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, Ownership Management, Time Management, DDM,
FOM Modules, Design patterns

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 by 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 of the tutorial where 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).

This paper presents an overview of the HLA Tutorial Part Two and describes the process and reasoning around the development of the tutorial.

1. Introduction

This paper focuses on part two of a freely available "HLA Tutorial" [1]. The development and philosophy of the first part is described in a previous paper [2]. Some background about why this effort was initiated, and some of the challenges with getting started with HLA [3], 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 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 carpenters 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, materials or blueprints of the house.

The tutorial has been extensively used in the "Smackdown" University outreach project [4], 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 hardcopy "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

Figure 1 shows the overall structure of the two parts of the tutorial.

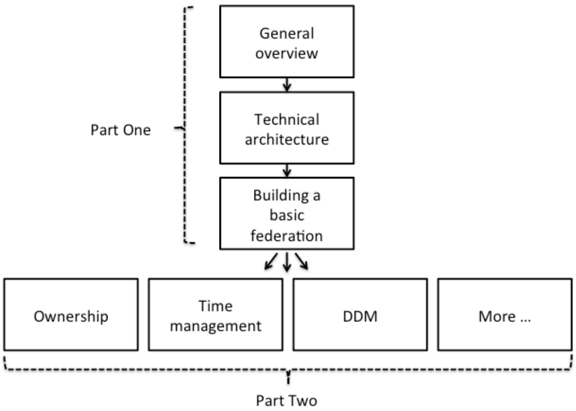

Figure 1: Overall tutorial structure

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 provide 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 is introduced, in small steps.

In addition to this, there are several appendices, most notably the Federation Agreement and the FOM of the sample federation. Source code and tools are also freely available for download for users that want to get 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 to test and debug federates and federations 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) [5].

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

3. More on the Main Sections

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

3.1 FOM Modules

The FOM Modules [6] 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, as shown in Figure 2. 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.

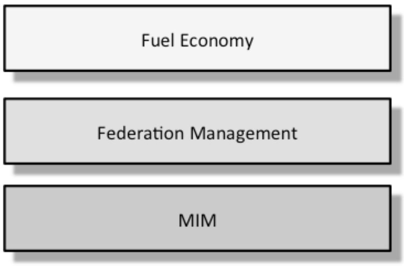

Figure 2: FOM Modules in the tutorial

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 of the tutorial describes the purpose and some typical use cases for ownership management. It also points out the implicit ownership that the creator of an object instance has. It 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 or no external data.

The ownership section then moves on to describing some fundamental principles, "push" and "pull", of HLA ownership. The focus of the tutorial is "pull" ownership, which the authors consider the best starting point for a novice. In most practical applications, the HLA ownership services alone are not enough for managing ownership transfer. Some kind of explicit management mechanism is usually added. 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. This is crucial for any fault tolerant federation.

3.3 Time Management

This is the most challenging part of HLA and it is presented in the longest chapter of the tutorial.

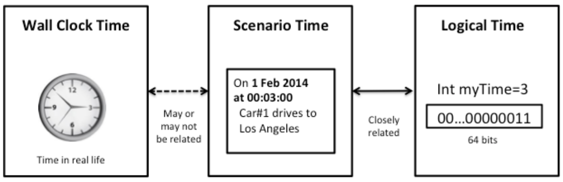

Figure 3: Three time concepts

It starts by presenting three important time concepts, as shown in Figure 3:

- 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 use 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 Lookahead and Greatest Available Logical Time.

3.4 Data Distribution Management

This section of the tutorial shows how to use Data Distribution Management, both with dimensions that can be considered "continuous", like the car position, and "discrete" dimensions, such as type of fuel. The tutorial emphasizes the use of normalization functions and that the important point that DDM does not work on the actual attribute values being updated, but needs to be provided separately – a common misconception among newcomers.

4. Discussion and experiences

This section of the paper provides some thoughts and experiences from the development of the tutorial.

4.1 Writing simple text for novices

The main challenge for experienced professionals, when writing a tutorial, is what not to write. It is crucial to focus on getting the key concepts absolutely clear. What may seem as interesting discussions, for example on variations and exceptions should be postponed for later chapters.

This may sound obvious, but for professionals with multi-faceted insights in HLA, these types of discussions may be very interesting. Once a beginner has mastered the main principles of HLA, he or she could start studying such discussions, but not before.

For many parts of the tutorial, this means that almost half the text was deleted during the final editing and many clarifications were made.

4.2 Spot the forest among the trees

The HLA specification is extremely detailed and describes every aspect of many small things. The problem for beginners is that it is very hard to figure out how all these things fit together. One example of this is Ownership management, which provides a very daunting state diagram, and eighteen different services that can fit together in many ways. When writing the HLA Tutorial, we focused on a small number of end-to-end examples of how to use ownership management.

4.3 Distributed algorithms and design patterns

HLA is a distributed architecture with several built-in distributed algorithms. These range from the simple sending and reception of interactions to the advancement of logical time and ownership transfer negotiations. On top of this, the federation usually needs to add its own patterns, for example for start-up, scenario handling, management of ownership transfer, etc. These concepts need to be introduced step-by-step as shown in figure 4.

