Achieving Interoperability in Military Simulation Applications Defining Models for HLA Utilization

Miguel Serna, Marta Beltrán, and Antonio Guzmán

***Abstract*—A generic High Level Architecture (HLA) interface provides solutions for distributed simulation problems but it does not provide use patterns to guarantee universal solutions for all the simulation applications. This is one of the main disadvantages of the HLA standard: the same problem can be solved in different ways.**

The SISO CSPI PDG standard can be considered as the first standard developed to standardize how HLA is used to support distributed simulation. This standard proposes four Interoperability Reference Models (IRM) to identify the typical interoperability problems in manufacturing and logistics environments and to avoid possible confusions using HLA.

Simulators interoperability is considered a critical capability for future joint forces. This paper proposes Interoperability Reference Models for the exclusive use of military applications, taking into account the SISO standard and the specific requirements of military environments. The types A, B, C and D of the SISO standard have been redefined, and a completely new IRM, type E, has been defined for Plan and Order Exchange situations.

***Index Terms*—Distributed Simulation, high level architecture (HLA), interoperability reference model (IRM)**

I. Introduction

The definition of the High Level Architecture (HLA) some years ago made possible interoperability for a wide range of simulation systems and applications [1], [2]. However, as there are different possible ways to use HLA for performing the same task; real and easy interoperability is still not a reality [3].

HLA was born as a military standard [4], but it soon became a very important tool in non-military applications. In fact, the industry has understood that more standardization efforts are needed to obtain true interoperability between the industry commercial-off-the-shelf simulation packages (CSP), defining Interoperability Reference Models in the SISO CSPI PDG standard (SISO Commercial-off-the-shelf Simulation Packages Interoperability Product Development Group standard). These models propose four universal patterns for using HLA in the most common interoperability environments for manufacturing applications [5].

The military simulation community recently has understood the importance of defining their own Interoperability Reference Models, taking into account the SISO CSPI PDG standards and the specific features of the military environments and applications [6].

This paper is focused on constructive simulators interoperability. A constructive simulation involves real people operating simulated systems. Real people make inputs to such simulations, but are not involved in determining the outcomes. This kind of simulation is used for the command and staff training level. The simulation recreates combat environments integrating all primary functions of command and other factors, such as electronic warfare, which are involved in the development of a tactical action or operation. And there is usually a computer database that determines the decisions taken by participants to solve the problems that arise during the exercises. Constructive simulators are based on Command and Control systems whose data exchange model standard is JC3IEDM 3.0.2 [7] defined by MIP (Multilateral Interoperability Program).

The main contribution of this paper is the definition of five new Interoperability Reference Models for military applications taking into account the specific needs of these environments, summarized by the JC3IEDM military standard. Some of these models can be defined adapting the manufacturing models proposed by the SISO CSPI PDG standard, but others must be defined from scratch. Specifically, a type E model has been created for the Plan and Order Exchange situation.

The rest of this paper is organized as follows. Section 2 presents the background needed to understand the presented research, specifically, this section summarizes the main concepts and ideas related to the SISO CSPI PDG standard. Section 3 defines the new Interoperability Reference Models for military applications and finally Section 4 summarizes the main conclusions of this work and some lines for future research.

II. Background

Commercial-off-the-shelf Simulation Packages (CSP) support the development and visualization of simulation models in production and logistics problems. Interoperability between commercial-off-the-shelf simulation packages is very difficult to achieve even with the advent of HLA. Although almost all the CSPs includes a generic HLA interface, different HLA interfaces cannot interoperate properly because there are different approaches to solve the same problems.

Thus, the main issue is to provide use patterns to solve the main interoperability problems with HLA in the same way. The standard SISO-STD-006-2007 2.0 defines a set of

Manuscript received July 10, 2012; revised August 16, 2012.

