

DEVS Modeling and Simulation of the Cellular Metabolism by Mitochondria

Roxana Djafarzadeh
School of Information
Technology and Engineering
University of Ottawa.
800 King Edward Avenue
Ottawa, ON, K1N 6N5, Canada
rdjafar@site.uottawa.ca

Gabriel Wainer
Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa, ON, K1S 5B6, Canada
gwainer@sce.carleton.ca

Tofy Mussivand
University of Ottawa Heart
Institute. Medical Devices Centre.
40 Ruskin Street
Ottawa, ON, K1Y 4W7, Canada
tmussivand@ottawaheart.ca

Keywords: DEVS, CD++, Mitochondria, Glycolysis, Krebs cycle.

Abstract: We present a simulation model for the metabolic pathways in the cells (namely, the Glycolysis and Krebs cycle), using the DEVS formalism and the CD++ tool. The hierarchical nature of DEVS makes it ideal for describing naturally hierarchical systems as the Cell, while its discrete-event approach improves performance due to the asynchronous nature of the events involved. DEVS timed-based nature can adequately represent the timing of the chemical reactions. We were able to create a precise and easy to use simulation environment for biological models and the results presented here show the potential of our approach.

INTRODUCTION

Simulation is becoming increasingly important in the analysis and design of complex systems such as those involving biological processes. The complexity of these systems makes computer simulation an adequate tool to study them under particular experimental conditions.

We are interested in the detailed analysis of the behavior of the mitochondrion, which fulfills different important roles in cellular metabolism [1]. A number of genetic diseases (including diabetes, deafness, heart, Alzheimer's, Parkinson, etc.) are associated with mutations in mitochondrial DNA; other fields of interest include the study of apoptosis and forensic science [2].

We want to provide an environment to understand and control dynamics of mitochondrial metabolism through computer simulation in a whole organelle scale. For this purpose, we built a mitochondrial model that included two biological pathways. The model puts emphasis on cellular metabolism and energy production aspects. Our approach is based on the use of the DEVS formalism [3]. DEVS provides a framework for the construction of hierarchical models in a modular fashion. The hierarchical nature of DEVS makes it ideal for describing naturally hierarchical systems as the Cell. Likewise, its discrete-event nature improves the execution performance of a model like this, due to the asynchronous nature of the events occurring in the cell. DEVS also uses explicit timing information;

hence, we can adequately represent timing of the reactions occurring in the cell. CD++ is a toolkit [4] that implements DEVS theories. We will show how we created a simulation environment for models of the cellular metabolism, based on DEVS and CD++, providing an experimentation environment with remote execution that can be used by researchers in this field worldwide.

BACKGROUND

Mitochondria are small double-membrane organelles found in the cytoplasm of eukaryotic cells. Mitochondria are responsible for converting nutrients into the energy yielding molecule, Adenosine TriPhosphate (ATP), to fuel the cell's activities [5]. Mitochondria can be divided into four components: outer membrane, inter-membrane space, inner membrane, and the matrix (see Figure 1).

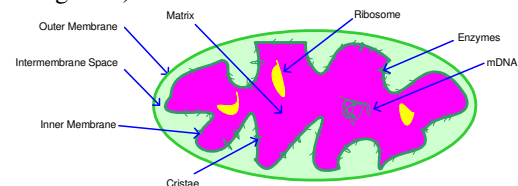


Figure 1. Scheme of the Mitochondrion.

The smooth outer membrane holds numerous transport proteins, which shuttle materials in and out. The components between the outer and inner membranes have important roles in electron transport and oxidative phosphorylation. The inner membrane has many folds called cristae, which are the sites of ATP synthesis. The cristae enclose a liquid-filled region known as the matrix, which contains mitochondrial genome. These gene products include various enzymes involved in the process of aerobic respiration, proteins necessary for the import of proteins, and proteins and nucleic acids required for the mitochondrial genome [6].

The chief function of the mitochondria is to create energy for cellular activity by the process of aerobic respiration. In this process, glucose is broken down in the cell's cytoplasm via the **glycolysis** process, to form pyruvic acid. In a series of reactions, part of which is called Krebs cycle, the pyruvic acid reacts with water to produce carbon dioxide and hydrogen. Energy is

released as the electrons flow from the coenzymes down the electron transport chain to the oxygen atoms. The enzyme ATPase, which is embedded in the inner membrane, adds a phosphate group to Adenosine Di-Phosphate (ADP) in the matrix to form ATP. Aerobic respiration is an ongoing process and mitochondria can produce hundreds of thousands of ATP molecules/minute. ATP is transported to the cytoplasm, where it is used for virtually all energy-requiring reactions. As ATP is used, it is converted into ADP, which is returned by the cell to the mitochondrion and is used to build more ATP [7].

