2017

Design of a Framework for Sharing and Generating Combat Damage Assessment(CDA) of a HLA/RTI Federation

Hongseon Park
University of Central Florida

DESIGN OF A FRAMEWORK FOR SHARING AND GENERATING COMBAT DAMAGE ASSESSMENT(CDA) OF A HLA/RTI FEDERATION

by

HONGSEON PARK

B.S. Korea Military Academy, 2007

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2017

Major Professor: Gene Lee

ABSTRACT

In this paper, a new framework for sharing Combat Damage Assessment(CDA) is proposed to find out the differences of each CDA system between military combat units belonging to their own federate in a HLA/RTI federation. When there are engagements in a battle among combat units belonging to their own federate in the HLA/RTI federation, each result of damage assessments is very different. This affects the HLA/RTI federation's confidence and needed to be overcome because it is also one of the major issues to generate reliable engagement data. Also, a RTI can generate only qualitative data about combat damage while quantitative data can be useful.

Therefore, the new framework for sharing CDA and generating quantitative CDA data is proposed to solve the problems with a CDA Module of one federate which is considered to have a standard engagement logic. The new framework is also tested through two case studies by using two federates of a HLA 1516 / MÄK RTI federation. This new framework will be helpful to increase the interoperability in a HLA/RTI federation, provide an environment in which all developers can reuse the proposed new framework, and generate quantitative engagement data through this new framework.

This work is dedicated to my parents and wife

Without their support, this work would have not been possible.

ACKNOWLEDGMENTS

I would like to express my profound gratitude to my supervisor Professor. Gene Lee for the guidance and support I received. At every stage help was available to deal with obstacles both large and small. Appreciation is also extended to my thesis committee, Professor. Luis Rabelo, and Professor. Ahmad Elshennawy for their valuable time and contributions.

I would also like to thank Dr. Sunghan Song, Dr. John Pastrana, Maj. Sookyoung Kim for their guidance, valuable comments and profound suggestions they offered towards my research and this thesis.

Special thanks to Yongkun Yoo, Kyungjin Park, Jaeho Kim for their friendship and support; I truly enjoyed the time we spent together. I'd like to express my gratitude to Maj. Wonil Jung who is a senior officer in my lab.

Most importantly, my deepest gratitude goes to my father, my mother and my wife who always proactively have supported me through all aspects in life, especially for the pursuit of this degree. Without my family, these accomplishments would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Research Motivation

This research focuses on development of a new framework in the context of sharing and generating Combat Damage Assessment(CDA) in a HLA/RTI federation. Each result of damage assessments is very different when there are engagements in a battle among combat units belonging to their own federate in the HLA/RTI federation, if each of the federates has their own engagement logic. This affects the HLA/RTI federation's confidence and needed to be overcome because it is also one of the concerning issues for virtual simulators to display correct results on the screen and generate reliable engagement data. Also, a RTI can generate only qualitative data about combat damage while quantitative data can be useful. Therefore, a new framework for sharing and generating CDA is proposed to solve the problems with a CDA Module of one federate which is considered to have a standard CDA logic. The new framework is also exercised through two case studies by using virtual and constructive simulations.

Combat Damage Assessment(CDA) is closely related to Combat Power(CP). The definition of CP is relative but it is needed to define the Combat Power in a way that is relevant to this research. Millett and Murray defined the Combat Power as "the ability to destroy the enemy while limiting the damage that he can inflict in return" (Millet et al., 1986).

The result of CP can be defined by CDA, so sharing one standard CDA logic is necessary to increase interoperability and generate reliable engagement data.

There are three supplementary explanations for the problem statement. Figure 1-1 and 1-2 show examples of the different engagement logics between two federates. Figure 1-1 is for SIMbox and 1-2 is for VR-Forces.

```
float killRadius=0.0f;
GET_ENTITY_ATT(explodedEtt, EntityWorld::ATT_KILL_RADIUS, killRadius);

float distance = (explotionPos-myPos).norma();

float damageFactor=0.0f;
GET_ENTITY_ATT(explodedEtt, EntityWorld::ATT_DAMAGE_FACTOR, damageFactor);

float armorFactor = 0.f;
GET_ENTITY_ATT(_pOwnerEntity,EntityWorld::ATT_ARMOR_FACTOR, armorFactor);

int damage = ( fKillradius – distance ) * fDamageFactor / fKillradius * (( 100.0f – fArmorFactor ) /100.f );
```

Figure 1-1. Engagement logic for SIMbox

```
(damage-table
  (front
    (angle-of-incidence
      (angle  0.5236) ;; 30 degrees
      (range-determinant
        (coefficients
          (catastrophic-kill -5.0000E-008   0.0000   1.0000)
        )
      )
    )
    (angle-of-incidence
      (angle  1.0472) ;; 60 degrees
      (range-determinant
        (coefficients
          (catastrophic-kill -5.0000E-008   0.0000   1.0000)
        )
```

Figure 1-2. Engagement logic for VR-Forces

In the engagement logic of SIMbox, the damage is decided by using Damage Factor, Armor Factor, and Kill Radius. Damage value is calculated as a quantitative format from 0 to 100.

In the engagement logic of VR-Forces, the damage is decided by using Probability of Hit(POH), Damage Model, and Armor Model. Damage value is determined as 0 (None), 1 (Slight), 2 (Moderate), 3 (Destroyed).

Figure 1-3 shows the message communication in a HLA/RTI federation and another explanation for the problem statement.

