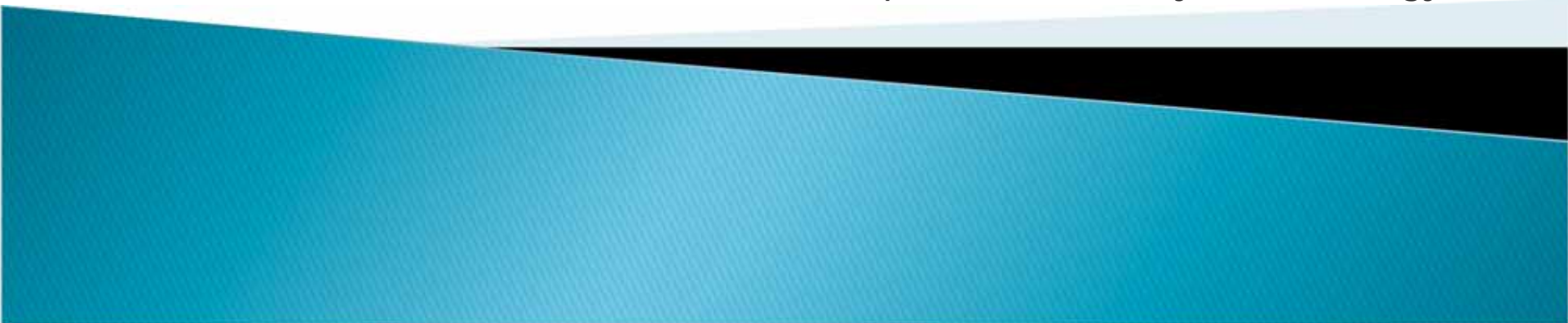


# From Computer Science to Systems Biology and vice-versa

New modeling challenges, approaches and tools

Ivan Mura

The Microsoft Research – University of Trento  
Center for Computational and Systems Biology



# Outline

- ▶ Context
- ▶ Modeling for systems biology
  - objectives
  - approaches
  - tools
- ▶ Challenges
- ▶ Solutions devised in systems biology
  - hooks for computer sciences
- ▶ Summary

# Context

# Biological research

- ▶ The scientific community of biologists outnumbers by far all others
  - a hot research area
  - big private investments (in 2006, Pharma and BioTech, 100B\$)
- ▶ Most of these resources are spent in experimental work in molecular biology studies
- ▶ Technological progress
  - increasing observability
  - speeding-up experiment execution
- ▶ A huge amount of experimental data is being generated
  - a fraction is available in various public repositories over the Internet

# Systems Biology into play

- ▶ The complexity of biological systems soon called for mathematical tools
- ▶ Computers support to mathematical biology approaches has generated two main areas of activity
  - Bioinformatics
  - Computational Biology
- ▶ Recently, the aim to integrate knowledge coming from traditionally separate areas of biology (genetics, proteomics, metabolomics) has led to **Systems Biology**
- ▶ The fundamental paradigm of Systems Biology
  - behavior is emerging from the dynamical interaction of components
  - systems should be studied with tools able to represent this

*...understand complex biological systems through the integration of experimental and computational research [H. Kitano, 02]*

# Research community size

A fast growing research area

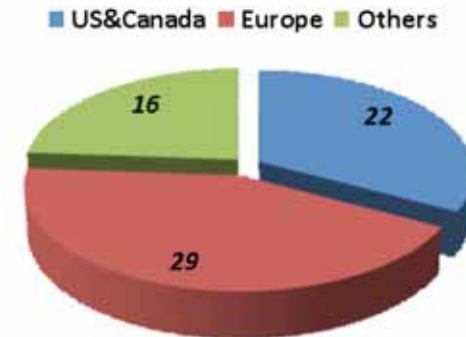
Around 70 international conferences and workshops in 2007 on related subjects

- 1100 attendants at the International Conference on Systems Biology 2008 in Gothenburg, Sweden

Standardization efforts

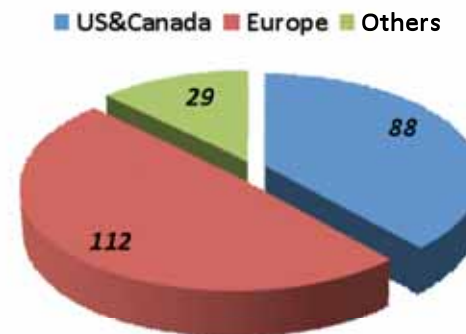
- SBML, CellML, BioPax
- SBGN
- SBO, SBRML
- 105 computational tools registered as SBML compliant

Systems Biology Institutes



[2007] source: [www.nature.com](http://www.nature.com)

