A TOOL FOR TEACHING MATHEMATICAL MODELING TO INFORMATION SYSTEMS STUDENTS

Reggie Davidrajuh, University of Stavanger, Norway, reggie.davidrajuh@uis.no
Istvan Molnar, Bloomsburg University of Pennsylvania, USA, imolnar@bloomu.edu

ABSTRACT

This paper talks about teaching mathematical modeling to information systems students. Mathematical modeling and simulation should be an important part of the IS curriculum because only by mathematical modeling can a comprehensive view of the relevant processes and mechanisms involved in the information systems be obtained. This paper also proposes the use of a tool, called general-purpose Petri net simulator (GPenSIM) for modeling information systems. This paper presents the basics of GPenSIM and emphasis is given to its favorable characteristics for teaching mathematical modeling of information systems, such as 1) being discrete event-based, 2) using simple mathematics, 3) being programming language neutral, and platform independent.

Keywords: Modeling Information Systems, Teaching Tool, GPenSIM, XML

INTRODUCTION

The term *information system* refers to a system, whether automated or manual, that comprises people, machines, and/or methods organized to collect, process, transmit, and disseminate data that represent user information [10]. Thus, an information system is not just about computers, programs, and programming, but also about many other issues in the environment, such as customer satisfaction (people), fail-safety (machines), and timing (methods).

Securing customer satisfaction, assuring fail-safety, and minimizing time taken for processing, are mathematical problems; collecting, processing, transmitting, and dissemination of large amounts of data requires mathematical and statistical modeling as information is stored all over the world in databases, on Internet sites, etc., in addition to the locally collected and produced data. The contemporary information systems extract important information from extensive databases using data mining, pattern recognition, classification methods and other related techniques. The organization and coordination of these databases is an important and challenging task. Thus, a study about information system should not be confined to information only; the ultimate goal should also be about understanding the processes that produce the data. Mathematical modeling provides a comprehensive view of the relevant processes.

The recent efforts defining the "simulationist" profession and determine the program and curriculum components resulted in an increased effort to define the content of the curriculum and the applied software tools at all basic application fields, including information systems. The authors' opinion is that a general theoretical framework and related platform-independent software tool can significantly contribute to curriculum internationalization and application of international educational standards.

This paper is structured as follows: The next section (Section 2) summarizes the main factors behind the search for a teaching tool, Section 3 describes the key functions of GPenSIM. Section 4 shows how GPenSIM can be used for modeling information systems. Section 5 discusses implications of using GPenSIM as a teaching tool to IS students.

LOOKING FOR A TEACHING TOOL

Results of mathematical models are calculated numerically using digital computers. In order to speed up the creation of computational models, diverse software tools are used. Since all the information systems are discrete event based, only the modeling tools that are applicable for discrete event systems are interesting to information systems students. There are also many types of tools available for modeling and simulation of discrete event systems: visual simulation tools (e.g., Extend, LabView), bond graphs, and modeling languages (e.g.. Visual SLAM, Simscript) are some of the tools used for this purpose. The tool that is useful for teaching modeling methodology for IS students should possess the following criteria:

1. *Modeling discrete event systems*: Obviously, because the domain of application is Information Systems, the tool should be applicable for modeling of information system.

2. *Simple mathematics*: Students following information systems program, may or may not possess mathematical skills, like the ability to handle differential equations, optimization methods (non-linear programming), etc. Thus, the tool should be based on simple mathematics; ideally, the tool should not demand anything other than basic algebra.

3. *Allow complex mathematics as option*: Though the tool should not demand advanced mathematical skills, it should provide extensions to do complex mathematical analysis, for example with stochastic mathematics. As any real life system is stochastic, the tool should provide facilities to model stochastic systems.

4. *Programming language neutral and platform independent*: A huge diversity of programming languages and operating systems is a barrier to extensive and cross-platform use of the modeling software. Thus, the tool should be based on a language and platform neutral standard, meaning it should accept inputs and create outputs in a standard interchange format, such as XML.

