International Journal of Sciences and Techniques of Automatic control & computer engineering
IJ-STA, Volume 5, N° 2, December 2011, pp. 1678–1691.

Web based Simulation environment of queuing systems

Graini Abid[1], Boubetra Abdelhak[2], Fillali Ferhat[2] and Graini Lazhar[3]

[1] Computer Science Department, University of Mohamed Boudiaf,
28000 Msila, ALGERIA.
grainiabid@yahoo.fr
[2] Computer Science Department University of Bordj Bou Arreridj,
EL Anasser 34000 BBA, ALGERIA.
boubetraabd@yahoo.fr
fillali.ferhat@caramail.com
[3] Institute of Optics and Precision Mechanics, Université Ferhat Abbas,
Sétif 19000 ALGERIA
aissa.oussa@gmail.com

Abstract. *The aim of this paper is to introduce the Web Based Simulation (WBS). The advantages of this paradigm are the basis of several applications accessible by any user involving remote real-time interactions and large scale visualization. We benefit from the Web technology to produce a Web-based discrete event simulations tool written in Java. This system is reachable through any internet browser and allows multiple types of users (simulation specialists, students,) to create simulation models, access to existing models, run simulations and share experiences. The main goal is to allow modeling and simulation of queuing systems, and compare the performances measured by simulator with analytical measurements. The problem that matters for all queuing systems simulators is to determine the number of replications to run before stopping simulation. We propose in this paper an iterative algorithm which attempts to maximize the trust of all performance measures by minimizing their standard deviation. Our system adopts the Local Simulation / Local Visualization Strategy, i.e. a simulation applet is downloaded from a server which makes the user involved in all phases of the simulation. The models created and the measures of performance are saved in XML format for easy reuse.*

Keywords: *Discrete Event Simulation, Web based simulation, Queuing systems.*

1. Introduction

Nowadays, many companies offer versions of software based on the web, which allows reaching new customers and providing continuous access to the most recent version of the software. These solutions based on the Web can be placed on multiple

This paper was recommended for publication in revised form by the editor Staff.
Edition: CPU of Tunis, Tunisia, ISSN: 1737-7749

distributed servers and allow an unlimited number of simultaneous users to access at any time. Web technology has the potential to significantly alter the ways in which simulation models are developed, documented, and analyzed [1]. After starting in 1996, research and development work on web-based simulation has exploded [2, 3]. As all web-based systems, the WBS should take the available standards into consideration. A WBS must be built on the basis of web standard technologies, such as HTML, XML, DOM, SOAP, etc making the application available to a wide variety of platforms [4]. In order to facilitate model creation, most WBS systems rely on component-based development which is an advantage of object-oriented paradigm (OOP) [4, 5].

Several benefits exist for web-based simulation. We can summarize them in the following points:

- Web-based simulators can be accessed from any computer using a web navigator that allows a computer running the simulation engine to become into a virtual simulation home.
- It allows modifications, updates and implementations to be made through the server.
- There is no explicit limitation in the amount of memory for storage of models.
- It allows access to server systems that can run the simulation faster (some parts of the simulation are executed on the server and other parts are executed on the local station).
- Using models repositories make them available to be shared and reutilized by many users.
- It is possible to build simulations where many users interact with each other (Distributed Interactive Simulation).

The main objective of this paper is to propose and implement a web based simulation tool of (M / M / s) queuing systems. We have designed and implemented a Java applet that allows a user to: compose new model, modify and run an existing one, save performance measures, etc. A simulation model can be instantly regarded as an animation on a web browser, allowing users to take advantage of the web environment, by offering the same level of interactivity and multimedia features as traditional simulators.

This paper is organized as follows : in Section 2, we describe web based simulation approaches. Section 3 is devoted to architectural design of the simulator. Section 4 describes the main components of the user interface. In section 5, we illustrate the proposed algorithm to determine the best number of replications to run. In section 6, we provide various experiments that demonstrate the effectiveness of our realization. Finally, section 7 concludes the paper.

