Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Using Existing Digital Tools for Efficient Metabolic Pathway Simulations

Luca Macchiarulo

Abstract— **The complexity of full-cell metabolic and regulatory pathways currently hinders high accuracy biological simulations with traditional Ordinary Differential Equation (ODE) solver-based methodologies. In this paper event-driven simulators used to tackle problems of similar complexity in Very Large Scale Integration (VLSI) digital designs are employed to trade-off complexity for accuracy. An event-driven model for typical enzymatically catalyzed reactions is proposed and compared with standard ODE solutions on single reactions and a biologically relevant portion of a metabolic pathway. The system shows good stability and controlled error propagation, and allows reduction of computational effort in systems where few concentrations are actively changing at any given time, as well as integration of other models such as on/off gene activation, stochastic and discrete behavior.**

I. INTRODUCTION AND PREVIOUS WORK

Recent research is bringing a wealth of details about metabolic and regulatory pathways and potentially allows simulation of entire cells with high degree of accuracy, but this comes at the cost of dazzling complexity. A commentary ([3]) to the paper [2] dealing with simulation issues reports that "Large genetic networks are currently out of reach for predictive simulations." In the same paper, the authors suggest that, in order to tackle the problem, some complexity reduction technique has to be found, and that even drastic simplifications (such as those of Kauffman networks [4]) might preserve much of the biologically relevant information. Most of the work in simulation, however, has been done under the common paradigm of ODE systems, with appreciable results (as, for example, the complete modelling of red blood cells [5], [6]), suggesting that "traditional" modelling at the concentration level cannot be dismissed entirely but should rather be integrated and/or improved. In this paper, prompted by an analogous problem at the simulation level for VLSI digital integrated systems, where nowadays tens of millions of transistors - each potentially exhibiting a mathematical complexity higher than that of a single catalyzed reaction - can be accurately predicted with high degree of accuracy, we suggest that an event-driven approach that discretizes the relevant variables *in value* and uses time rather than concentration as the critical variable could simplify the problem of integrating the two approaches and at the same time allow the use of existing tools optimized for fast simulations of digital objects. This paper differs from existing approaches (see, for example, [7] and [8]) in trying to model similar ODE-like behavior at a similar level of accuracy of the ODE solvers (used in most broad-spectrum biological simulators such as Gepasi [9], Jarnac [10], Virtual Cell [11], or SimTool [12]), while reducing the amount of computation needed by employing an event-based strategy. This will potentially allow the best of both "discrete-oriented" and "continuous-oriented" approaches to work together in complementary ways, thus reducing the simulation time for complex biological systems.

II. COMPLEX DIGITAL AND BIOLOGICAL SYSTEM SIMULATION

How are digital simulation tools able to deal with the complexity of system simulation of millions of elements? As in many other engineering feats, this is accomplished by a *divide et impera* approach, where the low level details (in this case precise values of currents, voltages, stored charges, and the like) are abstracted out to leave a structure of a graph of interconnected elements with highly simplified behavior, described by a few variables, and easily modelled as a reactive system (an object that responds to external stimulations in an unambiguous and uniform way, and remains idle otherwise). The main abstraction at play is that over value granularity: instead of representing each and every possible voltage level of the system (for example between 1.3 and 0V), the simulation allows only the two extreme values, represented by symbolic '1' and '0'. The time behavior of the system is still described accurately because the *delay* of the single elements can be modelled as a function of other known variables in the abstracted system.

In the biological domain, the scholars are not in general able to achieve such an extreme simplification in their descriptive variables (even though certain studies suggest that a massive semplification might be possible in studying the qualitative properties of the system [2], [3], [4]) but the level of accuracy needed (and compatible with the experimental data) is often way smaller than the one offered by ODE solvers. Therefore, it is interesting to see if an analogous reduction in simulation complexity is possible by limiting the concentration of the reagents and products to a finite number of discrete quantities. This done, it is easy to see that a single reaction can be modeled as a reactive system like a digital gate: its output (number of substrates converted into products or vice-versa) will change of a given quantity after a time which is dependent on the current distribution of substrates, products, and other partecipating species (such as activators or inhibitors). The following sections will show how such a suggestion can be developed into a full-fledged model for a biochemical reaction usable with existing simulation tools

L. Macchiarulo is with Faculty of Electrical Engineering, University of Hawaii at Manoa, 2540 Dole Str., Honolulu, HI 96822 lucam@hawaii.edu

1-4244-0033-3/06/$20.00 ©2006 IEEE.

