Abstract Behavioural Modelling of Biomolecular Circuits

Radu Muschevici, Reiner Hähnle

Software Engineering Group

Department of Computer Science, TU Darmstadt

— joint work with —

Thea Lotz, Beatrix Süß, Marc Vogel
Department of Biology, TU Darmstadt

2nd ABS Workshop
Darmstadt, 2018-05-29

CompuGene

- Inter-disciplinary collaboration between biologists, physicists, computer scientists, chemists, philosophers. . .
- Goal: develop & implement computer-based methods for the synthesis of robust complex genetic circuits in biological systems.
- Replace current trial-and-error approaches with more efficient ones.

Our role in the project

- Model and analyse genetic circuits
  using methods for formal specification of software.
- We use ABS, with all its advantages
  over programming languages & maths based techniques.

Biomolecular Engineering & Synthetic Biology

Biomolecular Engineering

- Manipulation of key biomolecules: nucleic acids, proteins, carbohydrates, etc.
- Applications: environment, agriculture, energy, food production, biotechnology, medicine, . . .

Synthetic Biology

- Construction of *synthetic* molecular and cellular machinery and systems
- *Engineering* of biology: synthesis of biological systems that display functions that do not exist in nature

Systems Biology: a Cynical View

- Systems Biology: the current **gold rush** for many **mathematical** and technical disciplines looking for **nutrition** (funding, self-esteem) in the **scientific food chain**
- Biophysics, Biomatics, Bioinformatics, Biostatistics...
- The story goes like this:
- I do X
- I do it for my pleasure, because I studied it, and anyway, this is the **only** thing I will do in my current incarnation...
- ...fortunately X is **very** useful for Biology
- When you have a **hammer**, everything looks like a **nail**

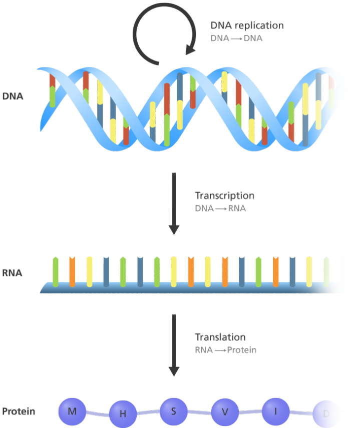
Systems Biology: an Arrogant View

- Biologists are essentially very **concrete** beings, spending most of their time in the **kitchen** doing manual work
- They were not selected (initially) based on ability to manipulate **imaginary concepts** or creativity and rigor in the abstract world of ideas but rather..
- ..based on their **rigor** and **efficiency** at the **bench**
- Now when they need to make a **real science** out of their details they need noble white collar brahmins, namely..
- ... physicists, mathematicians, computer scientists, as **spiritual guides**
- Like monotheists converting the pagans, these merchants of abstract methodologies try to impress the poor savage with their **logics** and **miracles**

Systems Biology: a Humble View

- Biologists are working with the most fascinating, complex and mysterious **real-life** phenomena
- Living systems are more complex than the **hydrogen atom** or the **electromagnetic field** (and are not effectively reducible to them)
- Living systems are more sophisticated than your **dumb terminal** or **smart phone** or mobile robot or car
- Living systems are more mysterious and primordial than the **prime numbers**, the **algebra of Boole** or the **free monoid**
- If some of our dry tricks can help them, even a bit, in their grand march toward..
- ..understanding something about **Life Itself** or helping doctors kill less patients
- We should be very happy and proud for doing, for once, something **meaningful**

The Central Dogma of Molecular Biology: Replication, Transcription, Translation



DNA & RNA

- Sequences of 4 types of *nucleotides*
- DNA molecule contains a living organism's *entire genetic code*
- RNA molecule encodes *individual gene*

Protein

- Sequence of 20 types of *amino acids*
- Performs the functions of a cell

DNA $\rightarrow$ RNA $\rightarrow$ Protein: *Gene expression*

code $\rightsquigarrow$ behaviour

Translation of mRNA to Protein

Transcription produces several types of RNA, including *messenger* RNA (mRNA).

