# AN APPLICATION FOR WEB-BASED MODELING AND SIMULATION

Yurena García-Hevia Mendizábal<sup>(a)</sup>, I. Castilla<sup>(b)</sup>, R.M. Aguilar<sup>(c)</sup>, R. Muñoz<sup>(d)</sup>

<sup>(a)</sup> Engineer at the Computer Science School, La Laguna University, Spain <sup>(b)(c)(d)</sup> Department of Systems Engineering and Automation and Computer Architecture, La Laguna University, Spain

<sup>(a)</sup>ygarcia@isaatc.ull.es, <sup>(b)</sup>ivan@isaatc.ull.es, <sup>(c)</sup>rosi@isaatc.ull.es, <sup>(d)</sup>rmglez@isaatc.ull.es

#### ABSTRACT

The disciplines of modeling and simulation are essential to the study of complex systems. The Department of Systems Engineering and Automation and Computer Architecture at the University of La Laguna has been developing a Java simulation library for Discrete Event Systems (SIGHOS). SIGHOS is intended to be used as part of tools that analyze the effects of proposed solutions on real problems. Using this library, however, requires programming skills which limit its effectiveness. In this paper we propose the introduction of a logic layer that separates the low-level simulation mechanisms of the modeling system. We achieve this through the creation of a web service that can run the simulation remotely via the Internet using SIGHOS. We then describe a service-oriented web application that encompasses a modeling and simulation environment through which a user can create models, process them (using the aforementioned web service), and observe the results without any knowledge of programming.

Keywords: web-based solution, simulation, SIGHOS

#### 1. INTRODUCTION

As discussed in previous articles (Aguilar, Castilla, Muñoz, Estévez, Martín and Moreno. 2005; Aguilar, Muñoz, Castilla, Martín, and Piñeiro 2006b) SIGHOS models through the use of a process-based methodology where the elements proceed via stages as determined by the state of the elements themselves. Within each stage, the elements perform different activities which make use of certain resources. These resources have to be available so as to devote part of their time to the execution of said activity.

Until now, the model to be simulated with SIGHOS could be described in one of two ways: by using the functions provided by the Java library, or by employing the XMLGHOS interface (Aguilar, R. Muñoz, Castilla, Martín, and V. Muñoz 2006a) which facilitates its use and automatically generates the models. XMLGHOS describes the system by using two files:

- *XMLModel* defines the resources, the activities, and the paths or *flows* the elements go through.
- *XMLExperiment* specifies the duration and number of simulations to be run, the type of

information to be gathered, and the way the elements flowing in the model are generated.

Although the use of XMLGHOS simplifies the modeling, those wishing to use the library have to be familiar with XML. Several new approaches are being addressed to improve the usability of SIGHOS. In this sense, web-based technology would allow the simulation models to be configured using forms which are more familiar to end-users. Moreover, this technology facilitates remote interoperability and accessibility, as well as the integration with a wide range of interfaces available on the market.

#### 2. THE WEB SERVICE

In recent years, web services have become one of the easiest and most efficient ways to access services remotely. The use of web services in simulation remains largely untapped. There are some papers on the topic, but most of them are limited to a specific operating environment (Orsini, Bruzzone and Revetria 2003; Moles, Alonso and Banga 2003; Bruemel, Vinichenko and Novickis 2007), such as assembly lines, transportation services, etc., or use outdated technologies (Taylor 2000) as applets.

Web services may be defined as a set of applications or technologies that are able to interoperate on the web (W3C 2004). These applications or technologies exchange data for the purpose of offering services. The providers offer their services as remote procedures and the users request the service by calling these procedures via the Web. The services provide standard communication mechanisms between the different applications, which interact among themselves to present dynamic information to the user. This interoperability is made possible by the use of open protocols.

The simulation processed by SIGHOS can thus be executed on a high-performance system (computer cluster, supercomputers, dedicated servers...) and accessed via the web service transparently by various clients. These can be written in different programming languages and be running on different hardware and/or software platforms thanks to the use of open standards.