2. Methods for Web-Based Simulation

There are three Web-based simulation and animation approaches suggested by Lorenz et al. (1997). In addition, Whitman et al. (1998) suggested three approaches. Here, we note that both suggestions have the same idea [6]. We show in this section an overview about each method.

2.1. Server Hosted Simulation

In this approach, a Common Gateway Interface (CGI) is used to transfer the model's parameters to the web server. When the simulation has finished, the CGI-script prints the results in an HTML page which is transmitted to the client's navigator. The figure Fig.1 is an illustration of such a server hosted simulation.

This type of simulation provides the advantage of using a familiar tool and enables the reuse of existing models [3, 5]. The primary disadvantage is that the client can only provide inputs and view pre-specified outputs at a specified time.

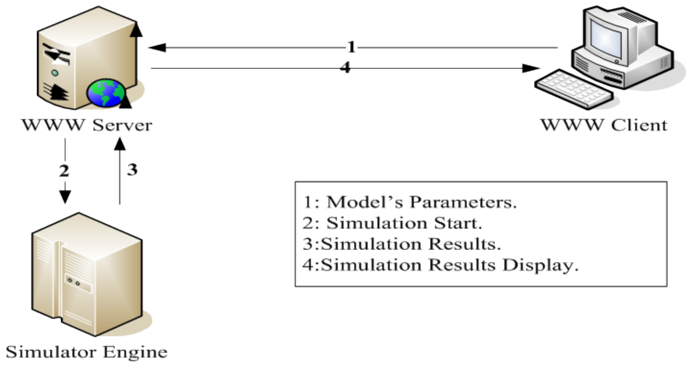

Fig. 1. Remote Simulation approach

2.2. Client Executed Simulation

This approach was introduced by Whitman et al. (1998) [6]. We shift the responsibility for execution completely from the server to the client as shown in Fig 2. Some simulation tools use an applets based approach, the user loads a Java Applet on his computer and the work is done on the client's machine. The performance of this type of simulation is limited by the client machine performance capability [7, 8, 9].

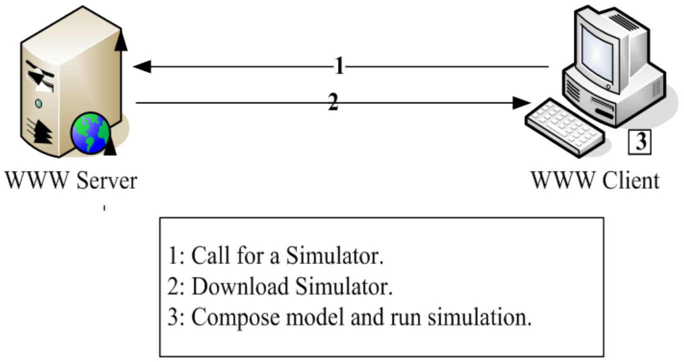

Fig. 2. Local Simulation / Visualization approach.

2.3. Hybrid Client/Server Simulation

The approach, first introduced by Berger and Leiner [1997] by combining the approaches of remote simulation and local visualization [4]. The simulation runs remotely on a simulation server. The results are transferred to the client and visualized locally. The animation can change continuously, delayed only by the executing simulation model and transmission time [6, 10]. Fig.3. is an illustration of the hybrid client/server simulation.

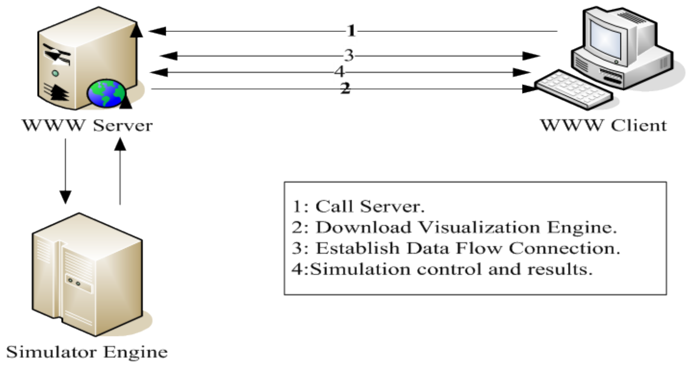

Fig. 3. Remote Simulation / Local Visualization approach.