Brief Overview of the Choices

Automata, Sateflow, and Petri nets, are the three tools that satisfy at least the first three criteria above. Though automata have a strong footing in computer science, they are not suitable for information systems students as they lack structure—the ability to modularize a system (decompose a system into modules) [2]. Stateflow is a commercial software that runs in MATLAB environment [8]. Stateflow is very similar to Petri net; converting a Petri net model of an information system into a Stateflow model and vice versa is easy. However, learning Stateflow, with its syntactic, semantic, and graphical details, is much more difficult than learning Petri net. In addition, Stateflow also demands some knowledge of MATLAB and Simulink.

Petri Nets Tools

Petri net is being widely accepted by the research community for modeling and simulation of event-driven systems, mainly due to its rigorous modeling techniques [2, 3, 6]. There is a number of Petri net tools available for free academic use; PNWorld [9] provides a list of tools. These tools are sophisticated tools flexible enough to model complex and large systems. However, these tools allow programming via graphical user interface (GUI), which can be very slow and differing from program to program.

GPenSIM [4] is a non-graphic program written in MATLAB. It is designed to accept XML based instructions, thus allowing the designer to concentrate on the modeling aspects, rather than on the graphical details. Though this paper presents the basic overall elements of GPenSIM, emphasis is given to its usage as a tool for teaching mathematical modeling to information systems students.

GPenSIM

A concise introduction to Petri net: The simple Petri net shown in Figure 1 is a model for business logic computation. The computation takes two database records and a business rule, and produces a business decision. In a Petri net, sources (like business rules and database records) and outputs (like business decisions) are called places, drawn as circles (e.g., Place-1). Computations (or events) are called transitions, drawn as vertical short bars (e.g., Transition-1). An arc connects a place to a transition, or a transition to a place, representing a path for a discrete part (e.g., data in packets) to flow. A place usually holds a number of parts, like database records. The number of parts inside a place is indicated by the markings black spots within a place.

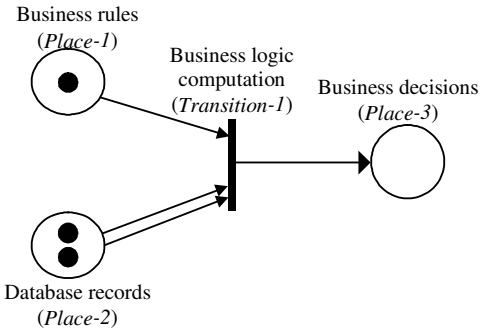

Figure 1. Petri net Model for Business Logic Computation

Any newer universally accepted tool should provide an industry-standard and vendor-neutral programming interface in order to satisfy the criteria of platform-independency. Hence, GPenSIM is supplemented with the capability to preprocess input in a standard interchange format, called Petri net markup language (PNML). PNML is based on extensible markup language (XML).

Petri Net Markup Language

The Petri Net Markup Language (PNML) is an ISO standard (ISO 15909-2) that is widely accepted as XML-based inter-change format for Petri nets [1, 5, 6, 11]. PNML is partitioned into different parts such as meta model, type definition interface, feature definition interface, etc. These details are not presented here; interested readers are referred to Billington et al [1].

Elements of PNML

A PNML document describes one or more Petri net. Each Petri net consists of objects, each of which has a unique identifier. The main objects are places, transitions and arcs. Table 1 shows the PNML elements and their attributes.

As indicated in Table 1, graphical information is enclosed with the tag <graphics>. Since GPenSIM is non-graphic, GPenSIM will ignore graphic information in PNML document.

Table 1. PNML Elements and Their Attributes

Entity	PNML Element	Attributes
Petri net Document	<pnml>	
Petri net	<net>	id: ID type: anyURI
Place	<place>	id: ID
Transition	<transition>	id: ID
Arc	<arc>	id: ID source: IDRef (Node) target: IDRef (Node)
Tool Info	<toolspecific>	tool: string version: string
Graphics	<graphics>	

USING GPenSIM

Figure 2 shows the scenario that deals with an information system for detecting business opportunities. In this scenario, the information system is fed with market data; for example, a share broker sends prices from a stock market. The information system will process the input data by first feeding it to the strategic module. If the strategic module finds out that the input data adheres to the strategic goals of the enterprise, then the broker will be informed immediately so that he can secure the business opportunity. In addition, the data will be fed into the tactical modules too, just to make sure that the input data adheres to the tactical goals also.