Translation of mRNA to protein

- *Ribosomes* slide along mRNA (5' → 3')
- Read 3 nucleotides (a *codon*) at a time
- For each codon, a specific amino acid is produced, appended to previous
- Complete chain of amino acids = protein

©Nature Education

The Suess Lab

Goals

- Study role of RNA in *regulating* gene expression
- Engineering of *riboswitches* that can regulate gene expression

Motivation

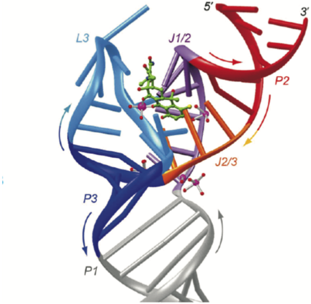
- Types of proteins produced determine the function & activity of a cell
- By controlling gene expression, one can influence how a cell functions

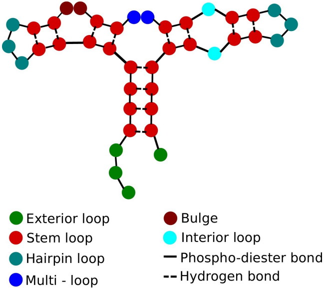
Approach

- Experiments in the wet-lab

Ribonucleic Acid (RNA)

- RNA molecule is *single stranded*.
- It folds into complex, stable 3-dimensional shapes (*secondary structure*).
- *Shape* influences *translation*.

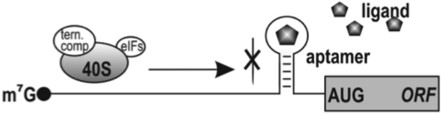


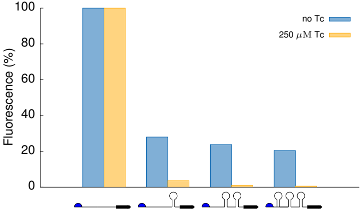


Research Question

An Improved Understanding of . . .

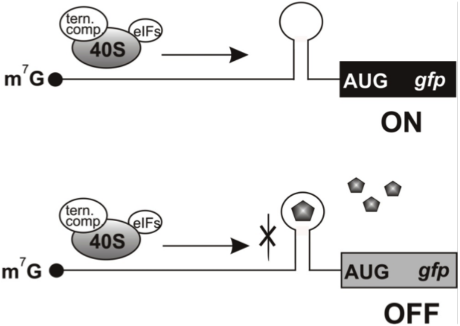
- the influence of mRNA secondary structures on the efficiency of translation
- specifically, the influence of *aptamers*





Why is This Important?

Create modular building blocks for biomolecular circuit design



Requirements

Successful construction of synthetic biological circuits requires

- precise understanding of intra-cell behavior
- predictable behavior
- repeatable outcome
- robustness (in a variable environment)

Why is This Difficult?

Huge Design Space for Precise Modeling

Aptamer-mediated translation influenced by many factors, including…

- nucleotide sequence of aptamers
- position of aptamers along mRNA
- number of deployed aptamers
- kind and concentration of ligand used

Why is This Difficult?

Mechanisms that cause observed behavior partially unknown. . .

- How does mRNA secondary structure affect ribosomal migration?
- How does ribosomal migration affect mRNA secondary structure?
- How do ligands alter secondary structure?

Difficulties in Experimentation

- Considerable effort to perform experiments (person days, manual effort)
    - sparse measuring points
    - slight variations in experimental set-up
- Direct observation impossible (instead, e.g., fluorescence)
- Measurement/analysis of single cell behaviour difficult
- Huge experimental design space
- How generalizable/transferable/comparable are experimental results?

Our Approach

Modeling biological organisms as concurrent, OO software

NOT: programming a simulator
BUT: the software behaves like the organism it models

- Experiment “in silico”, better “in ordinatro”
- Need suitable modeling language . . .

