Software technologies for the effortless development of geographically distributed simulators

Daniele Gianni
gianni@info.uniroma2.it

Dept. of Computer Science, Faculty of Engineering
University of Rome Tor Vergata
April 2007

Abstract

Distributed simulators are increasingly being used for their intrinsic advantages in terms of reusability, fault tolerance, performance, and geographic distribution. The development of a distributed simulator, however, requires significant investments in effort if compared to the development of traditional local simulators.

Although the most adopted distributed simulation standard, IEEE 1516 HLA (High Level Architecture), has introduced considerable improvements with respect to previous standards, it still suffers from shortcomings such as (1) lack of interoperability among different IEEE-compliant implementations, (2) absence of support in the development of individual federates and (3) reusability of federates only, with exclusion of smaller components.

This dissertation introduces two independent technologies to face such problems. The first technology is based on a CORBA-HLA architecture that overcomes limitation (1); the second one is based on SimArch, a layered architecture that overcomes limitations (2) and (3).

The CORBA-HLA architecture decouples federates from the specific HLA implementation so that federates can be effortlessly run on top of any HLA implementation that exposes services through an IEEE-compliant IDL interface. Preliminary CORBA-HLA performance measurements over national and intercontinental WANs are also presented.

The SimArch architecture introduces a set of abstraction layers on top of any distributed simulation infrastructure (e.g. HLA) in order to decouple simulation components from local/distributed execution environments and, furthermore, from the distributed platform in use.

A Java based implementation of SimArch is introduced for each layer. The first layer gives a Discrete Event Simulation (DES) abstraction on top of HLA. The second layer, named SimJ, an execution container for the DES simulation components. The third layer finally implements an high level simulation language on top of SimJ. In the specific case an extended queueing network language is implemented, called jEQN.

Supervisors:
Prof. Giuseppe Iazeolla, iazeolla@info.uniroma2.it
Prof. Andrea D’Ambrogio, dambro@info.uniroma2.it