DIS – HLA Gateway

Major James McRae, Senior Engineer, Land 134 CTC-LIS; Mr Joseph Steel, Royal Military College of Science
James.McRae@defence.gov.au

Abstract. There is always a great need for integration during the transition from one system to another. This is more so the case with the integration of simulation protocols were investment of money, time and intellect is too great to be abandoned for a new technology, despite the advancements gained. The Simulation and Synthetic Environment Laboratories (SSEL) at the Royal Military College of Science is an example of a synthetic environment site were the teaching and research simulations provided under the Distributed Interactive Simulation (DIS) standard can not be easily replaced. Therefore, integration with newly acquired High Level Architecture (HLA) federates is the preferred option to maximise the resources available for the conduct of exercises. This integration has been achieved for HLA federates that conform to the Real-time Platform Reference Federation Object Model (RPR-FOM, Version 1.0) through the use of a real-time gateway application.

The Gateway is versatile utility, which breaches the communication gap between HLA and DIS for applications that run in the SSEL. Written in C++, the application employs the latest Run-Time Interface (RTI) whilst maintaining portability and is compatible with the DIS versions used in the SSEL. Although requiring changes at the source code level, it is relatively straightforward to add functionality or modify the Gateway, for future changes in the standards used by the simulators and federates in the SSEL.

This paper looks at the development of the Gateway to achieve the aim of integration in the SSEL teaching environment. Knowledge of the two protocols is assumed is required insight into the consideration needed to develop a link between DIS and HLA. The options for integration are considered with the reasons for adopting the Gateway as a non-intrusive means of integration discussed. Notes on the design, coding and testing of the Gateway show that it is a viable solution to the integration of HLA and DIS in the SSEL and will perform in this environment with such efficiency that the DIS and HLA networks will appear seamless to the user.

1 INTRODUCTION

A marketing pitch for the software application described herein, might read: “The DIS/HLA Gateway converts between DIS PDUs and HLA Services in both directions. This gateway conversion occurs in “real-time” while the simulation exercise is in progress. The Gateway is the easiest way to implement HLA compliance as there is no modification required in the DIS compliant legacy simulator other than placing the Gateway between the legacy simulator and the HLA network, as shown at Figure 1. It will:

a. Translates attributes or parameters generated by a HLA federate that is based on the RPR-FOM, such as the SSEL CrewStations, to a DIS PDU, and promulgate it to all DIS simulations.

b. Translate every field in the DIS PDU, such as Entity ID to the appropriate attribute or parameter defined in the RPR-FOM and promulgate it to a subscribing HLA federate.

c. Provide a means for general DIS based simulations to interact with HLA federates and has specific application to the SSEL DIS simulations and HLA CrewStations.

d. Ensure HLA only receives new information, minimising network traffic.

e. Allow HLA objects to be ported into a DIS network, operating at real-time. The HLA objects are given DIS entity characteristics, such as a heartbeat.

f. Permits the joining and resigning of objects and entities from the synthetic environment at any time with graceful entry and exit.

g. Give the CrewStations that are HLA federates and based of RPR-FOM 1.0 an opportunity to be participants in DIS exercises, such as those involving ModSAF, and allow interaction with current SSEL DIS based utilities.

h. It is capable of crossing the current SSEL DIS gateway which links the 164 and 138 networks, permitting scalability of the environment.

i. It is fast, minimising latency between the two environments.

1 The Simulation and Synthetic Environment Laboratories (SSEL) at the Royal Military College of Science (RMCS).

2 The CrewStation is a Unix based MAK application that employs HLA and the RPR-FOM to create a virtual simulation based on a programmable dynamics model, such as a tank, light armoured vehicle or aircraft.

3 Protocol Data Unit.

4 ModSAF Version 5 is an entity level Synthetic Forces (SF) simulation developed for the US DoD. This application generates DIS PDUs.
j. It is sleek, taking advantage of both simulation environments to minimise the network traffic.
k. PC based, C++ Object-Orientated code.
l. Proven to allow MÄK[2] CrewStations, owned by the SSEL, to operate in mixed protocols, including HLA CrewStations with DIS based ModSAF 5.0.”

![Diagram of the Gateway and SSEL Network](image)

Figure 1: The Gateway and SSEL Network

The resultant Gateway is best described as a utility conceptually located between the transport and application layer of the network that caters for the data translation between DIS PDUs and HLA RPR-FOM objects. There is no better description in words of the principle of the Gateway than that shown Figure 2.