We want to be able to carry out simulation-based studies of this metabolism cycle. Our models have been created using DEVS (Discrete Event System Specification), a systems theoretical approach that allows the definition of hierarchical modular models [3]. A real system modeled using DEVS can be described as a set of atomic or coupled submodels. The atomic model is the lowest level and defines dynamics, while the coupled are structural models composed of one or more atomic and/or coupled models. An atomic DEVS model is defined as:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

Each atomic model can be seen as having an interface consisting of *input* (X) and *output* (Y) ports to communicate with other models. Every *state* (S) in the model is associated with a *time advance* (ta) function, which determines the duration of the state. Once the time assigned to the state is consumed, an internal transition is triggered. At that moment, the model execution results are spread through the model's output ports by activating an *output function* (λ). Then, an *internal transition function* (δ_{int}) is fired, producing a local state change. Input external events are collected in the input ports. An external transition function (δ_{ext}) specifies how to react to those inputs.

A coupled DEVS model is defined as:

$$CM = \langle X, Y, D, \{M_i\}, \{I_i\}, \{Z_{ij}\}, select \rangle$$

Coupled models are defined as a set of basic components (atomic or coupled), which are interconnected through the models' interfaces. The models' coupling defines how to convert the outputs of a model into inputs for the others, and how to handle inputs/outputs from/to external models

CD++ [4] is a modeling and simulation tool that was defined using the specifications of the DEVS formalism. CD++ is built as a class hierarchy of models related with simulation processing entities. DEVS Atomic models can be programmed and incorporated onto the Model basic class hierarchy using C++. Once an atomic model is defined, it can be combined with others into a multicomponent model using a specification language specially defined with this purpose. CD++ also provides client/server services, which provide remote access to a high performance simulation

server. Using these facilities, the users can develop and test their models in local workstations, and submit them in a remote CD++ high performance server. Then, they can receive, visualize and analyze the results on a local computer, improving model definition and execution.

A DEVS MODEL OF GLYCOLYSIS

Glycolysis, also called Embden-Meyerhof pathway, is a sequence of reactions used by virtually all cells to metabolize glucose [6]. The role of glycolysis is to produce energy. Glycolysis generates about 15% of the energy produced by aerobic respiration, and it is the basis for the metabolism in virtually all the living creatures. It consists of a sequence of ten reactions that converts a glucose molecule into two pyruvate molecules with the production of NADH and ATP. Specific enzymes control each of the different reactions, as showed in Figure 2.

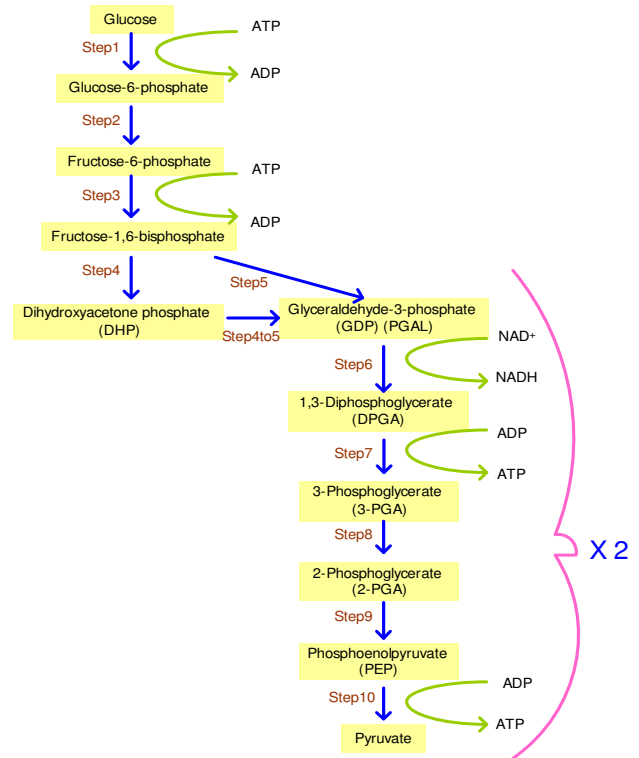


Figure 2. Glycolysis Pathway.

The glycolysis pathway was defined as a DEVS coupled model and it was implemented using CD++. We present, as an example, the definition of *Step1*, in which glucose is phosphorylated by ATP to form Glucose 6-phosphate and ADP [8]. The atomic model is defined by:

$$Step1 = \langle S, X, Y, \delta_{int}, \delta_{ext}, ta, \lambda \rangle$$

$$S = \{atpc, glucosec, ifhex, counter, phase, sigma\}$$

$$X = \{glucose, ATPi, hexokinase\}$$

$Y = \{ \text{glucose_6_phosphate}, \text{ADP}, \text{H} \}$
 δ_{int} , δ_{ext} , ta and λ are presented in the following tables, using CD++ implementation.

The external transition function presented in Table 1 is invoked every time *glucose*, *ATPi*, or *hexokinase* are received, and the reactions previously discussed of the step to the enzyme (*hexokinase*) or each of the inputs (*glucose* and *ATPi*), as previously described in Figure 2. Whenever a substance arrives, the value is added to a counter representing the number of molecules available in the mitochondria (*glucosec*, *atpc*; in the case of the enzyme *hexokinase*, only presence is needed, so, whenever we have the enzyme, a value is set to *true*). Then, the condition for the other substances is checked. For example, if the input is *glucose*, the counter is incremented by the value of the molecules entered, and then *atpc* and *ifhex* are checked. The internal transition function schedules an internal event after a preparation time describing the timing for the transfer. If there is glucose, ATPi and hexokinase in the system, then the reaction will happen.

