Languages & Notations for Systems Biology

Luca Cardelli
Microsoft Research
Cambridge UK

2004-09-07 NETTAB Camerino

www.luca.demon.co.uk
**Aims**

Modeling *biological systems*.  
By adapting paradigms and techniques developed for modeling *information-processing systems*.  
Because they have some similar features:  
- Deep layering of abstractions.  
- Complex composition of simpler components.  
- Discrete (non-linear) evolution.  
- Digital coding of information.  
- Reactive information-driven behavior.  
- Very high degree of concurrency.  
- "Emergent behavior" (not obvious from part list).  

2004-09-09
Methods

Model Construction *(writing things down precisely)*

Studying the notations used in systems biology.
Formulating process calculi, for various purposes.
Study their dynamics (semantics).

Model Validation *(using models for postdiction and prediction)*

Stochastic Simulation

Stochastic = Quantitative concurrent semantics.
Based on compositional descriptions.

“Program” Analysis

Control flow analysis
Causality analysis

Modelchecking

Standard, Quantitative, Probabilistic
Eukaryotic Cell

(10~100 trillion in human body)

Membranes everywhere

Structural Architecture

- Nuclear membrane
- Mitochondria
- Golgi
- Vesicles
- E.R.
- Plasma membrane (<10% of all membranes)
Functional Architecture

The Virtual Machines of Biochemistry
- Biochemical Networks - The Protein Machine
- Gene Regulatory Networks - The Gene Machine
- Transport Networks - The Membrane Machine

Systems Biology
1. “We (kind of) understand the components; but how does the system work?”
2. “Use high-throughput experiments to gather system data.”

Different chemical toolkits
Different instruction sets
Different programming models
Different notations

Gene Machine
Nucleotides

Regulation

Model Integration
Different time and space scales

Protein Machine
Aminoacids
Metabolism, Propulsion
Signal Processing
Molecular Transport

Membrane Machine
Phospholipids
Confinement
Storage
Bulk Transport

Holds genome(s), confines regulators
Directs membrane construction and protein embedding
Makes proteins, where/when/howmuch
Signals conditions and events

Implements fusion, fission
Holds receptors, actuators hosts reactions
1. The Protein Machine

• Complex folded-up shapes that:
  – Fit together, dock, undock.
  – Excite/unexcite, warp each other.
  – Bring together, catalyze, transform materials.
  – Form complex aggregates and networks.

• Mapping out such networks:
  – In principle, it’s “just” a very large set of chemical equations.
  – Notations have been developed to summarize and abstract.

An actual molecular interaction network.
(Nodes are distinct protein kinds, arcs mean that two kinds of proteins interact.)
Protein Structure

