Technical Resources

[How to develop

“smart connected” simulations with Simware]

Reference: Simware Technical Library

Date: April-2016

Version : 1

---

1 INTRODUCTION

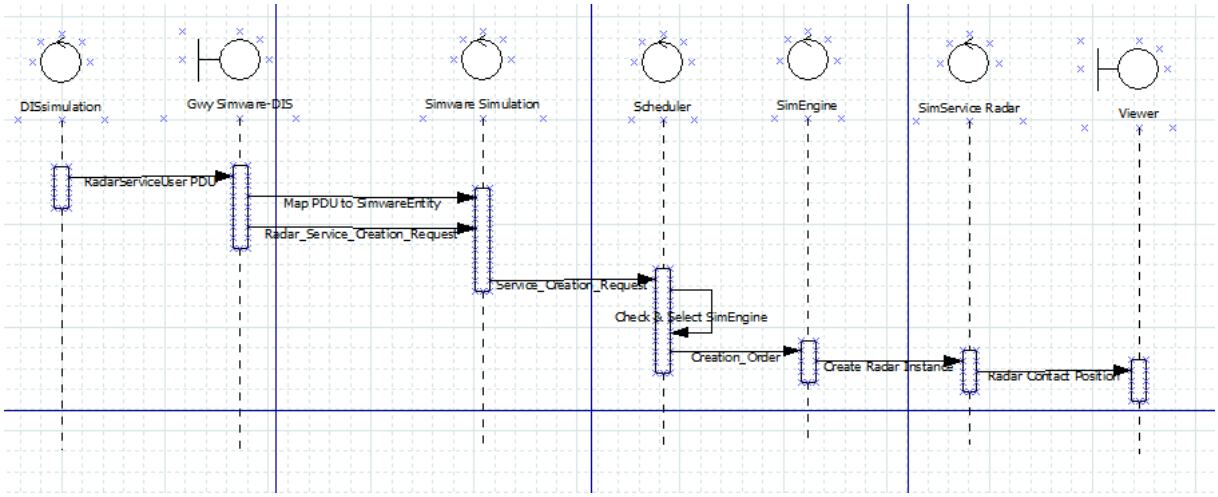
One of the most **powerful capabilities included in Simware is how to interoperate simulation entities by only leveraging the data-model**. Interoperability is achieved by exchanging data between the different simulation entities that are included in the virtual scenario. Data-models in Simware are based on HLA Object Model Templates or OMT as define by IEEE HLA standard. A key difference in Simware when compared with any other product in the market is that we don't only share a simulation data-model, with all the objects and interactions but we also share the state-machine of the simulation through a second data-model. This control data-model can be managed by any application using the Simware Control Library API provided in Simware Core. In this way, **interoperability is not limited only to exchange data about the behavior and dynamics of the simulation entities and it is extended to share information about the state and transitions in the simulation**. Leveraging this feature, it is very easy to have, for example, a virtual simulator, that is managed from an IOS located in the network. This level of integration is easily achieved with Simware by sharing the control data-model of the simulator. **Interoperability in Simware can be achieved at an entity level, using interactions to call the methods and procedures of other entities**. For example a fighter entity can use interactions to command external simulation services to launch flares, missiles or bombs. This is the way our simulation services integrated in SimwareLAB and also available as Simulink models in our Virtual Vehicle Repository are working. In this way, Simware is the only way right now in the market to get a level 5 of interoperability or "Dynamic Interoperability" for simulation as defined by LCIM (Levels of Conceptual Interoperability Model).