Systems Biology Departments

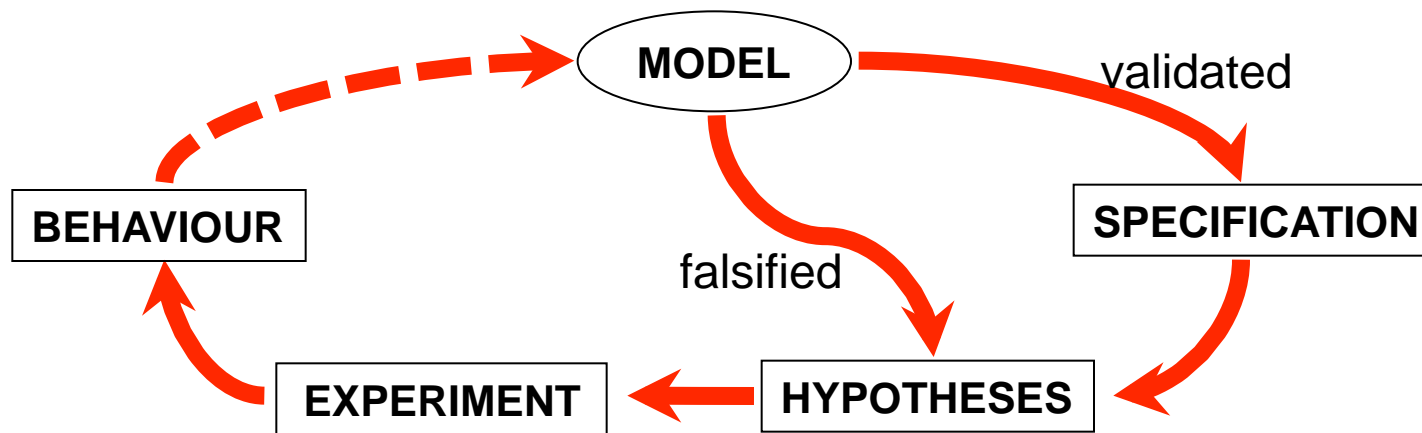


[today] source: [emb1.bcc.univie.ac.at](http://emb1.bcc.univie.ac.at)

# Objectives

# Models in a reverse engineering loop

- ▶ All in all, the main objective of modern biology is to solve a substantial problem of *Reverse Engineering*



- ▶ Model: a formal representation, which when
  - validated confirms the validity of the inferred knowledge used to build it
  - invalidated allows postulating new hypotheses and driving definition of experiments

# Predictive models

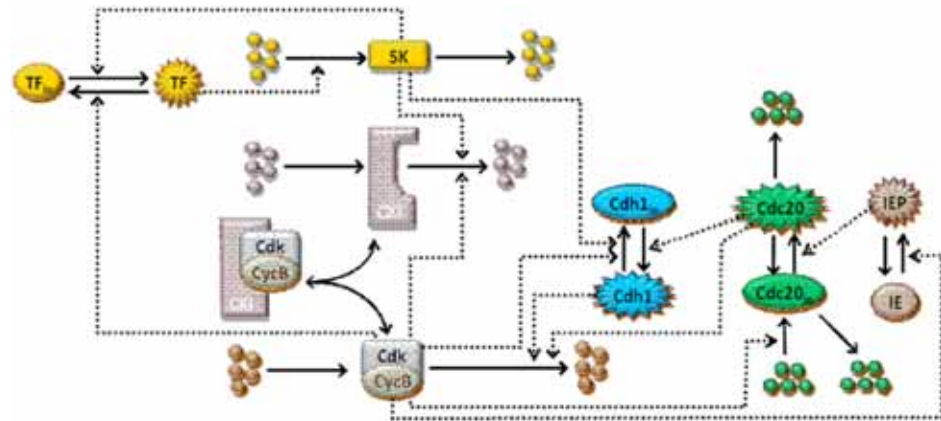
- ▶ Validated models are used for predictive purposes
  - refinement of available knowledge through deduction
- ▶ Many experimental scenarios hardly accessible in *wet-lab* experiments may be evaluated at low cost with the *in-silico* approach
- ▶ Example: gene silencing
  - in-silico: set one variable to FALSE
  - wet-lab: DNA engineering or RNA interference
- ▶ Experiments run on a model can significantly reduce the effort required in the lab

# Modeling approaches in SB

# Classical approaches

Biologists mostly use unstructured graphical models for encoding knowledge about systems

- unclear semantics
- lack of quantitative information
- generalizations totally overlooked



A more expressive reaction based specification language has been borrowed from chemistry