Primary
- The 20 Aminoacids

Secondary
- Alpha Helix, Beta Sheet

Tertiary
- Green Fluorescent Protein

Quaternary
- Triose Phosphate Isomerase

http://www.cmbi.kun.nl/gvteach/bioinformatica1/
### Some Allosteric Switches

<table>
<thead>
<tr>
<th>Domain architecture</th>
<th>Repressed state</th>
<th>Activated state</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(a)</strong></td>
<td><img src="diag1.png" alt="Diagram" /></td>
<td><img src="diag2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Input → Output</td>
<td>Off</td>
<td>On</td>
</tr>
<tr>
<td>SH3 → SH2 → Kinase</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(b)</strong></td>
<td><img src="diag3.png" alt="Diagram" /></td>
<td><img src="diag4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Input → Output</td>
<td>Off</td>
<td>On</td>
</tr>
<tr>
<td>SH2 → SH2 → Phos.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(c)</strong></td>
<td><img src="diag5.png" alt="Diagram" /></td>
<td><img src="diag6.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Input → Output</td>
<td>Off</td>
<td>On</td>
</tr>
<tr>
<td>EVH1 → B → GBD → pro → VCA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Domain architecture and autoinhibitory interactions in modular switch proteins. (a) Src family kinases contain N-terminal SH3 and SH2 domains, and a kinase domain flanked by intramolecular SH3-binding and SH2-binding sites (when the C-terminal motif tyrosine is phosphorylated by Csk). The crystal structures of several family members show that both intramolecular domain interactions function in concert to lock the kinase in an inactive conformation. Activating stimuli (red) include external SH2 or SH3 ligands. After initial activation, the kinase is maintained in an active state by autophosphorylation of its activation loop. (b) SHP-2 phosphatase contains two SH2 domains and a phosphatase domain. The crystal structure of the phosphatase shows that the N-terminal SH2 domain participates in an autoinhibitory interaction that directly blocks the phosphatase active site. Binding of external SH2 ligands activates by disrupting the autoinhibitory interaction. (c) N-WASP contains an Enahed WASP homology 1 (EH-1) domain, a B motif, a GBD, a proline-rich segment (pro) and an output region (VCA) that alone binds the Arp2/3 complex and stimulates its actin nucleation activity. The B and GBD motifs are required to repress activity and, by current models, are thought to participate in intracomplex interactions (only the structure of the GBD intramolecular complex for WASP is known). GTP-bound Cdc42 and FIP2 synergistically activate N-WASP.

Allosteric ("other shape") reactions modify accessibility.

**Kinase**
- donates phosphate P
- phosphorilates other proteins

**Phosphatase**
- accepts phosphate P
- dephosphorilates other proteins

**Logical AND**
at equal concentrations of the individual input stimuli, activation is much higher if both stimuli are present

"Phosphatase Kinase Kinase" = a kinase that activates a kinase that activates a phosphatase that deactivates a protein.

Humans have the same number of modular protein domains (building blocks) as worms, but twice the number of multi-domain proteins.

Taken from Wendell Lim
MIM: Molecular Interaction Maps (Kohn)

The double-arrowed line indicates that proteins A and B can bind to each other. The "node" placed on the line represents the A:B complex.

Asymmetric binding where protein A donates a peptide that binds to a receptor site or pocket on protein B.

Representation of multimolecular complexes: x is A:B; y is (A:B):C. This notation is extensible to any number of components in a complex.

Cova lent modification of protein A. The single-arrowed line indicates that A can exist in a phosphorylated state. The node represents the phosphorylated species.

Cleavage of a covalent bond: dephosphorylation of A by a phosphatase.

Proteolytic cleavage at a specific site within a protein.

Stoichiometric conversion of A into B.

Transport of A from cytosol to nucleus. The node represents A after it has been transported into the nucleus.

Formation of a homodimer. Filled circle on the right represents another copy of A. The node on the line represents the homodimer A:A.

z is the combination of states defined by x and y.

Enzymatic stimulation of a reaction.

General symbol for stimulation.
A bar behind the arrowhead signifies necessity.
General symbol for inhibition.
Shorthand symbol for transcriptional activation.
Shorthand symbol for transcriptional inhibition.

Degradation products

Taken from
Kurt W. Kohn
Molecular Interaction Maps

The p53-Mdm2 and DNA Repair Regulatory Network

Figure 6B: The p53-Mdm2 and DNA repair regulatory network (version 2) - May 19, 1999.

Taken from Kurt W. Kohn
The Protein Machine “Instruction Set”

On/Off switches

Protein

Binding Sites

Inaccessible

Inaccessible

cf. BioCalculus [Kitano&Nagasaki], κ-calculus [Danos&Laneve]

Each protein has a structure of binary switches and binding sites. But not all may be always accessible.

Switching of accessible switches.
- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.

Binding on accessible sites.
- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.
P Q \rightarrow R S \quad \text{Ordinary Chemical Reactions}

P Q \rightarrow P R \quad \text{Any combination of the above}
Notations for the Protein Machine

- **Stochastic π-Calculus**
  - Priami (following Hillston’s PEPA) formalizes a stochastic version of π-calculus where channels have communication rates.