3. Design and Implementation

A Web-Based Simulation System should provide functionality to process simulation models by using the benefits of the web technology. Basic functionalities are for storing simulation related data on a server and to download simulations applets from a server, which implies the use of a HTML-based interface [11]. The data are stored in XML format. The clients access the features of the simulation server using controls, which are implemented in HTML-documents; the users can transmit models parameters and data towards and from the server. The figure Fig.4 shows the main components building the WBSE.

A simulation applet is downloaded from a server which makes the user involved in all phases of the simulation. The models created and the measures of performance are saved in XML format for easy reuse.

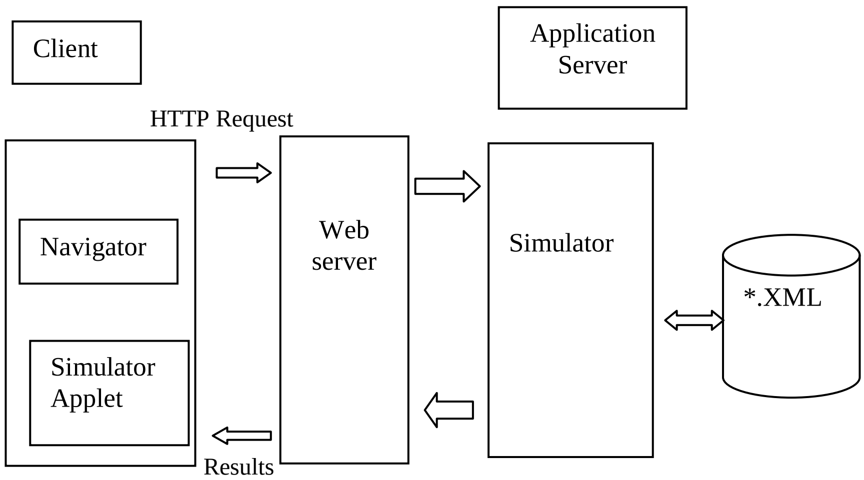

Fig. 4. Architectural design of simulation system.