(VHDL simulators will be considered with this study, see [1]).

The basic distinction between continuous-time and event-driven models is that, while continuos-time models choose (possibly adaptively) a time step for simulation and update the system at each of the time steps, an event driven tool modifies (any) of the variables only in response to some previously issued event, thus saving the "idle time" wasted by continuous-time models to advance the simulation time when nothing really happens.

The key to such reduction is the concept of "event": a change, with a time stamp attached, that triggers the elements of the system (logic gates, complex blocks or, in the suggested implementation, chemical reactions) that in turn cause new events to be scheduled **in future time**, in reaction to some previous event (typically, reaching a certain threshold) on their inputs.

III. Event-driven model of a substrate

The preliminaries of the previous section allow us to understand how a discrete event simulator could be used to represent accurately a chemical reaction with a Michaelis-Menten type of rate, or in general any catalyzed reaction kinetics. Considering the chemical species involved in the reaction first, we need to satisfy two different conditions:
1. Substrates and products are either introduced in or extracted from the reaction pool by external means (representing the external inputs or outputs of the system) or by reactions in which they appear as products or substrates.
2. The total concentration of a single species is determined by the factors above, plus the initial concentration.

It is extremely easy to model such behavior in an event-driven simulation environment, provided we have a correct description of reactions as entities that either subtract or add to the concentration of species at given times: Any chemical species to be analyzed can be represented by an object (**entity** in the VHDL jargon [1]) whose functionality is that of adding (with correct sign) the contribution of all reactions in which the species takes part, plus an external contribution that can both represent the initial concentration (at the beginning of the simulation) and successive external intervention to the system (additions/subtractions) The left part of figure 1 shows a graphical representation of such a system. The result of any addition/subtraction to the total concentration has immediate effect (no delay), because the delays are already taken care of by the reaction models (see below). In short, the output values of the components implementing the various chemical species are the estimates of their concentrations at any given simulation time.

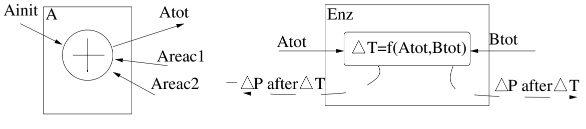

Fig. 1. Basic modelling components: Substates and Reactions

IV. Event-Driven model of a reaction

The core of the model is of course in the description of the reaction behavior. Every reaction is represented by a separate component, which is triggered (**sensitive** in the terminology of VHDL) by an event in its relevant variables (substrates or substrates and products in case of a reversible reaction), and proceeds in the following way (see figure 1 on the right):
1. The reaction rate is computed on the basis of the current information (on substrates, inhibitors, enzyme concentration, pH, temperature, ...).
2. the computation is repeated using a concentration of the substrates and products at the end of the supposed transition (that is, when ΔP substrates are transformed into products and vice-versa). The two rates are averaged.
3. On the basis of the averaged rate, the time needed to bring about the transformation of the ΔP substrates into products (or vice-versa) is computed using $\Delta T = \frac{Rate_{avg}}{\Delta P}$.
4. the contribution of the reaction to the substrates and products is set to be ΔP (with the appropriate sign) after ΔT seconds.

The averaging step is used to avoid excessive under- or over-estimation of the reaction rate inevitably introduced by the time discretization (and is analogous - but not equivalent - to a trapezoidal method for solving the related ODE). Each activation of a given enzyme will therefore bring a computational cost of 2 rate evaluations, that here and in the following we will take as the unit to assess the computational complexity of the scheme.

So, contrary to a traditional ODE solver, where the *time step* is fixed, and the variation in concentration is computed, in this method the *concentration step* (ΔP) is fixed (in value but not in sign) and the time to obtain such a variation is computed. This of course brings about different stability and stiffness properties but it has the advantages that:
- It is easily implemented in a usual event-driven simulation tool.
- It can be tailored to the precision in the output values that is needed.
- Each reaction can be modeled at the level of detail desired.
- Stoichometric relations between substrates and products are guaranteed.

