Simulation of Electronic Transactions using the DEVS Formalism for SCI 2001 / ISAS 2001

Iván Melgrati, Carlos Giorgetti, Ana Rosa Tymoschuk
GIRED Research Group – Systems Department
Universidad Tecnológica Nacional – Facultad Regional Santa Fe
Lavaise 610, (S3004EWB), Santa Fe, Argentina

ABSTRACT

Transactions over the Internet are rapidly becoming the tool of choice for users around the world to carry out an ever-increasing number of tasks. In part, this follows from satisfactory results of the first experiences in e-commerce of these users. After these first brush with Internet technology, they become more confident, and turn to on-line transactions more often.

The economic impact of electronic transactions is beyond any doubt one the most important aspects related to the use of Internet. These transactions display very distinctive features and cause, in turn, specific demands on the underlying infrastructure.

This presentation comprises the first stage of a research project focused on developing simulation models of e-commerce systems. On the basis of these models, different typical scenarios are tested. This paper introduces a general model for representing and simulating the components of the systems supporting electronic business transactions, using the DEVS (Discrete EVent System Specification) formalism.

The model considers a network composed of 2 Web Servers, one servicing static page requests and the other dynamic pages (ASP, Perl, CGI, etc.). Each resource is represented a DEVS atomic model, which are coupled to form the complete networking environment.

Keywords: simulation, models, electronic transactions, DEVS formalism, performance analysis, e-commerce.

1. INTRODUCTION

Transactions over the Internet are rapidly becoming the tool of choice for users around the world to carry out an ever-increasing number of tasks. Moreover, the explosion of the available Web sites makes the Internet a very valuable resource, providing almost any kind of information or service that can be thought of. Among these, are the all-too-common e-mail, discussion groups, virtual libraries, etc. As new technologies increase the capabilities of the Internet, new services arise: On-Line Learning, Streaming Audio and Video, E-Commerce, as well as many others.

E-commerce [1-2] has become a very effective means for doing business. Then, there is business-to-business (B2B) and business-to-customer (B2C). These transactions must be reliable, secure and efficient in order to guarantee the maximum profit and minimize risks. Therefore, the technology supporting them must comply with certain technological and functional standards to ensure an acceptable quality of service.

This paper introduces a general model for representing and simulating the components of the systems supporting electronic business transactions, using the DEVS (Discrete EVent System Specification) formalism. Once the models are completed, they are simulated under different load profiles so as to determine the impact of these workloads on the performance of the components of system and the system as a whole. These measurements provide insightful information that allow and support decision-making processes.

2. MODELING OF ELECTRONIC TRANSACTIONS

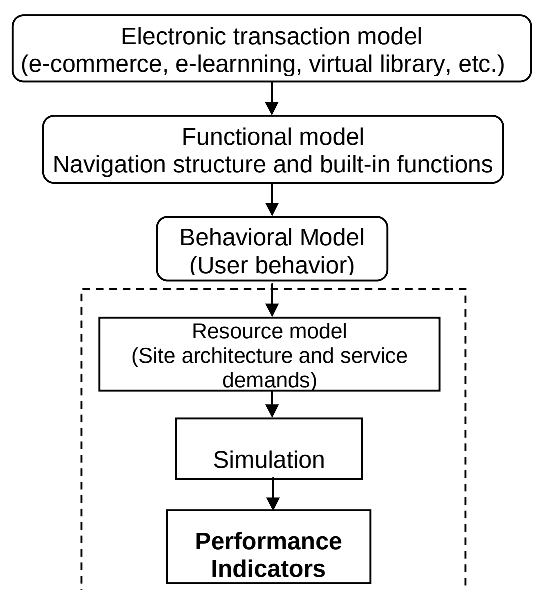

Figure 1

As Figure 1 shows, the modeling of the electronic transactions (and of any system in general) consists of several levels of abstraction. In the first stage the functional pattern of the proposed system is developed, depending on the system's actual transaction type. This accomplished by specifying its features and functions (access methodology to the system, users' validation, etc.)

Starting from these patterns and the users' preferences, it is possible to outline a model that reflects these characteristics, giving a formal specification of these patterns. This formalization process allows characterizing different workload profiles to which the system will have to respond.

