

Modeling biological systems in hybrid concurrent constraint programming

Alexander Bockmayr
Univ. Henri Poincaré, LORIA
B.P. 239, 54506 Vandœuvre, France
bockmayr@loria.fr

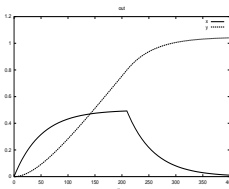
Arnaud Courtois
Univ. Henri Poincaré, LORIA
B.P. 239, 54506 Vandœuvre, France
acourtoi@loria.fr

ABSTRACT

Constraint programming is a new programming language paradigm that appeared in the late 80's and became more and more popular during the 90's [5]. In a constraint program, the user specifies a number of constraints. Each constraint defines a relation between the variables that describe the state of the system under investigation. The constraint programming tool provides constraint solving algorithms which infer new constraints from given ones and which compute solutions, i.e., valuations of the variables satisfying all the constraints. In *concurrent* constraint programming, different computation processes may run concurrently. Interaction is possible via the *constraint store*. The store contains all the constraints currently known about the system. A process may *tell* the store a new constraint, or *ask* the store whether some constraint is entailed by the information currently available, in which case further action is taken [4]. *Hybrid* concurrent constraint programming (*Hybrid cc*) is an extension of concurrent constraint programming which allows one to model and to simulate the temporal evolution of *hybrid systems*, i.e., systems that exhibit both discrete and continuous state changes [3]. Constraints in *Hybrid cc* may be both algebraic and differential equations. State changes can be specified using the combinators of concurrent constraint programming and default logic. The following example shows how to model in *Hybrid cc* an interaction between two genes.

```
x=0; y=0;
always { if (y < 0.8) x' = -0.02*x+0.01;
         if (y >= 0.8) x' = -0.02*x;
         y' = 0.01*x; }
sample(x,y);
```

Gene x activates gene y . Above a certain threshold, gene y inhibits expression of gene x . Executing this simple program yields the following time-dependant behavior:



Hybrid cc is well-suited for modeling and simulating systems in biology. Examples we have modeled include [1]:

- λ -phage response to UV irradiation
- Cell differentiation (*Xenopus laevis*)

- Stochastic behavior of protein-DNA complexation
- Transcriptional regulation of luminescence (*V.fischeri*)

The next table gives an overview of some important features of biological systems and their counterpart in *Hybrid cc*.

Biology	Hybrid cc
reaching thresholds	discrete events
time, concentration	continuous variables
kinetics	differential equations
gene interaction	concurrency
stochastic behavior	random numbers

Hybrid cc is a promising alternative to existing modeling and simulation tools in systems biology, both from the theoretical and the practical point of view. *Hybrid cc* is a declarative, compositional programming language based on the constraint programming paradigm. It allows the biologist to model biological systems in a natural and intuitive way, without having to be an expert in computer programming. For the computer scientist, it offers all the advantages of a full programming language. Current efforts to build a software platform for integrating different approaches in systems biology raise a number of fundamental questions about the nature of computation in living systems. The conceptual foundation of *Hybrid cc* and the corresponding semantics [2] may help to identify the key computational concepts required in systems biology.

REFERENCES

- [1] A. Courtois. Modélisation de systèmes biologiques en programmation par contraintes. Rapport de DEA (in French), Univ. Henri Poincaré, LORIA, July 2001.
- [2] V. Gupta, R. Jagadeesan, and V. Saraswat. Computing with continuous change. *Science of computer programming*, 30(1-2):3–49, 1998.
- [3] V. Gupta, R. Jagadeesan, V. Saraswat, and D. G. Bobrow. Programming in hybrid constraint languages. In *Hybrid Systems II*, Springer, LNCS 999, 1995.
- [4] V. A. Saraswat. *Concurrent constraint programming*. ACM Doctoral Dissertation Awards. MIT Press, 1993.
- [5] P. van Hentenryck and V. Saraswat. Strategic directions in constraint programming. *ACM Computing Surveys*, 28(4):701 – 726, 1996.