M. Serna is Captain of the Spanish Army (e-mail: mserca@gmail.com).

M. Beltrán and A. Guzmán are with the Computer Engineering School, Rey Juan Carlos University, 28933 Mostoles, Madrid, Spain (e-mail: marta.beltran@urjc.es; antonio.guzman@urjc.es).

Interoperability Reference Models (IRM) to create a common framework and to allow users to create distributed simulations consisting of CSPs and their models easily.

An IRM is defined as the simplest representation of a problem within an identified interoperability problem type. Each IRM is subdivided into different subcategories. To represent an interoperability problem the following general terms and definitions have been proposed by the standard:

- **Model (M)**: A model describes a real system that is executed by a CSP.
- **Federate (F):** The standard IEEE1516 defines a Federate as "Simulations, supporting utilities or interfaces to live systems". Usually a Federate runs in a single computer.
- **Event (E):** An event marks an instantaneous system transition between two different states in a time T.
- **Time (T):** It represents a specific simulation time in a model. Time is an integer value and in this standard does not have specific units.
- **Entity (e):** An entity is something that is processed. It goes through some queues and activities representing a system. The entity is defined by its attributes.
- **Queue (Q):** It is a queue of entities. Queues have some queuing discipline (LIFO, FIFO, etc).
- **Activity (A**): An activity is an action in a system with a known duration. The activity starts and ends with an event.
- **Resource (R):** A resource is something that is needed by an activity to begin.
- **Data Structure (D):** It is similar to a resource but there are modeling differences in terms of semantics.

The different behavior of these elements is the starting point to define the four interoperability reference model types:

Type A (Entity Transfer), **Type B** (Shared Resource), **Type C** (Shared Event) and **Type D** (Shared Data Structure). These four IRM types standardize the way in which CSPs interoperate using HLA in manufacturing and logistics simulations.

The IRM type A represents the interoperability problem that can occur when transferring an entity from one model to another. The entity transfer is between the Model A1 boundary activity and Model A2 boundary queue. This IRM has three sub-types:

- **IRM Type A1**: General Entity Transfer.
- **IRM Type A2**: Bounded Receiving Element.
- **IRM Type A3**: Multiple Input Prioritization.

The IRM type B represents the case where the state of a resource R shared across models must be consistent. This Reference model has only one sub-type.

The IRM type C represents the problem in which an event E is shared across models. The event must occur at the same simulation time in all the models in order to achieve the distributed simulation consistence.

The IRM Type D is similar to type B but there are some semantic differences because the shared element is a data structure instead of a resource. For example, it can be an inventory record or a bill of materials.

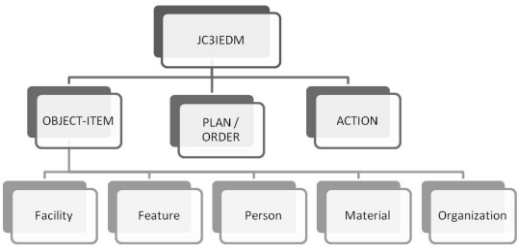

Fig. 1. JC3IEDM main entities

III. Interoperability Reference Models in Military Applications

A. Introduction

In almost all cases the purpose of constructive simulation in military environments is training, mission analysis or planning and operations support. Constructive simulators are characterized today by its major requirements for interoperability, especially in multinational operations, when the constructive simulators of the different countries must interoperate to build a distributed simulation. Interoperability has been defined by NATO as "the ability of systems, units or forces to provide or accept services from other systems, units or forces and use these services to operate together in an effective manner" [8].

Each country develops its simulators using an already developed RTI or creating its own infrastructure. Although most of them follow the IEEE1516 standard, which should ensure that they are fully interoperable, HLA has demonstrated to be a very complex standard that is not yet fully mature. The standard ambiguities lead to different interpretations in the development of RTIs and to compatibility problems across constructive simulators. Thus, without a common standard approach, interoperability hardly never is plug-and-play. The different armies use to build their simulators from scratch when interoperability is required.

