Airport Management Database in a Simulation Environment

Antonello Pasquarelli[1], Tomáš Hrúz[2]

[1] ALENIA MARCONI Systems, via Tiburtina km 12.400, 00131 Rome – Italy
Email: apasquarelli@lti.alenia.it
[2] Slovak Academy of Science, Mathematical Institute, Dúbravská 9, P.O. Box 56, 84000 Bratislava – Slovakia
Email: tomas@ifi.savba.sk

Abstract. This paper presents the SEEDS simulation environment for the evaluation of distributed traffic control systems. The description starts with a general overview of the simulator, completely developed in C++ according to the CORBA standard, and then focuses on the Airport Management Database (AMDB) implemented in the prototype as an add-on module. The emphasis of the AMDB is on the wide area network operation, and this leaded to the architecture centered on the Java system.
The paper shows the AMDB architecture, how it has been integrated in the core simulator and the interaction with it. The evaluation shows the ability of co-operation between Java and CORBA based C++ subsystems and the scalability of the whole system.

1. SEEDS Overview

SEEDS[1] (Simulation Environment for the Evaluation of Distributed traffic control Systems) is a distributed HPCN simulation environment composed of powerful workstations connected in a local network and it is targeted to the evaluation of Advanced Surface Movement Guidance and Control Systems (A-SMGCS). The simulation environment allows the definition and evaluation of technologies and performances needed to implement new functions and procedures of A-SMGCS, to mould new roles in the airport, to introduce new automatic tools and interfaces, to support A-SMGCS operator decisions.

The SEEDS consortium is composed of Alenia Marconi Systems (I), as coordinator, Sogitec (F), Artec (B), as industrial partners, University of Siena (I), LRR-TUM at Technische Universität München (D), Slovak Academy of Science (SK), as associated partners, and Sicta (I) as final user-partner. An European User Group, composed by the flight assistance administrations Sogel (L) and SEA (I), participated to all the phases of the project.

The software architecture of the simulator, defined using the Unified Modeling Language (UML) notation, is based on CORBA (Common Object Request Broker

[1] SEEDS (European Project Number 22691) has been partially funded by EC DGIII in the area of HPCN. SEEDS started the first January 1997 and finished the 30th September 1999.

Architecture) as communication middleware; the DIS (Distributed Interactive Simulation) protocol has been used for the image generation and distribution. This choice assures the scalability of the system, allowing the mapping of the different objects on heterogeneous workstations or PCs. Techniques for load balancing are used to reduce computing power, data rate and latency on the network. In particular centralized versus distributed architecture for traffic and image generation have been analyzed, and the distributed solution was adopted in SEEDS for performance reasons. Subscription and notification mechanisms are used to synchronize processes in the distributed environment.

The simulation environment is composed of commercial off-the-shelf components and of some proprietary software modules, and it is open to be connected to other ATM (Air Traffic Management) simulators. The main software modules of the SEEDS architecture, completely developed in C++ and reported in Fig.1, are: Scenario Generation and 2D-3D Visualization; Sensor Models; Airport Database; Surveillance, Control, Guidance and Planning Modules, Controller, Pilot and Driver models, Administration Station modules.

