Simplifying the Development of HLA-based Distributed Simulations with the HLA Development Kit (DKF)

Alberto Falcone, Alfredo Garro
Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES)
University of Calabria, Rende, Italy.
{first.lastname}@unical.it

Simon J. E. Taylor, Anastasia Anagnostou
Modelling & Simulation Group
Department of Computer Science
Brunel University London, UK.
{first.lastname}@brunel.ac.uk

***Abstract*— The IEEE 1516-2010 - Standard for Modeling and Simulation High Level Architecture (HLA) range of standards are widely used for distributed simulation, interoperability, reusability and the modeling of large-scale systems in many application domains. However, the development of distributed simulations based on the HLA standards remains a challenging task that requires a considerable effort in terms of time, cost and expertise. The HLA Development Kit software Framework (DKF) is a software framework that aims to facilitate the development of HLA-based distributed simulations. This tutorial presents an overview of the DKF and a methodology for its use. Examples of the DKF in action are given from the Simulation Exploration Experience (SEE), a worldwide project led by NASA that annually gives the opportunity to build collaboratively a distributed simulation of a Moonbase.**

Keywords—Distributed Simulation, High Level Architecture, Modeling, Simulation, Space

I. Introduction

Distributed simulation refers to technologies that enable the execution of a simulation program on distributed computer systems that are geographically distributed and connected by a communication network. Initial research on distributed simulation has been conducted in the military domain where the main objective was on how to achieve model reuse via interoperation of heterogeneous simulation components [1]. The U.S. Department of Defence (DoD) made huge investments in distributed simulation and played a key role in the developing of standards to facilitate interoperability of distributed simulation modules over a computer network. The IEEE 1516-2010 - High Level Architecture (HLA) is a well-known and accepted range of standards that specify the middleware, data interfaces and methods for distributed simulation [2]. Although it was initially developed for purely military applications, it has been widely used in non-military industries for its many advantages related to the interoperability and reusability of distributed simulation components. The development of distributed simulations based on the HLA range of standards remains a challenging task that requires a considerable effort in terms of both time, cost and expertise. The HLA Development software Kit Framework (DKF) aims to facilitate the development of HLA-based distributed simulations. This tutorial presents an overview of the DKF and a methodology for its use. Examples of the DKF in action are given from the Simulation Exploration Experience (SEE), a worldwide project led by NASA that annually gives the opportunity to build collaboratively a distributed simulation of a Moonbase.

II. Distributed Simulation

In the HLA standard a distributed simulation is called a *Federation* and it is composed of several HLA simulation entities, each called a *Federate*, which interact among them by using a *Run-Time Infrastructure (RTI)* that provides standard protocols and services to manage the communications and data exchange among *Federates*. Each *Federation* has a *Federation Object Model (FOM)* that is created in accordance with the *Object Model Template (OMT)* defined by the standard. A FOM contains specifications of *Object classes* (objects are instances – or entities – of object classes that have attributes that can be updated), *Interaction classes* (a message sent among objects that has parameters) and *Data types* (the technical specifications and semantics of attributes and parameters).

The design and development of HLA Federates are generally complex and resource-intensive not only because of the complexity of the HLA standards, but also due to the lack of proper how-to guides and ready-to-use examples. Moreover, developers have to spend a considerable effort to handle common HLA functionalities, such as the control of the simulation time; the synchronization process among Federates; the publish, subscribe and update of *Object class* and *Interaction class* elements along with related coders and decoders; and the management of RTI callbacks and associated exceptions.

The DKF is a general-purpose, domain-independent framework, fully implemented in the Java language and released under the open source policy Lesser GNU Public License (LGPL), which facilitates the development of HLA Federates [3][4]. The DKF has been designed and developed in the context of the research activities carried out within the SMASH-Lab (System Modeling And Simulation Hub - Laboratory) of the University of Calabria (Italy) working in cooperation with the Software, Robotics, and Simulation Division (ER) of the NASA's Lyndon B. Johnson Space Center (JSC) in Houston, Texas, USA. It has been successfully experimented in the Simulation Exploration Experience (SEE)

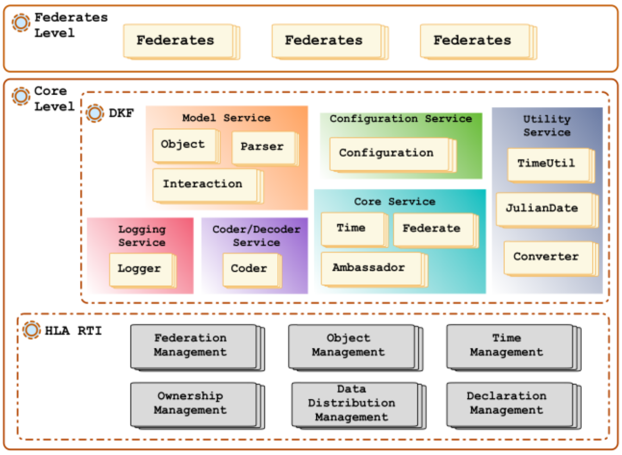

Fig. 1: Architecture of a HLA Federation based on the DKF.