4. The Simulator Model Builder

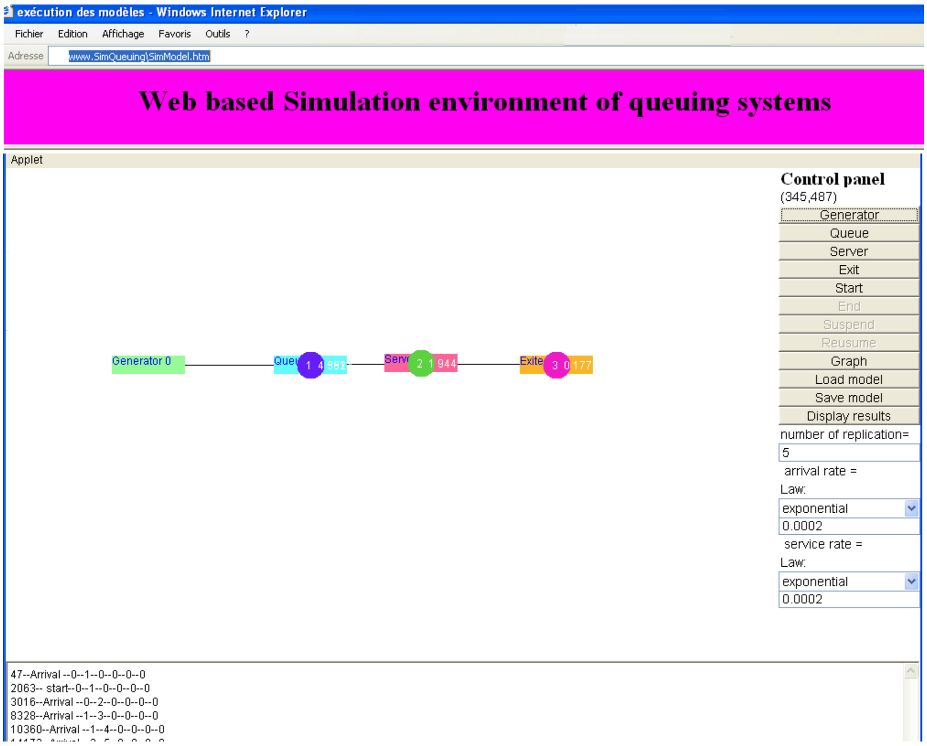

Fig. 5. Simulation applet view

The model builder allows construction of simulation model. The builder offers queuing system components for simulation model composition.

The use of a web-browser for model creation increases the accessibility of the simulation system by allowing any user with a browser access to model creation facilities. The users are simply allowed to run the simulation and view results.

4.1. Simulation Components

The builder provides a module of queue components to use in composition of queuing systems simulation models, the main components are the followings:

4.1.1. Generator Component

The component generator is a component that generates arrivals to be queued in the system queue. It creates clients according to a random inert-arrival time (uniform distribution, exponential …).

4.1.2. Queue

This component represents a component that queues items in a FCFS manner in the queuing system. The simulation system collects information on average waiting time in queue (Wq) and average number of clients in the queue (Lq).

4.1.3. Server Component

The server component is a component that provides a service station. The server served customers during a random time. When the service time has elapsed, the client is sent to the next component (Exit component) and the server moves into free state.

4.1.4. Exit Component

The last component is exit point which represents an exit from the queuing system. When a client reaches this component, it is removed from the queuing system and appropriates performance measures are updated.

5. Simulation’s run consideration

The simulation run is replicated many times using different random number seeds to obtain different independent observations for a performance measure. The average of these observations is taken as estimation of performance measures. But the difficulty resides in the determination of the number of replications to perform.

That is why, we propose two stopping criterions: attain the max number of replications or minimize a mean error or deviation on both performance measures (parameter introduced by the user). The following algorithm illustrates our approach:

```
1- Define two replication numbers max_rep and min_rep.
2- Initialize a deviation value on all performance
   measures.
3- While (i<min_rep) do
 begin
   Calculate the means of the replications i as
```

```
following:
meani = ∑xij /m ,  where m is the number of
observation of each replication.
 end
4- Calculate the overall estimate as follows:
     Estimate =∑ mean i/min _rep .
5- Calculate the deviation around the estimate by:
    dev = √(∑(mean i − Estimate)² /(n − 1)),    the sum is over
    i=1, 2, ..,n.
6- IF the average of all deviations is less than or
     equal to the desired deviation then stop simulation
     run.
    Else if min_rep <= max_rep
     Min_rep+=1;
     Return to 4.
```

Where $mean_i = \sum x_{ij} / m$, $Estimate = \sum mean_i / min\_rep$, and $dev = \sqrt{\sum (mean_i - Estimate)^2 / (n-1)}$.

The following performance measures are estimated:

- ρ = Server Utilization
- Wq = Average Waiting Time in Queue
- Lq = Average length of Queue
- W = Average Waiting Time in the System
- L = Average Number of clients in the System
- Ws = Average service time

6. Cases study

Experiment 1: In this case study, we consider a single server with a single queue (M/M/1 queuing system). It is assumed that the random variable corresponding to the inter-arrival times of clients from the component generator has exponential distribution with a rate of 0.045. The queue before the server follows the strategy first come, first served (FCFS). The server treats clients during a service time that is exponentially distributed with a rate of 0.05.