The web service involves adding a new abstraction layer to the SIGHOS simulation architecture, as shown in Figure 1. In this process, a method, which we designate simulate(), is published and made available on the Internet. This process requires a description of the parameters needed to carry out the simulation, namely the *XMLModel* and *XMLExperiment* previously described.



Figure 1: Web service layer (SIGHOSWS) over XMLGHOS and SIGHOS.

In order to facilitate the post-processing of data, the simulate() method returns the results of the simulation in XML format.

## 3. WEB-BASED MODELING AND SIMULATION ENVIRONMENT

One of the main objectives of our study is to make SIGHOS accessible to those users who are unfamiliar with the currently available interfaces (Java and XML). In this section, we describe the development process of a web application which constitutes a modeling and simulation environment and which allows these users to easily access the powerful characteristics of SIGHOS. This application is known as MOSINET (Modeling and Simulation on the Internet).

### 3.1. Analysis and design

analysis and design of the application consider the conditions or abilities that must be met by the software system to be developed so as to satisfy the specifications needed to increase the usability of the library.

The demands placed on traditional desktop applications by the user include necessities such as functionality (what the system must do), the storage of information, the ease of use and performance. Within a Web application, in addition to the above, it is also necessary to correctly define user accessibility requirements, its adaptability and scalability, its availability and participation (Cala 2007).

Often, the navigational structure of an application can be comfortable and intuitive for a Web designer who is familiar with the environment, but a user may not be able to identify all the requirements if he lacks an understanding of nodes, links, option menus, search engines, etc. It is important that these requirements be specified in terms:

- that the user knows and understands so as to improve communications between the analyst and the user; and
- that are sufficiently clear and concise so as to be correctly interpreted by the programmers of the web application.

When dealing with a constantly changing environment, the concepts of adaptability and scalability take on added significance. It is important that the system be able to grow without compromising the quality of the services offered, and also that it be able to respond to changes in lower levels of the application.

Since the use of the application will not be subject to strict time schedules and will be accessible to users in different time zones, it must be *available* at all times. This condition can be met through the use of web technologies.

#### 3.2. Application architecture

The application's architecture is structured in three layers, as shown in **Errore. L'origine riferimento non è stata trovata.** This separation allows for development to take place at several levels (by separating each layer from the whole). In case of a change, it is only necessary to modify the affected level without having to revise the entire code.



Figure 2: Web application layer structure.

Each layer's functions are as follows:

- *Presentation layer:* the graphical interface which interacts with the user. Used to show information to the user and to record the information provided by the user.
- *Business layer:* used to implement the services to be provided by the application. It receives requests from the presentation layer, processes

them and returns the results. It also communicates with the data layer to store or retrieve information from the database.

• *Data layer:* used to store and manipulate data. It consists of a database and a manager that receives information storage and retrieval requests from the business layer.

The application is built on the Java Spring and Hibernate frameworks. These frameworks simplify the development of the business and data layers by supplying the tools needed for user session management, object handling, framework-API interactions, etc.

### 3.3. What the user sees

The structure of the application's proposed presentation layer is displayed in **Errore. L'origine riferimento non è stata trovata.**, which shows four strictly classified groups of pages.



Figure 3: Site map of the web application

• On the start page, the user accesses the system by entering a unique username and password. Once validated, he selects a SIGHOS *XMLModel* with which to work (or creates a new one). A set of *XMLExperiments* is associated with each model. This work environment, in which just one *XMLModel* is used in each session, simplifies the reference checks made between *models* and *experiments*, that is, when reference is made to deleted element or resource types, when invalid identifiers are used, etc.

- On the pages used to define the model, the user can use forms to edit the SIGHOS components graphically. The data input are converted to XML and sent to the web service at the time of simulation. The components described here are the elements, the resource types and their availability, and the activities and work flows through which the elements will be processed.
- All the *XMLExperiments* associated with the model in use can be seen as a list. As with the *model*, the user can graphically edit the information associated with each SIGHOS experiment, including the way the elements are generated, the duration and the results to be recorded during the simulation. This information is stored in XML format and is used along with the model description file to carry out the web service simulations.
- The data from the processed simulations are shown on the results page. The user can select one simulation from a list, and thus see the result outputs returned by the simulator. These results are those which were selected in the experiment, and can be displayed graphically or downloaded on a file (XML, CSV, etc.).