- $\emptyset \rightarrow A, \emptyset \rightarrow B$
- $A+B \rightarrow C, C \rightarrow A+B+C$
- $C \rightarrow \emptyset$

Models based on systems of ordinary differential equations

- quantitative information expressed in the form of kinetic rate constants

# Intrinsic discreteness

- ▶ The truly molecular nature of biological interaction was considered hardly tractable
  - tracking single molecule state, location and movement is indeed quite heavy from a computational point of view
- ▶ This was considered to be true until 1976, when D. T. Gillespie
  - proved that the evolution of a well-stirred biochemical system can be accurately modeled by a continuous time discrete space Markov process
  - provided a very simple and extremely efficient simulation algorithm for computing realizations of such process
- ▶ Gillespie's algorithm (SSA) has paved the way for a number of discrete modeling approaches

# Algorithmic approaches

- ▶ Algorithmic biology aims at representing causality in biological transformations
- ▶ Fueled by Gillespie result, new modeling tools have been proposed
  - discrete state-space
  - stochastic reaction times

## Petri Nets

### Modeling metaphor

- tokens count the number of molecules of species
- transition model reactions

### Firing rates

- Transition rates always dependent on the marking of input places

## Process Algebra

### Modeling metaphor

- processes represent biological entities
- interactions are represented as communications on a channel

### Communications based on affinity

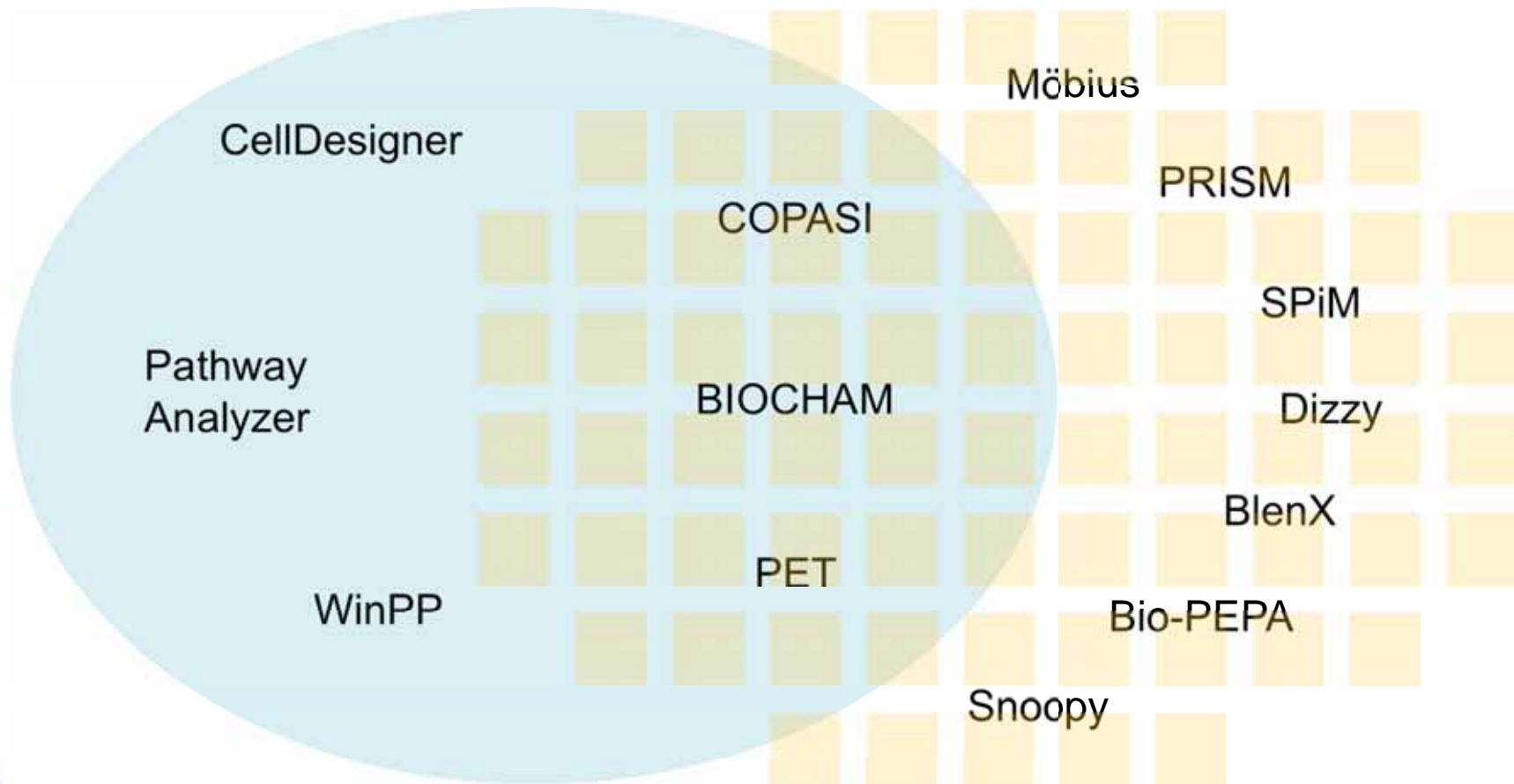
- interaction likelihood is defined through *affinities* of process