In military applications the models and simulations are not based on CSPs but on general purpose programming languages. And data are shared following the JC3IEDM data model (figure 1).

With all this information the IRM types necessary for military applications can be easily deduced:

- **Object-Item**: A military Object-Item defined in the JC3IEDM 3.0.2 standard can be an Entity, a Resource or a Data Structure in interoperability environments. Therefore, the IRMs type A, B and D of the SISO standard may be adapted for military applications.
- **Plan and Order**: The SISO Standard does not define any element related to the military Plan and Order concept, not present in manufacturing problems.
- **Action**: It is an event that can change the system state and the IRM type C of the SISO standard may help to solve its related interoperability problems.

Fig. 2. IRM Type A

As a conclusion, military Interoperability Reference Models for constructive simulators can be defined based on the SISO standard IRMs for Object-Items (IRMs type A, B and D) and for Actions (IRM type C). In order to handle the Plan and Order interoperability problems, a completely new IRM type E must be defined.

B. IRM Type A: Entity Transfer

In a tactical military environment the entity is defined as an Object-Item in the JC3IEDM model. SISO defines three subtypes for this IRM in the industrial standard. To define the military IRM type A these three sub-types can be defined again [6].

1) IRM Type A1: General Entity Transfer

This IRM subtype describes the following problem: an entity e1 (defined by an Entity Transfer Specification or ETS) leaves activity A1 in model M1 (Federate A) at time T1 and arrives to queue QB in model M2 (Federate B) at time T2 (figure 2). Always with $T1 \leq T2$.

This subtype can be defined with five different events:

- **Event 1**. Federate A publishes the entity ETS A that is going to be transferred and Federate B publishes the entity that is going to be received, ETS B.
- **Event 2**. Federate A subscribes to the ETS B (entity published by Federate B at event 1) and Federate B subscribes to ETS A (entity published by Federate A **at event 1).**
- **Event 3**. Federate A updates ETS A and its storage resources. Then, the Federate B receives the callback.
- **Event 4**. Federate B updates ETS B and its storage resources with the callback received at event 3. Then, Federate A receives the callback and it checks that ETS B has been updated correctly.
- **Event 5**. Federate A and Federate B unpublish their entities.

2) IRM Type A2: Bounded Receiving Element

This IRM subtype describes the following problem: an entity e1 leaves activity A1 in model M1 (Federate A) at time T1 and arrives to queue QB in model M2 (Federate B) at time T2 but the queue has no space to store the received entity. Always with $T1 \leq T2$.

These five events are needed to successfully complete the entity transfer in this case:

- **Event 1**. Federate A publishes the entity ETS A that is going to be transferred and Federate B publishes the entity that is going to be received, ETS B.
- **Event 2**. Federate A subscribes to the ETS B (entity published by Federate B at event 1) and Federate B subscribes to ETS A (entity published by Federate A at event 1).
- **Event 3**. To this point, the events have been the same that in the previous sub-type. But now, Federate A makes a reservation in the queue QB. Federate B receives the callback with this reservation.
- **Event 4**. Federate B checks if there is enough space in the queue QB to perform the entity transfer. If there is not enough space to receive the transferred entity, Federate B blocks the entity transfer and Federate A receives the callback of this blocking.
- **Event 5**. When the queue QB has enough space to receive the entity, Federate B unblocks the entity transfer and this leads to the event 3 of the IRM sub-type A1.

3) IRM Type A3: Multiple Input Prioritization

This IRM subtype describes the following problem: an entity e1 is transferred from Federate A to Federate B and an entity e2 is transferred from Federate C to Federate B too. Both entities arrive at the same time T to Federate B.