# 3.4. Use of the application

As described in the previous section, the user works with a single *XMLModel* and its associated *XMLExperiments*. This model can be new, or it can be edited using forms, or imported from an XML document.

Once the *model* is selected, *experiments* can be edited, created, imported and deleted; selected to be simulated; copied to create new experiments from existing ones, etc.

**Errore.** L'origine riferimento non è stata trovata. shows a prototype of the application. The screen has been divided in four areas:

- *Top pane*: it shows the application's logo, the login controls (you can log in, create an account or log out if you are logged in) and one link to access help (which explains how to use the application).
- *Left menu:* in this zone the user can select from among the four sets of pages previously described (start, models, experiments and results).
- *Main pane*: it shows the content of the page selected on the left menu.
- *Right pane*: there is a list of feature options which can be applied to the contents shown on the main pane (e.g. when viewing the list of available experiments, these options will include the possibility of deleting, copying, editing or exporting an existing experiment to XML, creating or importing a new experiment, etc.).

21141	/LL MOSINET	
Inicio	Inicio > Experimentos > Edición > Contenido General Contenido Nº experimentos: 5 Y Partir del estado final de: selección	✓ Options     Nuevo Listener     Nuevo Listener
Experimentes Resultados	Comienzo de la simulación: 5 V Final de la simulación: 5 V Listener: ActivityListener Descripción 1	unidad de tiempo: selección
Experimentos     Experimento 1     Experimento 2     Experimento 3		
Experimento 4  Nuevo	Generadores de elementos: Información Generador 1 Descripción 1 Generador 2 Descripción 2 Generador 3 Descripción 3	
Borner toda	Guardar Descartar Eliminar experimento	

Figure 4: Web application prototype

The experiment(s) to be simulated can be selected on the screen where they are displayed. This process then launches simulations which are calls to the published web service's simulate() method. Each selected experiment involves a call to simulate() with the same *XMLModel* and different *XMLExperiment*.

### 3.5. Results

MOSINET has progressed from the analysis phase into the development phase. To date, the XML interfaces for returning the results of the service have been developed and the web service has been published.

The first version of MOSINET is also available, and some screenshots can be seen in Figure 5 and Figure 6. This version allows users to perform the following operations:

- Upload models and experiments for storage in a database.
- Select a model and experiment to be simulated from among those available. This action uses the published web service.
- Once the simulation is completed, the results obtained are provided to the user in various formats, as shown in Figure 6. The results can also be downloaded in an XML format. The information is also used to generate graphs.

MOSINET		
upload		
ENTER THE MODEL AND THE EXPERIMENT FILES		
XML model name modelo XML model file Uploadl		
Upload Experiment XML experiment name experimento XML experiment file Upload Upload		
Select model file Choose One Select experiment file Choose One Simulate		

Salir Ayuda



A large portion of the data layer and the initial phase of the business layer has been implemented in the background of this application.



Figure 6: Web Interface for the simulation results

### 4. CONCLUSION

In this paper, we have presented a modeling and simulation framework which simplifies the use of a java simulation library.

A web service has been developed to make SIGHOS and the graphical interface independent, thereby enabling interoperability using standard protocols. This allows us to execute SIGHOS on a computer cluster, and the web interface on another server. Consequently, the capabilities of both two layers of the system are optimized.

In addition, a web interface called MOSINET has been designed to model and simulate with SIGHOS in a more convenient and easier way, hence improving the accessibility to the library for all types of users. The architecture of this interface is also divided into layers so that the data stored for each user and the services offered by the application remain separate.

#### ACKNOWLEDGMENTS

This work is being supported by a project (reference DPI2006-01803) from the Ministry of Science and Technology with FEDER funds.

Iván Castilla is being supported by an FPU grant (ref. AP2005-2506) from the Spanish Ministry of Education and Science.