# Tools in SB

# An historical perspective

- ▶ Since the beginning of the Human Genome project, computational support to biology has come through *bioinformatics* tools
  - String manipulation
  - Databases
  - Data mining
  - Statistical applications (clustering)
- ▶ The 90's have seen a spread of tools for continuous modeling borrowed from physics approach to biology
  - ODEs and PDEs solvers
  - Metabolix flux analysis
- ▶ During the last decade, tools developed within the computer science community started to be used
  - Petri Nets (1998, Goss-Peccoud) and Process Calculi
  - P-systems
  - Model checking

# The current situation



# Measures of interest

- ▶ Typical quantitative aspects of interest on biological systems
- ▶ How resilient is a system to perturbations? If a gene is silenced, what will change in
  - the probability of entering deadly states
  - the speed of metabolism
  - the patterns of genes activation
- ▶ What are the likely causes of a wrong system response?
  - which kinetic rate determine the observed phenotype
- ▶ How can we interfere on a system that is wrongly responding to bring it back into operation?
  - which reactions should be targeted by a drug
  - which entities should be removed

# Domain-specific challenges

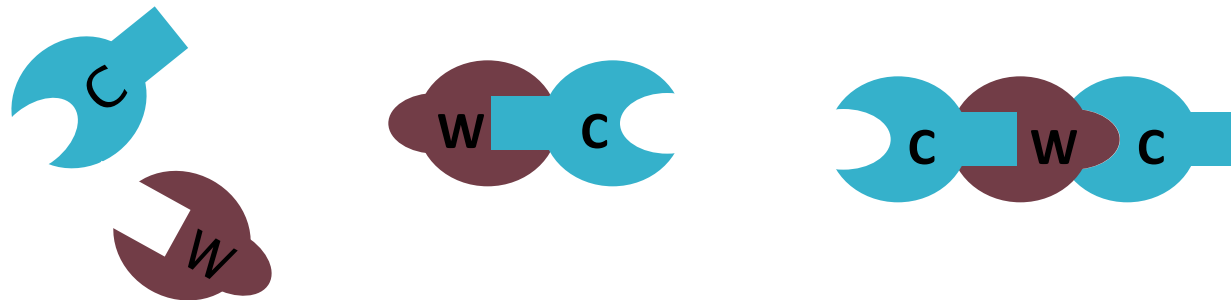
# Number of entities

- ▶ Biological systems have to deal with molecular noise
  - predictable behaviors emerge from large numbers effect
  - in the small volume of a cell nucleus there can be thousands of copies of a molecule type
- ▶ Different scales of multiplicities within a single system
  - 1 copy of a gene
  - $10^9$  molecules in one cell nucleus
  - $10^6$  synapses for one neuron
  - $10^{14}$  cells in the human organism
- ▶ Immediate consequences on state spaces
  - $10^{24}$  states in a toy cell cycle model

# Dynamic creation of entities

- ▶ Biological compounds have *sites* of interaction

- multiple sites can be present in the same entity
- bindings occur reversibly between **2** affine sites
- complexes of biological components can assemble without a precise order and can result in different topological structures
- example: protein C has 2 sites, both affine to 2 sites of protein W

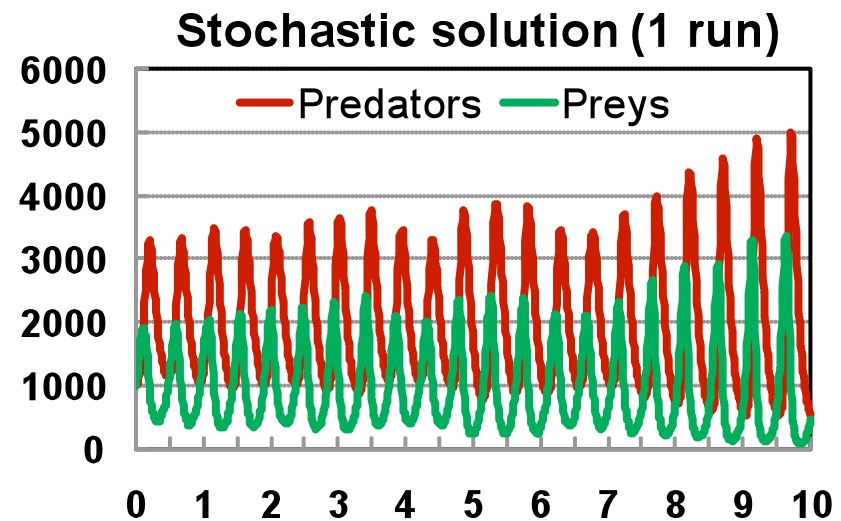


How many structures can form?

