SimWare-Kernel
Real Time Communication System for Simulation
(Paper by: Bruno Calvo, Ignacio Seisdedos)
http://www.nexteleng.com/

- Introduction
- Concepts
- Standards
- Implementation
- Interoperability scenarios
- Applications
Introduction

SimWare is a simulation framework based on:

- Distributed simulation architecture (eSim)
- Mission support architecture (MSOE)

Concepts:

- Publish / subscribe paradigm
- Use of a data model
Concepts

Publish-Subscribe

Client-Server vs Publish-Subscribe

How is data identified?
- Type: data types defined in data model (temp, pressure, …)
- Topic: data sent by an entity (tempX, pressureX)

Pressure X Temp X
Pressure Y Temp Y

Console 1
Console 2
Console 3

/Sensor/Temp*
/Sensor/*X
/Sensor/Pressure*
**Concepts**

**Data Model**

Objective: gather in a single document all data types used by the system

- Objects and interactions
- Attributes and parameters
- Transportation
- Frequency
- Entities

### PROTOSIM

<table>
<thead>
<tr>
<th>Objeto</th>
<th>Atributos</th>
<th>Rango / Unidades</th>
<th>Modo transporte</th>
<th>Frec (Hz)</th>
<th>Entidad publicadora</th>
<th>Entidades suscriptora</th>
</tr>
</thead>
<tbody>
<tr>
<td>CondAtmosferaEnPosicionAeronave</td>
<td>Variacion Magnetica</td>
<td>[-pi, pi] rad</td>
<td>DatSimSec</td>
<td>50</td>
<td>EAMB</td>
<td>EMOV</td>
</tr>
<tr>
<td>CondAtmosferaNivelMar</td>
<td>Presion Estatica</td>
<td>Pa</td>
<td>DatSimSec</td>
<td>50</td>
<td>EAMB</td>
<td>AERO</td>
</tr>
<tr>
<td>CondAtmosferaNivelMar</td>
<td>Temperatura Estatica</td>
<td>K</td>
<td>DatSimSec</td>
<td>50</td>
<td>EAMB</td>
<td>AERO</td>
</tr>
<tr>
<td>VientoEnPosicionAeronave</td>
<td>Direccion</td>
<td>[0, 2pi] rad</td>
<td>DatSimSec</td>
<td>50</td>
<td>EAMB</td>
<td>PI, VIS</td>
</tr>
<tr>
<td>VientoEnPosicionAeronave</td>
<td>Velocidad</td>
<td>m/s</td>
<td>DatSimSec</td>
<td>50</td>
<td>EAMB</td>
<td>PI, VIS</td>
</tr>
<tr>
<td>VientoEnPosicionAeronave</td>
<td>Vector Velocidad Ejes Horiz Local</td>
<td>m/s</td>
<td>DatSimSec</td>
<td>50</td>
<td>EAMB</td>
<td>AERO</td>
</tr>
<tr>
<td>RadioNavComSintonizadas</td>
<td>Receptor Balizas</td>
<td>ver Struct</td>
<td>DatSimSec</td>
<td>50</td>
<td>ERAD</td>
<td>VORI</td>
</tr>
<tr>
<td>RadioNavComSintonizadas</td>
<td>Receptor ADF</td>
<td>ver Struct</td>
<td>DatSimSec</td>
<td>50</td>
<td>ERAD</td>
<td>SADF</td>
</tr>
</tbody>
</table>
Standards

DDS

DDS: Data Distribution Service for Real-Time Systems

<table>
<thead>
<tr>
<th></th>
<th>Adopted</th>
<th>Finalized</th>
<th>Revised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2003</td>
<td>2004</td>
<td>2005</td>
</tr>
</tbody>
</table>

SimWare-Kernel

Bruno Calvo - bcalvo@nexteleng.es
Standards
HLA

Standards
HLA

HLA 1.3  IEEE 1516-2000 (1, 2, 3)  SISO DLC  IEEE 1516-2006 HLA Evolved


RTI Ambassador  Federate Ambassador  HLA

Bruno Calvo - bcalvo@nexteleng.es
Standards
Comparison DDS-HLA

<table>
<thead>
<tr>
<th>Similarities</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLA</td>
</tr>
<tr>
<td>DDS</td>
</tr>
<tr>
<td>Publish/subscribe data communication</td>
</tr>
<tr>
<td>User specified data types described in external file (FOM - IDL)</td>
</tr>
<tr>
<td>Multiple federations (DDS domains) can co-exist</td>
</tr>
<tr>
<td>Standard defined API</td>
</tr>
<tr>
<td>Content based filtering</td>
</tr>
<tr>
<td>Suitable for simulation applications</td>
</tr>
<tr>
<td>Tools for data specification, application development, debugging, monitoring</td>
</tr>
</tbody>
</table>