Table 1 – External transition function (δ_{ext})

```
Model &Step1::externalFunction
( const ExternalMessage &msg ) {
  if( msg.port() == glucose ) {
    glucosec = glucosec + msg.value() ;
    if ( ( atpc > 0 ) && ( ifhex == true ) )
      holdIn( active, Prep_Gluc );
  }
  else if( msg.port() == ATPi ) {
    atpc = atpc + msg.value() ;
    if ( ( glucosec > 0 ) && ( ifhex == true ) )
      holdIn( active, Prep_ATPi );
  }
  else if ( msg.port() == hexokinase ) {
    ifhex = true ;
    if ( ( glucosec > 0 ) && ( atpc > 0 ) )
      holdIn( active, Prep_Hexo );
  }
}
```

When the time interval expires, the *output function* is invoked and the first value in Step1 is sent through the corresponding output port. In this case, the output function is activated when all the conditions of the external function have been satisfied, that is, all three input events are in, and the reaction can happen. As a result, *ADP*, *glycose_6_phosphate*, and *H* will be sent out through the corresponding output ports.

Table 2 – Output Function (λ) for Step1

```
Model &Step1::outputFunction(InternalMessage
&msg) {
  if ( counter != 0 ) {
    sendOutput( msg.time(), ADP, counter );
    sendOutput( msg.time(),
      glucose_6_phosphate, counter );
    sendOutput( msg.time(), H, counter );
  }
}
```

After calling the output function, the internal transition function is invoked. The internal transition function will produce an internal state change according to the substances available in the mitochondria. The function updates the number of substances available according to the reaction, and it then *passivates*.

Table 3 – Internal Transition Function (δ_{int})

```
Model &Step1::internalFunction(const
InternalMessage &) {
  counter = 0;
  if ( ( atpc >= 1 ) && ( glucosec >= 1 )
    && ( ifhex == true ) ) {
    if ( atpc > glucosec ) {
      atpc = atpc - glucosec;
      counter=glucosec; glucosec=0;
    }
    else if ( atpc < glucosec ) {
      glucosec = glucosec-atpc;
      counter = atpc; atpc = 0;
    }
    else if ( atpc == glucosec ) {
      counter = atpc;
      atpc = glucosec = 0;
    }
  }
  passivate();
}
```

The remaining steps explained in Figure 2 (steps two to ten) were developed using a similar approach: the behavior of each component was carefully specified with an analysis of inputs and outputs for each step; then, each step was defined as a DEVS model following the specification. Finally, each model was implemented in CD++, and tested separately [9].

Following the description in Figure 2, we built a DEVS coupled model including all the steps previously defined as Atomic models, as seen in Table 4. First, the components of the coupled model are defined. Then, the input and output ports are included. Finally, the links show the translation functions.

Table 4 – Glycolysis Coupled Model

```
[top]
components : step1@Step1 step2@Step2
step3@Step3 step4@Step4 step4to5@Step4to5
step5@Step5 step6@Step6 step7@Step7
step8@Step8 step9@Step9 step10@Step10
out : H ADP NADH H2O pyruvate ATPo
in : glucose ATPi hexokinase phosphoglucosomerase
PFK isomerase aldolase G3PD NAD P
PGK PGM enolase pyruvate_kinase
Link : glucose glucose@step1
Link : ATPi ATPi@step1
Link : hexokinase hexokinase@step1
...
Link : aldolase aldolase@step4
Link : isomerase isomerase@step4to5
...
Link : ATPo@step7 ATPo
Link : H2O@step9 H2O
Link : pyruvate@step10 pyruvate
Link : ATPo@step10 ATPo
```

When we execute this model in CD++, we can study the model's behavior by analyzing the model's outputs. One simulation scenario we created validating the Glycolysis model is presented in the following Table.

Table 5 – Inputs/Outputs for Glycolysis Model

INPUTS	OUTPUTS
10:00 glucose 2	50:000 h 2
18:00 ATPi 3	72:000 nadh 2
50:00 hexokinase 1	72:000 h 2
51:00 phosphGlucoiSom 1	72:000 atpo 2
52:00 PFK 2	72:000 h2o 2
53:00 isomerase 1	72:000 atpo 2
55:00 aldolase 1	72:000 pyruvate 2
62:00 G3PD 1	
63:00 PGK 1	
64:00 PGM 1	
65:00 enolase 1	
67:00 pyruvKinase 1	
70:00 NAD 3	
72:00 P 2	

These simulation results accurately describe the reactions occurred during glycolysis [7], following the results of Figure 2. Table 5 and 6 show the input/output trajectories for the model and *Step1* respectively.

As we can see in Table 6, by time 30:00, we have all the three inputs required to produce a reaction. At time 30:00, two *glucoses* and six *ATPi* enter the system generating two *ADP*, 2 *glucose_6_phosphate*, and two *H* molecules.

Figure 3 shows the execution results for *Step1* using CD++ Modeler. CD++ allows the execution of the same model using a local application or a remote server. Once the simulation is done, the user can analyze the simulation results using different visualization tools. In this way, specialists located remotely are able

to run different experiments, and to analyze them locally using the visual tools here presented.