Based on the load parameters, the structure and configuration of the modeled system, the resulting behavior is analyzed, to determine the system's capabilities and avoid the potential bottlenecks.

This behavior is studied constructing a performance model where the resources are basically considered as centers of service with their associated waiting lines. The main objective is to evaluate the system's performance by assessing indicators such as response time, processing speed, among other. Its purpose is to identify the key areas for improvements, always aiming to increase the system's capabilities and reliability. In turn, it can also be used to tune or adjust its operational parameters, using the results to estimate the capacity necessary to meet the new requirements

3. THE DEVS FORMALISM

In 1976, Bernard P. Zeigler [3-6] proposed a hierarchical modeling and simulation mechanism, known as DEVS. This methodology applies system theory concepts to model continuous-time systems using a discrete events approach. This paradigm lets the modeler describe the model's behavior in a modular fashion, attacking complexity using a hierarchical approach.

This approximation provides a means to specify a mathematical object named system. A system is described as a set consisting of a time base, a set of inputs, a set of outputs, and a group of functions used to compute the model's successive state changes as well as informing other models about these changes.

DEVS is a universal formalism used to model and simulate DEDS (Discrete Event Dynamic Systems). The formalism defines how and when the state of a model changes, thus giving the modeler great flexibility when specifying the behavior of a system.

In addition, the time between events is variable. This allows different models execute their transition functions only if they have to, avoiding the overhead of checking whether a transition has to occur in every model, at each time-step (as in cellular automata, for instance). This affords better simulation speed since no transition function is activated unless it's necessary to do so.

DEVS models can be viewed as an entity with input/output ports that allow the model to communicate with its surroundings. The internal structure of the model is composed of:

- A set of state variables (that make up the model's current state)
- An internal transition function, indicating how and when internal changes take place
- An external transition function, which specifies how the model reacts to external messages
- An output function (which is used to communicate with other models).

The ability to "connect" atomic models together provides the means to create new, more complex models named coupled models. By applying a mathematical transform, coupled models can be treated as atomic models, thus allowing seamless development of models in a modular, hierarchical way. By repeatedly applying this transform, the modeler can develop very complex models in a hierarchical manner, by developing (and testing) each component separately and then combining them until the complete model is finished.

This provides an abstraction layer that allows continuous refinements to be applied to the model without affecting the global structure of the system. Moreover, it lets the modeler reuse the atomic model specifications without having to re-write them from scratch. This gives the modeler the ability to take models from a model database, and place them into a new one thus enhancing productivity and ensuring the correctness of the components without re-testing them.

To develop these atomic components the CD++ [7] simulation engine was used. For generating and simulating the coupled models, the Visual DEVS environment (developed by the authors of this paper) was applied. After completing the simulation runs, a performance analysis was performed to evaluate the impact of network configuration/topology, measured under different load levels, on the modeled system.

4. MODEL DESCRIPTION

Systems supporting transactions over the Internet can be seen as one or more servers which service a number of clients. These elements, along with operating systems,

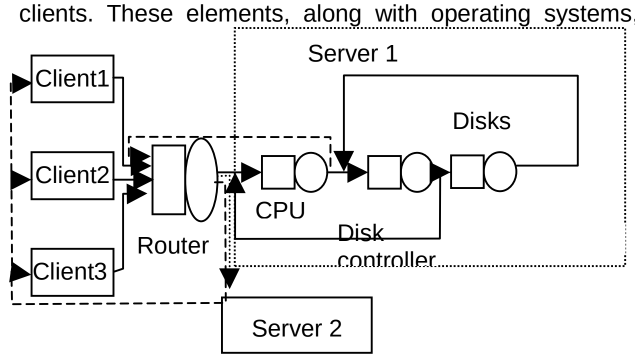

Figure 2

communication hardware and networking protocols work together to allow the effective exchange of information and content among the intervening parties.

Figure 2 depicts an outline of the modeled system, showing the structure for one of the servers and its physical resources. Circles represent a resource while rectangles identify queues of unfinished requests.

Each of these requests has three distinct components: processing time at the client, delay in the Intranet/Internet (round-trip) and the time to complete the task at the server.

