Developing the HLA Tutorial Part Two: Towards Federation Design Patterns

Björn Möller, Pitch Technologies, Sweden
Fredrik Antelius, Pitch Technologies, Sweden
Mikael Karlsson, Pitch Technologies, Sweden

bjorn.moller@pitch.se
fredrik.antelius@pitch.se
mikael.karlsson@pitch.se

Keywords:
HLA, Tutorial

ABSTRACT: *The first part of the HLA tutorial, based on HLA Evolved, released in 2012, was well received by both industry and academia. It is used, for example, for teaching distributed simulation in universities and also in the Smackdown project, originally initiated in SISO. The focus of the tutorial is to teach best practices for how to develop HLA federates and federations, rather than to cover all details of the HLA standard.*

The first part of the tutorial can be seen as the foundation for all federates, covering how to join a federation, publish, subscribe and work with objects and interactions. The recently released, second part, can be seen as a smorgasbord of techniques that can be used in federations, like time management, ownership, development of reusable FOM modules, fault tolerance and more. Few federations are likely to use all of the techniques but most federations will use some of them.

The biggest challenge is to explain the full potential of time management. The tutorial starts with a common and easily understood use case and gradually moves on to the general theory and more advanced time management topics.

One recurring theme of the tutorial is the best practices and design patterns typically used in federations. These are summarized in the end. Pointers are also provided to design patterns used in the NATO NETN design as well as in civilian HLA applications.

As a conclusion of the tutorial the concept of interoperability is revisited, based on the Layers of Conceptual Interoperability Model (LCIM).

1. Introduction

This paper focuses on part two of a freely available HLA Tutorial. The development and philosophy of the first part is described in a previous paper. Some background about why this effort was initiated and some of the challenges with getting started with HLA, using only the standards documents, is described in that paper.

1.1 Reactions to the HLA Tutorial Part One

The tutorial has been downloaded by thousand of readers. The authors have noted several interesting reactions to the first part of the tutorial.

Several people in the standards community have questioned if a tutorial is mainly a rewrite of the HLA specification. This may violate the copyright, in this case the IEEE copyright. A quick look at the tutorial reveals that it describes how to build federates and federations, not on the structure or details of the HLA specification. Not much detail is provided on each service call. Instead the tutorial points to the appropriate section of the HLA specification. If this was a carpentry tutorial, one might say that the focus is on how to build a house using the tools, not to provide a specification of a carpenters tools, material or building blocks.

The tutorial has been extensively used in the "Smackdown" University outreach project, originally initiated by SISO. The tutorial has been helpful in getting students up to speed with HLA. Since the first part does not cover HLA Time Management, students have used sample federates developed by NASA staff. Hopefully part two will help fill that gap.

Another interesting reaction is that several universities, for example in the UK, have adopted the tutorial as part of courses in distributed systems. This has happened without any direct involvement from the authors. A next step is probably to work more closely together with universities to develop additional lecture material.

Yet another interesting observation is that the authors have met many engineers from the simulation industry that maintain their own, neatly printed, copy of the tutorial. It is likely that a "print-on-demand" version of the HLA tutorial will be made available in the future.

2. Structure of the HLA Tutorial Part One and Part Two

Part one of the HLA tutorial consists of three main parts:

- An overview chapter that describes the origin and purpose of the HLA standard, the users of HLA as well as policy and market aspects.
- Two chapters that provides an overview of the architecture from a service-oriented perspective, where HLA can be considered a "Services Bus".
- Eight chapters that describe how to build a federation (federates and FOM), step by step. The structure of a FOM and federates are introduced, step-by-step.

In addition to this there are several appendices, most notably the Federation Agreement and the FOM. Source code and tools are also freely available for download for users that want to get s hands on experience or that need a starting point for their own development. Note that there is plenty of important information in the tutorial that is not provided at all in the HLA standard, for example how to structure a federate, how test and debug federates and the concept of object oriented HLA.

The second part of the HLA tutorial builds upon part one. It starts with four main sections:

- FOM modules and OMT data types
- Ownership
- Time Management
- Data distribution management

In addition to this there are several chapters with advice on interoperability and how to build HLA federations, including federation performance and fault tolerance. To promote a holistic view of federation development, the tutorial ends with a discussion around the Levels of Conceptual Interoperability Model (LCIM).

Part One of the tutorial is intended to be read in its entirety by a developer. Part Two can be read in any order, depending on the needs of the federation that the developer intends to develop.

3. More on the Main Sections

This section gives some additional insight in the main sections of Part Two of the tutorial.

3.1 FOM Modules

The FOM Modules section starts off with the monolithic FOM developed in Part One of the HLA tutorial. It then presents two main considerations when developing a FOM module:

- What is the purpose and scope of the FOM module
- What is the intended degree or reuse of the FOM module

This is then illustrated by splitting up the FOM into a general Federation Management module and a specific Fuel Economy FOM module. In most practical cases a FOM module needs some modifications to become generalized before it can be considered to be reusable. This is illustrated through a generalized scenario handling interaction.

One important FOM module that an HLA developer needs to understand is the predefined MIM module. It contains, among other things, some predefined building blocks for HLA Datatypes. How to build different types of Datatypes, such as Simple, Enumerated, Array and Record data types is described in detail.

3.2 Ownership Management

The principles of HLA ownership management are not obvious to many programmers with an object-oriented background. This section describes the purpose and typical use cases for ownership management. It also points out the implicit ownership that the creator of an object instance has. It also describes the importance of understanding that a distributed simulation needs to handle both locally created object instances as well as discovered, remotely registered object instances. The latter usually creates some confusion for developers that are used to develop code for simulations that receive little external data.