- **BioSPI**
  - Regev-Shapiro-Silverman propose modeling chemical interactions (exchange of electrons and small molecules) as “communication”.
  - Standard stochastic simulation algorithms (Gillespie) can be used to run in-silico experiments.
  - Complex formation is encoded via π-restriction.

- **PEPA**
  - Calder Gilmore and Hillston model the ERK pathway.

- **k-calculus**
  - Danos and Laneve (following Kitano’s BioCalculus) define a calculus where complex formation is primitive.

- **(Stochastic) Petri Nets**
  - S.Reddy’94 modeling pathways.
  - Srivastava Perterson and Bentley analyze and simulate E.coli stress response circuit.

- **Bio State Charts**
  - Harel uses State Charts to model biological interactions via a semi-graphical FSM notation.

- **Pathway Logic**
  - Talcott-Eker-Knapp-Lincoln use term-rewriting.

- **BioCham**
  - ChabrierRivier-Fages-Soliman use term-rewriting and CLT model checking.

- **Kohn Diagrams, Kitano Diagrams**

- **SBML (Systems Biology Markup Language)**
  - XML dialect for MIM’s:
    - Compartments (statically nested)
    - Reagents with concentrations
    - Reactions with various rate laws
  - Read and written by many tools via the Systems Biology Workbench protocol
    - Graph editors
    - Simulators (including simulation web services)
    - Databases
2. The Gene Machine

The “Central Dogma” of Molecular Biology

![Diagram of the Central Dogma]

- DNA → messenger RNA (transcription)
- messenger RNA → protein (translation)
- Protein → systems (interaction)

- 4-letter digital code
- 4-letter digital code
- 20-letter digital code
The Gene Machine “Instruction Set”

Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

Transcription produces molecules (RNA or, through RNA, proteins) that bind to regulatory region of other genes (or that are end-products).

**Human (and mammalian) Genome Size**
- 3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
  - Non-repetitive: 1Gbp 250MB
  - In genes: 320Mbp 80MB
  - Coding: 160Mbp 40MB
  - Protein-coding genes: 30,000-40,000

**M.Genitalium (smallest true organism)**
- 580,073bp 145KB (eBook)

**E.Coli (bacteria):**
- 4Mbp 1MB (floppy)

**Yeast (eukarya):**
- 12Mbp 3MB (MP3 song)

**Wheat**
- 17Gbp 4.25GB (DVD)
Gene Composition

Is a shorthand for:

Under the assumptions [Kim & Tidor]
1) The solution is well-stirred
   (no spatial dependence on concentrations or rates).
2) There is no regulation cross-talk.
3) Control of expression is at transcription level only
   (no RNA-RNA or RNA-protein effects)
4) Transcriptions and translation rates monotonically
   affect mRNA and protein concentrations resp.

Ex: Bistable Switch

Ex: Oscillator

Expressed
Repressed
Expressing
Gene Regulatory Networks

http://strc.herts.ac.uk/bio/maria/NetBuilder/

Maternal inputs and some upstream zygotic responses

Colored boxes indicate post-gastrular domains of gene expression

Skeletogenesis-structural genes
Pigment cell, mesoderm structural genes

Endoderm structural genes
Cell motility, ema 16

Begin coding region

A

module A

Bp

DNA

Or

And

Sum

Amplify

Gate

Time varying influence

scalar factor

Inhibitory switch

Taken from Eric H. Davidson
Indirect Gene Effects

No combination of standard high-throughput experiments can reconstruct an a-priori known gene/protein network [Wagner].

One of many bistable switches that cannot be described by pure gene regulatory networks [Francois & Hakim].
Structure of the Coding Region

The Central Dogma

DNA \[\xrightarrow{\text{transcription}}\] mRNA \[\xrightarrow{\text{translation}}\] Protein

RNA is not just an intermediary; it can:
- Fold-up like a protein
- Act like an enzyme
- Regulate other transcribed RNA
- Direct protein editing
- ...