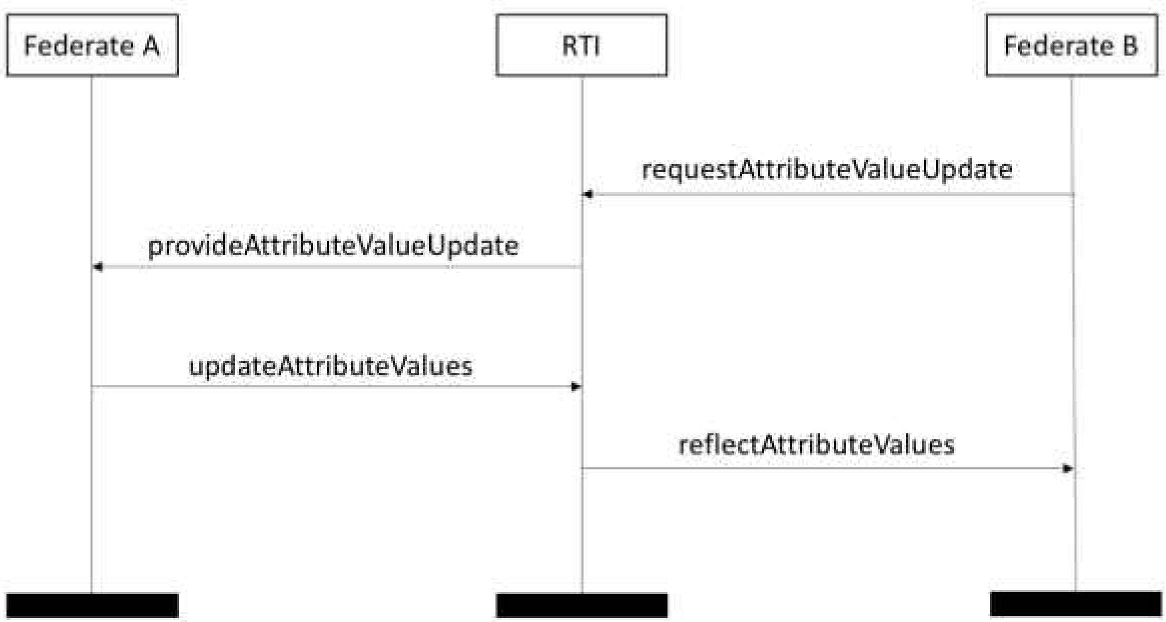

Figure 1-3. Communication in a HLA/RTI federation

"The Request Attribute Value Update service shall be used to update the values of specified attributes. When this service is used, the RTI shall solicit the current values of the specified attributes from their owners using the 'Provide Attribute Value Update' service"(IEEE, 2010).

Figure 1-4 shows the damage information on the Object Model Template(OMT) in an HLA/RTI federation.

```
<enumeratedData name="DamageStatusEnum32" nameNotes="18" representation="HLAinteger32BE" semantics="-NULL-">
<enumerator name="NoDamage" values="0"/>
<enumerator name="SlightDamage" values="1"/>
<enumerator name="ModerateDamage" values="2"/>
<enumerator name="Destroyed" values="3"/>
</enumeratedData>
```

Figure 1-4. Proposed the new CDA Framework

DamageStatus is shown as 0 (No Damage), 1 (Slight Damage), 2 (Moderate Damage), 3 (Destroyed), so quantitative engagement data cannot be generated through Runtime Infrastructure(RTI)

1.2 Research Questions

It is obvious that complex work and serious highly skilled effort are required for developing interoperation of simulations (Dahmann et al., 1999). There are difficulties of implementing consistent CDA in a HLA/RTI federation and generating reliable engagement data because of differences of CDA between combat units belonging to their own federate in a HLA/RTI Federation. Therefore, some questions arise for this research.

Q1. Why and when do a HLA/RTI federation require the proposed new CDA Framework?

Q2. How can the federation share one standard engagement logic together?

Q3. How can the federation generate quantitative engagement data after sharing a CDA in an HLA/RTI federation?

1.3 Research Objectives

The objectives of this research are:

- To develop in depth comprehension about HLA/RTI;
- To develop a new framework for sharing and generating CDA data; and
- To implement case studies for the new CDA framework of a HLA/RTI federation.

Figure 1-5 depicts the CDA conceptual structure. In this structure, the standard federate which is considered to have a standard CDA logic was assumed to be the Federate B.

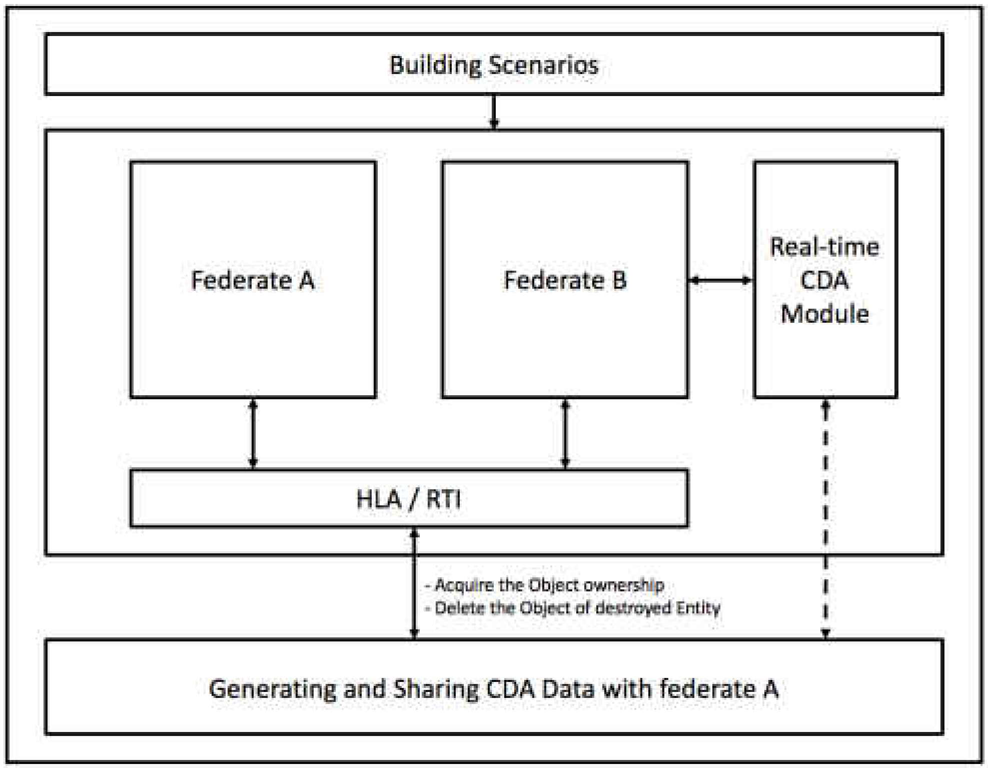

Figure 1-5. Proposed the new CDA Framework

1.4 Contribution

The contributions from this research work include the following:

- This research provides a new framework to share and generate the Combat Damage Assessment of a HLA/RTI federation. Reliable Combat Damage Assessment(CDA) is an important factor especially for military virtual simulators because the purpose of military virtual simulators is to develop a user's operational and technical skills to win combats.
- It is also meaningful to generate quantitative engagement data because the quantitative

engagement data can be used for feedback to develop a user's combat skills in detail.

- In addition, this new framework is also an unconventional approach to solve interoperability problems in a HLA/RTI federation. The framework does not follow the HLA rules but it can be adoptable in specific cases.

