WEB SIMULATION FOR THE MANAGEMENT OF THE OPERATIONS AT AN INTERNATIONAL AIRPORT

Miguel Poyatos[1], José M. Pastor[1], Javier Otamendi[2], Andrés García[3], Pablo García[3]

[1]University of Castilla-La Mancha, Polytechnic School,
Campus Universitario s/n, 16071 Cuenca, España
[2]University Rey Juan Carlos, Faculty of Social Sciences
Paseo Artilleros, s/n, 28032 Madrid
[3]University of Castilla-La Mancha, Industrial Engineering School
Avda. Camilo José Cela, s/n, 13071 Ciudad Real, España

miguel.poyatos@uclm.es (Miguel Poyatos)

Abstract

A new airport in Spain is looking for tools that will help not only with the scheduling of resources but also with their real time control during normal operations. The tool developed provides a graphical interface using both Gantt charts that will help visualize the future programs obtained by simulation and a real-time visualization of the online situation at the parking positions. The graphical interface runs a simulation model in multiple web browsers that are synchronized in the same database for the input and output data, using a Java server. The collaborative system aims at the improvement in the management of the resources allowing users to dynamically modify the Gantt chart bars as delays and other disturbances occur and immediately assess the impact in the future programs and make better decision accordingly.

Keywords: web-based simulation, resources allocation, visualization, JAVA.

Presenting Author's biography

Miguel Poyatos was born in Cuenca, Spain. He received the Telecommunications Technical Engineer Master, from the University of Castilla-La Mancha (UCLM) in 2009. In 2009 he begins his doctoral research in simulation systems. His research interests include RF Identification and tracking, factory automation, intelligent manufacturing systems and systems simulation and optimization. He works in Autolog group since 2007, where he developed his interests in J2EE applications orientated to analysis of resource allocation for incoming flight and RFID luggage tracking for Central Airport at Ciudad Real. His e-mail address is: miguel.poyatos@uclm.es and the Web-page of the Autolog association is: http://autolog.uclm.es.

1 Introduction

Nowadays Spain is undergoing important changes in the air service industry. Several new private airports are being built all over the country to relief part of the traffic in the main airports, as you can see in Fig. 1, and to attract low-cost companies. Ciudad Real is a city located 200 km south of Madrid and its new airport started operations during the second quarter of 2008.

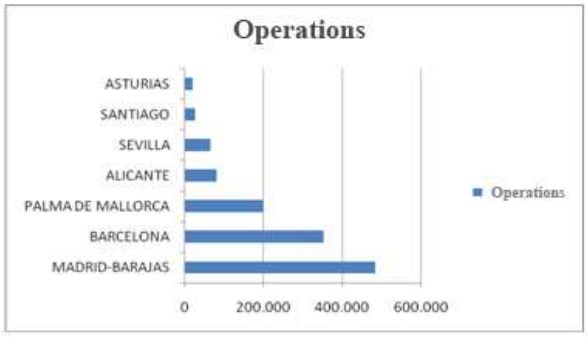

Fig. 1 Operations AENA.

The airport's management is encouraging the development of decision-support tools [1] that will help in the improvement of the runaway operations with the aim of optimally allocating resources, providing the management with the simulation [2] and visualization tools they need to properly schedule the resources; not only for long-term scheduling (shifts, requirement levels), but also for short term (weekly or daily assignments) and even on-line control.

The airport's operations include all the tasks that will be carried at the airplane parking positions at the airport. The loading of passengers includes the escorting of the passengers from the gate to the stairs of the airplane, while the unloading includes the walk from the airplane to the arriving lounge. The resources that are available are operators that escort either passengers or crew and the stairs to enter or exit the aircraft.

The management of the airport desires an easy-to-use tool with the following characteristics:

- Capability to draw Gantt charts with the resources and flights information stored in the database and to update them in real time if disturbances in the forecasts occur.
- Capability to statically visualize the situation at the parking positions at a specific moment in time to show the operations that are being carried out or prepared.
- Capability to show results of the utilization of the different resources and the timeliness or the service level that the handling units are able to provide to the carriers.

Therefore, the aim is to build an ad-hoc tool so that the staffing and the dynamic assignment of resources to tasks are conveniently performed in the required allotted time.

2 State of the art

Web-based simulation [3] provides a multiuser environment, which is indispensable for collaborative work.

