CATS 4.0
A component-based platform for simulation

Stefan Lundmark
Mandator Simulation & Training
Rusthållsgatan 21
S-253 61 Helsingborg
Sweden
Tel: +46 42 19 82 00
Fax: +46 42 14 29 63
E-mail: stefan.lundmark@mandator.se

Abstract

1. BACKGROUND

The development of simulators for training the Command and Control function at different command levels implies difficulties since every customer has requirements specific to their particular interpretation of how a simulator should be configured. The simulators are supposed to be used in widely different environments depending on the type of situation and role for which people are being trained for. A number of computers could be connected through a network, just like in any office network, or it could be a complete system configured for training. Examples of such systems could be the training system built into an air-traffic control tower to train air-traffic controllers or a simulation system built into a tank in order to train the crew under realistic conditions.

2. THE PROBLEM

The problem is the design of the architecture for the simulator platform. It must be possible to adapt an existing system with a minimum of changes in order to meet a specific customer requirement but still be able to benefit from all the experience and know-how inherent in an existing product or simulator platform. Also, the customer often wants to determine, in part or completely, the design of the user interface. Certain parts of the system need to be changed in order to provide training corresponding to the real situation. In addition, communication need to be established with external equipment which is used in a real situation such as radar systems, telephones and computer systems using different protocols..

3. ARCHITECTURE

3.1. Basic principle

The simulator platform is divided into components individually performing a specific task. It is possible for the components to execute using a computer at random within a network as well as communicating with each other via open interfaces without knowing where the other

components are situated. This also implies the duplication and distribution of databases and other data sources to meet the requirement for real-time situations in the simulator.
Mandator has developed a system architecture for simulators based on using a mix of client/server system, component-based software development and dynamically down-loadable plug-in modules within a distributed system which changes or expands the functionality of the simulator by configuration.

3.2. Game engine

The simulator is built-up around a game engine which can handle all types of scenarios for Command and Control training and simulations. The function of the game engine is to drive the simulation forward and to synchronise events occurring during the simulation. The game engine does that by offering a number of open interfaces in order to extend its functionality. This is realised by using dynamically connected plug-in modules offering different types of services, to the game engine.

3.3. Plug-in module

A plug-in module is a binary module, application or DLL, offering one or several services via an interface and which can be dynamically connected to a game engine. It is divided into two parts, a server plug-in module containing all activity-codes for the service being offered and a client plug-in module containing user-interface to control and operate the module. Also, the interface between the server and the client are open providing great advantages for a customer who is technically well-versed in the use of computers.
A pure client/server relationship exists between the client plug-in module and the server plug-in module concerning the number of clients/servers, different rights of access depending on the status of the user, data updates, etc.

3.4. Events bus

All server modules are interconnected in a bus system into which all events signifying a change of the scenario are reported and the time recorded relative to the game clock. By recording all events stored in the events bus it is possible after the termination of an exercise or simulation to replay the entire scenario for evaluation. It is also possible to stop the replay midway and continue the simulation under manual control using different decisions from those used in the earlier simulation. This is of particular use in the training of various complex situations.
In order to simplify for both the trainee and the exercise directing staff the user interface contains only the functionality which the trainee can control. The client plug-in modules are for example down-loaded automatically into the client thus expanding the menues and possibilities to control the scenario.