1.5 Thesis Overview

This research has six overall chapters. The motivation and the context of this research are described in Chapter 1. In Chapter 2, the background of this research topic is explained. In Chapter 3, the research methodology on development of the Combat Damage Assessment framework of a HLA/RTI federation is introduced and components of the new framework are described in detail. Two case studies are used to prove the framework in Chapter 4 and Chapter 5. At last, research summary, limitations, and future research are discussed in Chapter 6.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The main objective of this chapter is to provide a review about basic concept of M&S, HLA and RTI before discussing about the proposed new framework. More specifically, Chapter 2 describes the other approach to solve the problems of a federation which has different engagement logics and studies related to a comparison to gain knowledge about similar and different aspects which are necessary for the development of the new CDA framework.

2.2 Modeling and Simulation (M&S)

Modeling and simulations have always been a major part of human history. Modeling can be defined by “the process of producing a model; a model is a representation of the construction and working of some system of interest.” and simulation can be defined as “A simulation of a system is the operation of a model of the system” simply (Maria, 1997).

Modeling and Simulation(M&S) is used to simulate real system’s objectives by modeling components, simulation steps and process and implementing produced models in a time flow. The area of M&S was extended from War Game to Task Request, Weapon Acquisition, Decision, Analysis and Military Training. Also, efficient operation of massive simulation and interoperability between complex systems has been studied.

2.2.1 *Stand-alone Simulation and Federated Simulation System*

Simulation systems can be divided into two kinds of systems. They are stand-alone simulation systems and federated simulation systems. The stand-alone simulation has a parallel simulation environment. Most simulation systems were developed as a stand-alone simulation system in an initial phase. Figure 2-1 shows the stand-alone simulation system.

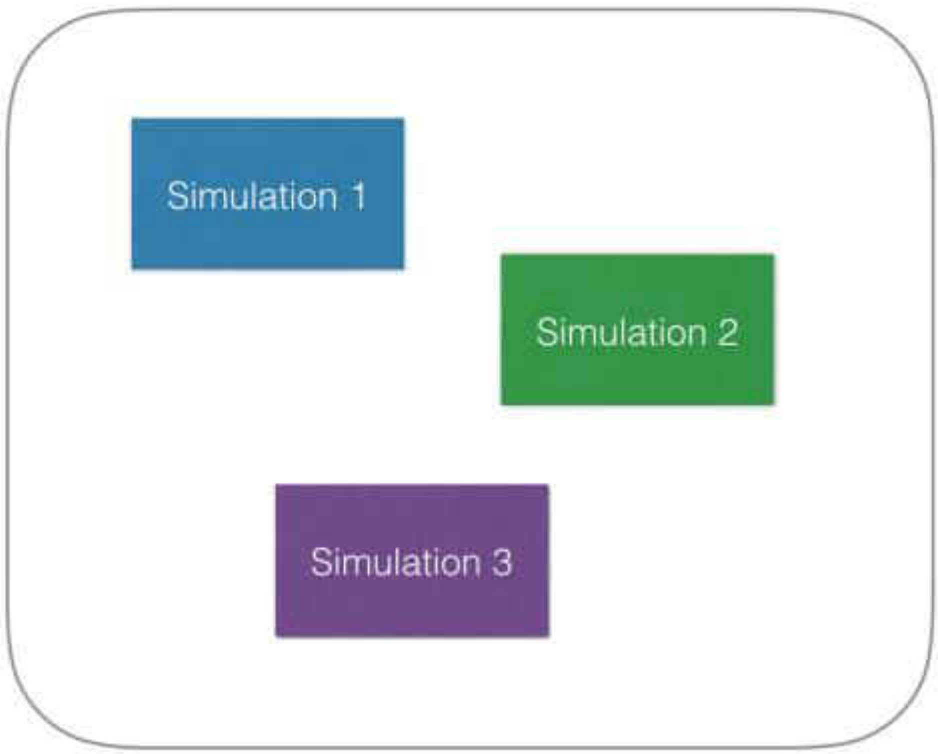

Figure 2-1. Stand-alone simulation example

As time went by, the federated simulation systems were developed to take advantages about reuse and budget. The standard example of federated simulation systems is High Level Architecture(HLA). The HLA uses a Run Time Infrastructure(RTI) software to interface between federates. Figure 2-2 shows the example of a federated simulation system.

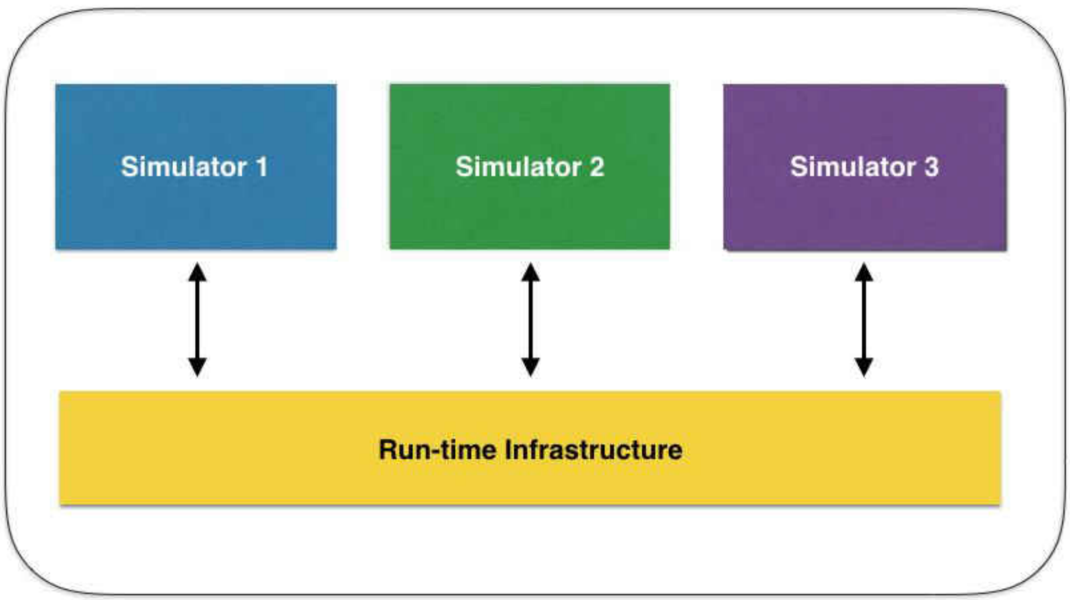

Figure 2-2. Federated simulation example

2.2.2 *Live, Virtual and Constructive Simulations*

Live, Virtual and Constructive are three types of distributed simulations and they also can be three different types for the simulation systems of military warfare. The three types of classifying simulation systems are broadly used (*MODELING AND SIMULATION (M&S) MASTER PLAN*, 1995).