This section then moves on to describing some fundamental principles, "push" and "pull", of HLA ownership. The focus of the tutorial is "pull" ownership. In most practical applications, the HLA ownership services alone are not enough for managing ownership transfer. An example of a typical design pattern is provided; in this case a centrally managed "pull transfer of entire instance" is described.

The acquisition of ownership of attributes where the registering federate has been lost is also covered.

3.3 Time Management

This the most challenging part of HLA and it is presented in the longest chapter of the tutorial. It starts by presenting three important time concepts:

- Wall clock time
- Scenario time
- Logical time

A straightforward implementation of HLA Time Management in the Fuel Economy federation is then shown. A frame-based approach is used, meaning that all federates uses a fixed and equal time-step. In each "frame" the state of next "frame" is calculated. The cycle with granted/advancing state is shown together with the flow of outgoing and incoming time stamped events.

Simulation speed and pacing is then described and solutions are described for real-time, scaled real-time and as-fast-as-possible simulation.

Once the practical example is understood the theory of HLA Time Management is presented, introducing Look-ahead and Greatest Available Logical Time.

3.4 Data Distribution Management

This section shows how to use DDM with both dimensions that can be considered "continuous", like the car position, and "discrete" dimensions, such as type of fuel.

4. Discussion

4.1 What parts of HLA do we need?

Part One of the tutorial can be seen as the foundation of any HLA federation. The knowledge and the services described will be used by anyone that intends to build a federation.

Part Two of the tutorial provides a "smorgasbord" of HLA features. Most federations will use some of these features but few federations will use all of them

4.2 Towards design patterns

One recurring theme in the practical examples is that overarching distributed algorithms, or "design patterns" are usually developed, where HLA services can be considered building blocks. Some examples of the Fuel Economy Federation are the scenario management, the execution management and the management of how and when ownership transfer is performed. The NATO Education and Training Network groups (NMSG-068, NMSG-106) has also worked extensively with design patterns, for example for providing services, such as logistics and refueling, between simulated entities. The authors would like to argue that this is a trend that will continue. SISO should be one of the main forums for exchanging experiences from distributed design patterns for simulations.

5. Conclusion

The purpose of the HLA tutorial project is to increase the interest in distributed simulation in general and HLA in particular by lowering the barrier to the HLA standard. This is a long-term project, but the interest in the tutorial and the number of downloads already indicates success.

The next step is to collect feedback on the best practices presented and perform one more revision of both Part One and Part Two during 2014.

The authors also hope to see an increased activity around design patterns for distributed systems within SISO the coming years.

References

[1] "The HLA Tutorial", www.pitch.se, September 2012

[2] "High Level Architecture Version 1.3", DMSO, www.dmso.mil, April 1998

[3] IEEE: "IEEE 1516, High Level Architecture (HLA)", www.ieee.org, March 2001.

[4] SISO: "Dynamic Link Compatible HLA API Standard for the HLA Interface Specification" (IEEE 1516.1 Version), (SISO-STD-004.1-2004)

[5] IEEE: "IEEE 1516-2010, High Level Architecture (HLA)", www.ieee.org, August 2010.

[6] Frederick Kuhl, Richard Weatherly, Judith Dahmann: "Creating Computer Simulation Systems: an Introduction to the High-Level Architecture", Prentice Hall PTR (2000), ISBN 0130225118

[7] DMSO: "RTI 1-3-Next Generation Programmer's Guide Version 3.2", September 2000, US Departement of Defense: Defense Modeling and Simulation Office

[8] "Porting a C++ Federate from HLA 1.3 to HLA 1516" & "Porting a Java Federate from HLA 1.3 to HLA 1516", March 2003, http://www.pitch.se/support/pitch-prti-1516

[9] "Migrating a Federate from HLA 1.3 to HLA Evolved", November 2010, http://www.pitch.se/technology/about-hla-evolved

[10] "FEAT PDG - Federation Engineering Agreement Template", www.sisostds.org

[11] SISO: "Real-time Platform Reference Federation Object Model 2.0 ", SISO-STD-001 SISO, draft 17, www.sisostds.org

Author Biographies

BJÖRN MÖLLER is the Vice President and co-founder of Pitch Technologies. He leads the strategic development of Pitch HLA products. He serves on several HLA standards and working groups and has a wide international contact network in simulation interoperability. He has twenty years of experience in high-tech R&D companies, with an international profile in areas such as modeling and simulation, artificial intelligence and Web-based collaboration. Björn Möller holds an M.Sc. in Computer Science and Technology after studies at Linköping University, Sweden, and Imperial College, London. He is currently serving as the vice chairman of the SISO HLA Evolved Product Support Group and the chairman of the SISO Real-time Platform Reference FOM PDG.

FREDRIK ANTELIUS is a Senior Software Architect at Pitch and is a major contributor to several commercial HLA products, including Pitch Developer Studio, Pitch Recorder, Pitch Commander and Pitch Visual OMT. He holds an M.Sc. in Computer Science and Technology from Linköping University, Sweden.

MIKAEL KARLSSON is a is the Infrastructure Chief Architect at Pitch overseeing the world's first certified HLA IEEE 1516 RTI as well as the first certified commercial RTI for HLA 1.3. He has more than ten years of experience of developing simulation infrastructures based on HLA as well as earlier standards. He also serves on several HLA standards and working groups. He studied Computer Science at Linköping University, Sweden.