![Diagram showing Gateway and DIS/HLA network](image)

Figure 2: The Gateway and SSEL Network

1.1 METHOD

The development of the Gateway was conducted over three phases which were:

a. Review of literature and the project concept. Training in C++ and Microsoft© Visual Studio and the HLA Hands-on Practicum were conducted in this phase.
b. Design and coding of the Gateway.
c. Testing of the Gateway and the HLA and DIS environments were then conducted to complete the report.

1.1.1 Review Phase

Prior to the decision to take on this project, some literature search and discussions were conducted to confirm feasibility of the topic. Upon commencement of the research, a more detailed literature search was undertaken. In particular, the DIS and HLA references were studied in detail and the resources of the DMSO website sourced for HLA tools.

This literature search was in compliment to the MSc courses conducted at RMCS. Although all courses that comprised the MSc¹ were used, particular knowledge to the dissertation topic were obtained from the courses:

b. Software Engineering for Scientific Applications.
c. Networked and Distributed Simulation.

It was essential during this phase to obtain skills in the chosen programming language. As C was a major language introduced in the course, C++ was chosen as the preferred language which is supported by the RTI².

Furthermore, HLA training received on the MSc needed to be complemented in order to understand the application of the RTI interfaces. This was conducted near Washington DC through the DMSO sponsored HLA Hands-on Practicum.

1.1.2 Design and Coding

A design for the Gateway was developed using software-engineering techniques learnt on the MSc. This design was then adapted into code. The initial code was written on Windows 98 with the DMSO RTI 1.3NG. As the code matured, it was ported to the SSEL Windows 2000 Simlab 13 workstation, which was networked with the SSEL CrewStations.

Properties of the CrewStations running in DIS and HLA were observed.

1.1.3 Testing

The code was completed with modifications to support the MÄK RTI and to be compatible with the CrewStations implementation of DIS and HLA. Tests were conducted on various configurations and code modified as required. Built in timers to the Gateway were used to capture performance measures and this was used in conjunction with the available network, DIS and HLA tools to monitor and evaluate the Gateway. Finalising this report was a means to validate the Gateway, cross checking what it does, against with what it is suppose to do.

¹ Master of Science in Defence Simulation and Modelling.
² Run-Time Interface, generically refers to all RTI that is compatible with the DMSO RTI-NG 1.3.
2 RPR-FOM

At the time of producing the Gateway, RPR-FOM version 1.0[1], which differs greatly from previous 0.x versions, appears to be established base document with future versions (2.0 and 3.0 are in un-released draft form) building on it rather than modifying it. The layout of the RPR-FOM complies with the HLA Object Model Template (OMT) guidelines[4]. This layout is relatively easy to follow with detailed tables and linked notes to describe the Classes, attributes and parameters. In the following description, the term ‘fields’ refers to the DIS PDU fields (such as the fields that make up a PDU header include timestamp and DIS protocol) defined in IEEE1278.1-1995, ‘attributes’ refers to HLA object attributes and ‘parameters’ refer to HLA interaction parameters.

Essential to adopting the RPR-FOM into the Gateway was access to the following documents, for the reasons of:

a. IEEE1278.1-1995[5] for the description of a DIS PDU, its characteristics and field requirements; and

In essence there are two main components to the RPR-FOM, which come in the form of files. The first is the RPR-FOM.omt file that was used for developing the Gateway as it contains all the tables required by the OMT. The second was the RPR-FOM.fed, which is used by the RTI to create a reflection of the objects to be used during a federations execution.

2.1 Mapping Objects to Entity State and other PDUs

The RPR-FOM has created object classes based on the enumeration of the Entity Type field in PDUs. For example, Table 1: EBV-DOC Entity Type, shows some of the Entity Type enumerations with their mapping into the respective RPR-FOM object classes shown in Figure 3.

The object classes defined in Figure 3 inherit the attributes of the higher classes, for example Platform inherits any and all attributes from the class PhysicalEntity. The ‘P’ and ‘S’ after each class name states that the intention of that class is to be either Publishing or Subscribing respectively. The Gateway should therefore not publish attributes as class PhysicalEntity.

3 RESOURCES NEEDED TO BUILD THE GATEWAY

In order to design, develop and test the Gateway, a number of resources needed to be obtained or accessed. Table 2 reflects the resources that would be needed for the Gateway and would be required for any similar HLA-DIS integration utility.