Table 2-1. Types of distributed simulations

Category	Live	Virtual	Constructive
People	Real	Real	Simulated
System	Real	Simulated	Simulated

2.3 High Level Architecture(HLA)

HLA (High Level Architecture) is a software architecture which can be reusable for execution of distributed simulation applications. HLA consists of rules, interface specification and object model template.

HLA has ten rules that governs how federates and federations are constructed.

1. Federations shall have a HLA Federation Object Model (FOM), documented in accordance with the HLA Object Model Template (OMT).
2. In a federation, all representation of objects in the FOM shall be in the federates, not in the run-time infrastructure (RTI).
3. During a federation execution, all exchange of FOM data among federates shall occur via the RTI.
4. During a federation execution, federates shall interact with the run-time infrastructure (RTI) in accordance with the HLA interface specification.

5. During a federation execution, an attribute of an instance of an object shall be owned by only one federate at any given time.

6. Object Model (SOM), documented in accordance with the HLA Object Model Template (OMT).

7. Federates shall be able to update and/or reflect any attributes of objects in their SOM and send and/or receive SOM object interactions externally, as specified in their SOM.

8. Federates shall be able to transfer and/or accept ownership of attribute dynamically during a federation execution, as specified in their SOM.

9. Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of attributes of objects, as specified in their SOM.

10. Federates shall be able to manage local time in a way which will allow them to coordinate data exchange with other members of a federation.

The interactions between federation and federates are governed by an interface specification with the Runtime Infrastructure. Object Model Template(OMT) is a role to document major information about simulations. No single, monolithic simulation program can satisfy all people's needs. This is the premise of the HLA and the reason to adopt a reusable HLA comprising simulation federations.

HLA has been developed from Aggregate Level Simulation Protocol(ALSP) and Distributed Interactive Simulation(DIS) in the history and focused to improve the interoperability in distributed simulations. However, supporting the semantic interoperability in a HLA was not considered, so Simulation Interoperability Standard Organization(SISO) developed a Real-time Platform Reference Federation Object Module(RPR-FOM) to support it such as velocity, location, and damage status.

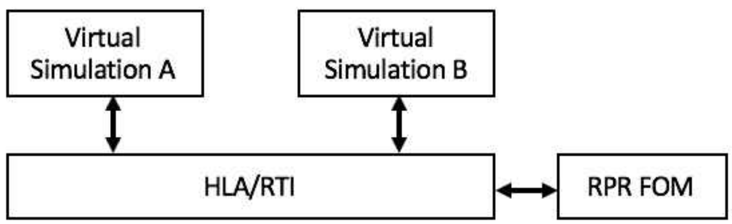

Figure 2-3. The structure of PRP FOM

RPR FOM is a kind of Common Foundation Reference FOM(CFR-FOM) that is a collection of abstract data used in a federate and is a set of object attributes and interactions used in federations generally. PRP FOM is to organize the Protocol Data Units(PDUs) of DIS by HLA objectives and interaction class. Therefore, PRP FOM support a data format which help to interconnect with real-time simulations based on platforms such as fighters, vessels, units and weapons developed in a DIS environment. PRP FOM also follows all HLA rules and services.

2.4 Runtime Infrastructure(RTI)

HLA federation is one set of different federates and each of the federates can interact through a Runtime Infrastructure(RTI) and a Federation Object Model(FOM). Each federate can be such applications as:

1. Simulations
2. Data Logger like MAK Data Logger
3. Passive(Stealth) Viewers
4. Live Entity Surrogates

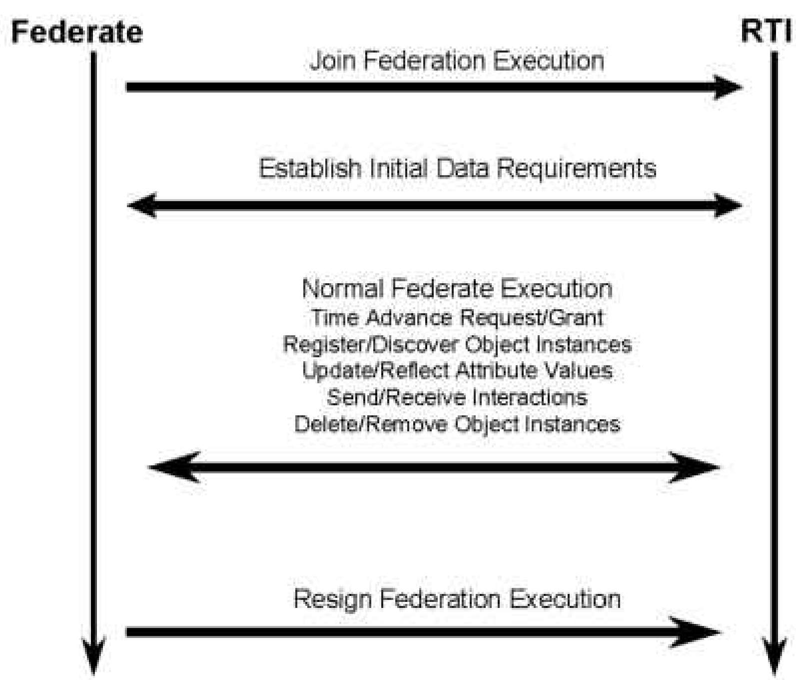

Figure 2-4. Overall view of federate-to-RTI relationship

Source: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Federate Interface Specification

Federates can communicate together by services from RTI. RTI is used to support for the measurement of interoperability. Also, Interoperability request the commonality between FOMs of participated simulations. RTI is a software implementation of specified services in the HLA interface specification. RTI is also a software aggregate to provide commonly required services to simulation systems. Figure 2-5 depicts the concept of a Runtime Infrastructure (RTI).