Table 6 – Inputs and Outputs for Step1

INPUTS	OUTPUTS
15:00 hexokinase 1	30:000 adp 2
30:00 glucose 2	30:000 glucose_6_phosph 2
30:00 ATPi 6	30:000 h 2
40:00 glucose 4	40:000 adp 4
40:00 ATPi 1	40:000 glucose_6_phosph 4
55:00 glucose 1	40:000 h 4
55:00 ATPi 1	55:000 adp 1
65:00 glucose 1	55:000 glucose_6_phosph 1
	55:000 h 1
	65:000 adp 1
	65:000 glucose_6_phosph 1
	65:000 h 1

In Figure 3, we see that at time 40:00, four *glucose* molecules enter the system generating four more outputs of each of the *ADP*, *glucose_6_phosphate*, and *H* molecules.

In [10], we presented CD++/Maya, an application to generate 3D visual representation of DEVS models. Maya [11] is a software application for 3D digital animation and visual effects. It provides a suite of tools for 3D world creation, including animation, rendering, dynamics, etc. In CD++/Maya, the log files generated from CD++ can be translated into 3D representations.

We used CD++/Maya to create 3D graphics and animations for the glycolysis model. Figure 4 shows a snapshot of the animation. In this case, we show that, while the enzymes are entering the reactions, their names are being added as text to the top of the screen for clarity.