Multiple protocols are available to communicate clients among them with a single server: sockets, CGI post and get methods, RMI, Corba, etc.

Secure Socket Layer allows secured communications easily protecting the communication against eavesdropping techniques.

Java platform includes a powerful language for client simulation [4], applets, which allows for the possibility of running simulations and visualizations from different computers into the Web-browser [5], with good graphics capabilities. Also Java integrates the connection with databases via concurrency techniques and SSL encryption protocol in the default library.

3 Description of the tool

The tool that has been developed has three main modules, as shown in Fig. 2: a database which stores flights and resource data, a web server controlling concurrent access and multiple simulations [6] and visualizations running in the web clients.

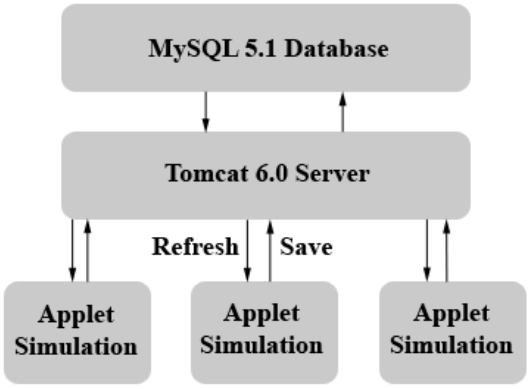

Fig. 2 Web multiple simulation system.

3.1 Database

The database incorporates the data related to the resources: name, description, number of available resources, shifts, etc. and the most relevant data about flights: company, flight number, arrival time/date, departure time/date.

This data feeds simulation algorithm, therefore, flights and resources tables are sending to the Applets

thought Java server. The data is stored back before the simulation runs in the clients.

3.2 Server

Server manages the applet's input and output data using an SSL socket [7], sending input data from the JDBC database connection and storing results into the database again. In this process the server program takes care of the synchronization of threads coming from the applet simulations [8].

Only one thread can store information at the same time, during this routine the other threads can't read any data.

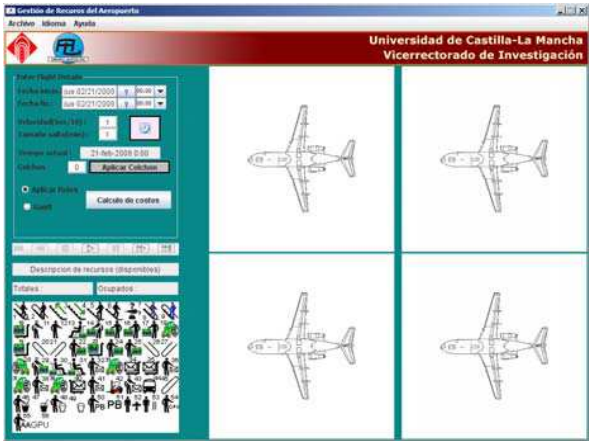

Fig. 3 Desktop version of the simulation tool.

SSL socket is used instead other method to maintenance the compatibility with old versions based on a traditional desktop application.

These previous versions, as seen in Fig. 3, use the database connecting and sending SQL queries directly from a JApplication running in the Java Virtual Machine.

When the multiuser tool was designed, the synchronization was needed to ensure the full tables integrity. Therefore, server was designed to receive the incoming threads from desktop applications. For this reason, later web versions based on Applets connect with a server application using sockets to read and store the simulation data.

On the other hand, the Applet class from latest versions can be run in an Applet viewer as local application without a web browser.

3.3 Simulation algorithm

Applets provide a graphical interface to show the simulation results and implement the simulation algorithm.

The input data of the algorithm is:

Flights:

- Flight number.
- Arrival Date Time.
- Departure Date Time.
- Passengers.

Resources:

- Code.
- Resource Name.
- Quantity available.
- Cost per hour per unit.
- Relative start time.
- Relative finish time.
- Relative to ADT or DDT: Boolean indicating if the resource's start and end time are relative to ADT or DDT.

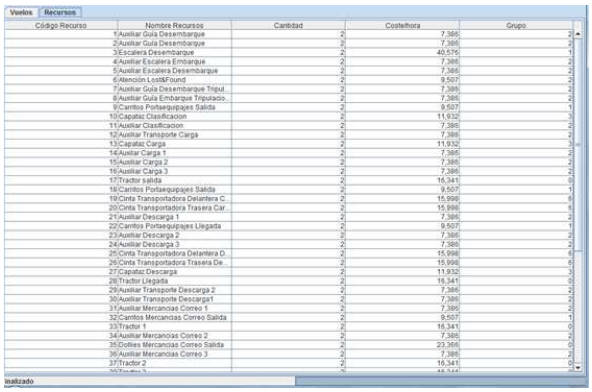

Fig. 4 Input costs and number of resources available.