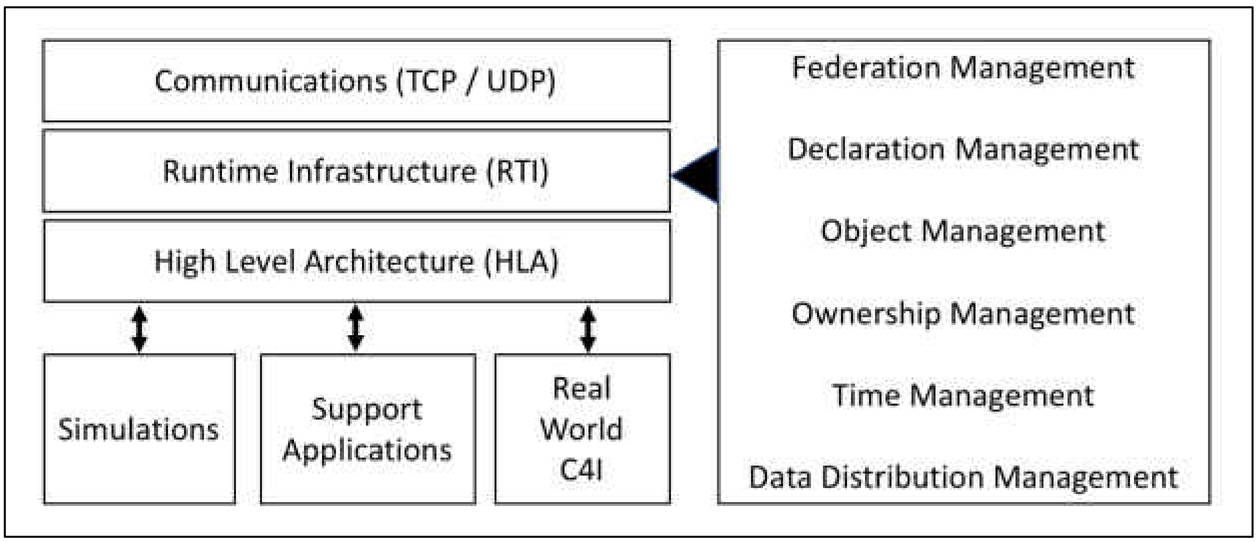

Figure 2-5. The concept of Runtime Infrastructure(RTI)

These commonly required services can be divided into six parts. These are Federation Management, Declaration Management, Object Management, Ownership Management, Time Management, and Data Distribution Management.

Table 2-2 summarized the six major required services of a Runtime Infrastructure.

Table 2-2. The six major required services of Runtime Infrastructure

Components	Description
Federation Management	• Providing services: Generation or degeneration of federations • Defining the implementation of federations: Implementation generation and federate join or resign • Operating federation: Check point generation, restoration and synchronization.
Declaration Management	• Providing methods to achieve efficient data exchange between federates • Declaring publish or subscribe of object attributes between federates
Object Management	• Using for real exchange of data: registration of new instance of object class or update of instance's attributes • Using for subscription of updating value of other federates' interconnections and instances' attributes • Using for detection of new instances • Using for controlling of data transfer method
Ownership Management	• Managing of updating responsibility and transferring of object attributes between federates • Providing services to acquire ownership and to divest ownership to other federates • Managing of mutually exclusive authorities when sharing about updating responsibility and deleting authority of object attributes between federates
Time Management	• Controlling logical time process of all federates • Deciding the time management degree of each of the federates
Data Distribution Management	• Using for reducing useless transmission or receive between federates in a federation • Adopting Region concept

2.4.1 *Object Management: Request Attribute Value Update*

To update specified attributes' values, the Request Attribute Value Update service should be used. By using this service, the RTI can get the desired values of the specified attributes by using the "Provide Attribute Value Update" from other federates which has ownership of the attributes service (IEEE Std 1516.1-2000). Message communication method is used to request and update the values between each federate and RTI.

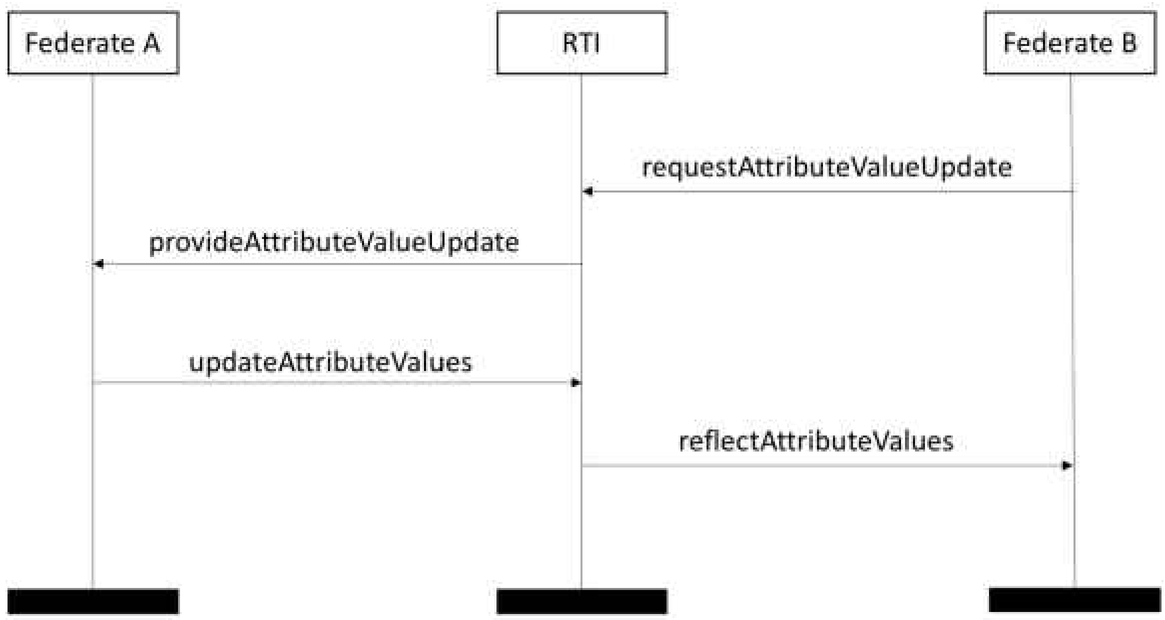

Figure 2-6. Update Attribute Value Update sample

2.4.2 *Ownership Management*

Ownership management is one of the managements for RIT services. The ownership management is related to control and interconnect each attribute's specified values in a HLA federation. By following this management principle, the ownership of instance attributes can be transferred by each federate in a HLA federation and RTI services. Figure 2-7 depicts the method to establish ownership of instance attributes in a HLA federation.