Everyday users exhibit some characteristics that allow the modeler to set them apart based on criteria such as frequency of requests, types of requirements, etc. These in turn generate different workloads on the system that must be accounted for in order to analyze its performance and avoid potential bottlenecks. In this case, it is useful to divide requests based on the kind of content they request: static Web pages, dynamic content and finally, a mixture of both.

This model proposes 3 types of clients, each of them falling into one of the categories mentioned above. Hence, there is a client that request only static pages, another one access exclusively dynamic content (such as ASP pages, PHP-based scripts, etc.) while the third one represents a user that accesses a mixture of either type.

5. DESCRIPTION OF THE VISUAL-DEVS TOOL

The integrated development environment (IDE) Visual-DEVS (developed by I. Melgrati) was designed for modeling and simulating discrete event models. The tool was written in Visual Basic and uses the CD++ simulation engine [7] to perform the actual simulation.

The multi-document interface (MDI) included in this tool allows to work at the same time on several, independent models. The specifications of the atomic models (parameters, input/output ports) are stored in a database, from which they are extracted whenever a new component is added to the working model.

The purpose of this environment is to ease the design of DEVS coupled models, in a graphical way, avoiding the need to use formal descriptions based on text files.

The system allows adding models and couplings in a dynamic way, and at the same time lets the user include new descriptions of atomic components. Consequently, new models can be integrated to the tool giving the modeler the ability to adapt the tool to various requirements of his/her work.

The tool has an interface to the simulation engine. Simulation parameters (simulation time, accuracy, output files, etc.) can be specified and passed to CD++. In figure 3, a typical view of the tool is shown.

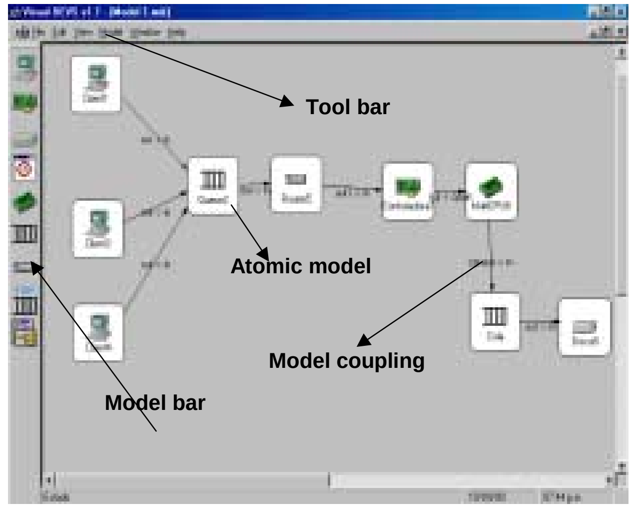

Figure 3

The model bar allows adding atomic components to the coupled model, with a single click. When adding a model, a dialog box appears where the parameters of the new component can be written down. These properties can be modified at any moment, by double clicking on the model icon.

To link two icons, the user selects them, then a dialog box is activated, where the origin and destination ports are chosen from list boxes.

The tool bar lets the user carry out activities such as exporting the outline from the coupled model to WMF or EMF files, as well as to copy it to the Clipboard. The size of the graph can also be adjusted (zoom), with options to adjust the scales in different ways (fit to screen, 1:1, etc.)

6. RESULTS

The output variables of the chosen simulation model are the system's processing speed and response time under each of the simulated load profiles.

Case 1 shows a comparison of the selected indicators against the time between clients' requests to assess the behavior of the system as the requests from clients or the actual number of active clients -of each kind- connected to the network increase. The input variables of the system are the mean time between requests in each client; all other parameters remaining unchanged. Figures 4a and 4b show the results of the simulation runs performed under these conditions.

In this case, the inter-arrival time was varied among the successive petitions of the clients to examine the server's response as the number of requests grows.

Just as it could be expected, the graphics demonstrate that as it increases the petitions in the system, the speed of prosecution of the works diminishes, at the same time that the time of answer of the system increases visibly.

The performance measures improve as the requirements become more frequent until arriving to a maximum (good) where the indicators are degraded due to the overload of the system and the biggest times of wait in the lines associated to the devices.