project since the 2015 edition). The SEE project, which has been organized, since 2011, by the Simulation Interoperability Standards Organization (SISO) in collaboration with NASA and other research and industrial partners, aims at providing undergraduate and postgraduate students with practical experience of participation in international projects related to M&S and, especially, to the HLA standard and compliant tools.

III. The HLA Development Kit Framework

The DKF allows to easily conceptualize, define and build an HLA distributed simulation [5]. Developers can focus on domain-specific aspects of a simulation module since, the common HLA functionalities, which are generic across domains, are managed by the core components of the DKF. The *HLA Development Kit Framework* does not represent another implementation of the HLA standard but it is a software framework placed on the available HLA/RTI implementations such as, Pitch RTI, VT MÄK and PoRTIco that provides an additional layer of abstraction suitable to hide the complexity of the standard and facilitate the development of HLA Federates. The Kit comes with all the services needed to develop HLA Federates; *technical documentation* that describes the architecture of the DKF along with its services; *user guide* to support developers in the use of the kit; a set of reference *examples* of HLA Federates created through the use of the DKF; and, some *video-tutorials* that show how to create both the structure and the behavior of an HLA Federate. The main benefit of the DKF is that it allows developers to operate under the paradigm "write once and run anywhere". The architecture is shown in figure 1. It consists of a *Federates Level* (application-specific services) and a *Core Level* (a set of APIs that are used to develop and manage a HLA Federate). In the tutorial, examples from the SEE event will be given to demonstrate how the DKF is used to ease the development of HLA-based distributed simulations.

Developers use the DKF to first create the *HLA ObjectClass* classes with their attributes and coders (as specified in the FOM) by using the annotation mechanism provided by the DKF to build various coders. To link the simulation, an extension of the *DKFAbstractFederate* abstract class must be created as well as implementing methods according to the Federate life cycle that is provided and completely managed by the DKF core components. The Federate Ambassador must then be built. Finally, the *Main* class needs to be created. This class manages the setup of the HLA simulation infrastructure by reading a *.json* configuration file that contains all the parameters necessary to create and manage the HLA Federation execution.

The main differences between federation development without the DKF and with the DKF are summarized below:

Object/Attribute declaration
Without DKF: Manually declared in Federate Class
With DKF: Annotated in Object Class
Interaction/Parameter declaration
Without DKF: Manually declared in Federate Class
With DKF: Annotated in Interaction Class
Attribute/Parameter update
Without DKF: Manually for each element
With DKF: Collectively for each Object/Interaction
Data Types Coders
Without DKF: Explicitly stated in FOM
With DKF: Using DKF coder package
Time advance
Without DKF: Scheduled and managed in simulation
With DKF: Managed by HLA/RTI via DKF

IV. Summary

This paper has introduced the DKF as a toolkit to reduce the complexity of implementing HLA-based distributed simulations. The tutorial will demonstrate its use through several use cases taken from the SEE events over the past three years.

References

[1] Richard M. Fujimoto, "Parallel and distributed simulation systems," *WILEY Ser. Parallel Distrib. Comput.*, vol. 1, p. 162, 2001.

[2] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules," *IEEE Std 1516-2010 (Revision IEEE Std 1516-2000)*, pp. 1–38, Aug. 2010.

[3] A. Falcone, A. Garro, A. Anagnostou, N. R. Chaudhry, O.-A. Salah, and S. J. E. Taylor, "Easing the development of HLA federates: The HLA development kit and its exploitation in the SEE project," in *Proceedings - 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2015*, 2016.

[4] A. Garro, A. Falcone, N. R. Chaudhry, O.-A. Salah, A. Anagnostou, and S. J. E. Taylor, "A Prototype HLA development kit: Results from the 2015 simulation exploration experience," in *SIGSIM-PADS 2015 - Proceedings of the 3rd ACM Conference on SIGSIM-Principles of Advanced Discrete Simulation*, 2015.

[5] A. Falcone, A. Garro, S. J. E. Taylor, A. Anagnostou, N. R. Chaudhry, and O. Salah, "Experiences in simplifying distributed simulation: The HLA development kit framework," *J. Simul.*, vol. 10, no. 37, pp. 1–20, 2016.