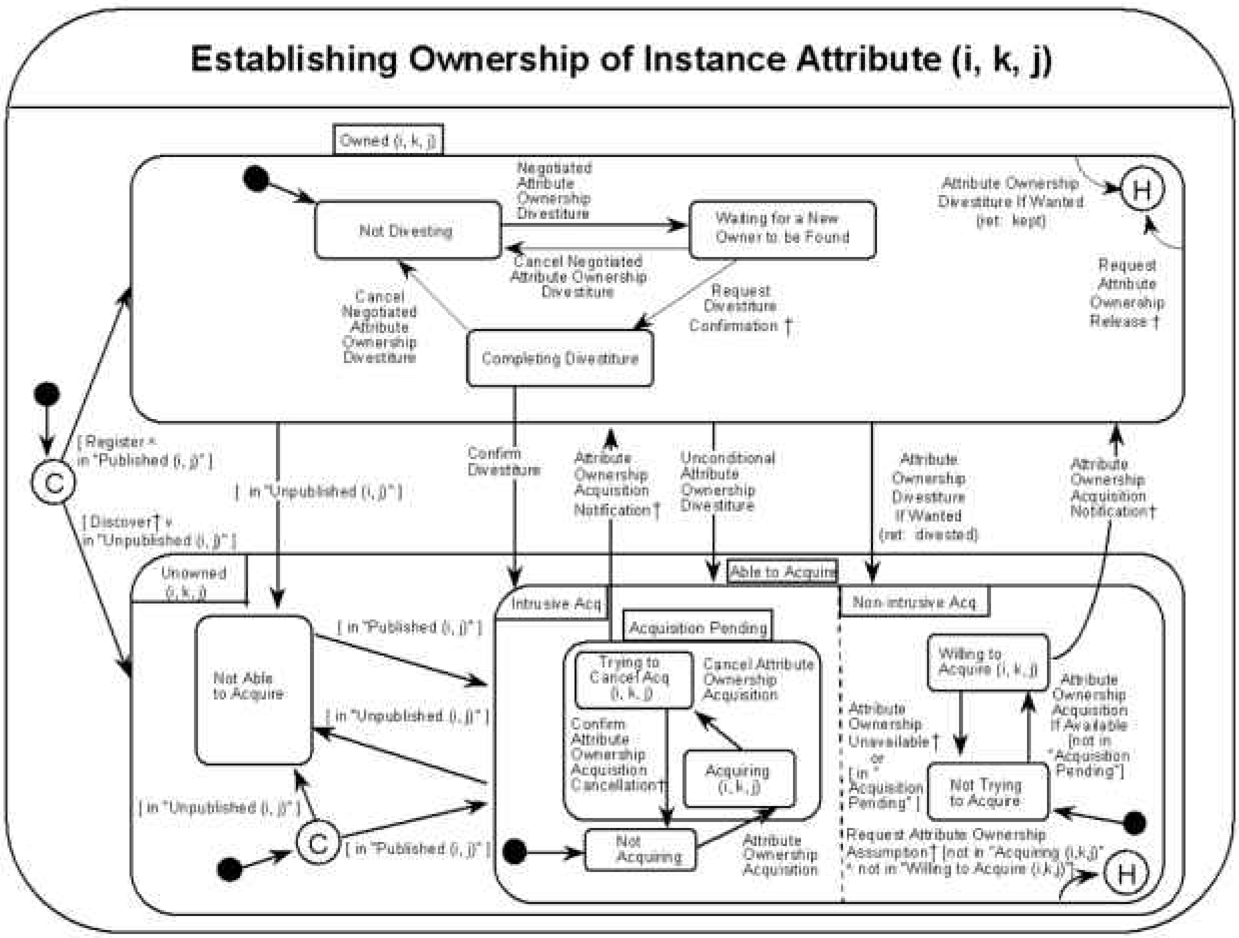

Figure 2-7. Establishing ownership of instance attribute (i, k, j)

Source: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HL A) Federate Interface Specification

2.5 Decision Tree Method

The decision tree method was suggested to overcome the difference of combat damage assessments between combat units belonging to their own model in Combined Arms Integrated Interoperability System(CAIIS) (Moon, 2011).

The CAIIS consists of five wargame models and the study focused on two major models among them. Figure 2-8 describes the CAIIS.

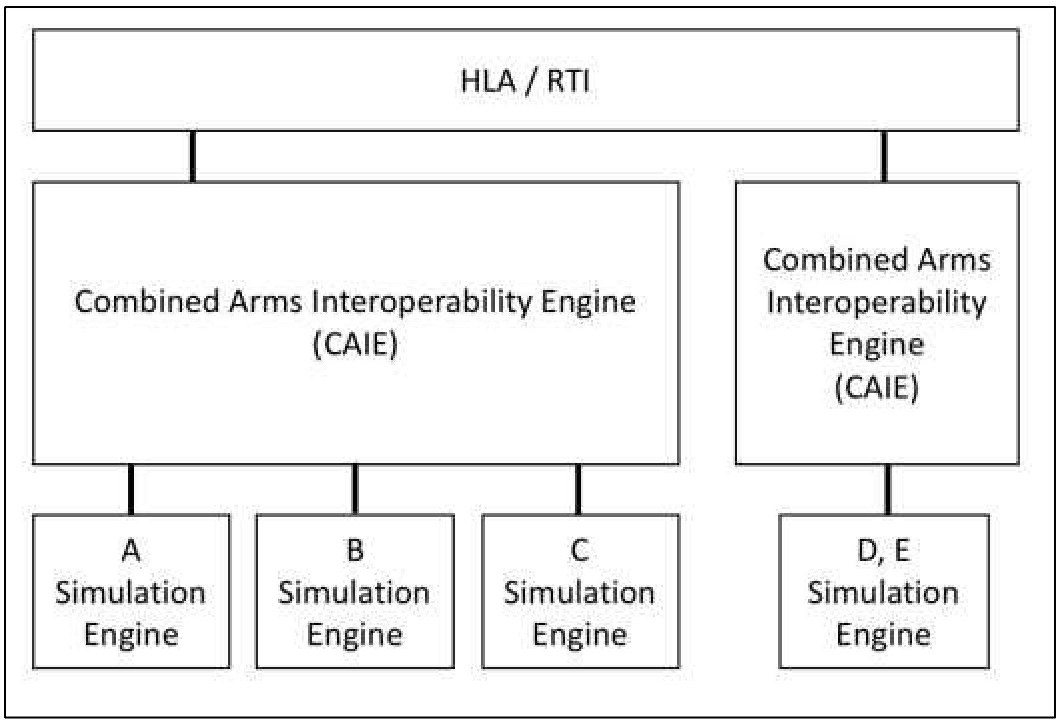

Figure 2-8. CAIIS structure

Combat damage assessment of simulation Engine A, B in the Figure 2-8 are calculated by use of formula in Figure 2-9, but each simulation engine has different own formulas of “Firing participate multiplier” and “Vulnerability multiplier”. Two simulation engines have same formula to calculate the damage assessment, but final values are different form each other. There are big differences for the final value between them.

Damage assessment = Basic loss rate x Decreasing factor x Firing participate multiplier x Vulnerability multiplier

Figure 2-9. Damage assessment formula

Decision tree method was suggested to solve this problem. Decision tree is an analysis method by classification, prediction, and segmentation techniques. This method has an advantage to make researcher easy to understand the analysis process and explain because the analysis process is presented by tree structure (Breiman, 1984). Figure 2-10 depicts the decision tree method.