Table 1: EBV-DOC Entity Type

<table>
<thead>
<tr>
<th>ENTITY KIND</th>
<th>DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Kind</td>
</tr>
<tr>
<td>0</td>
<td>Other</td>
</tr>
<tr>
<td>1</td>
<td>Platform</td>
</tr>
<tr>
<td>2</td>
<td>Munition</td>
</tr>
<tr>
<td>3</td>
<td>Life form</td>
</tr>
<tr>
<td>4 - 9</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 3: RPR-FOM Object Classes

Table 2: Resources Needed for the Gateway

<table>
<thead>
<tr>
<th>RESOURCE</th>
<th>SOURCE</th>
<th>REQUIRED FOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Visual C++</td>
<td>SSEL</td>
<td>Development of the application. C++ preferred as C was a major language taught on the MSc course, it was available in the SSEL and the RTI provided C++ API. Also contains header files and libraries, such as Winsock, for network communications.</td>
</tr>
<tr>
<td>HLA RTI</td>
<td>DMSO MÅK</td>
<td>Requirements include: Allow access to RTIexec and Federexec for joining federations. Obtain the libRTI for RTI ambassador and Fed ambassador Classes. Programmer’s Guide.</td>
</tr>
</tbody>
</table>
### RESOURCE SOURCE REQUIRED FOR

- **RPR-FOM**
  - OMD Library
  - omi.msiac.dmo.mil
  - Also SSEL
  - Federation Object Model for HLA based on the DIS PDU information. Is crucial for the interaction of HLA and DIS.

- **VR-Link (FOM)**
  - MAK
  - Similar to the RPR-FOM, with additions for the MAK RTI internal management.

- **OMDT**
  - DMSO
  - Assist in analysis of the VR-Link and RPR-FOM .fed files.

- **MAK CrewStations**
  - SSEL
  - Can be executed as either a HLA federate or DIS entity. Prime tool for development and testing the Gateway.

- **DIS Logger**
  - SSEL
  - MAK also available.
  - DIS data collection for post analysis of the actual HLA to DIS conversions against expected.

- **DIS Monitor or Stealth Viewer**
  - SSEL
  - Run time monitoring of the DIS entities created.

- **CNA Network Analysis**
  - SSEL
  - Run time monitoring of the network traffic.

- **HLA Logger**
  - DMSO
  - MAK
  - HLA data collection for post analysis of the actual DIS to HLA conversions against expected.

- **Windows based Workstation**
  - SSEL
  - Point of development and execution of the Gateway.

- **HLA Hands-On Practicum**
  - DMSO
  - A free HLA RTI training course run near Washington DC that provides programming training for use of the RTI. Very worthwhile.

- **Enumeration & Bit Encoded Values Document (EBV-DOC)**
  - SSEL
  - The RPR-FOM incorporates enumeration directly as attributes.

- **IEEE 1278.1-1995**
  - SSEL and other IEEE sources
  - Defines the DIS PDUs that need to be replicated in the Gateway.

- **IEEE 1278.2-1995**
  - SSEL and other IEEE sources
  - Defines the DIS communication services that the Gateway will need to incorporate.

- **IEEE P1516 and IEEE P1516.2**
  - DMSO web site
  - Reference only for the HLA Framework, Rules and OMT

### RESOURCE SOURCE REQUIRED FOR

- **IEEE P1516.1**
  - DMSO web site
  - HLA Interface Specification reference – the RTI Programmer’s Guide was the practical reference used.

### 4 DESIGN ISSUES

#### 4.1 Byte Ordering

Different machine architectures sometimes store data using different byte orders. For example, Intel-based machines store data in the reverse order of Macintosh (Motorola) machines and the Silicon Graphics Unix-based computers hosting the CrewStations in the SSEL. Intel’s byte order, called “little-Endian,” is also the reverse of the network standard “big-Endian” order. Table 3 explains these terms.

#### 4.2 Dead Reckoning or Not.

DIS implements dead reckoning[5], which is a method of position/orientation estimation. It is used to limit the rate at which simulations must issue state updates for an entity.

HLA assumes future technologies will permit network bandwidth to support updates of attributes at whatever speed is required. Moreover, the OMT[4] required the frequency of update to be spelt out. The Update Type and Update Condition columns of the OMT record the update policies for an attribute. The update type may be specified as *static*, *periodic*, or *conditional*. When the update type is *periodic*, a rate of number of updates per time-unit may be specified in the Update Condition column. Attributes with a *conditional* update type may have the conditions for update specified in the Update Condition column.
The intent is that a federate does not need to maintain a model of remote objects but will reflect a ‘ghost’ based on the current attributes provided. Of course, if a federation calls for each federate to maintain dead reckoning or an equivalent scheme, then this can be implemented in the FOM, and is so with the RPR-FOM.

