and Analysis of Mathematical Models of Infectious Disease

R. Norman and C. Shankland
Department of Computing Science and Mathematics, University of Stirling, UK.
{ces,ran}@cs.stir.ac.uk

Abstract. We introduce a series of descriptions of disease spread using the process algebra WSCCS and compare the derived mean field equations with the traditional ordinary differential equation model. Even the preliminary work presented here brings to light interesting theoretical questions about the “best” way to define the model.
\[
S = ?i(t); I \\
I = !i(t); I \oplus ?i(t); I \oplus \tau_r; R \\
R = ?i(t); R
\]

\[
S + I \rightarrow^+ I + I \\
I + I \rightarrow^+ I + I \\
I \rightarrow^r R \\
R + I \rightarrow^+ R + I
\]

\[
[S]^* = -t[S][I] \\
[I]^* = t[S][I] - r[I] \\
[R]^* = r[I]
\]

Automata match the standard ODE model!

The Kermack-McKendrick, or SIR model!
Simplified Model

\[ S = ?i(t); I \\
I = !i(t); I \oplus \tau_r; R \\
R = 0 \]

\[ S + I \rightarrow ^{+} I + I \\
I \rightarrow ^{r} R \]

\[ [S]^* = -t[S][I] \\
[I]^* = t[S][I] - r[I] \\
[R]^* = r[I] \]

Same ODE, hence equivalent automata models.

Not totally obvious that one could have simplified the automata model.
Lotka-Volterra
An unbounded state system!
ODE

Lotka-Volterra Equations

The Lotka-Volterra equations describe an ecological predator-prey (or parasite-host) model which assumes that, for a set of fixed positive constants $A$ (the growth rate of prey), $B$ (the rate at which predators destroy prey), $C$ (the death rate of predators), and $D$ (the rate at which predators increase by consuming prey), the following conditions hold.

1. A prey population $x$ increases at a rate $\frac{dx}{dt} = A \cdot x \cdot t$ (proportional to the number of prey) but is simultaneously destroyed by predators at a rate $\frac{dx}{dt} = -B \cdot x \cdot y \cdot t$ (proportional to the product of the numbers of prey and predators).

2. A predator population $y$ decreases at a rate $\frac{dy}{dt} = -C \cdot y \cdot t$ (proportional to the number of predators), but increases at a rate $\frac{dy}{dt} = D \cdot x \cdot y \cdot t$ (again proportional to the product of the numbers of prey and predators).

This gives the coupled differential equations

$$\frac{dx}{dt} = A \cdot x - B \cdot x \cdot y \tag{1}$$

$$\frac{dy}{dt} = -C \cdot y + D \cdot x \cdot y \tag{2}$$

Automata match the Lotka-Volterra model (with $B=D$)
Laws by ODEs
Idle Delay Law by ODEs

\[ A = \tau_\lambda; A \oplus \tau_\mu; B = A = \tau_\mu; B \]

\[ A = \tau_\lambda; A \oplus \tau_\mu; B \]

\[ A \rightarrow^\lambda A \]
\[ A \rightarrow^\mu B \]

\[ [A]^* = -\mu[A] \]
\[ [B]^* = \mu[A] \]

\[ A = \tau_\mu; B \]

\[ A \rightarrow^\mu B \]

\[ [A]^* = -\mu[A] \]
\[ [B]^* = \mu[A] \]
Idle Interaction Law by ODEs

\[ A = \text{?c}; B \]
\[ C = \text{!c}; C \]

\[ A + C \rightarrow_r B + C \]

\[ [A] \cdot = -r[A][C] \]
\[ [B] \cdot = r[A][C] \]
\[ [C] \cdot = 0 \]

It may seem like \( A \) should decrease half as fast, but NO! Two ways to explain:
- State \( A \) is memoryless of any past idling.
- Activity on \( c \) is double.

\[ A = \text{?c}; A \oplus \text{?c}; B \]
\[ C = \text{!c}; C \]

\[ A + C \rightarrow_r A + C \]
\[ A + C \rightarrow_r B + C \]

\[ [A] \cdot = -r[A][C] \]
\[ [B] \cdot = r[A][C] \]
\[ [C] \cdot = 0 \]