For scheduling purposes, the quantity of the resources and the unit cost are the critical information, as seen in Fig. 4.

Output data:

- Max simultaneously access. It shows the number of calls that have simultaneously been placed for each of the resource time, if unlimited resources were available.
- Cost per resource.
- Cost per flight per passenger.
- Real scheduling of the resources including delays.
- Total costs, calculated by adding the penalty costs associated with delays to the traditional direct costs.
- Occupation level, which might be close to a given target, usually around 70-80%.
- Other critical ratios.

The simulation algorithm, as you can see in Fig. 5, is a backtracking process that delays the resources when it's impossible the assignation because of there isn't enough units. On each delay the planning changes

modifying the availability for the rest of the simulation time.

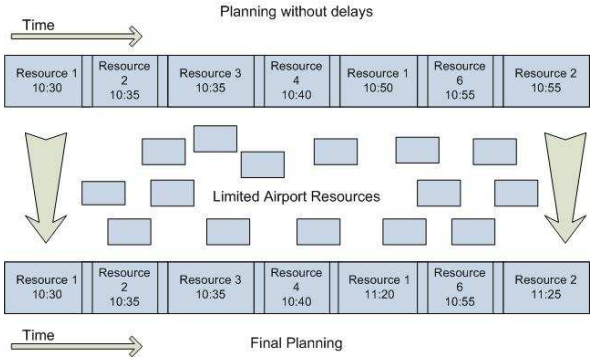

Fig. 5 Simulation algorithm

3.4 Applet visualization

The Applet interface, Fig. 6, allows the user to show the results on real time visualization of the parking positions and on Gantt chart visualization as well as details of the executed simulation.

When the applet loads initially, it shows an empty interface where the user can "Refresh", "Run" or "Save". By clicking the "Refresh" button, users can load, if the concurrency threads make it possible, the resources and flights data. With the "Run" button, users can start the simulation algorithm. The simulation uses the start and the finish date and time set in the JDatePickers located at the left of the "Refresh", "Run" and "Save" buttons.

When it's finished, the visualizations panels show the Gantt chart, see Fig. 7 in the next page, a real time status and runtime statistics. Finally, by clicking on the "Save" button, users can store the results back to the database.

3.4.1 Real-time visualization

The real-time visualization, as you can see in Fig.6, shows the state of the landside operations, it shows four parking positions. These positions can have three states:

- Empty: The icon of the airplane is white and there isn't any flight number or company logo, the parking position is free, no resource is attending any plane at this position.
- Reserved: Airplane's icon is white and it shows the company logo and the flight number, the parking position is occupied because some resources are involved in a process for an incoming flight.
- Busy: Airplane's icon is black and it has got a logo and the flight number. The plane is at the parking position and is being attended.