The next subsections will illustrate a few examples of such models with progressively higher complexity of the reaction, and its comparisons with a classical accurate ODE-solver solution.

A. Irreversible Michaelis-Menten type reaction

This kind of reaction is represented by the simple well-known reaction rate formula

$$v = \frac{K_{cat}[Enz][S]}{[S] + K_m}$$

The reaction rate is dependent on the enzyme and substrate concentration only. An example of simulation of such modelis shown in figure 2, together with a 4th order Runge-Kutta solution of the same system, using ΔP values of 0.1 mMol and 0.01mMol respectively (the staircase graphs represent the event-driven simulations, the continuous lines represent the ODE solver results, that uses a small fixed simulation step to ensure accuracy of results). It is clear from the figure that, even using a large ΔP, the reaction is modelled quite accurately, and the computation effort is kept to a minimum (see table I for a comparison of computational efforts). Furthermore, the length of the time steps are implicitly computed by the method, and not laboriously calculated via error estimates as in more traditional Runge-Kutta-Fahlberg methods.

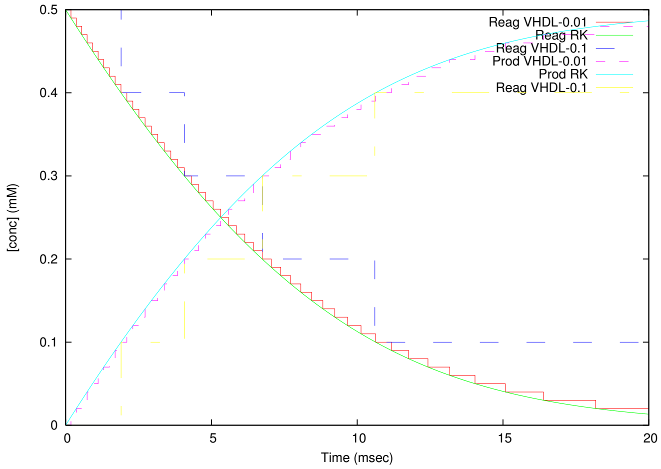

Fig. 2. Irreversible Michaelis-Menten type reaction

B. Reversible reaction

Reversible reactions can be readily modeled with a similar approach, whose only differences are the dependence of the time rate on the product as well as the substrate, and the corresponding triggering of the reaction from the two sides as well. The rate rule used is the classical:

$$v = [Enz]\frac{(K_{cat}/K_{mS})[S] - (K_{catR}/K_{mP})[P]}{1 + [S]/K_{mS} + [P]/K_{mP}}$$

where K_{catR} represents the turnover number of the enzyme for the reverse reaction, while the two Michaelis constants model the affinity of the substrate and the product with the enzyme. It is interesting to note that here also the simulations (figure not shown) agree with an error which is smaller than the ΔP value.

C. Inhibition, Multiple Substrates, allosteric and further effects

Modelling of inhibition and multiple substrates and products was also integrated,

also in this case very good match with the expected behavior is observed (results not shown).

Further complications on the enzyme kinetics, including allosteric effects, temperature-controlled activities, and pH-sensitive reactions can be also integrated into the model, as long as they can be expressed with an instantaneous rate law. Therefore, all systems that can be modelled as ODEs are amenable to solution with the mechanism proposed. In addition, effects that cannot be easily integrated into ODEs are more easily dealt with an event-driven strategy, such as the low concentration effect (the finite and non-continuous nature of molecular interaction that might deviate from ODEs assumption for low numbers of molecules), the stochastic nature of the interactions, and an on/off activation of regulatory mechanisms.