- It may be cumbersome or even impossible to specify such a behaviors in many formalisms

# Oscillatory behaviors

- ▶ Many biological systems achieves equilibrium conditions that are not commonly found in artificial systems
  - living systems keep oscillating
- ▶ Many systems have transient oscillation that stop abruptly
  - dead
- ▶ This poses issues in
  - defining adequate measures that can characterize cyclic system behavior
  - comparing similar but different systems



# Partial system knowledge

- ▶ **Known unknowns**
  - many biological entities are only partially characterized
  - interaction among entities are not always observable and thus values of many parameters to be used in models are unknown
- ▶ **Unknown unknowns**
  - not all the entities participating in an interaction network are known
  - we may not know which abstractions are actually used when defining models
- ▶ **Modularity is only apparent**
  - the number of roles and functions of entities keeps growing
  - one input rarely corresponds to a single response

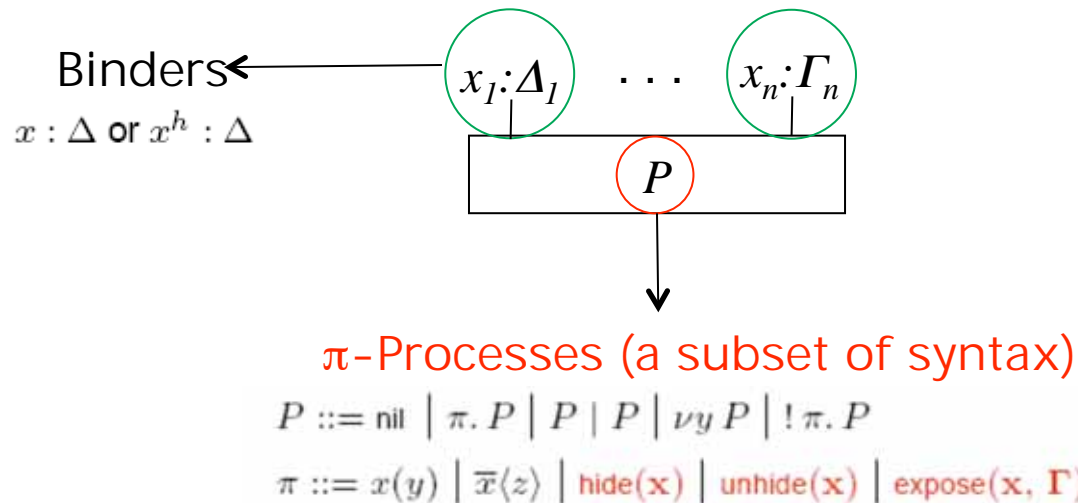
# Solutions devised in SB

# To handle big size populations

- ▶ **Continuous approximation**
  - the number of entities is approximated into a proportional concentration
  - variations of concentrations are modeled as changes in their first derivative
  - models are sets of non-linear ordinary differential equations, solved through numerical integration
- ▶ **Many tools exist for continuous ODE modeling**
  - reaction-based languages are commonly used for specification
  - ODEs are automatically obtained from reactions
  - efficient numerical solvers handle large/stiff models
  - time-dependent, equilibrium, vector fields and bifurcation analyses
- ▶ **Work in progress...**
  - some theoretical and experimental results show interesting relationship between results of discrete and continuous models

# To handle dynamic creation of new entities

- ▶ Interaction-based modeling languages based on process algebra
- ▶ BlenX encapsulates  $\pi$ -calculus processes into boxes with interaction capabilities