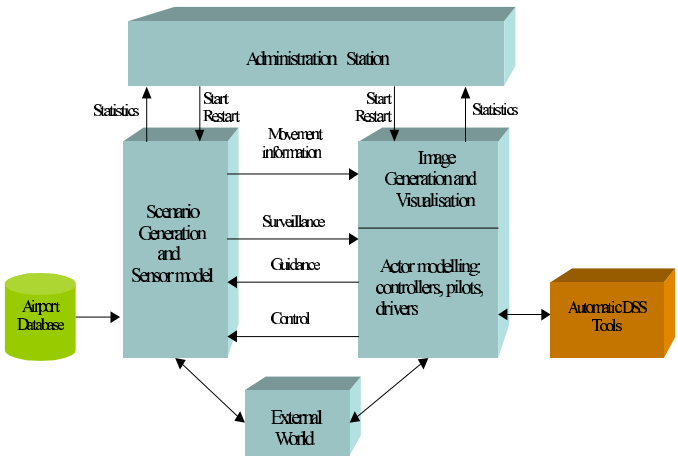

Fig. 1. SEEDS Simulator Architecture

The scenario generator is in charge to animate the scene according to the stimuli coming from the actors of the simulation and from the External World. The actors can be simulated (software processes) or real (human beings). The 3D visualization reproduces the scene as seen by the actors' eye (eye model); the 2D visualization reproduces the scene as seen by the sensors present in the airport (ASDE radar, GPS, DGPS, Magnetic/Dynamic sensor error models). The controllers have a set of DSS (Decision Support Tools) which help them to plan the aircraft surface movements. The other functions of A-SMGCS (Surveillance, Control and Guidance) are also implemented. An Administration Station is responsible to configure, start-up, stop, restart the simulation, and it collects application level and system level statistics.

2. Airport Management Database Architecture

The purpose of the Airport Management Database (AMDB) module is to add a database system to the core simulator, which implements the main aspects of a real airport database model, such as: Meteorological situation, Flight data list, Initial Climb Procedures, Instrument Approach Chart, Standard Instrumental Departure (SID) procedures, Standard Approach Route (STAR). For each procedure, the database contains textual description and visual information.

In this manner, all information are available to the SEEDS core simulator, through a CORBA interface, and can be updated by the airport agencies (Meteo Services, Central Flight Management Unit, etc.).

The main aspect of the AMDB module design is a wide area network (WAN) operation emphasis, which leaded to the architecture centered on the Java system. This solution provides the possibility of WAN access to the external aspects of the airport simulations.

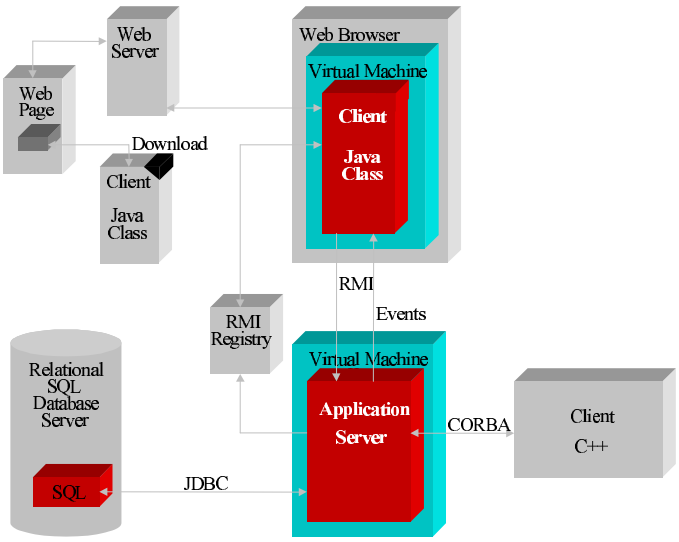

Fig. 2. Airport Management Database Architecture

In order to achieve an high level of flexibility, the AMDB architecture is designed as a three-tier, or three level, system whose main modules are the SQL database server, the application server and several clients (See Fig. 2).

Considering that the AMDB is a complex system containing various hardware and software platforms, open software tools where adopted for the development of this module.

2.1. Database

The data repository has been modeled and implemented using a relational database subsystem able to handle large data sets, which occur if the airport simulator developed during the SEEDS project were used for simulation of traffic comparable to real airports. In our case the database server is the PostgreSQL system.

2.2. Client

The AMDB system has two main kind of clients: the Java clients of the AMDB module and the C++ clients of the core simulator. The Java clients can handle the meteorological information and they also offer support for chart maps processing and administration.

The meteorological client allows to change the meteorological situation on the airport and to store all changes and operations to relational database for further on-line or off-line processing.