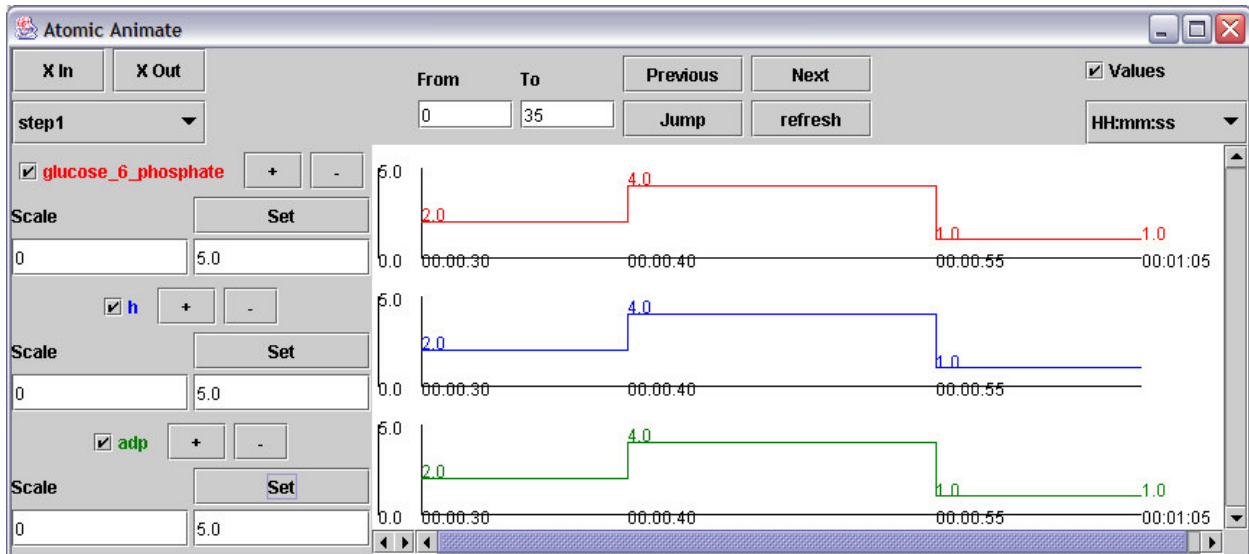


Figure 3. Atomic Animation of Step1 of Glycolysis

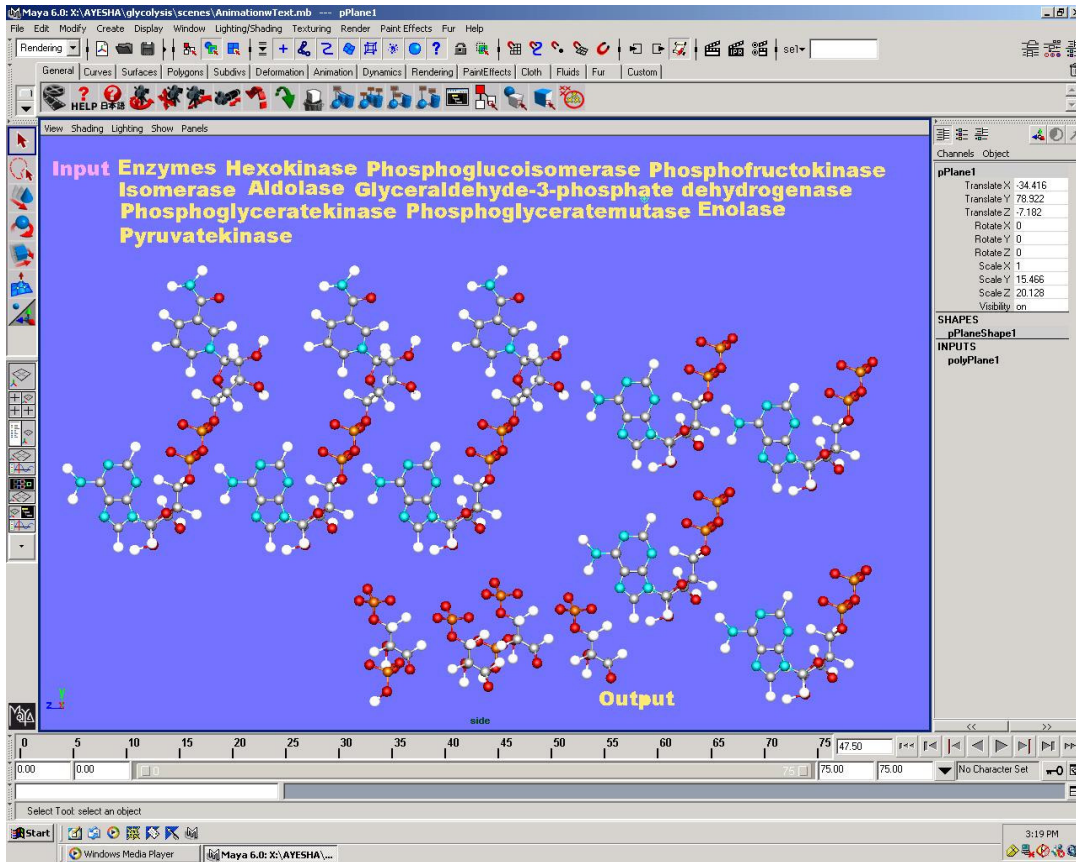


Figure 4. Visualization of the Glycolysis execution in CD++/Maya: Step 6.

KREBS' CYCLE MODELING

The Krebs cycle, also called the Tri-Carboxylic Acid (TCA) cycle and the Citric Acid Cycle (CAC), oxidizes pyruvate formed during the glycolytic pathway into CO_2 and H_2O . This cycle is a series of chemical reactions of central importance in all living cells that utilize oxygen. The citric acid cycle takes place within the mitochondria in eukaryotes, and within the cytoplasm in prokaryotes. For each turn of the cycle, 12 ATP molecules are produced, one directly from the cycle and 11 from the reoxidation of the three NADH and one FADH_2 molecules produced by the cycle by oxidative phosphorylation [12]. Glucose is converted by glycolysis into pyruvate. Pyruvate enters the mitochondria, linking glycolysis to Krebs's cycle. This step (StepA), as seen in Figure 5, is also called the bridging step. Pyruvate dehydrogenase – a complex of three enzymes and five coenzymes – oxidizes pyruvate using NAD^+ to form acetyl CoA, NADH and CO_2 .

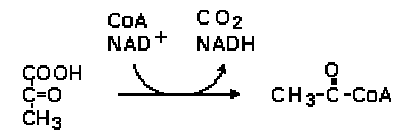


Figure 5. StepA of Krebs' Cycle [8]

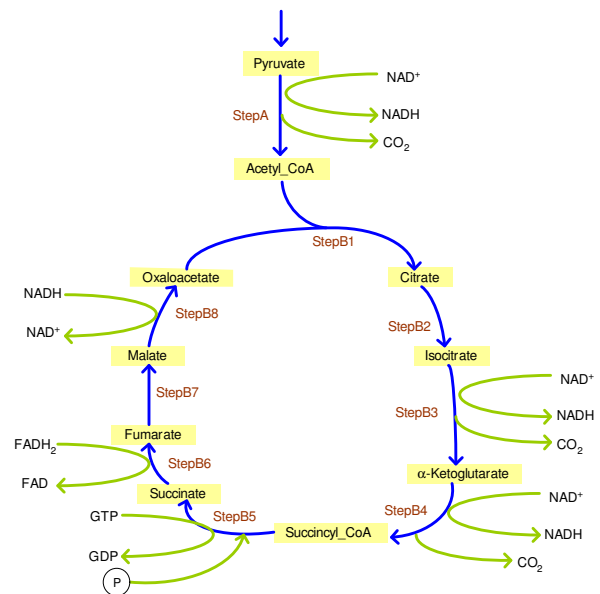


Figure 6. Krebs' Cycle Reactions

We defined a model of the Krebs' cycle (depicted in Figure 6), using identical principles than the ones used in the previous Section. In the *StepA* of Figure 6, Pyruvate is degraded and combined with coenzyme A to form acetyl coenzyme A. NADH and CO_2

are released during this process, as seen in Figure 5 (*StepA* is the link between glycolysis and Krebs's cycle). The DEVS atomic model for *stepA* is defined as:

$$StepA = \langle S, X, Y, \delta_{int}, \delta_{ext}, ta, \lambda \rangle$$

$S = \{pyruvatec, pyruvateDehydrogenase, hscCoaic, nadc, counter, phase, sigma\}$

$X = \{pyruvate, pyruvateDehydrogenase, HSCoAi, NAD\}$

$Y = \{acetyl_CoA, NADH, H, CO_2\}$

$\delta_{int}, \delta_{ext}, ta$ and λ are presented in detail in [9].