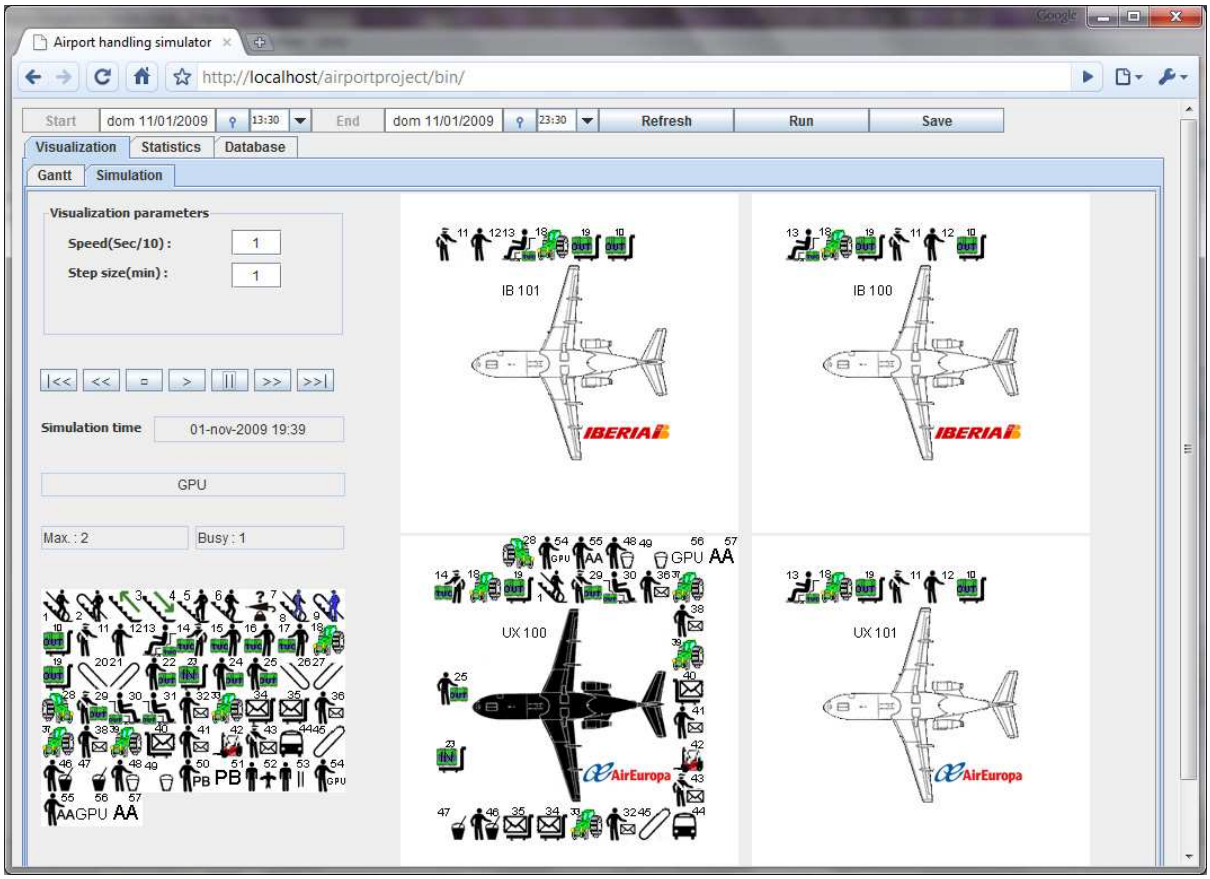

Fig. 6 Real-time visualization and simulation controls.

The resources appear and disappear depending on the results of the simulation and the time selected in the visualization. At this point, users can accelerate or stop the visualization of the simulation results using the left side controls.

The controls include, from top to bottom:

- Visualization parameters:

  Speed (sec/10): Number of tenths of a second that takes a minute in the simulation.

  Step size: Number of minutes that runs in the simulation for each Speed step.

- Play controls:

  Go to start simulation time.

  Rewind.

  Stop.

  Play.

  Pause.

  Forward.

  Go to the finish time.

- Simulation time of the visualization.
- Max number of resources available in the airport.
- Number of busy resources at the current simulation time.
- Resources panel, showing all resources.

The separated simulation and visualization allows the user to perform unusual activities in a simulation like going backwards in the simulation time. The visualization shows the data related to a simulation time not by calculating the simulation from the beginning, it calculates whole output data and shows the selected result in the time.

A JTimer class handles simulation time in the visualization. That class sets resource's state in the JPanels extended classes that implements methods to set the state and description of each resource and airplane. Also, when the mouse is over a resource, users can see the number of resources busy at the moment and total resources available.

The related interface is made using Swing library, which provides great graphics capabilities, as double-buffered panels (JPanel) which is for displaying images on time, JTable to show data in a sortable grid, or a common rich user interface controls as JButton, JTabPane, JDatePicker, etc.

3.4.2 Gantt chart visualization

The Gantt chart visualization shows the planning for multiple flights being attended over the time. The flights buttons surround the resources bars, where each bar use a color code:

- Gray: The resource isn't available on its start time.
- Green: The resource is attending the plane in the original planning time.
- Red: The resource still attending at the airplane after its original finish time.