## REFERENCES

- Aguilar R.M., Castilla, I, Muñoz, V, Estévez, J.I., Martín, C.A., Moreno, L., 2005. Predictive Simulation for Multiagent Resource Distribution: An Application in Hospital Management. International Mediterranean Modeling Multiconference (EMSS 2005). Marseille- France.
- Aguilar, R.M., Muñoz, R., Castilla, I., Martín, C.A., Muñoz, V., 2006. An interface to integrate workflow diagrams into discrete event simulation. International Mediterranean Modeling Multiconference. Barcelona- España.
- Aguilar, R.M., Muñoz, R., Castilla, I., Martín, C.A., and Piñeiro, J.D., 2006. Verification and Validation in Discrete Event Simulation: A case study in hospital management. (EMSS 2006) Barcelona-Spain
- Bruemel, E., Vinichenko, S., Novickis, L. 2007. Model-Based essential logistics principles for creating a web-portal of transport services consumers. I.J. of SIMULATION Vol. 8 No 1. Available from: http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-8/No-1/Paper-1.pdf [accessed 9 April 2008].
- Cala, C., Rodríguez, A.L., and Barroso J.A. 2007. Reflexiones sobre el Framework de desarrollo del Consejo Superior de Investigaciones Científicas. Available from:
- Moles, C.G., Alonso, A.A., Banga, J.R., 2003, Desarrollo de Herramientas de Simulación Vía Web Basadas en Ecosimpro. February 24-25, Madrid (Spain)
- Orsini, A., Bruzzone, A.G., Revetria, R. 2003 Framework Development For Web-Based Simulation Applied to Supply Chain Management. I.J.of SIMULATION Vol. 4 No. 1. Available from: http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-

4/No-1&2/Orsoni.pdf [accessed 9 April 2008].

- Taylor, S. 2000. Netmeeting a Tool for Collaborative Simulation Modelling. I. J. of SIMULATION Vol. 1 No 1-2. Available from: <u>http://ducati.doc.ntu.ac.uk/uksim/journal/issue-1/SimonTaylor/SimonTaylor.pdf</u> [accessed 9 April 2008].
- W3C, Web Services Architecture. 2004. Available from: <u>http://www.w3.org/TR/ws-arch/</u> [accessed 15 April 2008].

# AUTHORS BIOGRAPHY

**YURENA GARCÍA-HEVIA MENDIZÁBAL** is a Technical Computing Systems Engineer from the University of La Laguna. She is currently a student of Engineering Computer Science and a research grant holder at the Department of Systems Engineering and Automation at the University of La Laguna. Her fields of interest are the design and implementation of simulation environments for service organizations, modeling and simulation, and business intelligence. Email to <<u>ygarcia@isaatc.ull.es</u>>

IVÁN CASTILLA was born in La Laguna, Tenerife and attended the University of La Laguna, where he studied Engineering Computer Science and obtained his degree in 2004. He is currently working on his PhD with the Department of Systems Engineering and Automation at the same university. His interests include discrete event simulation and business intelligence. Email to <<u>ivan@isaatc.ull.es</u>>

**ROSA M. AGUILAR** received her MS degree in Computer Science in 1993 from the University of Las Palmas de Gran Canaria and her PhD degree in Computer Science in 1998 from the University of La Laguna. She is an associate professor in the Department of Systems Engineering and Automation at the University of La Laguna, Canary Islands, Spain. Her current research interests are decision making based on discrete event simulation systems and knowledge-based systems, intelligent agents, and intelligent tutorial systems. Email to <<u>rosi@isaatc.ull.es</u>>

**ROBERTO MUÑOZ** was born in La Laguna, Tenerife and obtained his degree in Engineering Computer Science from the University of La Laguna in 2006. He works as an analyst in Nivaria Innova and collaborates with the Department of Systems Engineering and Automation at the University of La Laguna, where he is working on his PhD. His research interests include the modeling, simulation and optimization of business processes. Email to <<u>rmglez@isaatc.ull.es</u>>