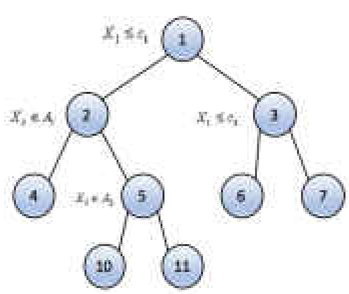

Figure 2-10. Decision tree example

Decision tree method has many advantages to overcome the difference of combat damage assessments between combat units belonging to their own model, but it has also disadvantages, such as less accuracy (Kaushal, 2014).

CHAPTER 3: OVERVIEW OF NEW FRAMEWORK

3.1 Introduction

The purpose of this chapter is to provide an overview of new framework, termed CDA framework, for sharing and generating CDA in a HLA/RTI federation. The framework offers useful guidance for sharing an engagement logic in a HLA/RTI federation and generating quantitative combat damage assessment data.

The two major contributions of this section include: (1) new framework that guides the user through the interconnection of simulations which has each own engagement logic to solve different CDA problems and (2) new method that generates quantitative data from a HLA/RTI federation without modification of each federate internal code.

3.2 Overview of the Framework

This section provides a brief overview of the conceptual CDA framework. The CDA Module is a major key of this framework to share and generate real-time quantitative CDA data in a HLA/RTI federation. Figure 3-1 depicts a diagram of the CDA framework divided in six main steps.

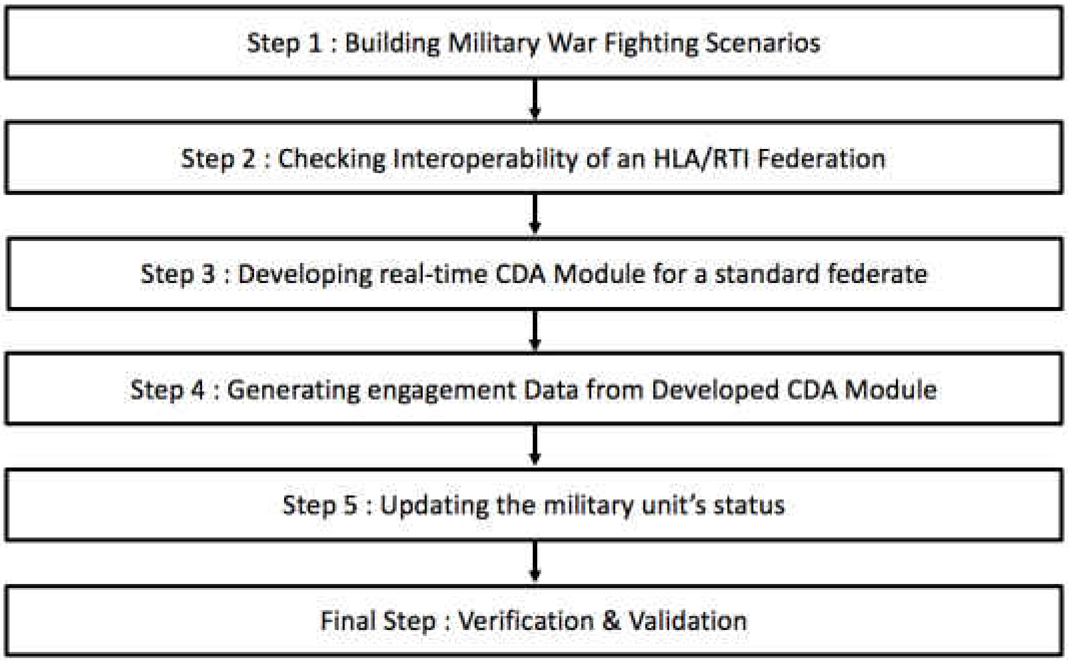

Figure 3-1. Six main steps to develop CDA Framework

Step 1 includes the conceptual phase to build military war fighting scenarios. All developers should consider many aspects to make authentic, relevant scenarios. Step 2 is a phase to check the interoperability of a HLA/RTI federation. Step 3 describes the development of a CDA Module for a standard federate to operate the actual framework construction. The CDA Module is a core program to control CDA factors in a standard simulation and generate quantitative CDA data in real-time. Step 4 is an essential phase to generate quantitative CDA data from CDA Module. Step 5 describes a principle how the generated damage data from Step 4 can share and update each military unit's status. Verification and Validation (V&V) process is the final step for this new framework.

3.3 Step 1: Building Military War Fighting Scenarios

All military simulations are operated based on War Fighting Scenarios. There are four ingredients for a successful scenario based on military training. They are 1) authentic, relevant scenarios, 2) pressure situations that tap user emotions and force them to act, 3) a sense of unrestricted options and, 4) re-playability (Aldrich, 2004). Building military war fighting scenarios is a fundamental step for a successful military simulation training.

Scenarios based on military training are the overall task approach that focuses on performance and learning from it (Reigeluth, 1999). Kindley (2002) also stated that learning from it in the simulation environment employs real-world issues as the basis of learning because it concentrates on the trainee's performance results like reflection of the real-world results. "Train as you will fight" is the one fundamental principle of Marine Corps military training (USMC, 1996). Therefore, many aspects to build military war fighting scenarios should reflect real-world military training by the fundamental principles.

3.4 Step 2: Checking Interoperability of a HLA/RTI Federation

This section describes the interoperability measurement of a HLA/RTI federation. This step is also a fundamental phase to generate reliable data from military simulations in a HLA/RTI federation. Figure 3-2 shows the result when the interoperability of two programs is considered (Morris, 2004). Types of interoperability are introduced in the Figure 3-2.