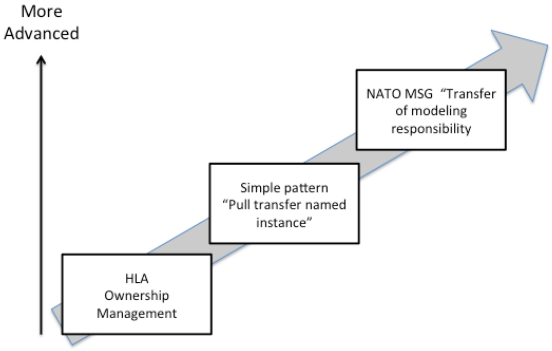
More Advanced
NATO MSG "Transfer of modeling responsibility
Simple pattern "Pull transfer named instance"
HLA Ownership Management

Figure 4: Patterns, step-by-step

One example is ownership management, where the tutorial introduces a very simple distributed algorithm for managing the ownership transfer. After this, a more advanced pattern, Transfer or Modelling Responsibility ("TMR" from NATO MSG-068/106 [7]) is presented.

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.

4.4 Challenges of a problem-driven approach

It is common to teach HLA by describing each service group and then explain how it can be used. The HLA tutorial takes another approach, a more problem-driven approach, where typical federation problems are presented and solutions, using HLA services, are offered.

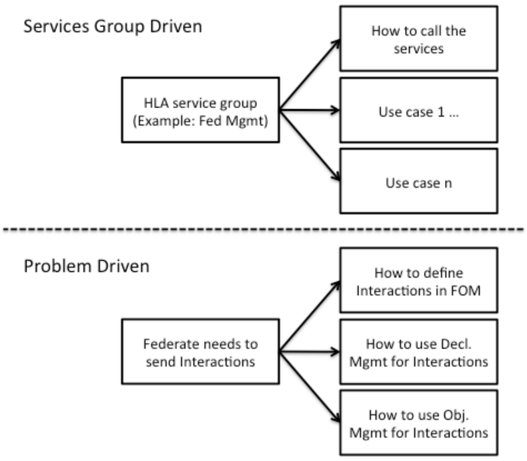
Services Group Driven
HLA service group (Example: Fed Mgmt)
How to call the services
Use case 1 ...
Use case n
Problem Driven
Federate needs to send Interactions
How to define Interactions in FOM
How to use Decl. Mgmt for Interactions
How to use Obj. Mgmt for Interactions

Figure 5: Problem driven approach

For example, the tutorial presents the need to send interactions in the federation, then describes how to define the interactions in the FOM and then how to handle them in the federate code. Finally it discusses the services used and some general principles.

This approach turned out to be difficult to implement for time management. In this case it was actually necessary to provide some general theory before showing how it was to be used in the federation.

4.5 Many ways to use Time Management

HLA Time Management is very powerful. Its general and abstract design allows for many ways to handle time in a federation. The drawback of the general approach is that it can be challenging for a beginner. The focus of the main sample is on frame-based federates, where the entire state of a federate is updated for equal length time steps. All federates are both regulating and constrained, which is a typical conservative approach in federations.

One of the main target audiences for the HLA Tutorial is the SISO Smackdown activity, now renamed to "Simulation Exploration Experience". This federation uses an asymmetric scheme, where some federates (from NASA) are both regulating and constrained and some other federates (developed by students) are only constrained. The reason for this is that student federates, if incorrectly implemented, should not be able to slow down or stop the entire federation. This approach is somewhat unusual and had to be covered as an advanced time management topic.

There are more challenges with teaching time management. Some issues that a federate developer needs to consider, like how to make a process sleep, to avoid “busy waiting”, is done differently in different operating systems. The use of time management may also affect how the main loop of a simulation is structured, which may be a challenge when adapting existing simulations to HLA.

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 in the coming years.

References

[1] “The HLA Tutorial Part One”, www.pitch.se/hlatutorial, September 2012

[2] Björn Möller, Steve Eriksson, Åsa Wihlborg: “Developing an HLA Tutorial: Philosophy and Best Practices” Proceedings of 2012 Fall Simulation Interoperability Workshop, 12F-SIW-047, Simulation Interoperability Standards Organization, September 2012.

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

[4] Simulation Exploration Experience, SEEE2014, http:// http://exploresimulation.com

[5] Tolk A, Muguira JA. "The levels of conceptual interoperability model". Fall Simulation Interoperability Workshop. Orlando, Florida: Simulation Interoperability Standards Organization, 2003.

[6] Björn Möller, Björn Löfstrand. “Getting started with FOM Modules”, Proceedings of 2009 Fall Simulation Interoperability Workshop, 09F-SIW-082, Simulation Interoperability Standards Organization, September 2009.

[7] RTO Technical Report: “NATO Education and Training Network”, TR-MSG-068, Feb 2013, NATO

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.