4.3 Data Flow

Design of the Gateway was greatly simplified with the Data Flow Diagram shown at Figure 4. Having a detailed understand of the Data Flow Diagram led to two fundamental design questions:

a. What will be the programs control flow?
b. What C++ classes are needed?

The solution to the first question was supplied in the form of the requirements dictated by the RTI and the FEDEP. In particular, the Gateway must commence with federation management, establish declaration management and conduct object management. HLA requires objects and federates to gracefully leave the federation, and therefore the closure of the program must be removing from the object management, un-publishing and un-subscribing from the declaration management and resign from the federation management.

![](image)

**Figure 4:** Data Flow of the Gateway

The resulting design for the body of the Gateway is an repetitive cycle of:

a. DIS Manager: Receive DIS network traffic.
b. DIS Manager: Process entity state PDUs into the DIS Entity List Manager.
c. DIS Manager: Process interactions directly to the RTI.
d. DIS Entity List Manager: Review all the entities in the DIS Entity List Manager and post any new entity state PDUs to the DIS Manager and send on the DIS network.
e. Tick() the RTI.
f. Process any reflectAttributes() directly to the DIS Manager.
g. Process any receiveInteraction() directly to the DIS Manager and send on the DIS network.
h. HLA Object List Manager: Review all the entities (derived from the HLA objects) in the HLA Object List Manager and post any new entity state PDUs to the DIS Manager and send on the DIS network.
i. Every five seconds, scan all the entities (derived from the HLA objects) in the HLA Object List Manager and create entity state PDUs for each, pass to the DIS Manager and send on the DIS network.
j. Repeat until user exits.

5 LIMITATIONS OF THE GATEWAY

Limitations are defined as an imposition or constraint that occurs due to the lack of functionality.

5.1 Inherited Constraints

The definition above is extended to those constraints imposed by the following, which are inherited by the Gateway:

5.1.1 DIS

It is anticipated that the current IEEE standards and the enumerations defined in the EBV-DOC, will not vary to any extent to impact on the Gateway. If more recent protocol version is used (defaulted at 2.0.4), then the user can change this through a command line argument on starting the Gateway. DIS is the legacy system and therefore the principle driver for any constraints placed on the Gateway and the simulation network, which includes the ‘best-effort’ communication policy.

5.1.2 HLA

Impacting on the Gateway includes the rules and how the interface specification is implemented by the supporting RTI. As HLA, as an architectural concept, can be molded to suit the simulation environment, it can be said that it has no limitations to impose on the Gateway. Hence, the major limitations are derived from those imposed by the developers of the (or any) FOM. However, the practical application of the HLA through the current RTI technologies introduces constraints and weaknesses.

5.1.3 RFR-FOM

Errors, non-conformity, ambiguous descriptions and incompleteness become inherited by the Gateway and other federates adopting the RPR-FOM. Changes to the RPR-FOM will most likely require changes to the source code.
5.2 Closing Notes on the Gateway

The main purpose of this section is to list the limitations, their cause and any possible future resolution. This section is provided in lieu of a recommendations section. It is difficult to create recommendations that will be relevant tomorrow. The Gateway’s usefulness is intrinsic to the future changes in HLA, the advancement of the RPR-FOM and the SSEL resources or teaching requirements.

The gateway is a successful application, but as can be appreciated by the nature of the DIS and HLA protocols presented above, limitations are imminent. The following limitations are imposed by the Gateway:

a. Real-time simulation only. This is derived from the DIS standard and imposes the requirement of HLA federates. The Gateway is not designed to accommodate any changes for faster or slower than real-time simulation.

b. Requires RPR-FOM 1.0 or compatible .fed file. This constraint is imposed on all HLA federates that are required to participate in a DIS exercise at the SSEL.

c. To be effective for large simulations, so as not to inflict latency, the Gateway will require a dedicated workstation. This offers the advantage of centralising the Gateway and the RTIexec on a single workstation, remove the overhead from a simulation’s host.

d. Current implementation only support Entity State, Fire and Detonation PDUs. This is sufficient to achieve integration with the current HLA federates operated in the SSEL (namely, the HLA CrewStations). Additional PDUs that are supported by the RPR-FOM can be easily implemented.

e. Current implementation supports a single network, however, an option for multiple networks was explored and is possible.

REFERENCES