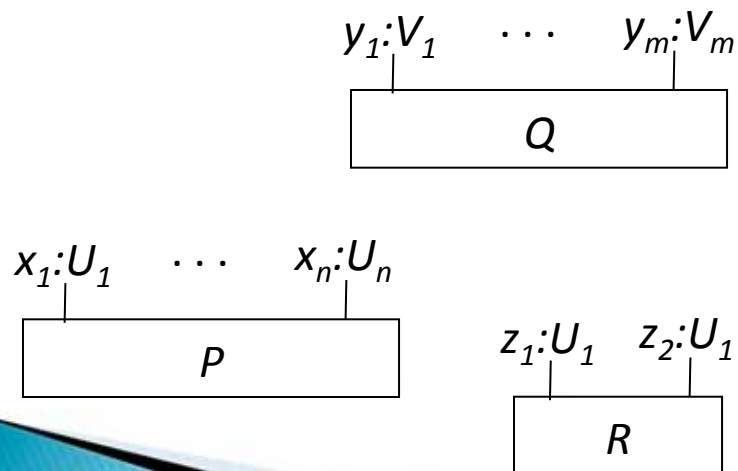


- ▶ Evolutions of the internal process change the state of the box and of its interaction capabilities

# A separate specification

## INTERFACES

- ▶ The set of interaction capabilities of entities are modeled by binders
- ▶ At any moment, interaction can only happen through visible binders
- ▶ Binders are typed



## COMMUNICATION RATES

- The rate at which interaction happen through binders is specified by a type affinity table
- Multiple rates can be used to specify rate of start, failure, completion of the interaction

affinities			
$U_1, V_1$	$0$	$0$	$0$
$U_1, V_2$	$r_{12}$	$0$	$0$
$U_1, V_3$	$r_{13}$	$k_{13}$	$c_{13}$
....			

# An example: Web services

- ▶ Web services use standardized XML messaging
- ▶ Allow for self-descriptive and discoverable services

## WDSL

XML language to specify

- messages: types of data exchanged
- ports: sets of abstract operations defining offered services

## WSCI

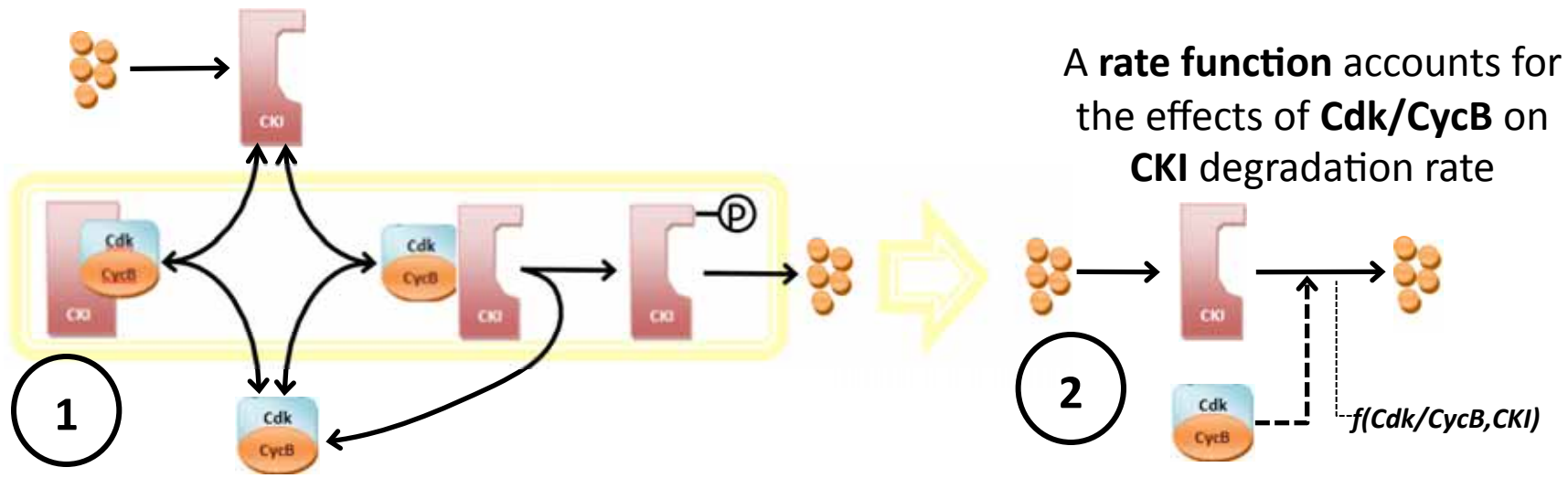
XML language to specify

- a refinement of WDSL ports detailing on externally visible interfaces
- who can participate in an interaction

- WDSL and WSCI specifications can be parsed to automatically obtain a BlenX model
- Quantitative information can be added to the model to conduct simulations

# To manage unknowns

- ▶ An ideal abstraction usage: we want to simplify ①



- ▶ The real abstraction usage: current knowledge only allows building ②
- ▶ However, a good news is that we can obtain rate functions inferred from wet-lab experiments