V. Example: Start of Glycolysis

In this section, a small but meaningful portion of a metabolic pathway is used to illustrate the simulation of a system of interacting reactions, rather than a single one (albeit complicated). We consider the first three reactions of the glycolitic pathway (Glucose phosphorilation, glucose1-P to fructose-1-P isomerization and fructose-1-P phosphrilation), involving three enzymes (Hexokinase, Isomerase and Phosphofructokinase) and 6 chemical species (Glucose (GLU), Glucose-1-Phosphate (GLU1P), Fructose-1-Posphate (FRU1P), Fructose-1,6-Disphopshate (FRU16DP), ADP and ATP) that presents a series of interesting features: Two reactions are multi-substrate and multi-products, common substrates and products (ADP, ATP) introduce competitive effects, and one reaction is bi-directional.

The first simulation results (pseudo-Gly in table I) uses reaction coefficients of the same order of magnitude, that makes the system non-stiff (the system is rather stiff with the naturally occurring enzymatic parameters). In this case, we can use a non-stiff solver (as the 4th Runge-Kutta here used for comparison) without reducing too much the computation step. Using a fairly small ΔP guarantees close matching with the ODE solver. The final equilibrium concentrations are accurate with an error of ΔP, and correspond to the expected values. Increasing the ΔP value gracefully degrades the accuracy of the solution, as it will be shown in the next section, allowing consistent savings in simulation complexity. On the other hand, increasing the simulation step for the ODE solver after a certain value produces instability and completely erroneous results (the moderate stiffness of the system allows this value to be relatively large).

For the second simulation, the enzyme coefficients were defined according to [8], and [5], while the initial concentrations were left as before (and, still according to [5], far from typical intercellular values. In particular the GLU values are abnormally high). This simulation aimed at showing performance to unusual initial conditions. The results for three different values of ΔP follow qualitatively well the real behavior (figure 3), and this is more relevant if we notice that the transformation of GLU into FRU16 passes through two substrates always present in very low concentrations (GLU1 and FRU1), and are therefore approximated pretty roughly at high values of ΔP. Even with this modelling problem, both the final concentrations and the transition of FRU16 is modelled accurately. We attribute this fact mainly to the strict enforcement of the mass action implied by the VHDL reaction models. The Runge-Kutta solver has to employ a small time step to avoid stiffness issues in this case, while the two results of the following section show the good graceful degradation properties of the event driven simulation.

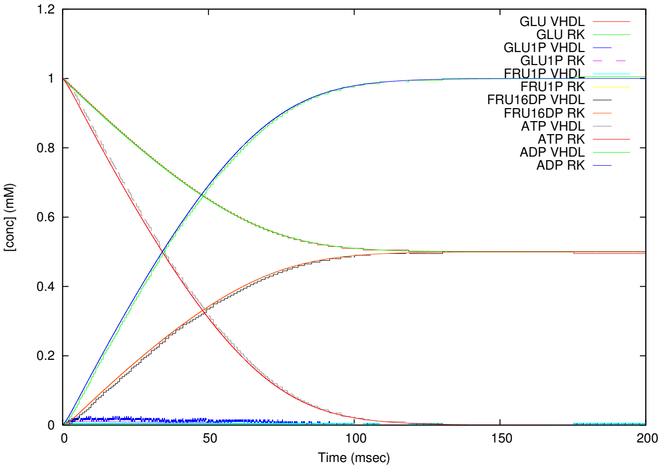

Fig. 3. Comparison of VHDL and RK simulations of First Glycolysis reactions -high GLU

The last simulation has been performed with lower (more physiologically accurate) GLU levels. Even in this case the curves match fairly well (compared with the ΔP value).

VI. Computational Analysis and Discussion

Considering the various examples of reactions described in the paper, we analyze the accuracy/computational cost trade-offs of the VHDL-based method and compare it with the ODE solver. Table I summarizes the results of the experiments: in the first column we report the simulated system The second column reports the value of the ΔP parameter for the VHDL-based simulations. The third column shows the number of times the ODE solver has to evaluate any rate equation for an "optimal" choice of its time step. This choice is made, for the single reaction cases, such that the accuracy is at least as good as the one provided by the VHDL-based method (In these cases a great increase of time step is possible, the single equation being non-stiff). In the case of the Glycolysis simulations the number reported is the number of evaluations of the solver that guarantees a stable solution (the system is stiff in these cases). The fourth column reports the number of evaluations of the rate equations in the VHDL-based simulation, and its number is an indication of potential savings in execution time when compared to ODE solvers (the value reported in the previous column). Note that each ODE iteration, in this case, implies 4 evaluations of rates, while only two are needed in the VHDL simulation. The last column represents the maximum error among all the concentration values compared with a ODE solver solution with small timestep. A few points are worth considering:

- Computational savings vary from case to case, but in general are more substantial with larger values of ΔP and with longer "idle" periods, due to the reactive nature of the computation.