Six events are needed to manage this situation in a military environment:

- **Event 1**. Federates A and C publish the entities that are going to be transferred, ETS A and ETS C.
- **Event 2**. Federate B subscribes to ETS A and ETS C (entities published by Federate A and Federate C at event 1).
- **Event 3**. Federate A and Federate C make a reservation in the Federate B queue. Each reservation has the next fields:

  Source Federate: The federate sending the entity.
  Entity Volume: The transferred entity volume.
  Logical Reservation Time: The logical federation time the reservation is made at.
  Priority: It is the entity priority, usually depending of the kind of sending federate.

  Federate B receives the callback with these reservations and save it in the LLD (Load List Data).
- **Event 4**. Federate B publishes two entities ETS BA and ETS BC (one for each federate that made a reservation at event 3).
- **Event 5**. Federate A subscribes to ETS BA and Federate C subscribes to ETS BC (entities published by Federate B at event 4).

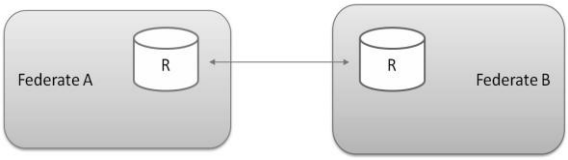

Fig. 3. IRM Type B

- **Event 6**. Federate B loads the first entry in its Load List Data and unblocks the transfer with the source federate. This Federate receives the callback to start the entity transfer from event 3 of the IRM sub-type A1.

C. IRM Type B: Shared Resources

This IRM type describes the following problem: a resource(described with a Resource Shared Specification or RSS) is shared by two Federates (figure 3). In a tactical military environment this resource is defined as an Object-Item in the JC3IEDM model. This model can be defined with five different events:

- **Event 1**. Federate A publishes the resource RSS A that is going to be shared and Federate B publishes the resource RSS B similar to the resource published by Federate A.
- **Event 2**. Federate A subscribes to the RSS B (resource published by Federate B at event 1) and Federate B subscribes to RSS A (resource published by Federate A at event 1).

- **Event 3**. Federate A updates RSS A and its storage resources. Then, the Federate B receives the callback.
- **Event 4**. Federate B updates RSS B and its storage resources with the callback received at event 3. Then, the Federate A receives the callback and checks that RSS B has been correctly updated.
- **Event 5**. Events 3 and 4 are repeated while the resource is shared. To finish this IRM Federate A and Federate B unpublish their resources.

D. IRM Type C: Shared Events

This IRM type describes the following problem: an event (described with an Event Information Package or EIP) is shared by two Federates that have an interest in it. In a tactical military environment this event is specified with an Action defined by the JC3IEDM, Action-Task if it has been programmed or Action-Event if it is unexpected.

The Synchronization Table defined by the Federation FOM use tags to keep track of the logical time when each event occurs. Federate A is supposed to be the owner of the event because it is the federate that programs the event or that observes it, that is the reason why this federate publishes and updates the information related to the event. This IRM can be defined with five events:

- **Event 1**. Federate A and Federate B advance their logical time to the time when the event occurs to start the IRM. This event time is specified with a tag in the Synchronization Table.
- **Event 2**. Federate A publishes the EIP that is going to be shared.
- **Event 3**. Federate B subscribes to the EIP (information published by Federate A at event 2).
- **Event 4**. Federate A updates the EIP. Then, Federate B receives the callback.
- **Event 5**. Events 3 and 4 are repeated while the event is shared. To finish this IRM Federate A unpublishes its EIP.

E. IRM Type D: Shared Data Structures

This IRM type describes the following problem: a data structure (described with a Data Structure Specification or DSS) is shared by two Federates. In a tactical military environment this structure is specified again with an Object-Item defined by the JC3IEDM.