Challenging the Dogma (in higher organisms)

97-98% of the transcriptional output of the human genome is non-protein-coding RNA.
30-40,000 "protein genes" (1.5% of genome)
60-100,000 "transcription units" (>30% of genome is transcribed)
Structure of a Regulatory Region

C  Module A functions:

Vegetal plate expression in early development:

Synergism with modules B and G enhancing endoderm expression in later development:

Repression in ectoderm (modules E and F) and skeletogenic mesenchyme (module DC):

Modules E, F and DC with LiCl treatment:

Fig. 1. Endo16 cis-regulatory system and interactive roles of module A. (A) Diversity of protein binding sites and organization into modular subregions [modified from (7)]. Specific DNA binding sites are indicated as red blocks; modular subregions are denoted by letters G to A (Bp, basal promoter). Proteins binding at the target sites are indicated in this work are indicated: Otx, SpOtx-1 (12); SpGCF1 (14); the proteins CG, Z, and P, which are not yet cloned; and protein C [a CREB family protein (18)] in subregion F. Proteins for which sites occur in multiple regions of the DNA sequence (indicated by the black line) are shown beneath. (B) Sequence of module A and location of protein binding sites. Sites are indicated in the same colors as in (A). A fragment containing CG3 and CG4 sites as well as Bp has no endoderm-specific activity and serves other upstream cis-regulatory systems promiscuously; similarly, the Endo16 cis-regulatory system functions specifically with heterologous promoters substituted for Bp (5, 8, 19). Boxed sequences indicate conserved core elements of the target sites (7, 12, 14), not the complete target site sequences. (C) Integrative and interactive functions of module A (5, 8). Module A communicates the output of all upstream modules to the basal transcription apparatus. It also initiates endoderm expression, increases the output of modules B and G, and is required for functions of the upstream modules E, F, and DC. These functions are repression of expression in nonendodermal domains and enhancement of expression in response to LiCl.

[Image of the structure of a regulatory region with proteins and DNA binding sites labeled. The text explains the functions of module A in different developmental stages and includes diagrams illustrating the interactions.]
Function of a Regulatory Region

\[ \text{DNA} \quad \text{Begin coding region} \]

\[ \text{And} \quad \text{Begin coding region} \]

\[ \text{Sum} \quad \text{Begin coding region} \]

\[ \text{Amplify} \quad \text{Begin coding region} \]

\[ \text{Gate} \quad \text{Begin coding region} \]

\[ \text{time varying influence} \quad \text{Begin coding region} \]

\[ \text{scalar factor} \quad \text{Begin coding region} \]

\[ \text{inhibitory switch} \quad \text{Begin coding region} \]

\[ \text{B} \]

\[ \text{If } (F = 1 \text{ or } E = 1 \text{ or } CD = 1) \text{ and } (Z = 1) \hspace{1cm} \text{Repression functions of modules } F, E, \text{ and } \]
\[ \text{DC mediated by } Z \text{ site} \]

\[ \alpha = 1 \]

\[ \text{else } \alpha = 0 \]

\[ \text{if } (P = 1 \text{ and } \text{CG}_{1} = 1) \hspace{1cm} \text{Both } P \text{ and } \text{CG}_{1} \text{ needed for synergistic link} \]
\[ \beta = 2 \]

\[ \text{else } \beta = 0 \]

\[ \text{if } (\text{CG}_{1} = 1 \text{ and } \text{CG}_{2} = 1 \text{ and } \text{CG}_{3} = 1) \hspace{1cm} \text{Final step up of system output} \]
\[ \gamma = 2 \]

\[ \text{else } \gamma = 1 \]

\[ \xi(t) = B(t) + G(t) \]

\[ \zeta(t) = \beta \xi(t) \]

\[ \text{if } (c(t) = 0) \hspace{1cm} \text{Positive input from modules } B \text{ and } G \]
\[ \zeta(t) = \Omega(t) \]

\[ \text{else } \zeta(t) = \sigma(t) \]

\[ \text{if } (\alpha = 1) \hspace{1cm} \text{Synergistic amplification of module } B \]
\[ \tau(t) = 0 \]

\[ \text{else } \tau(t) = \xi(t) \]

\[ \theta(t) = \gamma^{*} \tau(t) \]

\[ \text{Repression function inoperative in endoderm but blocks activity elsewhere} \]

\[ \text{Final output communicated to BTA} \]