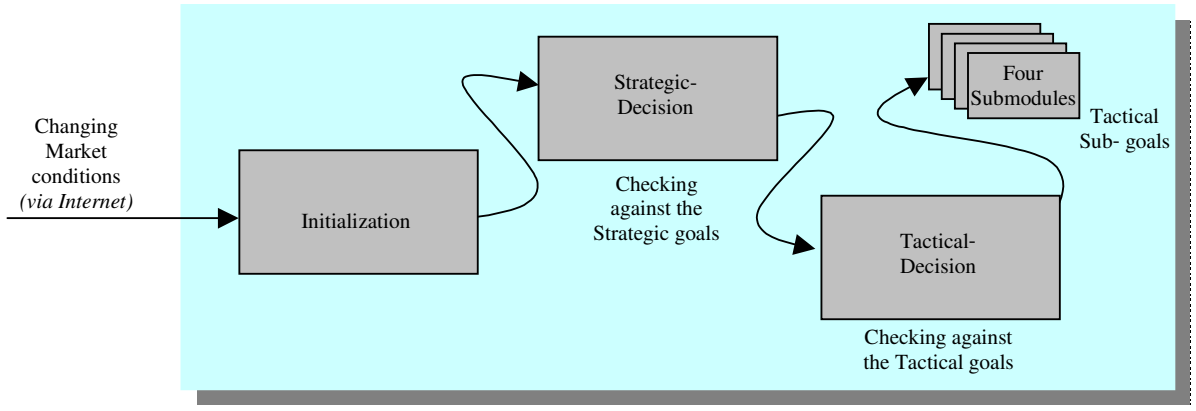

Figure 2. The Information System

The results of the analysis at the tactical module will be sent to the broker too. If the analysis at the tactical modules were affirmative, then the broker may pay a higher price, otherwise less. Thus, when a broker (or client, in client-server paradigm) sends a share price (request) to the information system (server), he gets two responses: the first is from the strategic module, which informs the broker whether or not to go for the opportunity. The second response is from the tactical module, which informs the broker how well the opportunity accommodates to the tactical goals, so that he can pay a higher price or a lower price for the opportunity. Details of the strategic and tactical modules are not shown here; the interested reader is referred to Ma and Davidrajuh [7].

The Modeling Approach

Mathematical modeling is to calculate the timing of the two consecutive responses from the information system to the broker. How long the broker has to wait before he receives the first response that may suggest him to go for the business opportunity, and what is the time gap before he receives the second response that will inform him how attractive the opportunity is. To calculate the timing, first the information system should be mathematically modeled, and then simulations should be run on the model. More precisely:

Step-1. *Mathematical modeling*: The information system that is shown in Figure 2 should be converted into a Petri net model. The Petri net model is shown in Figure 3. It is relatively easy to convert a discrete event based system like the information system shown in figure 1 into a Petri net model: all the processing nodes are replaced by transitions with an input and an output place. Interested readers are referred to the standard textbooks on Petri nets.

Step-2. *Creating a PNML document*: The Petri net model shown in Figure 3 is graphical and is drawn on a piece of paper. The information shown in the figure should be converted into a set of instructions. This is done using PNML. The resulting program is a PNML document. The PNML document for the Petri net is given in Figure 4.

Step-3. *Simulations with GPenSIM*: The PNML document is fed to GPenSIM, which will run the simulation and return the results. Simulation results are shown in Figure 5. Figure 5 indicates that the client (share broker) gets the first response (strategic decision) in 10.2 seconds and the second response (tactical decision) in 10.3 seconds.