The external transition function will be invoked every time *pyruvate*, *pyruvateDehydrogenase*, *HSCoAi* or *NAD* arrive. In that case, we add the number of molecules received for each substance to a counter kept for that input (*pyruvatec*, *pyruvateDehydrogenase*, *hscCoaic* and *nadc*). Once the value of the counter of the input event is incremented, the condition for the other requirements is checked. For example, if the input event is *pyruvate*, the pyruvate counter value is incremented by the value of the pyruvates entered, and then it is checked against other input counters: *hscCoaic*, *nadc* and *pyruvateDehydrogenase*. If there is enough *NAD*, *HSCoAi*, and *pyruvat-*

eDehydrogenase in the system, the reaction will occur. In the case of *pyruvateDehydrogenase*, since it is an enzyme and its presence is only needed for the reaction, its value is set to true.

When the preparation time interval expires, the output function is invoked and the *Acetyl_CoA*, *NADH*, *CO2*, and *H* have to be sent out through the output ports. The internal transition function updates the counter of molecules available in the mitochondria, and passivates the model waiting for the next arrival of a substance.

All the remaining steps were developed using a similar approach. To do this, the behavior of each component was carefully specified with an analysis of inputs and outputs. Each step was defined as a DEVS model following the specification. Afterwards, each model was implemented in CD++, connecting all the submodels previously defined, as in Figure 7.

CD++ permits defining coupled models like this one using a graphical user interface. Figure 8 presents the visual input creation of this coupled model, which is exported to CD++ notation to be executed.