---

Taken from Eric H Davidson
Gene Machine Programs

• All that goes to show that:
  - The faithful description of even a simple genetic network is probably going to require writing a fairly substantial “program”/model.
Notations for the Gene Machine

• Many of the same techniques as for the Protein Machine apply.
  - Process Calculi, Petri Nets, Term-Rewriting Systems...

• But the “programming model” is different.
  - Asynchronous control.
  - Biologically poorly understood.
  - Network “motifs” are being analyzed.

• Specific techniques:
  - Hybrid Petri Nets
    • [Matsuno, Doi, Nagasaki, Miyano] Gene Regulation
    • Genomic Object Net
      www.genomicobject.net

• Gene Regulation Diagrams
• Mixed Gene-Protein Diagrams
3. The Membrane Machine

Molecular transport and transformation through dynamic compartment fusion and fission.

Well, what is all that for?
"Given the complicated pathways that have evolved to synthesize them, it seems likely that these [modified proteins] have important functions, but for the most part these functions are not known" [MBP p.609]
Membrane Fusion

**Aggressive fusion (virus)**

By unknown mechanisms, the exoplasmic leaflets of the two membranes fuse" [MCB p745]

**Cooperative fusion (vesicle)**

“Fusion of the two membranes immediately follows prefusion, but precisely how this occurs is not known” [MCB p742]
Membrane Fission

Assembly and disassembly of the clathrin coat

Vesicle Formation

“Nonetheless, the actual process whereby a segment of phospholipid bilayer is ‘pinched off’ to form a pit and eventually a new vesicle is still not understood” [MCB p.746]

Cytokinesis (Mitosis)
Notations for the Membrane Machine

- **“Snapshot” diagrams**
  - In biology literature.

- **P-Systems**
  
  http://psystems.disco.unimib.it/.

- **BioAmbients**
  - An extension of BioSPI along Ambient Calculus lines (with more bio-relevant mobility primitives) to model dynamic compartments.

- **Brane Calculi**
  - Computation on the membrane...
The Membrane Machine “Instruction Set”

Arbitrary subsystem

Endo → Exo

Zero case

Pino

One case

Phago

Arbitrary subsystem

Mito

Mate

Zero case

Drip

One case

Bud
Ex: Viral Reproduction
Virtual Machines of Biochemistry

Gene Machine
Nucleotides

Protein Machine
Aminoacids

Membrane Machine
Phospholipids

Regulation

Makes proteins, where/when/howmuch
Signals conditions and events
Directs membrane construction and protein embedding
Holds genome(s)
Confines regulators
Holds receptors, actuators hosts reactions
Model Integration
Different time and space scales

Metabolism, Propulsion
Signal Processing
Molecular Transport

Confinement
Storage
Bulk Transport

Implements fusion, fission
Modeling Biological Processes
Storing Processes

• Today we represent, store, and analyze:
  - Gene sequence data
  - Protein structure data
  - Metabolic network data
  - Compartmentalized reaction data (SBML)
  - …

• How can we represent, store, and analyze biological processes?
  - Scalable, precise, dynamic, highly structured, maintainable representations for systems biology.
  - Not just huge lists of chemical reactions or differential equations.
Computational Modeling Approaches
-- Diverse Spectrum

SPECIFIED

differential equations

Markov chains

Booleans models

Bayesian networks

relationships

ABSTRACTED

* mechanisms

(including structure)

influences

* statistical mining

Pacific Northwest National Laboratory
U.S. Department of Energy
Where are the scalable, precise, dynamic, highly structured, maintainable representations of biological processes?
A process calculus (chemistry, or SBML)

\[
\text{Na} + \text{Cl} \rightarrow_{k_1} \text{Na}^+ + \text{Cl}^- \\
\text{Na}^+ + \text{Cl}^- \rightarrow_{k_2} \text{Na} + \text{Cl}
\]

This graphical representation degenerates into spaghetti diagrams: precise and dynamic, but not scalable, structured, or maintainable.