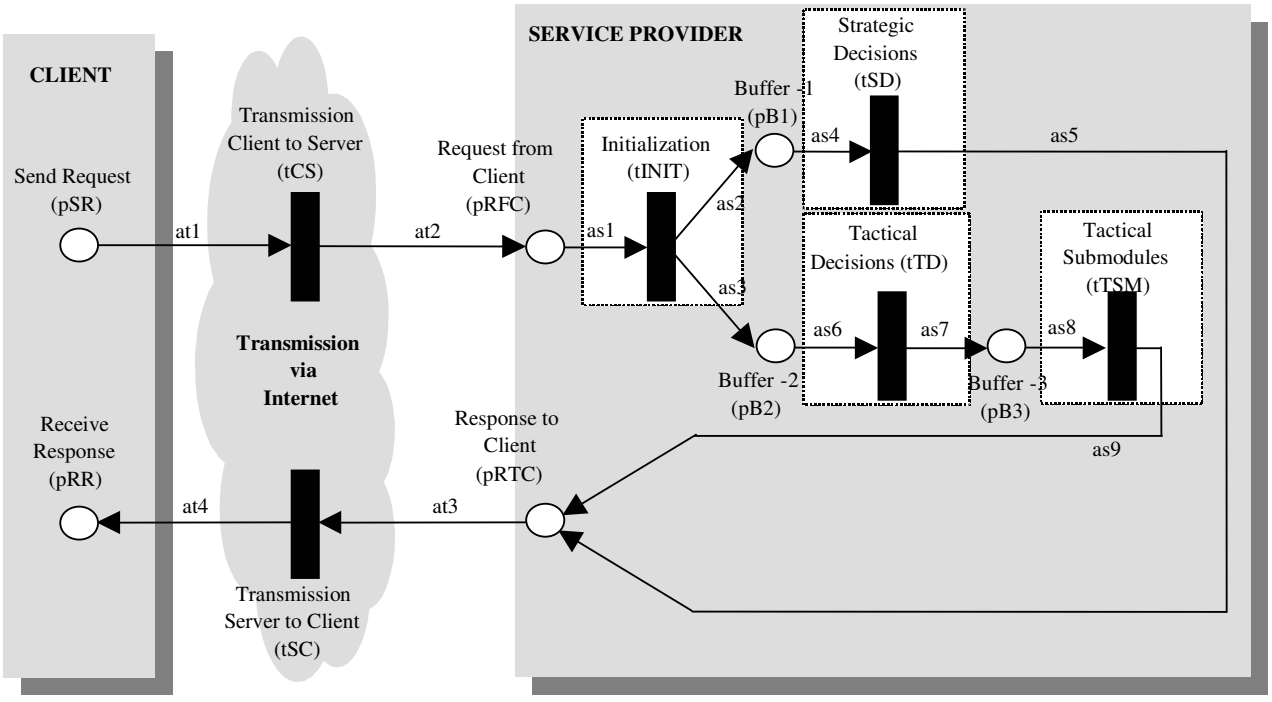

Figure 3. The Petri Net Model of the Information System

```
<pnml xmlns="http://ied.ux.his.no/Davidrajuh/PNML/pnml">
        <net id="pnml -net-id1"
                type="http://ied.ux.his.no/Davidrajuh/PNML/pnml/Simple1">
<!--- name of the Petri net ------ >
                <name> <text> Petri net model of the Information Systems (IACIS -2006) </text> </name>
<!--- declaration of places --->
                <place id="pSR"> <!--- Send Request --->
                        <initial_marking> <text> 1 </text> </initial_marking>
                </place>
                <place id="pRR"> </place> <!--- Receive Response --->
                <!--- similarly, the other 6 places (pRFC, pRTC, pB1, pB2, pB3) are also declared. -->
                <!--- For brevity, these details are not shown here --->

<!--- declaration of transitions --->
                <transition id="tCS"> <!--- trsnmission Client_to_Server --->
                        <firing_time> <!--- guassian distribution: mean 5 sec, std.dev: 50 ms --->
                                <text> normrnd(5000,50) </text>
                        </firing_time>
                </transition>
                <transition id="tSC"> <!--- transmission Server_to_Client --->
                        <firing_time> <text> normrnd(5000,50) </text> </firing_time>
                </transition>
                <transition id="tINIT"> <!--- Initialization --->
                        <firing_time> <!--- uniform distribution between 280 ms – 320 ms --->
                                <text> unifrnd(280, 320) </text>
                        </firing_ time>
                </transition>
                <transition id="tSD"> <!--- Strategic decisions --->
                        <firing_time> <text> unifrnd(80, 100) </text> </firing_time>
                </transition>
                <transition id="tTSD"> <!--- Tactical sub decisions --->
                        <firing_time> <text> unifrnd(10, 15) </text> </firing_time>
                </transition> <firing_time> <text> unifr nd(25, 35) </text>
<!--- declaration of arcs (establishing connections) ------>
                <arc id="at1" source="pSR" target="tCS"> </arc>
                <arc id="at2" source="tCS" target="pRFC"> </arc>
                <arc id="as1" source="pRFC" target="tINIT"> </arc>
                <!--- similarly, the other 10 arcs (at3, at4, as2 –to- as9) are also declared. -->
                <!--- For brevity, these details are not shown here --->
                … …
        </net>
</pnml>
```