Bruno Calvo - bcalvo@nexteleng.es
## Comparison DDS-HLA

<table>
<thead>
<tr>
<th>Differences</th>
<th>HLA</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designed for simulation</td>
<td>Designed for real-time</td>
<td></td>
</tr>
<tr>
<td>Hierarchical object models</td>
<td>DDS supports hierarchical object graphs (DLRL)</td>
<td></td>
</tr>
<tr>
<td>Persistent objects</td>
<td>DDS has “durability” on topics</td>
<td></td>
</tr>
<tr>
<td>No implied QoS semantics</td>
<td>Rich QoS semantics</td>
<td></td>
</tr>
<tr>
<td>Logical time management</td>
<td>No implied time management semantics</td>
<td></td>
</tr>
<tr>
<td>No standard wire protocol</td>
<td>Open standard RTPS wire protocol</td>
<td></td>
</tr>
</tbody>
</table>
Implementation

Motivation for a DDS-based HLA

The similarities between DDS and HLA make DDS fit perfectly as the low-level layer of an enhanced version of HLA/RTI.

Benefits of this implementation

- Enhanced performance, which meets new interoperability requirements (optimized for RealTime, available on different RT OS, ...)
- Direct connection DDS/HLA (simple DDS data from sensors to HLA federations)
- Present to the user new DDS QoS parameters.
- Easy HLA migration, since Runtime Infrastructure shall meet the latest HLA Dynamic link compatible API.
- Standardize the wire-protocol (RTPS).

DDS is also available for real-time simulations, when there is no need of HLA semantics.
Implementation

SimWare-Kernel

Middleware based on publish-subscribe paradigm, offering in a single API, access to either DDS domains or HLA federations.

- **Kernel (DDS)**
  - Used inside simulator
  - QoS parameters

- **Kernel (HLA)**
  - Used to join simulators
  - Interoperability with current applications

**DDS Simulation**

**HLA Simulation**

Bruno Calvo - bcalvo@nexteleng.es
Advantages

- Allows an easy distribution of components between HLA and DDS simulations
- Unifies HLA applications of many Runtime Infrastructure vendors through the use of the same interface.
- Allows an uniform use of data.
- Allows dynamic (from XML file) generation of data types.
Kernel has a user-level API based on:

- **Simplicity**: focusing on just sending/receiving data.

- Having an advanced API to offer **HLA services**:
  - Declaration management (data discovery)
  - Instance based publications, instance discovery
  - Synchronization points
  - Save / restore
  - Attribute based ownership
  - Durability
Basic services
- QoS Transportation types: defined in the data model
- QoS Frequency: defined in the data model
- Persistence: volatile (DDS) + transient (Kernel)
- Ownership: HLA property management
- Domains/Federations: many independent simulations at the same time
- Liveliness: declaration management, discovery, error management, etc.
- Dead Reckoning

Future
- Content filters
- Variable frequency
- Simulation discovery
Implementation

Data Model use options

Source Code Generator

ModeloDatos.hpp

Application

ModeloDatos.hpp

Lib. Kernel (generic)

Specific application (simulation modules, etc)

Data model

Application

Lib. Kernel (generic)

Generic application (gateway, recorder, etc.)
Implementation

Gateway

Application using Kernel middleware designed mainly to join DDS and HLA applications.

- Joins DDS domains and HLA federations:
  - DDS-DDS
  - HLA-HLA
  - DDS-HLA

- Manages mapping of data between data models

- Filters, in order to allow only relevant data to be sent from one simulation to the other

- Joins not only user data, but also HLA services:
  - Ownership.
  - Durability
  - Entity discovery

Bruno Calvo - bcalvo@nexteleng.es
Interoperability

Scenarios

DDS simulation
- Performance improved simulations
- QoS parameters

DDS simulation, offering filtered data to HLA
- High performance in DDS simulation, allows real-time
- Offer only relevant data to HLA federates, at lower frequency

SimWare-Kernel

Bruno Calvo - bcalvo@nexteleng.es
Interoperability

Scenarios

Reusing HLA applications into DDS simulations

Distribution of applications. Reuse of components.

SimWare-Kernel

Bruno Calvo - bcalvo@nexteleng.es
Interoperability Scenarios

Reduce integration time

Join many HLA federations with different HLA/RTIs (DMSO, Mäk, ...)

HLA (DMSO)  HLA (Mäk)
Examples
Flight Simulator

Cockpit

I/O

Executive

eHost

Kernel DDS

Scenario

IOS

Recorder

eLogistic

SimWare

Bruno Calvo - bcalvo@nexteleng.es
Examples
Missile Mistral Simulator

DDS0
HLA
DDS1
DDS2
Summary

A DDS-based HLA could bring important benefits:

- Lower Latency
- Better communication determinism and RT behavior
- Enhanced QoS as defined by the DDS standard
- Wire interoperability
- Communication with non-simulation platforms

The SimWare Kernel is platform for distributed simulation that provides the best of both worlds:

- Offers a publish-subscribe that abstracts (joins) DDS and HLA
- Provides a powerful, unified single data model
- Enables many deployment scenarios which may combine existing HLA implementation platforms