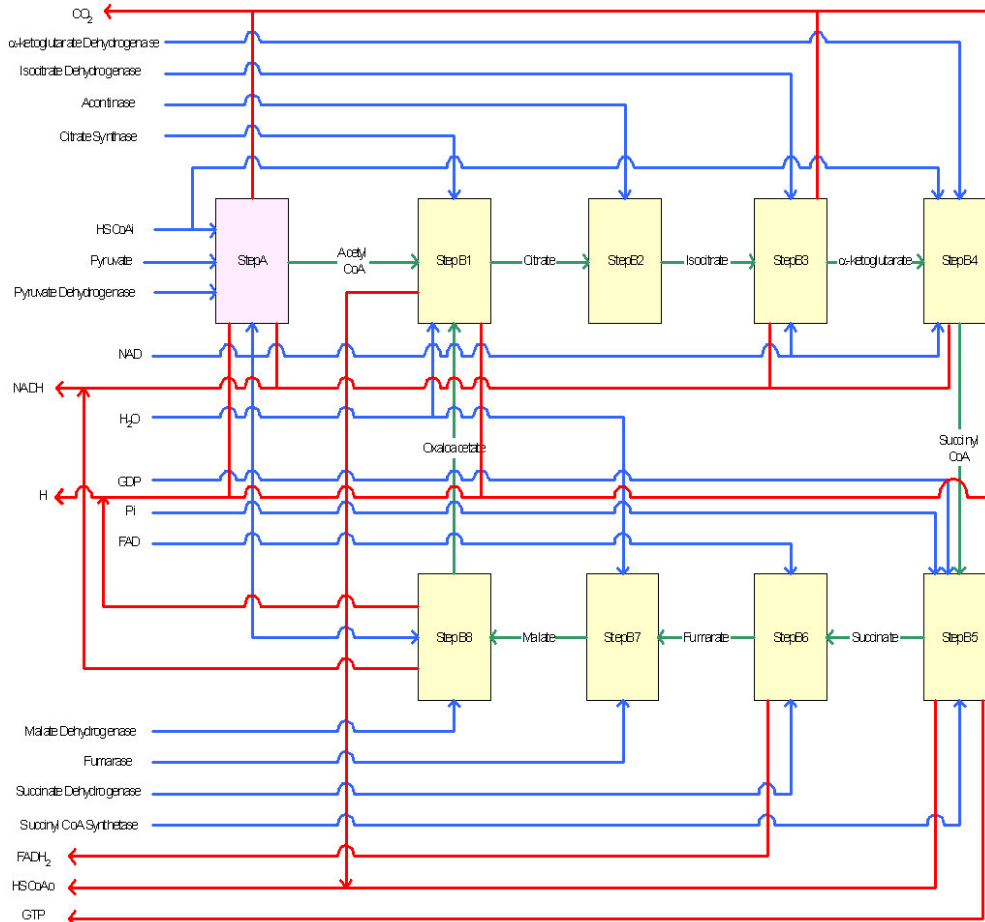


Figure 7. Coupled Model of Krebs' Cycle

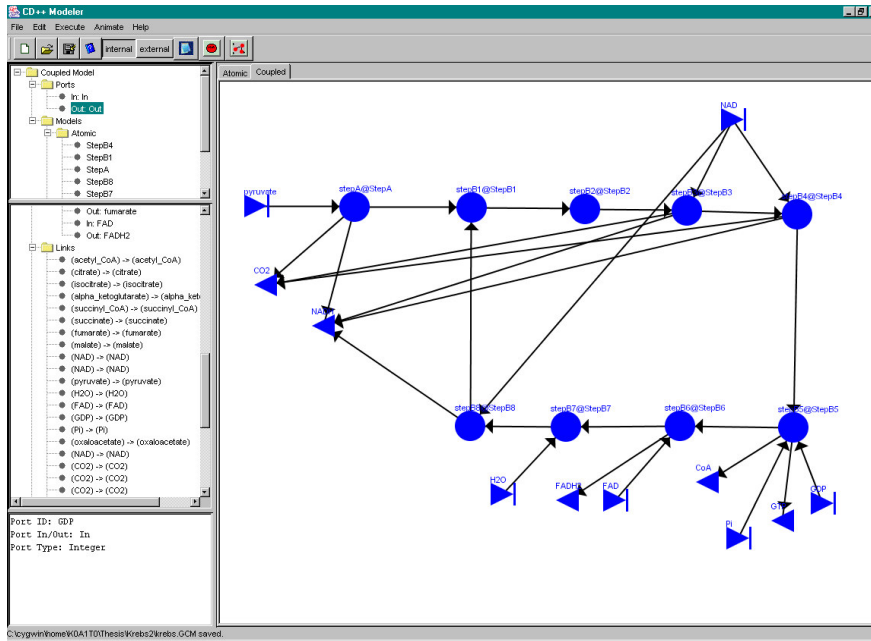


Figure 8. Defining the Krebs coupled model using CD++ Modeler.

Table 7 – Inputs and Outputs for Krebs Model

INPUTS	OUTPUTS
00:10:00 pyruvate 1	01:11:000 nadh 1
00:25:00 NAD 2	01:11:000 co2 1
00:40:00 H2O 2	01:11:000 h 1
00:57:00 HSCoAi 2	02:01:000 hscoao 1
01:11:00 pyruDehydrog 1	02:01:000 h 1
01:14:00 citrateSynthase 1	02:30:000 nadh 1
02:01:00 oxaloacetate 1	02:30:000 co2 1
02:14:00 acontinase 1	02:30:000 nadh 1
02:30:00 isocitrDehydrog 1	02:30:000 co2 1
02:30:00 alKetDehydrog 1	02:30:000 h 1
02:50:00 succCoASynth 1	03:11:000 hscoao 1
03:10:00 GDP 1	03:11:000 gtp 1
03:11:00 Pi 1	03:28:000 fadh2 1
03:15:00 FAD 1	03:45:000 nadh 1
...	...

Table 7 shows a list of the input and output events for Krebs' model. These simulation results accurately describe the reactions occurred in the mitochondria, and mimics the results described in Figure 6. This figure demonstrates the execution results for this set of inputs.

Figure 9 shows snapshots of some of the reactions in the Krebs Cycle Animation done in CD++/Maya. Figure 9 a) shows the beginning of the reaction, in which one pyruvate and four NAD⁺ appear. Figure 9 b) shows the formation of Acetyl CoA, and the production of Carbon dioxide and NADH as byproducts.

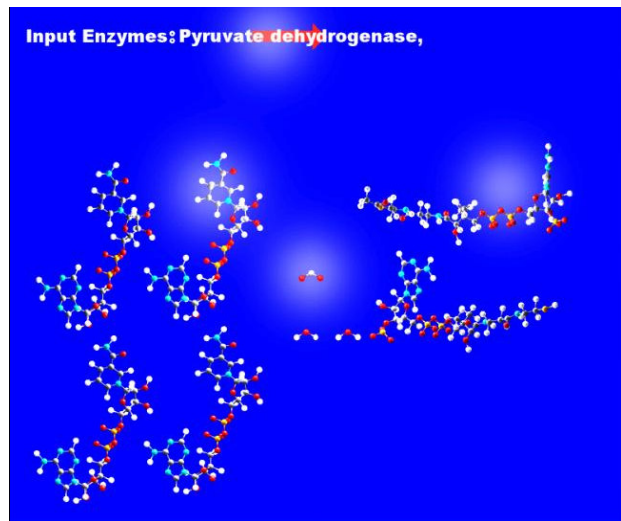
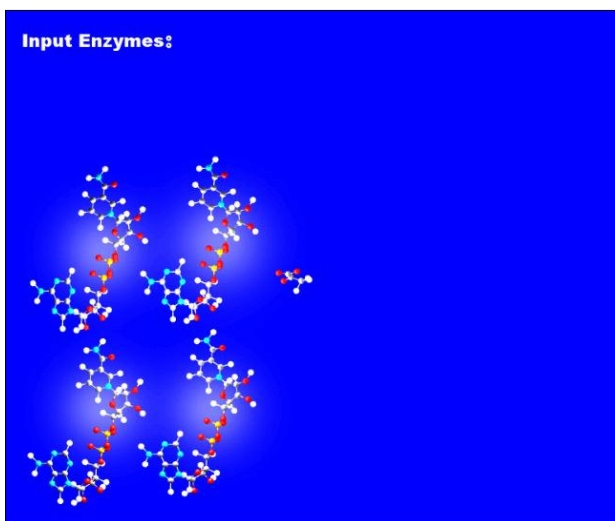


Figure 9. (a) The Krebs cycle begins (b) Acetyl CoA is formed.