<table>
<thead>
<tr>
<th>0</th>
<th>200</th>
<th>400</th>
<th>600</th>
<th>800</th>
<th>1000</th>
<th>1200</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1200</td>
<td>1000</td>
<td>800</td>
<td>600</td>
<td>400</td>
<td>200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>280</td>
<td>260</td>
<td>240</td>
<td>220</td>
<td>200</td>
<td>180</td>
</tr>
</tbody>
</table>

new c@1.0:chan

let \( A() = \text{?c}; B() \)
and \( B() = () \)
and \( C() = \text{!c}; C() \)
run \( (C() | 1000 \text{ of } A()) \)
Asynchronous Interleaving

\[ \tau_{\lambda}; B \mid \tau_{\mu}; D = \tau_{\lambda}; (B \mid \tau_{\mu}; D) + \tau_{\mu}; (\tau_{\lambda}; B \mid D) \]

Amazinly, the B’s and the D’s from the two branches sum up to exponential distributions.
Want to show that B and D on both sides have the “same behavior” (equal quantities of B and D produced at all times)

[B] and [D] have equal time evolutions on the two sides provided that [A1]=[Y+A2] and [C1]=[Y+C2]. This imposes the constraint, in particular, that [A1]₀=[Y+A2]₀ and [C1]₀=[Y+C2]₀ (at time zero). The initial conditions of the right hand system specify that [A2]₀=[C2]₀=0 (since only Y is present). Therefore, we obtain that [A1]₀=[C1]₀=[Y]₀.

So, for example, if we run a stochastic simulation of the left hand side with 1000*A1 and 1000*C1, we obtain the same curves for B and D than a stochastic simulation of the right hand side with 1000*Y.
Polymerization
Bidirectional Polymerization

new c@μ new stop@1.0

\[ A_{\text{free}} = \]
\[ \text{!c}(rht); A_{\text{brht}}(rht)) + \]
\[ ?c(lft); A_{\text{blft}}(lft) \]

\[ A_{\text{blft}}(lft) = \]
\[ \text{!c}(rht); A_{\text{bound}}(lft,rht)) \]

\[ A_{\text{brht}}(rht) = \]
\[ ?c(lft); A_{\text{bound}}(lft,rht)) \]

\[ A_{\text{bound}}(lft,rht) = ?\text{stop} \]

Communicating Automata
Bound output !c(r) and input ?c(l) on automata transitions to model complexation.

Polymerization is iterated complexation.
Bidirectional Polymerization

Circular Polymer Lengths

Scanning and counting the size of the circular polymers (by a cheap trick).
Polymer formation is complete within 10t; then a different polymer is scanned every 100t.

100x$A_{\text{free}}$, initially. The height of each rising step is the size of a separate circular polymer.
(Unbiased sample of nine consecutive runs.)
Actin-like
Poly/Depolymerization

new c@μ

A_free = !c("lft,: A_{blft}(lft)) + ?c(rht): A_{brht}(rht)

A_{blft}(lft) = !lft; A_free + ?c(rht): A_{bound}(lft,rht)

A_{brht}(rht) = ?rht; A_free

A_{bound}(lft,rht) = !lft; A_{brht}(rht)

Monomer
Automatic
Free
!

?c(r)

Bound left

?c(r)

Bound right

Bound both

1000 monomers settle to ~100 polymers of size ~10

1000.0

directive sample

directive plot

val lam = 1.0 (* dissoc *)
val mu = 1.0 (* assoc *)
new c@μ chan(chan)

let Af() =
(new lft@lam:chan run
  do !c(lft); Al(lft) or ?c(rht); Ar(rht))

and Al(lft:chan) =
  do !lft; Af() or ?c(rht); Ab(lft,rht)

and Ar(rht:chan) =
  ?rht; Af()

and Ab(lft:chan, rht:chan) =
  !lft; Ar(rht)

run 1000 of Af()
Conclusions
Conclusions

- **Stochastic Collectives**
  - Complex global behavior from simple components
  - Emergence of collective functionality from “non-functional” components
  - (Cf. “swarm intelligence”: simple global behavior from complex components)

- **Artificial Biochemistry**
  - Stochastic collectives with Law of Mass Interaction kinetics
  - Connections to classical Markov theory, chemical Master Equation, and Rate Equation

- **Properties of collective behavior**
  - Simulation
  - Systematic translation to ODEs from parametric process “libraries”
  - Correspondence (or not) between stochastic and deterministic behavior

- **Interdisciplinary connections**
  - Process descriptions vs. chemical descriptions
  - Process descriptions vs. ODE descriptions