8 replications were performed. Figures show (6, 7, 8) the theoretical and experimental values of server's occupation, average length of queue component and average waiting time in the system

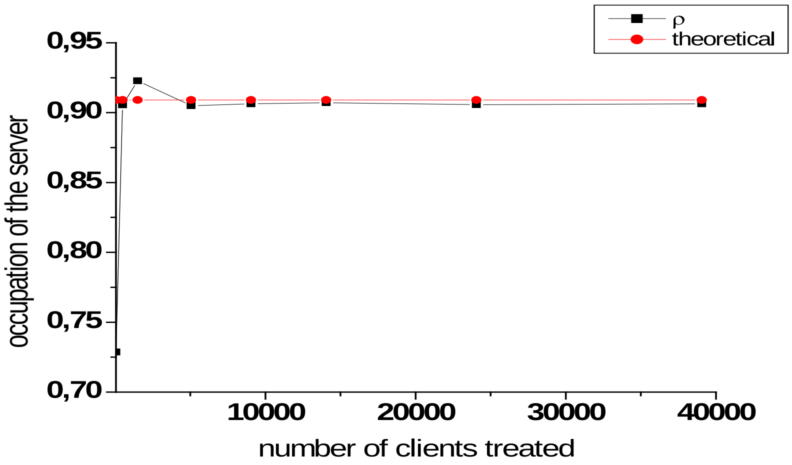

Fig. 6. Average server load

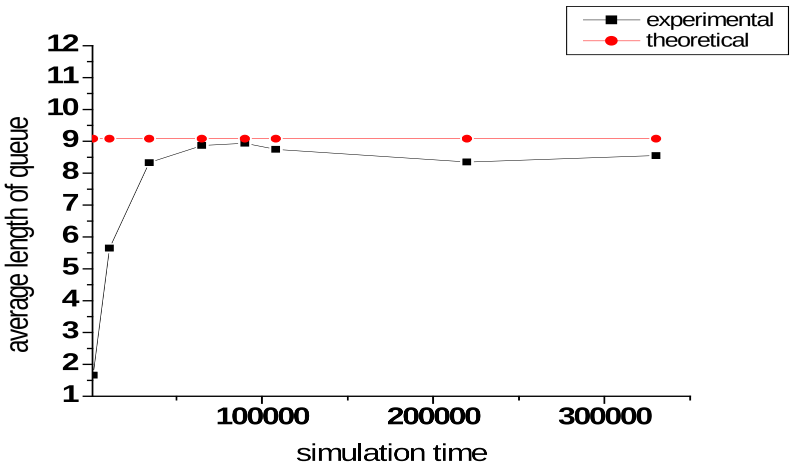

Fig. 7. Average length of queue

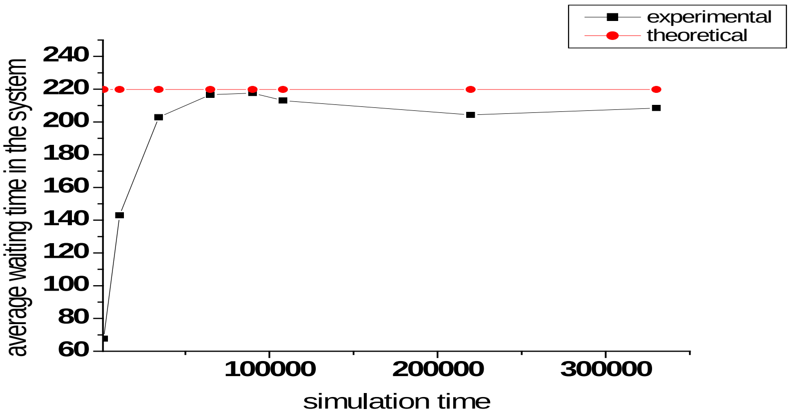

Fig. 8. Average number of clients in the system

The following table shows comparative test results for M/M/1 queuing system with 39050 clients across eight replications.

Table 1 Comparative test results for M/M/1 queuing system

Performance Measure	Analytical Solution	Estimated Value by Simulation	% Deviation
ρ	0,90910000	0,90633333	0,28239264
Wq	200 ,02200220	188,489548	5,65153704
Lq	9,09200011	8,5570032	5,75998049
W	220 ,02200220	208,4990017	5,132859375
L	9,989	9,08793034	9,02061928
Ws	20	20,0110569	0,05528428

The rates of deviation shown in the table above are reasonable based on the variance of each parameter, which increases our confidence about the accuracy of our system.