CONCLUSION

During the last several decades, computer simulation has become an integral part in both the basic and applied fields of biological research. The hierarchical and discrete-event capabilities of DEVS makes it an ideal method for modeling biological events. As illustrated herein, CD++ can be used to model and simulate biological pathways using a systematic method with models that consist of sets of lower-level interactions.

We have shown that these more complex simulations can be built and validated incrementally using these simple sub-components. The use of DEVS provides a system that allows even non-computer science specialists the ability to develop complex simulation models for specific research problems. The approach also enables reuse of simulation components and allows seamless integration of these components into more complex simulation models. This capability would allow for sharing and inter-operability of model components thus aiding the development of a wide variety of model implementations for the broader medical research field.

REFERENCES

- [1] KRAUSS S. "Mitochondria: Structure and Role in Respiration". In: *Nature Encyclopedia of Life Sciences*. Nature Publishing Group. June 2001. <http://www.els.net/> [doi:10.1038/npg.els.0001380]
- [2] POULTON J. and BINDOFF L. "Mitochondrial Respiratory Chain Disorders". In: *Nature Encyclopedia of Life Sciences*. Nature Publishing Group. May 2000. <http://www.els.net/> [doi:10.1038/npg.els.0002287]
- [3] ZEIGLER, B.; KIM, T.; PRAEHOFER, H. "Theory of Modeling and Simulation". Academic Press. 2000.
- [4] WAINER G., "CD++: a toolkit to define discrete-event models". *Software, Practice and Experience*. Vol. 32, No.3. pp. 1261-1306. November 2002.
- [5] ALBERTS, B.; BRAY, D.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WATSON, J. "Molecular Biology of the cell", Third Edition. Garland Publishing. 1994.
- [6] THORSNESS P. and HANEKAMP T. "Mitochondria: Origin". In: *Nature Encyclopedia of Life Sciences*. Nature Publishing Group. January 2000. <http://www.els.net/> [doi:10.1038/npg.els.0001381]
- [7] CURTIS HELENA and BARNES N. SUE, "Biology", Fifth Edition. W. H. Freeman. 1989.
- [8] MABER, J. "Step by step Glycolysis" Dept Biochemistry & Molecular Biology, the University of Leeds, <http://www.jonmaber.demon.co.uk/glysteps>. [Last checked on Oct. 25, 2004].
- [9] DJAFARZADEH, R. "Modeling and Simulation of Cellular and Energy Production by Mitochondria", M.A.Sc. Thesis. Ottawa-Carleton Institute for Electrical and Computer Engineering. 2005.
- [10] KHAN A., WAINER G., "Advanced visualization of DEVS and Cell-DEVS models in Maya". Proceedings of the SISO Spring Interoperability Workshop, San Diego, CA. 2005.
- [11] ALIAS Corp. "Maya 6 Features in Detail," http://www.alias.com/eng/products-services/maya/file/maya6_features_in_detail.pdf [2004 Oct. 25],
- [12] HAMES B. D. and HOOPER N. M., "Instant Notes in Biochemistry", 2nd Edition. Springer. 2000.

ROXANA DJAFARZADEH received a B. A. Sc. in Computer Engineering (1998) and M. A. Sc. in Electrical and Computer Engineering (2004) both from the University of Ottawa, Ottawa, Canada.

GABRIEL WAINER received the M.Sc. (1993) and Ph.D. degrees (1998, with highest honors) of the Universidad de Buenos Aires, Argentina, and Université d'Aix-Marseille III, France. He is Assistant Professor in the Dept. of Systems and Computer Engineering, Carleton University (Ottawa, ON, Canada). He was Assistant Professor at the Computer Sciences Dept. of the Universidad de Buenos Aires, and a visiting research scholar at the University of Arizona and LSIS, CNRS, France. He is author of a book on real-time systems and another on Discrete-Event simulation and more than 100 research articles. He is Associate Editor of the Transactions of the SCS, and the International Journal of Simulation and Process Modeling (Inter-science). He is Associate Director of the Ottawa Center of The McLeod Institute of Simulation Sciences. He has been awarded Carleton University's Research Achievement Award (2005-2006).

TOFY MUSSIVAND, FRSC, is Professor of Surgery and Engineering and Chair of the Medical Devices Centre of the University of Ottawa. He received his undergraduate education and training in engineering and management. He received his doctorates in Medical Engineering and Medical Sciences at the University of Akron and North Eastern Ohio University College of Medicine. Thereafter, Dr. Mussivand joined the internationally acclaimed Cleveland Clinic Hospital and Research Foundation. Currently, he is the Director of the Cardiovascular Devices Division of the University of Ottawa Heart Institute. Dr. Mussivand holds academic appointments as Professor of Surgery, Faculty of Medicine, and Professor of Engineering, School of Information Technology and Engineering, at the University of Ottawa as well as Professor in the Department of Mechanical and Aerospace Engineering, Faculty of Engineering, at Carleton University. In addition, he serves as the Chair of Medical Devices and Biomedical Engineering, which was recently established at the University of Ottawa. He has published over 250 papers, books and technical articles and is an editorial board member on several international medical and engineering journals.