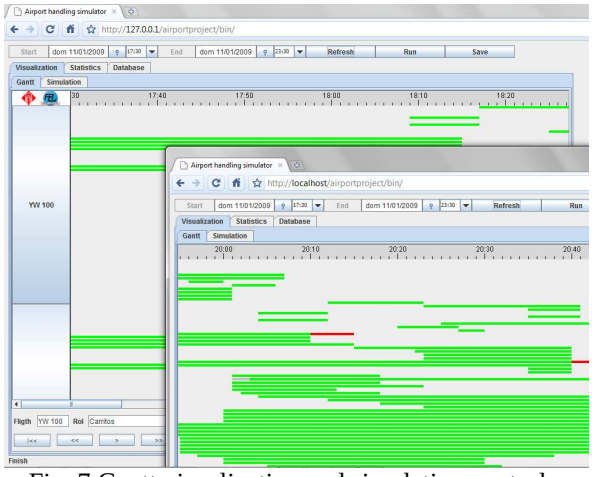

Fig. 7 Gantt visualization and simulation controls.

When the simulation is finished, users can modify the resources schedule dragging the start or the end point in the bar of the Gantt chart and running again the simulation. It modifies the scheduling of this resource changing the start or the end time changing the input data and simulating the new result for this planning.

Using this method user can improve the planning dimensioning the number of resources necessary and optimizing the resource's schedule.

3.4.3 Statistics

The airport management staff has another tab in the web application to control the output by data tables including, by tab order:

- Real scheduling after simulation.
- Max number of simultaneously access.
- Cost per resource.
- Cost per flight.
- Total cost including delays and other critical ratios.

Figure 8, in the next page, shows the output screen that summarizes the calculation of the delays per resource, including:

- Flight number.
- Resource's name.
- Start time.
- Finish time.

- Real start time.
- Real finish time.
- Delay the and also the time at which the delay happened.

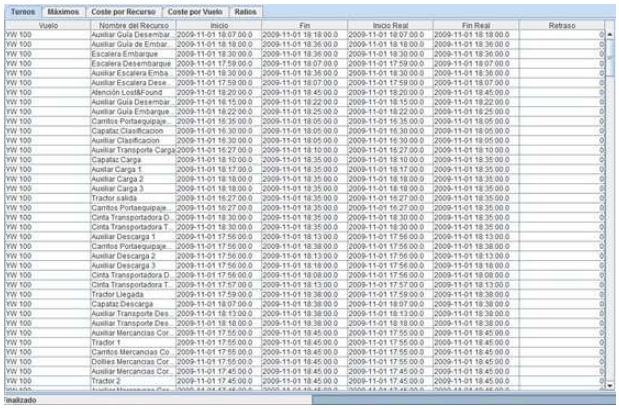

Fig. 8 Output resource planning

4 Conclusion

A collaborative dynamic management tool has been developed in Java which allows for the execution and visualization of simulation models with the aim of optimizing the process of assigning and scheduling the resources, both for long-range planning and for online decision making.

The tool can be used in a collaborative environment thanks to the multiple simulations in the clients synchronized in the same database.

5 References

[1] A. G. Bruzzone and P. Giribone. Decision-Support Systems and Simulation for Logistics: Moving Forward for a Distributed, Real-Time, Interactive Simulation Environment. 31st Annual Simulation Symposium, Boston, Massachusetts, April 1998.

[2] W. Krug, Th. Wiedemann, J. Liebelt, and B. Baumbach. Simulation and optimization in manufacturing, organization and logistics. *Proceedings 14th European Simulation Symposium*, 2002.

[3] J. Byrne, C. Heavey and P. Byrne. A review of web-based simulation and supporting tools, Simulation Modelling Practice and Theory, doi: 10.1016/j.simpat.2009.09.013, 2009

[4] R. A. Kilgore, K. J. Healy and G. B. Kleindorfer. The future of Java-based simulation. *30th Winter Simulation Conference*, Washington D.C., United States, 1998.

[5] P. Lorenz, H. Dorwarth, K. Ritter and T. J. Schriber. Towards a web based simulation environment. *29th Winter Simulation Conference*, Atlanta, Georgia, United States, 1997.

[6] M. Dahlin, A. Brooke, M. Narasimhan, and B. Porter. Data Synchronization for Distributed Simulations. *2001 European Simulation Interoperability Workshop*, 2001.

[7] J. Guitart, D. Carrera, and V. Beltran. Session-Based Adaptive Overload Control for Secure Dynamic Web Applications. International Conference on Parallel Processing, Univ. of Oslo, Norway, June 14-17, 2005

[8] R. M. Fujimoto. Parallel and Distributed Simulation Systems. Jhon Willey & Sons, Inc., New York, USA, 2000.