Figure 4. The PNML Document

```
Petri Net Model for Information Systems
Initial Markings:
pSR       pRFC      pRTC      pRR       pB1       pB2       pB3
 1         0         0         0         0         0         0
step:2     Firing event: tCS     (Starting time:   0)  Firing Time: 4916.72
Current markings:
pSR       pRFC      pRTC      pRR       pB1       pB2       pB3
 0         1         0         0         0         0         0

[PRINT OUT DETAILS FOR STEPS 3 -6 ARE DELETED FOR BREVITY …]
…
step:7
Firing event: tSC     (Starting time: 5272.40)  Firing Time: 10278.67
Current markings:
pSR       pRFC      pRTC      pRR       pB1       pB2       pB3
 0         0         0         1         0         0         0
step:8
Firing event:  tSC     (Starting time: 5313.31)  Firing Time: 10327.70
Current markings:
pSR       pRFC      pRTC      pRR       pB1       pB2       pB3
 0         0         0         2         0         0         0
Completion time: 10327.70
```

Figure 5. Simulation Results

CONCLUSION

The mathematical modeling course of an IS curriculum should also provide a powerful tool for supporting comprehensive view and model-based analysis of information systems. The presented Petri-net based discrete simulation tool, GPenSIM, uses Petri Net Markup Language for model description and standard XML for data interchange. The software provides high-level platform independency and can be applied in a networked environment. The example model presented demonstrates how easy to use the features of the tool and shows what level of student support is provided in the learning process.

Beyond the basic course objectives of improving problem solving skills and thinking processes, especially abstraction, GPenSIM helps students to develop an in-depth understanding of the use of software tool and of the related computational problems.

Future work will focus on further extension of software support in the modeling process as such and will reflect the product life cycle view.

REFERENCES

1. Billington, J., Christensen, S., Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., & Weber, M. (June 2003). The Petri Net Markup Language: Concepts, Technology, and Tools. *Proceedings of the 24th International Conference on Application and Theory of Petri Nets 2003*, Springer LNCS 2679, 483--505.
2. Cassandras, G. & LaFortune, S. (1999), *Introduction to Discrete Event Systems.* Hague, Kluwer Academic Publications.
3. Davidrajuh, R. (2000). *Automating Supplier Selection Procedures*. PhD thesis, Norwegian Univ. Science and Technology (NTNU).
4. GPenSIM (2006). Available: http://ilab16.ilab.ux.his.no/GPenSIM/
5. Jungel, M., Kindler, E., & Weber, M. (Oct 2000). The Petri Net Markup Language. *Petri Net Newsletter*, *59,* 24-29.
6. Kindler, E. (2004). Using the Petri Net Markup Language for Exchanging Business Processes: Potential and Limitations. *Proceedings of the First GI-Workshop on XML Interchange Formats for Business Process Management (XML4BPM)*, Marburg-Germany, March 2004
7. Ma, H., & Davidrajuh, R. (2005). An iterative approach for distribution chain design in agile virtual environment. *Industrial Management and Data Systems, 105*(6), 815-834.
8. MATLAB (2006). Available: http://www.mathworks.com
9. Petri net world (2005). Available: http://www.daimi.au.dk/CPnets/
10. Wikipedia (2006). Available: http://www.wikipedia.org
11. Weber, M. & Kindler, E. (2003). The Petri Net Markup Language. *Petri Net Technologies for Modeling Communication Based Systems*, LNCS 2472, 124-144.