Experiment 2: In this case study, we consider a single queue and several servers (M/M/3 queuing system). It is assumed that the random variable corresponding to the inter-arrival times of clients from the component generator has exponential distribution with a rate of 0.1408. The queue before the server follows the strategy first come, first served (FCFS). The server treats clients during a service time that is exponentially distributed with a rate of 0.05.

Seven replications were performed. Figures show (9, 10, 11) the theoretical and experimental values of server's occupation, average length of queue component and average waiting time in the system

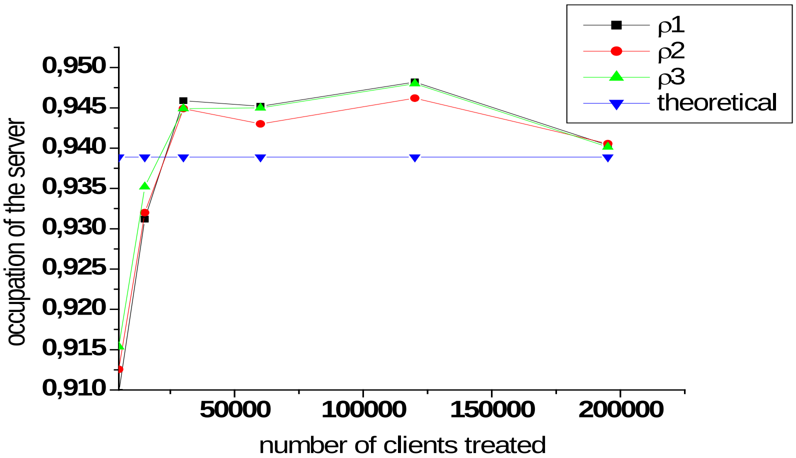

Fig. 9. Average server load

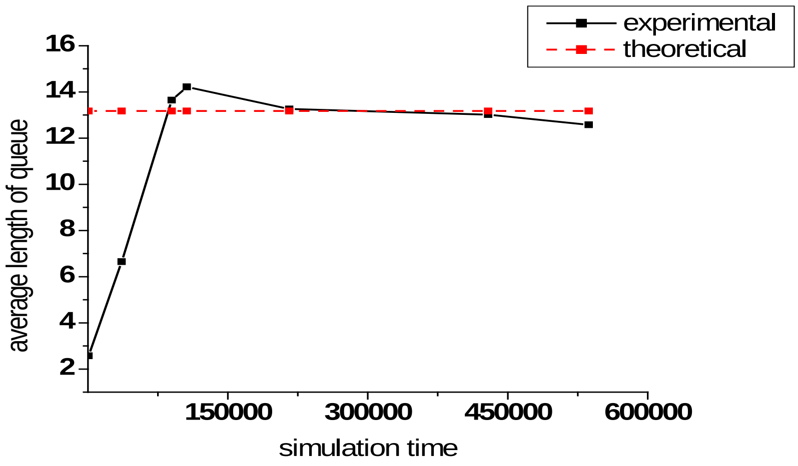

Fig. 10. Average length of queue

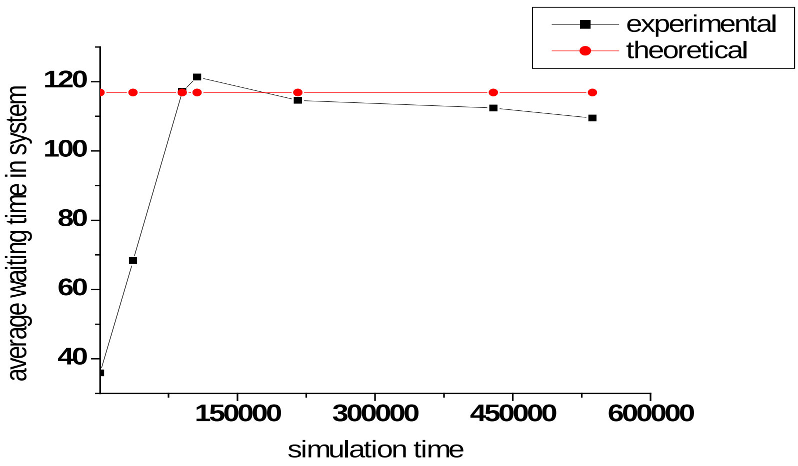

Fig. 11. Average waiting time in system

The following table shows comparative test results for M/M/3 queuing system with 195100 clients across seven replications.