ABS:
An Abstract Modeling Language

The Abstract Behavioral Specification (ABS)* Language, 2009ff

- Formal, executable, concurrent modeling language
- Object-oriented style, Java-ish syntax
  - Domain modeling with data types or with objects/classes
- Eclipse-based and web-based IDEs available
- Concurrent execution designed to be verifiable
- Complete ABS models can be (non-deterministically) simulated
- (Partial) models can be analyzed:
  - Resource consumption, including termination
  - Formal verification against (safety) properties

*`http://www.abs-models.org/`

Models as Programs

Rationale

1. Programming languages are notations that express the behavior of a (computing) system in a concise manner

Models as Programs

Rationale

1. Programming languages are notations that express the behavior of a (computing) system in a concise manner
2. Decades of research into programming languages resulted in well-understood and practically useful concepts:

Data abstractions:	data types, object-oriented models, . . .
Computation abstractions:	procedure calls, iterators, local variables, . . .
Concurrency abstractions:	processes, synchronization, scheduling, . . .

Models as Programs

Rationale

1. Programming languages are notations that express the behavior of a (computing) system in a concise manner
2. Decades of research into programming languages resulted in well-understood and practically useful concepts:

Data abstractions: data types, object-oriented models, . . .
Computation abstractions: procedure calls, iterators, local variables, . . .
Concurrency abstractions: processes, synchronization, scheduling, . . .

Abstract Behavioral Modeling Principle

1. Design a “programming language” tailored to modeling
2. Model elements+behavior of real systems as “programs” in that language

Work Flow of Behavioral Modeling

System to be modeled

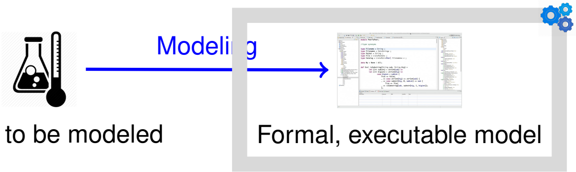
Work Flow of Behavioral Modeling

System to be modeled

Formal, executable model

Work Flow of Behavioral Modeling

Does the model behave like the modeled system?



System to be modeled

Modeling

Formal, executable model

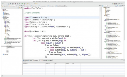
Execute model:

Simulation

Calibration

Work Flow of Behavioral Modeling

Modeling



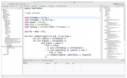
System to be modeled

Formal, executable model

Conjectured/stipulated property

Work Flow of Behavioral Modeling

Modeling



System to be modeled

Formal, executable model

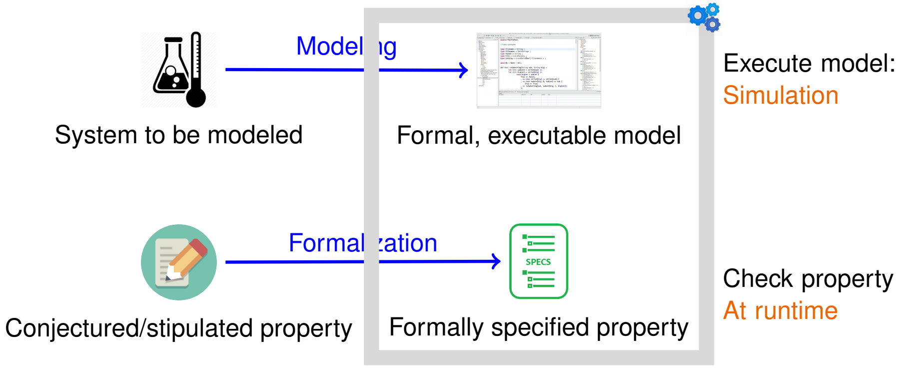
Formalization

Conjectured/stipulated property

Formally specified property

Work Flow of Behavioral Modeling

Does the model respect the conjectured property?