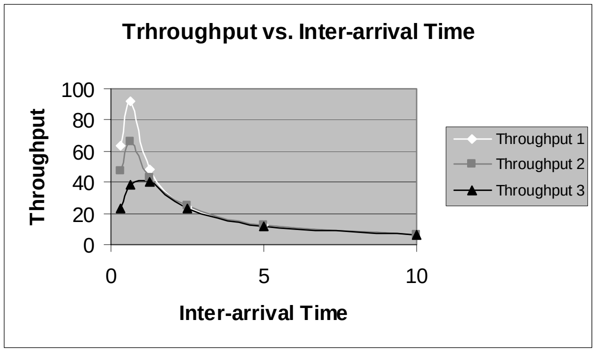

Figure 4a

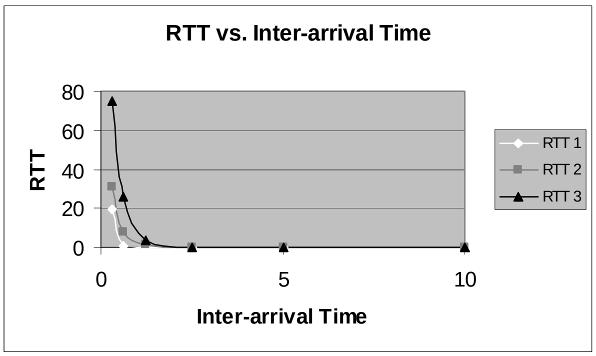

Figure 4b

It is important to point out that dynamic-pages-only clients' performance (black curve) degrades much faster than that of the other clients' because of the higher requirements imposed on the CPU and disks serving those requests.

In Case 2 the servers' relative load (% of requirements to one server over the other), is the changing factor. It is necessary to highlight that in this case, the parameters of the three client groups were equaled to isolate the influence of the relative load from any other factors. Figures 5a and 5b show the results obtained during the simulation under these conditions.

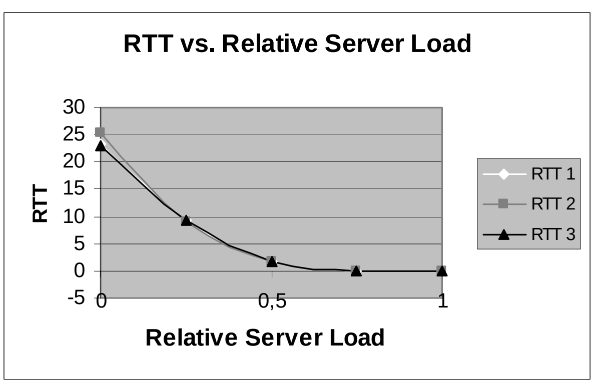

Figure 5a

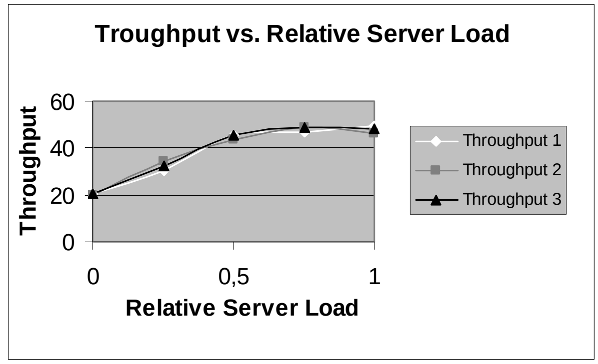

Figure 5b

In this case, as the percentage of requirements of static pages is increased (Server S1), the global performance increases due to a smaller average time to complete the transactions.

As the three load profiles possess the same operating parameters, the measures are almost identical, except for small fluctuations product of the random nature of the modeled system.

In Case 3 the changes in the influence of the delay of transmission at the router is considered.