The IRM type D may be confused with the IRM type B. But there are important semantic differences between them, type D is more flexible than the IRM B. A resource has a well-defined structure but a data structure can contain different kinds of information. This IRM type can be defined with five different events:

- **Event 1**. Federate A publishes the structure DSS A that is going to be shared and Federate B publishes the structure DSS B, similar to the structure published by Federate A.
- **Event 2**. Federate A subscribes to the DSS B (Data Structure published by Federate B at event 1) and Federate B subscribes to DSS A (Data Structure published by Federate A at event 1).
- **Event 3**. Federate A updates DSS A. Then, the Federate B receives the correspondent callback.
- **Event 4**. Federate B updates DSS B with the callback received at event 3. Then, the Federate A receives the callback and checks that DSS B has been correctly updated.
- **Event 5**. Events 3 and 4 are repeated while the data structure is shared. To finish this IRM Federate A and Federate B unpublish their data structures.

F. IRM Type E: Plan and Order Exchange

The JC3IEDM defines the concept Plan and Order and it does not fit any of the previously defined IRMs, because it is not an Object-Item or Event and because the interoperability problem can not be described as a transfer or a sharing. In fact, it is a sharing followed by a transfer.

When a unit A takes part of an operation it must prepare a plan. The plan expresses the commander's maneuver plan and often contains different annexes. The core of the plan is prepared by the staff of the unit A. This first draft is subsequently amended by subordinate units, since each annex of the plan specifies the specific work of each subordinate unit in the operation [7].

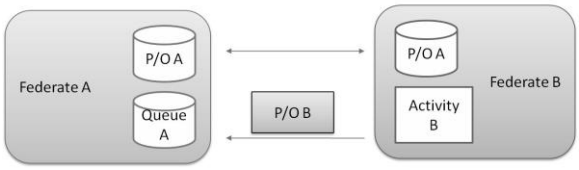

Fig. 4. IRM Type E

In terms of interoperability, the unit A shares a core plan with a subordinate unit that is in the plan environment. The subordinate unit elaborates its annex for this plan and sends it back to the unit A for the commander approval. When unit A receives the annex, and the commander approves it, the shared core plan is updated with this annex (figure 4).

Therefore the IRM type E describes the following problem: a Plan and Order (described with a Plan Exchange Specification or P/O) is shared by two units (unit A and unit B, subordinated to the first one), then, unit B must send its annex for the plan to the unit A.

This IRM has two different phases. The first phase describes a P/O sharing and the second describes a P/O transfer. Nine events are needed to define the model in this case:

- **Phase 1: Event 1**. Federate A publishes the P/O A that is going to be shared.
- **Phase 1: Event 2**. Federate B subscribes to the P/O A (resource published by Federate A at event 1).
- **Phase 1: Event 3**. Federate A updates P/O A. Then, the Federate B receives the callback.
- **Phase 2: Event 4**. Federate B publishes the P/O B that is going to be transferred. This plan P/O B, is based in the shared plan P/O A, in fact, it is an annex for it.
- **Phase 2: Event 5**. Federate A subscribes to the P/O B (resource published by Federate B at event 4).
- **Phase 2: Event 6**. Federate B updates P/O B. Then, Federate A receives the callback.
- **Phase 2: Event 7**. Federate A updates P/O A with the annex for the plan (information received in the callback at event 6). Then, the Federate B receives the callback.

- **Phase 2: Event 8**. Federate B unpublishes the P/O B. The phase 2 will be repeated whenever Federate B needs to update the P/O A.
- **Phase 2: Event 9**. To finish the IRM the Federate A unpublishes the shared plan P/O A.

IV. Conclusions

As advances in technologies and standards continue to boost our ability to solve the typical interoperability problems caused by the HLA specification, military forces have understood the need of incorporating these advances to their applications.

The four standardized HLA use patterns defined by the SISO CSPI PDG standard cannot be used in military applications. These use patterns, called Interoperability Reference Models (IRM), encapsulate HLA functionalities in a user-friendly way preventing the problems caused by the complexity of the HLA standard and by the ambiguities and insufficiencies of its specification in order to achieve real interoperability among manufacturing and logistics simulators.

There are important differences between the typical interoperability problems in these applications and the problems that arise in military applications. Furthermore, the military models are not based on COTS packages such as the manufacturing and logistics simulators.