System to be modeled

Formal, executable model

Execute model:
Simulation

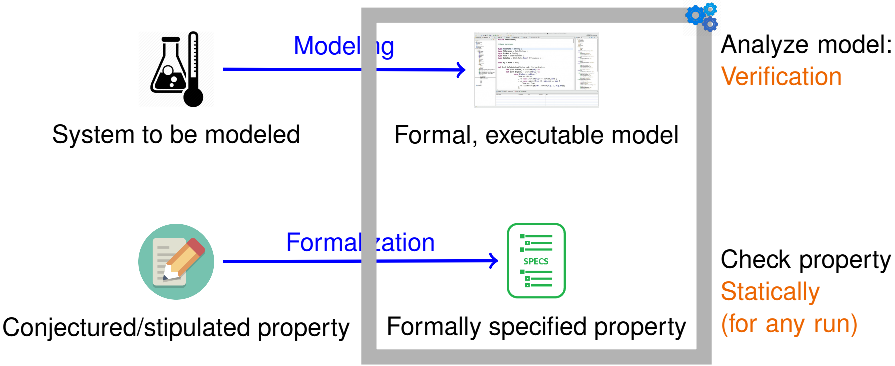
Conjectured/stipulated property

Formally specified property

Check property
At runtime

Work Flow of Behavioral Modeling

Do all possible model behaviors respect the conjectured property?



Modeling

System to be modeled

Formal, executable model

Analyze model:
Verification

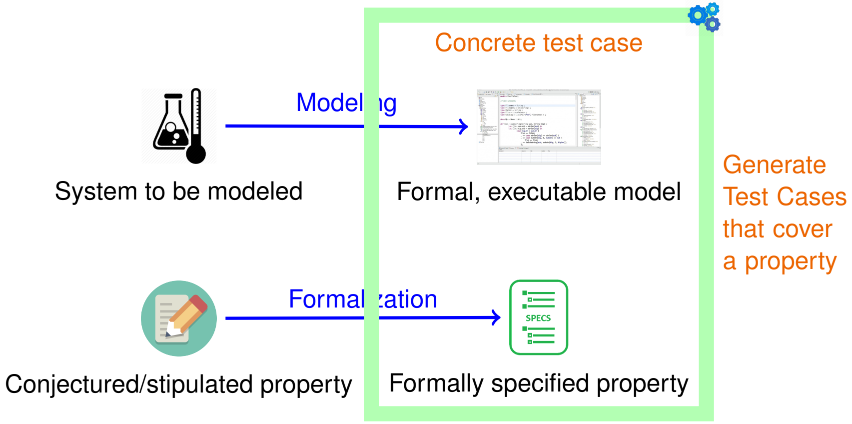
Formalization

Conjectured/stipulated property

Formally specified property

Check property
Statically
(for any run)

Work Flow of Behavioral Modeling



Concrete test case

Model ng

System to be modeled

Formal, executable model

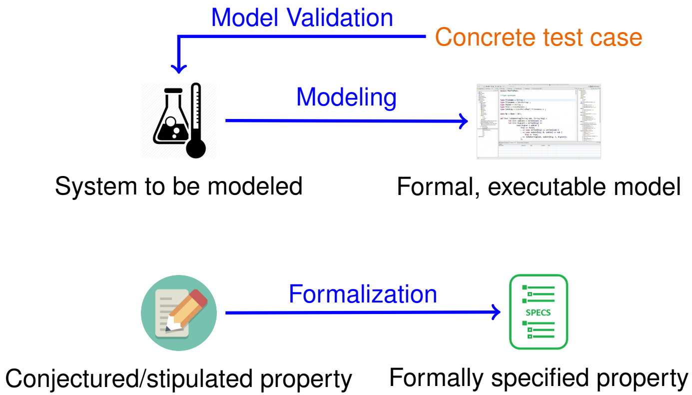
Generate
Test Cases
that cover
a property