2 EXAMPLES OF SMART CONNECTED SIMULATIONS

This article explains with a couple of examples how entities in Simware interact through the middleware NCWare Sim

2.1 DIS SIMULATION REQUESTING THE CREATION OF A RADAR SIM SERVICE IN SIMWARE.

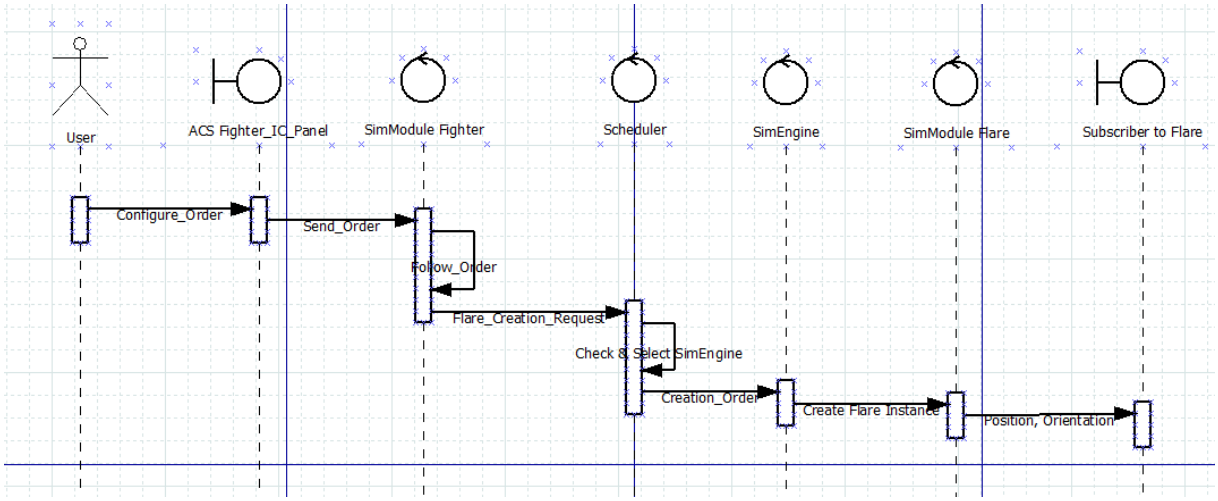
In this example, a **DIS simulation will use RadarServiceUser PDU to request the creation of a new instance of a radar in a simulation server** that is providing Radar simulation services.



As you can see in the sequence diagram shown above, process start with the DIS simulation sending a PDU requesting the service of a Radar simulation. This PDU will be received by the DIS gateway and will be translated to the interaction Radar_Service_Creation_Request that will be published in the simulation domain using NCWare Sim. This request will be managed by the Scheduler and the SimEngine in Simware runtime infrastructure to **command the creation of a new instance of a radar that will be managed by SimService Radar**. SimService Radar will publish the radar contact information to the middleware and this middleware will distribute the data to all the subscribers of this specific simulation object.

2.2 COMMANDING THE CREATION OF INSTANCES FROM ACS OR SCL.

The same flow of data is shown in the next example, in which the instructor is using ACS HMI to order a fighter to launch a Flare. In this case, **Fighter and Flares and different entities and because of that Fighter needs to use an interaction to request the creation of a new instance of a Flare** to the Sim service doing the physical simulation of the flares.



In this second scenario the basic flow of data is the same as in former example with a bit more of complexity because in this case the **connection is between two independent simulation services running in the network**: SimModule Fighter and SimModule Flare. Fighter will use an interaction defined in the simulation datamodel to order the creation of a new instance of a flare to an external application (SimModule Flare).

A variation of this sequence diagram is when **you are not using ACS but you have built your own Control station using Simware Control Library** (SCL) API. In this case you will use functions included in SCL::Communication layer Class to request to order the creation of the new instance of the flare to the fighter (see SCL documentation for further details).

3 WRAPPING UP

This examples shows how easy is to créate fully open, distributed and uncoupled simulation scenarios with hundreds of entities in the network that are sharing data by exchanging simulation objects and interoperating by exchanging interactions as orders to command the behavior of the entities in the network.

Combining this interoperability capabilities with the distributed computing capabilities offered by the support of multiple SimEngines in the network, allows to build and deploy virtual scenarios with full interaction between all the simulated entities.