This paper proposes five IRMs specifically defined for military applications taking into account the typical interoperability environments for constructive simulators. The IRMs type A, B, C and D are redefinitions of the SISO CSPI PDG models considering the JC3IEDM data model, but a completely new model, the IRM Type E, has been defined for Plan and Order Transfer situations.

These IRMs have been tested on a real world interoperability scenario, implementing all the models with the Portico RTI developed with the participation of the Australian Defense Simulation Office (ADSO) [9], [10], and demonstrating how the new IRMs can be easily implemented on top of real existing RTIs allowing plug-and-play military distributed simulations.

References

[1] *IEEE standard for modelling and simulation High Level Architecture (HLA) - Framework and rules*, IEEE Standard 1516-2000.

[2] *IEEE standard for modelling and simulation High Level Architecture (HLA) - Federate interface specification*, IEEE Standard 1516.1-2000.

[3] S. Taylor, N. Mustafee, S. Strassburger, J. Ladbrook, S. Turner, and M. Low, "The SISO CSPI PDG Standard for commercial off-the- shelf simulation package Interoperability Reference Models," in *Proc. of the 2007 Winter Simulation Conference*, 2007, pp. 594–602.

[4] K. Morse, M. Lightner, R. Little, B. Lutz, and R. Scrudder, "Enabling simulation interoperability," *IEEE Computer*, vol. 3, no. 3, pp. 115–117, 2006.

[5] *Standard for commercial-off-the-shelf simulation package Interoperability Reference Models*, SISO Standard 006-2007-DRAFT-2.0.

[6] M. Serna, F. Sevillano, M. Beltran, and A. Guzman, "Defining the entity transfer interoperability reference model for military applications," in *Proc. of the 2010 Spring Simulation Conference (MMS)*, 2010, pp. 354-361.

[7] *The Joint C3 Information Exchange Data Model JC3IEDM*. MIP DMWG Working Group Standard.

[8] NATO policy for interoperability. NATO UNCLASSIFIED [Online]. Available: http://www.nato.int/docu/interoperability/html_en/interoperability01.html (last visited September 2009)

[9] Portico developer documentation. Portico Open Source Project [Online]. Available: http://porticoproject.org/ (last visited June 2009)

[10] Portico user documentation. Portico Open Source Project [Online]. Available: http://porticoproject.org/ (last visited June 2009)

M. Serna is Captain of the Spanish Army (Specialty of Artillery), he received the master degree in Information Technology and Communication Systems for Defense in 2007, the master degree in Advanced Hardware and Software Systems in 2010 and the Ph.D. from the Computing Department, Rey Juan Carlos University, Madrid, Spain, in 2011.

His current research interests are related to distributed simulation, C4 systems, exchange technologies and interoperability in military environments.

Dr. Serna collaborates actively with the GAAP research group at the Rey Juan Carlos University and is the national representative in different groups of the MIP program.

M. Beltrán received the master degree in electrical engineering from University Complutense of Madrid, Spain, in 2001, the master degree in industrial physics from UNED, Spain in 2003 and the Ph.D. degree from the Computing Department, Rey Juan Carlos University, Madrid, Spain in 2005.

She is currently working with this department as a lecturer. Her current research interests are high performance computing, heterogeneous systems and interoperability.

Dra. Beltrán is the leader of the GAAP research group and has extensively published in high-quality national and international books, journals and conference proceedings in the areas of computer architecture and parallel and distributed systems.

A. Guzmán received the master degree in applied physics from University Autonoma of Madrid, Spain, in 1999 and the Ph.D. degree from the Computing Department, Rey Juan Carlos University, Madrid, Spain in 2006.

He is a coauthor of more than 25 papers in international conference proceedings and journals and his current research interests are high performance computing, interoperability and security.

Dr. Guzmán is currently an associate professor in the Computing Department of Rey Juan Carlos University.