The flight data list client is used for editing, processing and store the flight information into the database. AMDB system contains also the database tables to describe the airport structures, the navigation charts, the aircraft description etc.

2.3. Application Server

The application server is a Java application which manages the data contained in the database. It receives the requests coming from the set of currently running clients and it is the only agent allowed to mediate them towards the database, in order to assure an high data consistency. It exhibits also a CORBA interface in order to exchange data with the C++ modules. The event notification has been implemented in order to have a correct situation awareness in all kind of clients. The application server architecture is shown in Fig. 3.

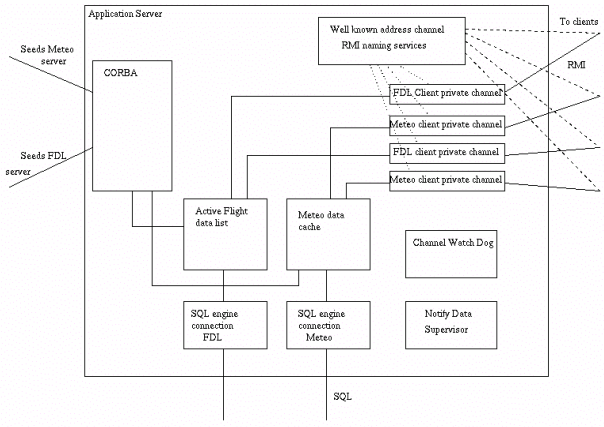

Fig. 3. The Application Server Architecture

The application server manages the connections with the clients using a time-out mechanism. Considering that the AMDB module is designed with a WAN computing oriented structure, security aspects have also been taken into account.

3. Interaction with the Core SEEDS Environment

The core of the SEEDS simulator is written in C++ and uses the CORBA standard to communicate between different system modules. The AMDB module is entirely written in Java. This situation has been an excellent occasion to test the ability of co-operation between Java and CORBA based C++ subsystems. The introduction of new platform into the project has stressed the necessity to stick very closely to CORBA standards otherwise the reusability of the software modules could be decreased. In the prototype the AMDB is hosted on a Digital Alpha station. The Apache web server has been used to manage the interaction with the Java clients.

3.1. Java Clients

The code of clients is stored in signed archives. Clients are downloaded from the web server to the browsers where they are run on browser virtual machine. In the prototype configuration the browsers used are Netscape browser version 4.x or higher and Microsoft Explorer version 4.x or higher together with Java Plug-in from Sun. The Plug-in implements the features of Java 1.1 system, which are not implemented by the major browsers.

3.2. C++ Clients

The application server is connected to the core SEEDS system through the CORBA interface. The mechanism of event notification has been implemented between the application server and the C++ clients using the existent subscription methods.

4. Conclusions

This paper has presented the SEEDS simulation environment with particular attention to Airport Management Database, which is an add-on module of the core simulator. The chosen approach using Java to develop the module has been shown appropriate. In fact it has been a good occasion to see whether the speed of recent Java Virtual Machine (JVM) implementations are able to cope with tasks arising in such complex applications as is an airport simulator. The results of these tests have been very encouraging and they provide a good basis to extend this approach in future. Furthermore, the use of Java allowed to release any special requirement on the client machines, other than the presence of a standard web browser.

References

1. S. Bottalico. SEEDS (Simulation Environment for the Evaluation of Distributed Traffic Control Systems). A Simulation Prototype of A-SMGCS. In *Proceedings of the Internationa Symposium on Advanced Surface Movement Guidance and Control System.* Stuttgart, Germany, 21-24 June 1999.
2. Lewis, G., Barber, S., Siegel, E., Programming with Java IDL, John Wiley & Sons, 1998.
3. Flanagan, D., Java in a Nutshell, O'Reilly, 1997.
4. Official SEEDS Web page. http://www.lti.alenia.it/EP/seeds.html