A different process calculus

\[
\text{Na} = e_{k_1}!. e_{k_2}?. \text{Na} \\
\text{Cl} = e_{k_1}?. e_{k_2}!. \text{Cl}
\]
Write Things Down!

Sydney Brenner: "When you want to have a predictive science, you have to be able to calculate."

When you want to calculate, you have to be able to write things down.

- Write biological systems as programs, as if they were software systems
  - Software is a precise (yet not quite predictable) notation for systems of high structural and combinatorial complexity.
  - Small programs can express highly complex behavior. Especially true in nondeterministic concurrency (and in deterministic chaos).
  - We don’t use differential equations to write operating systems.

- Write them as text, to better describe dynamic behavior
  - Not as cartoons or diagrams
  - Need to choose a syntax
    - Always a food fight.
    - But needed for tools to work on: simulation, analysis, storage, search.
  - In C++, Prolog, Etc.?
    - Not likely... We need highly concurrent analyzable formal languages.
  - Representing processes, not just data. Concurrency, stochasticity.
Stochastic π-calculus Executive Summary

- A textual process calculus:
  - The modular representation of concurrent (and stochastic) processes of all kinds.
  - Cuts down to CTMCs in the finite case (not always), then standard tools are applicable.
  - Can be given friendly automata-like scalable graphical syntax (work in progress).
  - Is directly executable (e.g. via Gillespie).
  - Is analyzable (large body of literature, at least in the non-stochastic case).

Figure 2. Regulating Gene Expression by Positive Feedback [8]

Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF
### Molecule vs. Process

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction capability</td>
<td>Channel</td>
</tr>
<tr>
<td>Interaction</td>
<td>Communication</td>
</tr>
<tr>
<td>Modification</td>
<td>State change</td>
</tr>
</tbody>
</table>

Cellular Abstractions: Cells as Computation  
Regev&Shapiro NATURE vol 419, 2002-09-26, 343

This mapping works well both for the “protein machine” (synchronous communication) and the “gene machine” (asynchronous communication). But is not enough for the “membrane machine”.
BioAmbients

- An extension of Sto-π-calculus
  - Dynamic membranes: operations for merging, splitting, interacting through membrane channels.
  - Good abstraction:
    - partitions subsystems
    - models membranes as a whole
  - Implemented by Aviv Regev.
- An adaptation of Ambient Calculus
  - A process language for dynamic containers (mobile agents, distributed locations, etc.)

BioAmbients: An Abstraction for Biological Compartments.
TCS, Special Issue on Computational Methods in Systems Biology. Elsevier.
(Elsevier Articles in Press)
Multi-Level Organization

Hormone transport

Neuron

Brain area

Hormone transport

Fig. 10. Hypothalamic pathways for weight regulation. A partial view of molecular pathways, neurons and nuclei involved in weight control. Oreogenic (weight gaining) signals are in green, anorexic (weight loss) ones are in red. For further details see the main text. Adapted from [17].

Weight regulation system from the literature.

Example chosen because it involves several levels of biological organization: molecular, cellular, and anatomical.

BioAmbients representation of the same system.

(Schematic representation of a BioAmbients script, hand-drawn)

Stochastic simulation
(1 neuron per functional area, ~100 receptors per neuron)

N.B.: discrete processes: thousands of components are enough, not billions.

2004 © Regev
So You Modeled Something

And then what?
Model Validation: Stochastic Simulation

- **Basic algorithm: Gillespie**
  - Exact (i.e. based on physics) stochastic simulation of chemical kinetics.
  - Can compute concentrations and reaction times for biochemical networks.