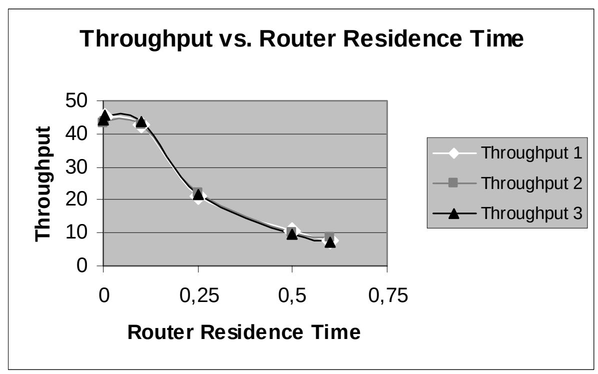

Figure 6a

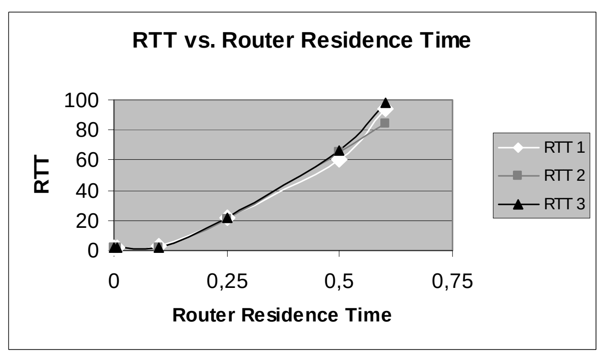

Figure 6b

From the observation of the results in Figures 6a and 6b it follows that as the processing speed of the router drops, the performance of the system is significantly hindered.

Case 4 shows the impact caused by the variation of the delay on the Internet on the performance of the system. It is necessary to underline that both servers are configured to operate at the same speed. This avoids the influence of different server speeds yielding a more accurate set of results.

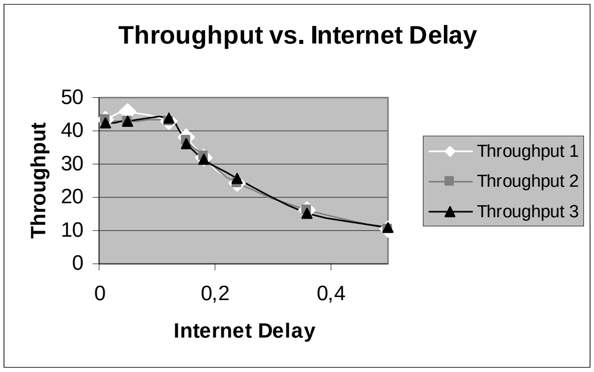

Figure 7a

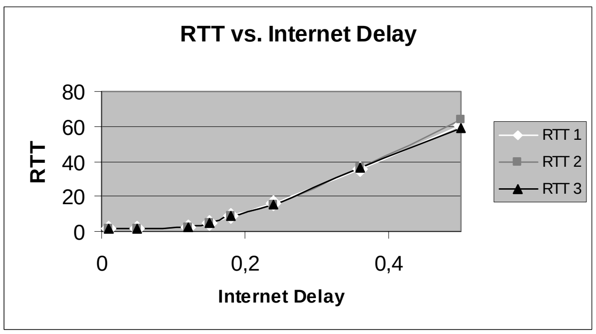

Figure 7b

From the analysis of the results it is inferred that, as in the previous case, the lag imposed by the Internet provokes a sound decrease in the efficiency of the transactions. The optimization of the Internet portion of the system is of vital importance to improve its overall quality of service.

In the last Case the reaction of the system is shown in the light of changes in the processing power of the dynamic-pages server's CPU. It is worth to mention that during this series of simulations the speed of the CPU of the static-page server remained constant.