# To speed-up stochastic simulation

- ▶ Gillespie's family of Stochastic Simulation Algorithms
- ▶ Fundamental hypothesis
  - times of occurrence of every reaction in the system follow a negative exponential distribution
- ▶ Let
  - $R_1, R_2, \dots, R_m$  the reaction set
  - $X(t) = \mathbf{x}$  the state of the system
  - $a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_m(\mathbf{x})$  the reaction rates, also called *propensities*
  - $a_0(\mathbf{x})$  defined as  $\sum_j a_j(\mathbf{x})$

# Direct method (1976)

- ▶ Given  $X(t)=\mathbf{x}$  , the probability that the next reaction happens in the infinitesimal time interval  $[t+\tau, t+\tau+dt]$  and is a reaction of type  $j$  is

$$a_j(\mathbf{x}) \cdot \exp(-a_0(\mathbf{x}) \tau)$$

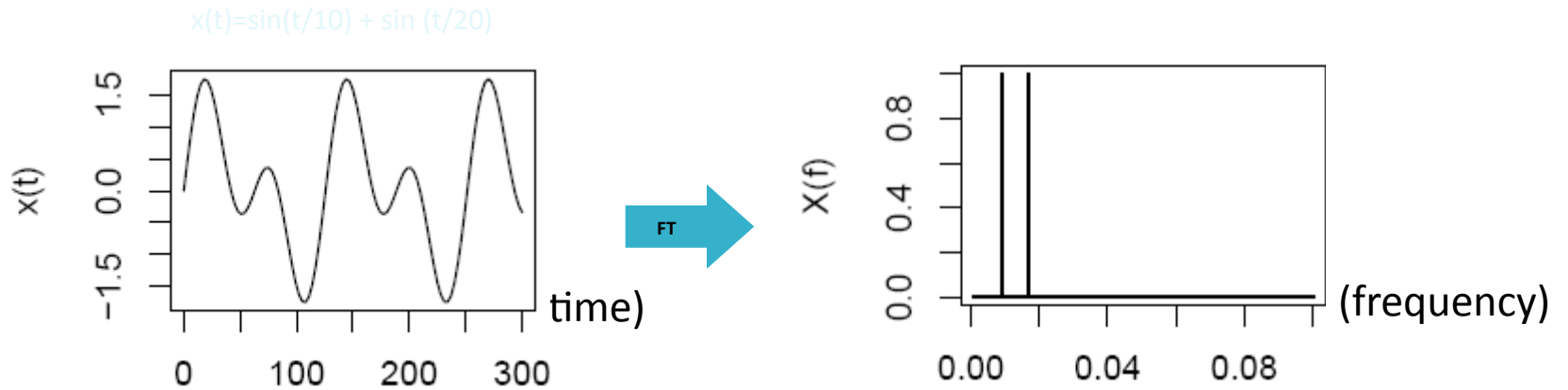
- the time  $\tau$  to the next reaction is an exponential random variable of mean  $1/a_0(\mathbf{x})$
  - the probability that next reaction is of type  $j$  is  $a_j(\mathbf{x}) / a_0(\mathbf{x})$
- ▶ At each simulation step, 2 uniform r.n.  $u$  and  $v$  are drawn
    - $\tau$  is chosen to be  $\ln(u^{-1})/a_0(\mathbf{x})$
    - $j$  is chosen as the smallest integer satisfying  $\sum_{i=1}^j a_i(x) > v \cdot a_0(x)$