- Stiffness of the system don't pose limits to the reduction in cost (at least for the intervals investigated).

- The final accuracy appears to be connected to the ΔP value: In all cases the errors are typically within a few times ΔP, thus giving the experimenter an easier "knob" to control the trade-off accuracy/simulation time.

TABLE I

Number of rate evaluation and accuracy for various reaction simulations.

Reaction	ΔP	eval(RK)	eval(VHDL)	err
Uni Irreversible	0.01	600	102	0.003
Uni Irreversible	0.05	120	22	0.006
Uni Irreversible	0.1	60	14	0.01
Uni Reversible	0.01	2400	52	0.0014
Uni Reversible	0.05	400	12	0.007
Uni Reversible	0.1	200	18	0.05
Pseudo Gly	0.005	1200	1614	0.016
Pseudo Gly	0.01	1200	834	0.03
Pseudo Gly	0.05	1200	194	0.18
Gly ([Glu]=1mM)	0.005	345000	32196	0.019
Gly ([Glu]=1mM)	0.01	345000	25068	0.031
Gly ([Glu]=1mM)	0.05	345000	14484	0.11
Gly ([Glu]=0.01mM)	0.0001	31000	4836	0.0008
Gly ([Glu]=0.01mM)	0.005	31000	562	0.009

VII. CONCLUSIONS AND FUTURE WORK

In this paper a methodology allowing the application of standard VLSI fast event-based tools to metabolic simulation is proposed, validated on real-life biological cases, and compared with more traditional approaches. Future work will concentrate on mathematical elucidation of the properties of the models, stiffness and stability analysis, comparison with state of the art stiff-oriented ODE solvers and simulation of entire metabolic systems.

References

[1] P. Ashenden, *The Designer's Guide to VHDL*,

[2] Avi Ma'ayan,1 Sherry L. Jenkins, *et al*, "Formation of Regulatory Patterns During Signal Propagation in a Mammalian Cellular Network," *Science* Vol. 309. no. 5737, pp. 1078-1083, 12 August 2005.

[3] Stefan Bornhold, "Less Is More in Modeling Large Genetic Networks," *Science*, Vol. 310, no. 5747, pp. 449-451 , 21 October 2005.

[4] S.A. Kauffman, "Metabolic stability and epigenesis in randomly constructed nets," *Journal of Theoretical Biology*, vol. 22, pp. 437467, 1969.

[5] A. Joshi,B. O. Palsson "Metabolic dynamics in the human red cell. Part IV–Data prediction and some model computations." *J. Theor Biol.* 1990 Jan 9;142(1):69-85.

[6] N. Jamshidi, B. O. Palsson, "Systems biology of the human red blood cell," *Blood Cells, Molecules, and Diseases* 36 (2006) pp. 239247.

[7] V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman, "Petri net representations of metabolic pathways." Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB '93), pages 328–336, 1993.

[8] X. Q. Xia and M. J. Wise, "DiMSim: A Discrete-Event Simulator of Metabolic Networks," *J. Chem. Inf. Comput. Sci.* (2003), 43, pp.1011-1019.

[9] P. Mendes, "GEPASI: a software package for modeling the dynamics, steady states and control of biochemical and other systems." *Comput. Appl. Biosci.* 9, pp. 563-571 (1993).

[10] H. .M. Sauro, "Jarnac, a System for Interactive Metabolic Analysis," in Hofmeyr *et al* (eds.), *Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics*, pp.221-228.

[11] J. Schaff, *et al* "A general computational framework for modeling cellular structure and function," *Biophys. J.* 73, 1135-1146.

[12] T. Aho, "Simulation tool for genetic regulatory networks," M.Sc. thesis, Department of Information Technology, Tampere University of Technology, Tampere, Finland.