



A computational modeling language for complex laboratory experiments



Adriana Compagnoni^{b,*}, Paola Giannini^{a,*}, Christopher Kelley^b

^a Computer Science Institute, DiSIT, Università del Piemonte Orientale, viale Teresa Michel 11, 15121 Alessandria, Italy

^b Department of Computer Science, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken, NJ 07030, United States

ARTICLE INFO

Article history:

Received 5 May 2015

Accepted 21 May 2015

Available online 30 May 2015

Keywords:

Computational biology

Probabilistic simulation

Process algebra

ABSTRACT

BIOSCAPE^P, a probabilistic version of BioScape, is a new modeling language for a state-dependent stochastic simulation of parallel processes in three dimensional space. Normally a modeling language describes an initial concentration of entities after which all changes are driven by the simulation of reactions. We instead design **conditional simulation commands** which depend on a **global state**. Our contribution is a new command in the form `when R run A1 ··· An`, which will cause A₁ ··· A_n entities to be added to the system when property R in the context of a global state is satisfied. Commands and reactions are further equipped with **user-defined update functions** to produce **side effects** on the global state. The global state for simulation is defined to include at least a simulation clock to enable time dependent computation.

BIOSCAPE^P matches the realistic nature of experimentation, by defining uncertainty from two sources: stochastic movement generating reactions on proximity, and probabilistic choice, where an entity has the ability to be involved in more than one reaction.

To capture the richer notion of state dependent conditional commands, which need to be evaluated periodically, we must define a system of **multi-level semantics** consisting of two layers: **World Level Semantics** and **Individual Level Semantics**. Both levels take turns to evaluate their respective domains. The World Level Semantics evaluates the aforementioned `when-run` conditional commands, while the Individual Level Semantics simulates reactions, entity movement, and maintains the simulation clock and timed entities.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Computational models have long been used to generate virtual versions of laboratory experiments where initial concentrations react over a period of time producing time-stamped simulation results. Although this mechanism captures a large class of experiments, it does not model the conditional events that often alter complex wet-lab experiments with external stimuli.

Our earlier work on BIOSCAPE [1] helped us understand the needs of virtual experimentation in a 3D environment. Experimentalists, however, rely on time and other observable conditions to tailor their experiments, and for safety reasons, drugs are often administered in appropriate doses over time. Motivated by the lack of formalisms and tools to design virtual experiments depending on a global state including the simulation clock, we define BIOSCAPE^P. BIOSCAPE^P extends the syntax and semantics of BIOSCAPE to enable clock dependent simulation, and more generally state dependent computational modeling. Fig. 1 shows an example of a state dependent experiment. A simulation command is a collection of conditional

* Corresponding authors.

E-mail addresses: adriana.compagnoni@gmail.com (A. Compagnoni), giannini@di.unipmn.it (P. Giannini), ckelly@stevens.edu (C. Kelley).

<http://dx.doi.org/10.1016/j.nancom.2015.05.001>

1878-7789/© 2015 Elsevier Ltd. All rights reserved.

```

when (clock = 0) runU Bac1 ... Bacn AmA1 ... AmAk;
when (clock=12hs ∧ nBac>1) runU1 AmA1 ... AmAk;
when (clock=24hs ∧ nBac>1) runU1 AmA1 ... AmAk;

```

Fig. 1. Antibiotics every 12 h while infection persists.

$T ::= \text{fl} \mid \text{int} \mid \text{bool} \mid \top \mid \text{chan}\{T\} \mid T_1 * \dots * T_n$	Expression Types
$V ::= x_1 \mapsto c_1, \dots, x_n \mapsto c_n$	Global State
$P, Q ::= \mathbf{0}$	Empty Process
$\mid X(\delta)$	Entity Instance
$\mid P \mid Q$	Parallel Composition
$\mid (\nu a^{rd} : \text{chan}\{T\}).P$	Restriction
$\pi ::= \text{delay}^{p, \delta, u}$	Delay
$\mid !u(\delta)^u$	Output
$\mid ?u(x)^{p, \delta, u}$	Input
$\mid \text{mov}^u$	Move
$M ::= \pi.P \mid \pi.P + M$	Choice
$N ::= M \mid (\nu a^{rd} : \text{chan}\{T\}).N$	Restricted Choice
$u ::= a \mid b \mid \dots \mid x \mid y \mid \dots$	Identifiers
$\delta ::= u \mid c \mid x \mid y \mid \text{op}(\delta) \mid \delta_1, \dots, \delta_n \mid () \mid \delta.i$	Expressions
$v ::= a \mid b \mid \dots \mid c \mid () \mid v_1, \dots, v_n$	Expression Values
$D ::= \emptyset \mid D, X(x : T) = N^{\xi, \omega, \sigma}$	Entity Definitions
$C ::= \text{when } \delta \text{ run}^u X(\delta) \dots X(\delta) \mid C; C \mid \text{skip}$	Commands
$E ::= \emptyset \mid E, a^{rd} : \text{chan}\{T\}$	Channel Declarations
$S ::= \langle D, E, C, V \rangle$	System

Fig. 2. Syntax of BIOSCAPE^P.

events (when-run). The predicate guarding the run command may refer to global variables. In this example the variable nBac corresponds to the number of bacteria in the system. At time 0, initial concentrations of bacteria (Bac) and antimicrobial agents (AmA) are added to the system. The remaining conditional events will cause the initial dose of antibiotics to be repeated every 12 h for 24 h, if there are still bacteria present in the system ($nBac > 1$). u and u_1 are user-defined state update functions that modify global variables depending on the relevant observations in each model. For example, they could keep track of the number of doses of antibiotics delivered.

This paper is a revised and improved version of [2] where the language has been enriched with

- an expression sub-language,
- a type system checking the correctness of the whole systems,
- durations of reactions modifiable at run-time and
- generalized update functions.

Moreover, the details of the probabilistic semantics have been specified in full.

The paper is structured as follows. In Section 2 we introduce the syntax of the modeling language BIOSCAPE^P with

its type system. An example of modeling with BIOSCAPE^P is presented in Section 3.1. In Section 4 we define the two level semantics of the language followed by the main result on the correctness of the execution of programs and a graphical example of the execution of a simple system. Finally, in Section 5, we present work related to our proposal and draw some conclusions.

2. Syntax of modeling language

The syntax of BIOSCAPE^P appears in Fig. 2. It includes constant types fl, int, bool, and \top as primitive types, as well as channel types, $\text{chan}\{T\}$, and tuple types, $T_1 * \dots * T_n$. We assume that the *global state*, V , is a mapping between global variables, denoted by x and y , and values of primitive types. Global variables include the simulation clock, giving the programmer access to experimental time and counters for species' populations. Among the novel aspects of BIOSCAPE^P is a collection of commands made possible in part by the new concepts of experimental time and global state in a language-based modeling platform.

The language describing the entities behavior is a π -calculus in which we have a non-deterministic and

Download English Version:

<https://daneshyari.com/en/article/10343462>

Download Persian Version:

<https://daneshyari.com/article/10343462>

[Daneshyari.com](https://daneshyari.com)