# Reformulations of the method

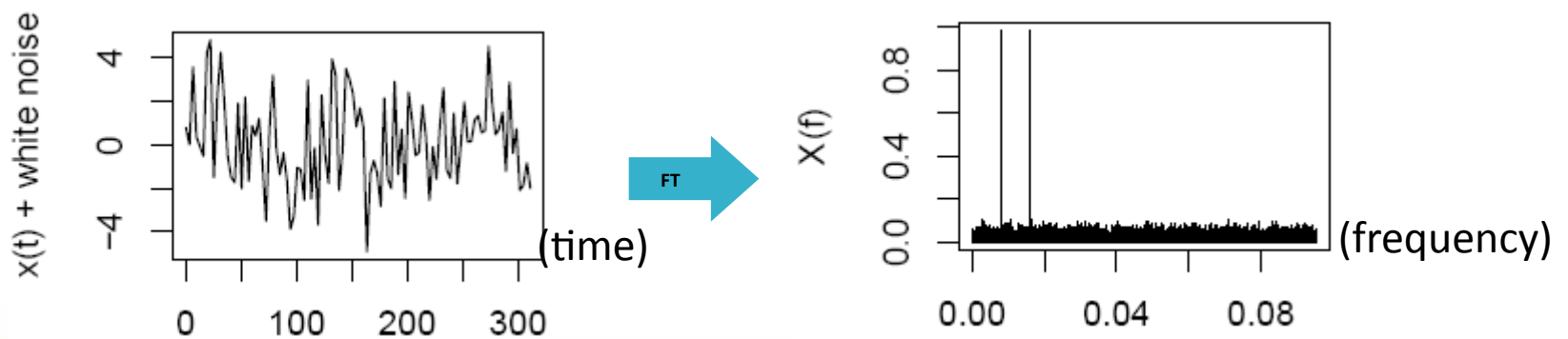
- ▶ First reaction method (1976)
  - at each simulation step, draw  $m$  uniform r.n. and compute  $\tau_1, \tau_2, \dots, \tau_m$ , the putative time of all reactions
  - choose  $\tau$  as the  $\min(\tau_1, \tau_2, \dots, \tau_m)$
  - choose  $j$  as the index of the minimum above
- ▶ Next reaction (2000)
  - same as the above one, but the putative times are saved in an indexed binary tree so that the minimum is always at the top
  - a dependency graph is used to keep track of coupling among reactions to determine when putative times in the tree have to be resampled
- ▶ Modified direct method (2004)
  - a pre-run to determine a suitable order of reactions to minimize cost of step 2)
- ▶ Sorting direct method (2006)
  - self-adaptive version of the one above, no pre-run

# To analyze oscillatory regimes

- ▶ Convert time series to frequency spectra

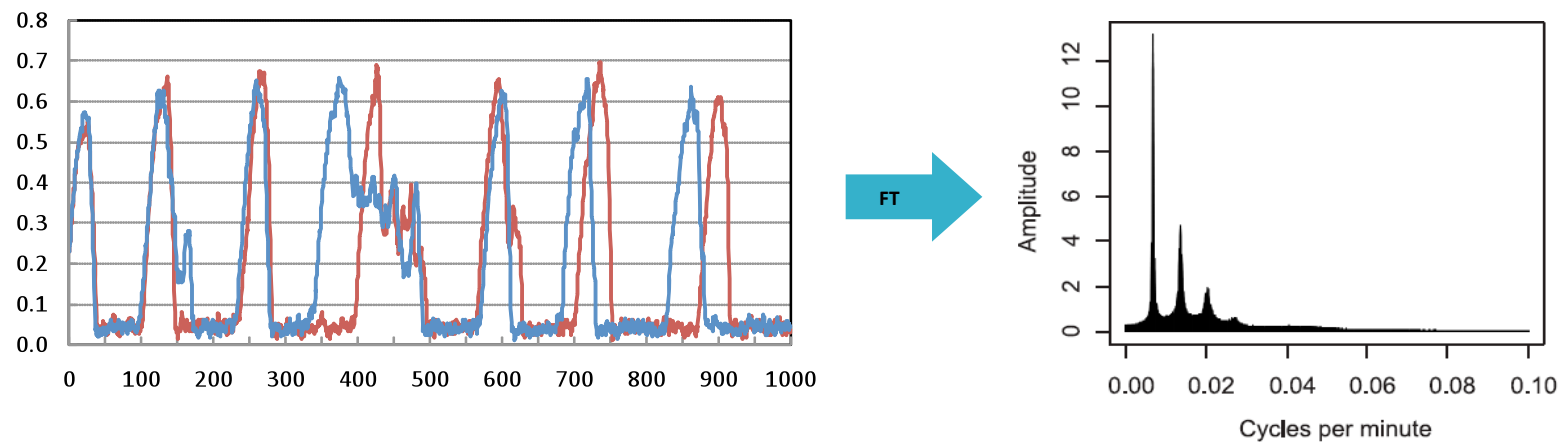


- ▶ Widely used in hardware



# Statistical measures over FA

- Spectra of multiple stochastic runs are averaged



## ▶ Three measures

$$\rho1 = \log_2(\max(f_{1..N-1})/\langle f_{1..N-1} \rangle) \quad \text{log(peak/mean)}$$

$$\rho2 = \sigma(f_{1..N-1})/\langle f_{1..N-1} \rangle \quad \text{coefficient of variance}$$

$$\rho3 = \sup |F_{0..N-1}^1 - F_{0..N-1}^2| \quad \text{Kolmogorov - Smirnov statistic}$$

$f_\omega = \omega^{\text{th}}$  complex frequency component,  $F =$  cumulative frequency distribution of  $f$

# Summary

- ▶ Models play a key role in Systems Biology
- ▶ Some modeling challenges are shared with computer science, some others are domain specific
- ▶ Approaches and tools are in an explorative phase
- ▶ Some solutions independently devised may be useful/improve over current practice in computer science