Formal zation

Conjectured/stipulated property

Formally specified property

Work Flow of Behavioral Modeling



Model Validation

Concrete test case

Modeling

System to be modeled

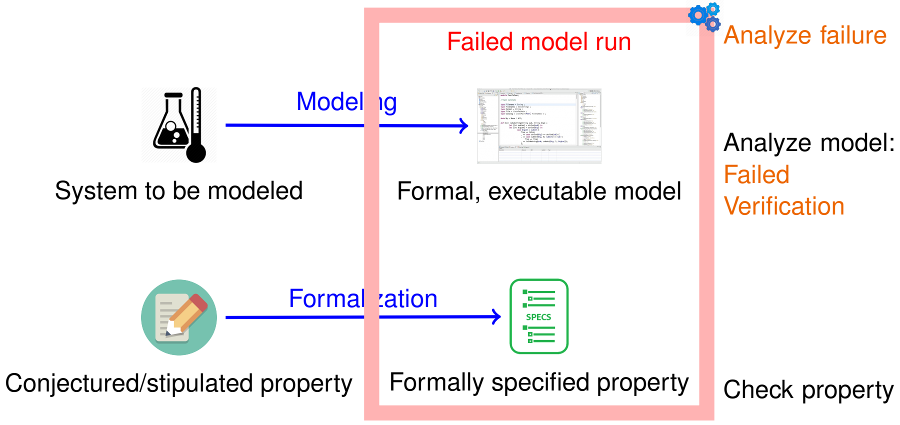
Formal, executable model

Formalization

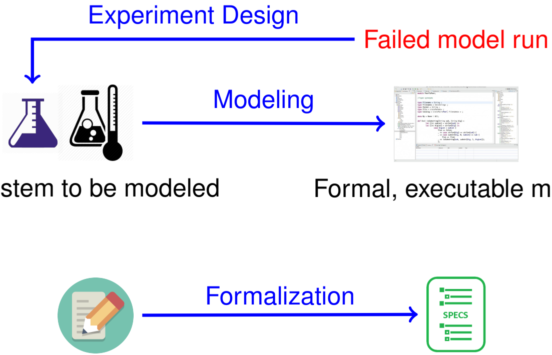
Conjectured/stipulated property

Formally specified property

Work Flow of Behavioral Modeling



Work Flow of Behavioral Modeling



Experiment Design

Failed model run

Modeling

System to be modeled

Formal, executable model

Formalization

Conjectured/stipulated property

Formally specified property

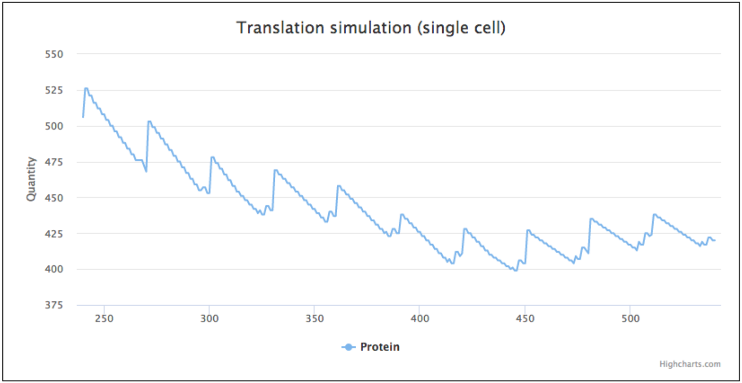
Advantages (technical)

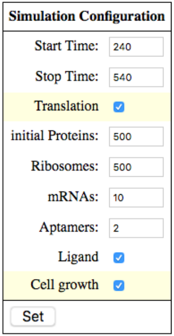
OO model easy to understand for (some) Bio domain experts

```
class Ribosome(Cell cell) implements Ribosome {
    Unit run() {
        while (active) {
            Mrna mrna = await cell!obtainMrna();
            Bool attached = await mrna!attach(this);
            if (attached) {
                this.scan(mrna);
                this.decode(mrna);
                await mrna!detach(this);
            }
        }
    }
    Unit scan(Mrna mrna) {...}
    Unit decode(Mrna mrna) {...}
    ...
}
```