- **Stochastic Process Calculi**
  - BioSPI [Shapiro, Regev, Priami, et. al.]
    - Stochastic process calculus based on Gillespie.
  - BioAmbients [Regev, Panina, Silverma, Cardelli, Shapiro]
    - Extension of BioSpi for membranes.
  - Case study: Lymphocytes in Inflamed Blood Vessels [Lecaa, Priami, Quaglia]
    - Original analysis of lymphocyte rolling in blood vessels of different diameters.
  - Case study: Lambda Switch [Celine Kuttler, IRI Lille]
    - Model of phage lambda genome (well-studied system).
  - Case study: VICE [U. Pisa]
    - Minimal prokaryote genome and metabolism of VIrtual CELl, in stochastic π-calculus.
Causality Analysis

- **Biochemical pathways**, ("concurrent traces" such as the one here), are found in biology publications, summarizing known facts.
- This one, however, was automatically generated from a program written in BioSpi by comparing traces of all possible interactions. [Curti, Priami, Degano, Baldari]
- One can play with the program to investigate various hypotheses about the pathways.

Control Flow Analysis

- Flow analysis techniques applied to process calculi.
- Overapproximation of behavior used to answer questions about what "cannot happen".
- Analysis of positive feedback transcription regulation in BioAmbients [Flemming Nielson].

Probabilistic Abstract Interpretation

- [DiPierro Wicklicky].
Model Validation: Modelchecking

• **Temporal: NuSMV**
  [Chabrier-Rivier Chiaverini Danos Fages Schachter]
  - Analysis of mammalian cell cycle (after Kohn) in CTL.
    - E.g. is state $S_1$ a necessary checkpoint for reaching state $S_2$?

• **Quantitative: Simpathica/xssys**
  [Antioniotti Park Policriti Ugil Mishra]
  - Quantitative temporal logic queries of human Purine metabolism model.
    
    $$\text{Eventually}(\text{Always} \ (PRPP = 1.7 * PRPP1))$$
    implies
    $$\text{steady\_state()}$$
    and $$\text{Eventually}(\text{Always} \ (IMP < 2 * IMP1))$$
    and $$\text{Eventually}(\text{Always} \ (hx\_pool < 10*hx\_pool1)))$$

• **Stochastic: Spring**
  [Parker Normal Kwiatkowska]
  - Designed for stochastic (computer) network analysis
    • Discrete and Continuous Markov Processes.
    • Process input language.
    • Modelchecking of probabilistic queries.
What Process Calculi Do For Us

We can write things down
- We can modularly describe high structural and combinatorial complexity ("do programming").

We can calculate and analyze
- Directly support simulation.
- Support analysis (e.g. control flow, causality, nondeterminism).
- Support state exploration (model checking).

We can reason
- Suitable equivalences on processes induce algebraic laws.
- We can relate different systems (e.g. equivalent behaviors).
- We can relate different abstraction levels.
- We can use equivalences for state minimization (symmetries).

Disclaimers
- Some of these technologies are basically ready (medium-scale stochastic simulation and analysis, medium-scale nondeterministic and stochastic model checking).
- Others need to scale up significantly to be really useful. This is (has been) the challenge for computer scientists.
And There Are More...

- Many other approaches, same basic philosophy, tools being built.
  - State Charts, Live Sequence Charts [Harel]
    - Hierarchical automata.
    - Scenario composition.
  - Charon language [UPenn]
    - Hybrid systems: continuous differential equations + discrete/stochastic mode switching.
  - Etc.

May the fittest survive...
Conclusions

Q: “The data are accumulating and the computers are humming, what we are lacking are the words, the grammar and the syntax of a new language…”
D. Bray (TIBS 22(9):325-326, 1997)

A: “The most advanced tools for computer process description seem to be also the best tools for the description of biomolecular systems.”
E. Shapiro (Lecture Notes)
References


Papers

*BioAmbients*
- a stochastic calculus with compartments.

*Brane Calculi*
- process calculi with computation “on” the membranes, not inside them.

*Bitonal Systems*
- membrane reactions and their connections to “local” patch reactions.

www.luca.demon.co.uk/BioComputing.htm