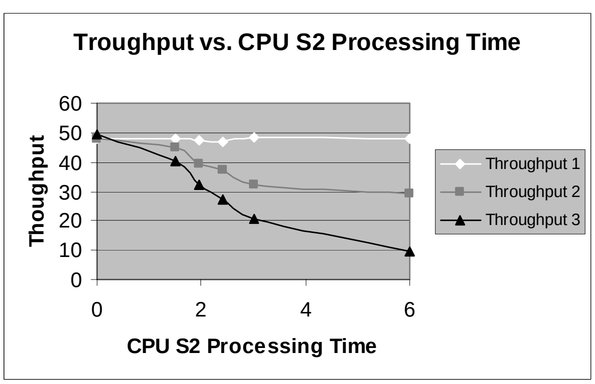

Figure 8a

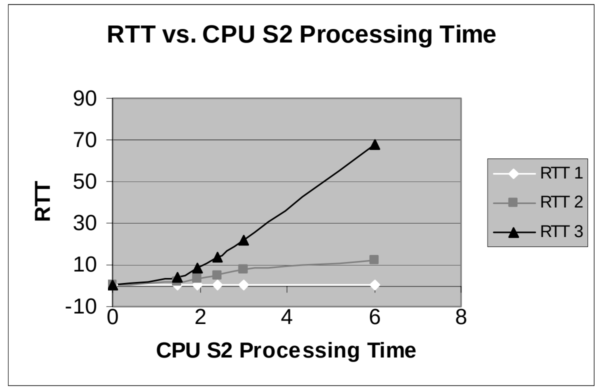

Figure 8b

As figures 8a and 8b confirm, the variation of the performance of the CPU of the dynamic-page server significantly affects those users that perform transactions using that server, especially those who only operate with it.

7. CONCLUSIONS

A simple model of electronic transactions operating over the Internet was implemented using l the DEVS formalism as the selected modeling tool.

From the simulation runs, results consistent with characteristic measures of typical systems of this kind were obtained.

From these results the following conclusions were extracted:

- The router poses a potential bottleneck, since all the requests and replies must go through it. Hence, it is vital to achieve the highest possible speed in the routing process in order to minimize delays.

- The optimization of the communications link to the Internet greatly improves the system throughput.

In general, all these factors have a tremendous impact on user satisfaction. Therefore, it is crucial to take every possible action to improve the overall quality of service. Simulation models are a very effective way to detect potential bottlenecks as well as to test new scenarios without having to physically implement them.

The use of the DEVS formalism has proven to be an effective means to develop powerful, expressive simulation models. Moreover, the use of this methodology has allowed a simple, effective way to modify these models, which in turn allowed for an easy adaptation to the proposed scenarios.

The use of graphical tools (such as Visual DEVS) is a very compelling means to develop DEVS models, for the modeler can better visualize its overall structure and dependencies. On the other hand, these tools avoid the error-prone procedures which are common in text-based specifications, helping to achieve a greater degree of accuracy and expressiveness.

8. ONGOING AND FUTURE WORK

This paper outlines some of the work carried out until the moment by the GIRED Research Group (Networks Research Group), in the area of modeling and simulation of computer networks through the use of the DEVS formalism.

Other areas of development of the group include the monitoring of computer networks in a controlled atmosphere (Connectivity Lab at the Systems Department).

Also, Petri Nets are used to model these networks, to assess the applicability of this modeling paradigm to real-world environments.

Below is a summary of the Group's current and future areas of work:

- Improvements to the of Visual DEVS tool's GUI, such as the inclusion of coupled models to the toolbar, thus allowing multi-level, hierarchical modeling.

- Performance monitoring on real-world systems is continued to obtain typical-data feeds, thus aiding to refine and validate output generated by simulation models.

- After completing a basic model-base, the Group is planning to expand it by including new, more specific models (hosts, network interfaces, routers, etc.)

9. REFERENCES

[1] D. Menasce, V. Almeida, "Scaling for E-Business. Technologies, Models, Performance and Capacity Planning." Prentice Hall, 2000.

[2] D. Menasce, V. Almeida, "Capacity Planning for Web Performance. Metrics, Models, and Models." Prentice Hall, 1998.

[3] B. Zeigler, "Theory of modeling and simulation". Wiley, 1976.

[4] B. Zeigler, "Multifaceted Modelling and discrete event simulation". Academic Press, 1984.

[5] B. Zeigler, "Object-oriented simulation with hierarchical modular models". Academic Press, 1990.

[6] B. Zeigler, D. Kim, "Design of high level modelling / high performance simulation environments". Technical Report, Department of Electrical and Computer Engineering, University of Arizona. 1995.

[7] D. Rodríguez, G. Wainer, N-CD++: Manual del Usuario, Departamento de Computación, FCEN, UBA, Argentina, 1999.