Table 2 Comparative test results for M/M/3 queuing system

Performance Measure	Analytical Solution	Estimated Value by Simulation	% Deviation
ρ 1	0,93896667	0,94743569	0,9019515
ρ 2	0,93896667	0,94488733	0,63055103
ρ 3	0,93896667	0,94743093	0,90144456
Wq	96,8750358	89,4880712	7,625251
Lq	13,6443644	12,5799409	7,80119545
W	116,8750358	109,4934632	6,315782142
L	16,4612644	14,291343	13,1819848
Ws	20	20,005392	0,02696

The table above shows that after seven replications, the simulated values (W, L, Ws, Wq, ρ 1, ρ 2, ρ 3) are all very close to the theoretical values and asymptotically converge to these values, which increases our trust about the accuracy of the system.

7. Conclusion and perspectives:

We have considered in this paper the new concept of web-based simulation. The Web presents a new environment for the construction and execution of simulation tools. The new web technologies have the potential to change the manners in which simulation models are crated and executed. Our system QWebSim demonstrates some important concepts of Discrete Events Web Based Simulation, in particularly the simulation of M/M/n queuing systems. The experiments show that the measured Markov quantities are difficult to stabilize even after a large number of clients treated. As a future work, many enhancements can provide better performances and functionalities such as:

- Use of web services technology by introducing the SBS paradigm (Service Based Simulation).
- Enabling most complex queuing systems, characterized by the existence of several queues, several generators, etc ...
- Benefit of the parallel computing which allows distributing simulation across multiple processors.

References

1. Soliman, H., Alfantookh, A.: Web-Based Simulation: An Overview. Proceedings of the Gulf Internet 99 Symposium (GI'99), Dammam, Saudi Arabia, (1999) 96-99.
2. John A, Miller., Andrew F, Seila., Xuewei Xiang.: The JSIM web-based simulation environment. Future Generation Computer Systems, (2000) 19–133.
3. Benjamin Ping-Chang Yen.: Web-Based Simulation Tools, Proceeding of Industrial Engineering and Engineering Management, Hong Kong August (1997).
4. David, S., Myers.: An Extensible Component-Based Architecture for Web-Based Simulation Using Standards-Based Web Browsers. Master thesis, (2004).
5. Michael Pidd., Noelia Oses and Roger J., Brooks.: Component Based Simulation On the Web. Proceedings of the Winter Simulation Conference, 1999.
6. Asim El Sheikh., Abid Thyab Al Ajeeli., and Evon M., Abu-Taieh.: Simulation and Modeling: Current Technologies and Applications. IGI publishing 2008.
7. Charles, Marr., Christopher, Storey., William, E. Biles.: A JAVA-Based Simulation Manager for Web-Based Simulation. Proceedings of the Winter Simulation Conference, 2000.
8. Sven Meyer zu Eissen, Benno Stein.: Realization of Web-based simulation services. Computers in Industry (2006) 261-271.
9. John, A,. Miller and Paul, A,. Fishwick.: Research and commercial Opportunities in Web-Based Simulation. March 2001
10. Ricki, G,. Ingalls. : Introduction to Simulation. Proceedings of the Winter Simulation Conference. 2002.
11. Peter, Lorenz., Heiko, Dorwarth, Klaus-Christoph Ritter., Thomas, J,. Schriber.: Towards a Web Based Simulation Environment. Proceedings of the 1997 winter simulation conference, pp 1338-1344.