Advantages (technical)

OO model easy to modify, instrument, visualize




Simulation Configuration

Start Time: 240

Stop Time: 540

Translation ☑

initial Proteins: 500

Ribosomes: 500

mRNAs: 10

Aptamers: 2

Ligand ☑

Cell growth ☑

Advantages (technical)

Parallelization is easy

- Active, concurrent objects; deployment components
- ABS compiles to Erlang, Haskell; potential to run on MP hardware

©Argonne National Laboratory

Advantages (methodological)

- Find out which in vivo experiments are likely to gain insights before actually performing them:
  - reduce # of required experiments
  - design more targeted experiments
- Model-based explanation and prediction
  - Attempt to use model for finding explanations (more detailed observations)
  - Test new conjectured causation in model first
  - Attempt to predict result of in vivo experiments
- Permits static analysis of all possible behaviors
  - gain insight into inherent properties of a model

Overall Approach

1. Model: single cell with translation mechanism
2. Calibrate: fit model to experiment
3. Validate: can we correctly predict known behavior? ⟵ we are here
4. Cash in on opportunities mentioned earlier

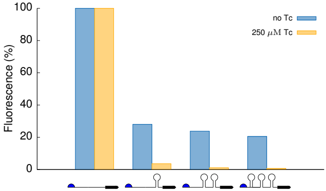
Next Concrete Steps

Use our model to better understand regulatory mechanisms for gene expression.

Some questions to be answered:

- Why is effect of multiple aptamers non-linear?
- Why does aptamer position influence regulation?
- How to fine-tune translational regulation?

Conjecture and test possible explanations in model.



Kötter et al., Nucleic Acids Research, Vol. 37, 2009.

ABS Model Characteristics

A typical yeast (*Saccharomyces cerevisiae*) cell has

- 200,000 ribosomes — we currently simulate 500
- 20,000–60,000 mRNAs — we simulate 10
- 12 Million DNA nucleotides — we currently don't go there

Model:

- Cell, ribosomes, mRNAs are active objects
- Primary molecular structure represented as list
- Decoding of mRNA by ribosome implemented with asynchronous messages

```
while (...) {
    Nucleotide n = await mrna!readNucleotide(position);
    canProceed = await mrna!canAdvance(this, position);
    position = await mrna!advance(this, position);
}
```

Modelling Biology in ABS: Model vs. Reality

Example: Ribosome–Cell interaction

- Model: Ribosomes *message* the cell for an mRNA molecule available for decoding – cell processes messages sequentially.
- Reality: Ribosomes *bump into* an mRNA in their proximity – can happen concurrently.

Example: Ribosomes decoding mRNA

- Model: Ribosomes *message* the same mRNA before they can advance – mRNA processes messages sequentially, performance bottleneck.
- Reality: the molecule itself is distributed in space – many ribosomes decode the same molecule concurrently, advance by “trying” – can happen concurrently.

Modelling Biology with ABS
Preliminary Assessment

ABS Advantages

- Complex, fine-grained models that are easy to understand/maintain/evolve
- Abstract time highly useful (but don't let your time slices become too small)
- HTTP API great for communicating with domain experts (biologists)
- ABS usability has improved tremendously over past year.

Proximity to Application Domain

- Communication: by message passing in ABS — by diverse means in nature
- Asynchronous message exchanges are not cheap
- Reality is more concurrent than we can (reasonably) model in ABS
- But models don't need to match detail level of reality to be useful.

(Possible) future directions

- Floating point numbers will facilitate data I/O.
- Increase level of concurrency of model with available means (active objects, concurrent linked lists?); use high performance hardware
- Explore more flexible concepts of concurrency, communication (multiactive objects, finer-grained notion of space (distance, movement), concurrent data structures, . . .)