- programmatic: interoperability between different program offices
- constructive: interoperability between the organizations that are responsible for the

  construction (and maintenance) of a system
- operational: interoperability between the systems

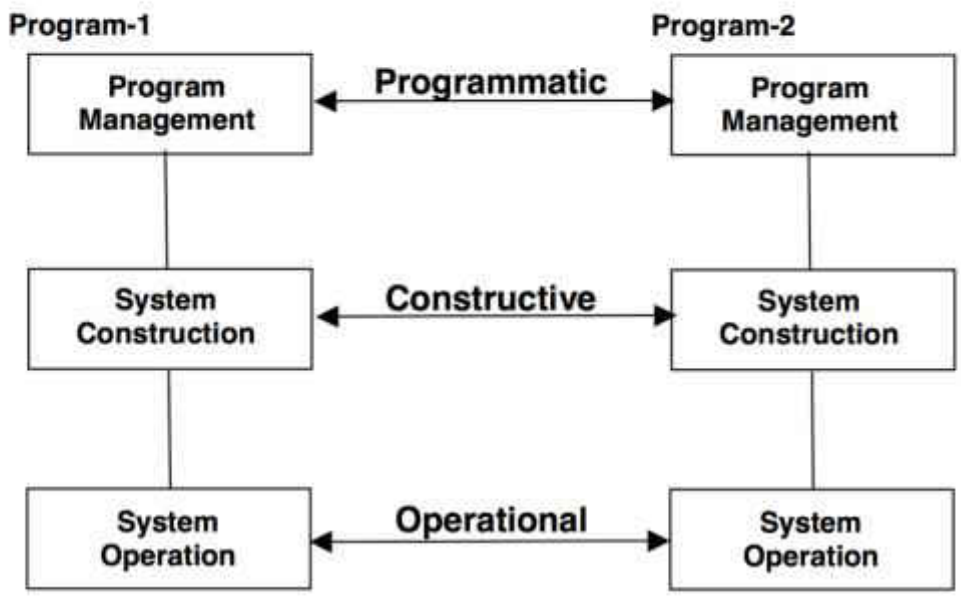

Figure 3-2. Different Types of Interoperability

Source: Simulation model validation

All developers should consider all types of interoperability to measure the degree of interoperability between two programs, but this research only focuses on the limited situation to use a HLA/RTI.

3.5 Step 3: Developing Real-time CDA Module for standard federate

This section describes the Real-time CDA Module for a federate which has a standard CDA logic. Deciding the standard federate which has standard engagement logic is important because we cannot use different kinds of CDA logics in one HLA/RTI federation to generate reliable data. The functional requirements of developing a real-time CDA Module include the following:

- Shall show real-time engagement result data while the federation is operating;
- Shall generate and store real-time military units' damage status; and
- Shall control engagement factors related to any engagements between military units in a HLA/RTI federation.

The CDA Module is a key program to develop the new CDA framework because all processes of generating and controlling of CDA data are operated through the CDA Module.

There is another important thing to keep for the new framework. One of important pre-requirements to implement this CDA Module is to set an undestroyed function for all military units in a non-standard federate. At this point, the possible problem is that the combat unit of non-standard federate cannot be destroyed. Therefore, additional process to destroy the combat unit of non-standard federate is needed. Additional federate will be attached to the RTI and destroy the combat unit after receiving the engagement data from CDA module and acquiring its ownership from non-standard federate.

3.6 Step 4: Generating Engagement Data from Developed CDA Module

This section describes the process to share the generated data from developed CDA Module between all federates in a HLA/RTI federation. The generated CDA data can be transferred or received between federates by RTI.

At this step, someone can cast a doubt upon the infringement of the HLA rules. That is "During a federation execution, all exchange of FOM data among federates shall occur via the RTI". However, considering HLA rules in this situation is not mandatory to follow because this framework is a simple supporting structure to share CDA data between two federates and generate quantitative CDA data only when there is a difference CDA between two federates. Also, FOM data is never exchanged between CDA Module and the standard federate.

The case study in Chapter 4 shows this process by using HLA/RTI based on virtual and constructive simulations.

3.7 Step 5: Updating Military Unit's Status

Entity is defined as "any distinct person, place, thing, event or concept where information is maintained or something which exists as a particular and discrete unit" (SISO, 2007). This section describes the process to update military units' damage status from CDA Module to other federates.

If there are destroyed military units in a standard federate, the information can be shared by CDA module through RTI and the military units also can be destroyed by the ownership management of RTI. The case study in Chapter 5 shows this process by using an example of MÄK RTI.

3.8 Step 6: Verification and Validation

This section describes the verification and validation(V&V) process of the new CDA framework. This is the final step for the new CDA framework and the most important phase. All developers usually are concerned with whether their developments and these results are correct or not. These concerns are related to V&V. The definition of Verification is “ensuring that the computer program of the computerized model and its implementation are correct” (Sargent, 2005). Validation can be defined as “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model” (Schlesinger, 1979).

Verification is a process that confirms that design synthesis has resulted in a physical architecture that satisfies the system requirements. Whereas validation is a process of confirming that a set of requirements, designs, or systems, meets the intent of the developer. The differences of Verification and Validation can be distinguished by these critical questions as:

1) Is the right software being built for the need? 2) Is the software being built rightly? (Fisher, 2003).

This process usually takes place before the documentation of results and after the implementation of the simulation to ensure credibility of simulation.

The simplified version of the modeling process is suggested in Figure 3-3 (Sargent, 2005).

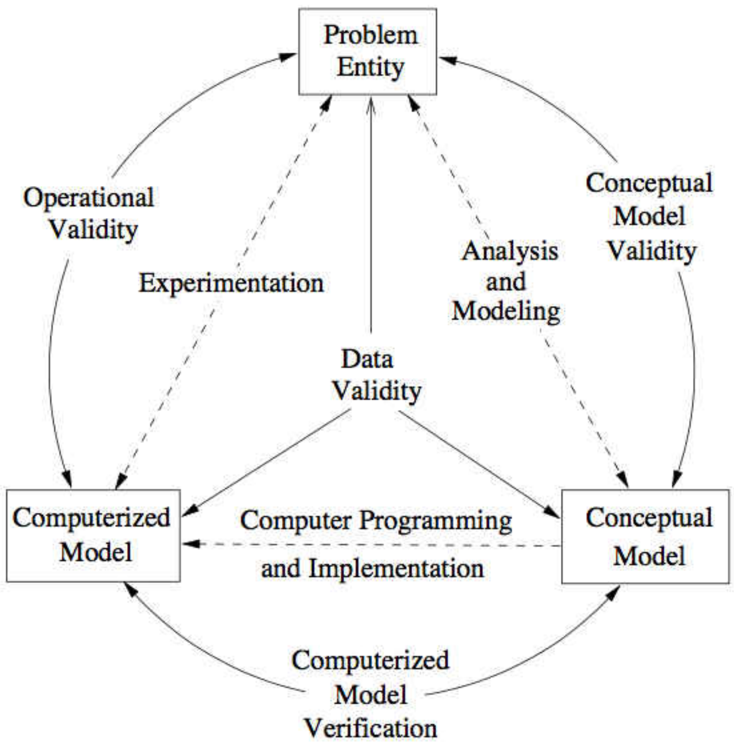

Figure 3-3. Simplified Version of the Modeling Process

Source: Simulation model validation

By following this Process, computerized model verification, operational validity, and conceptual model validity are needed to use for this final step.

CHAPTER 4: CASE STUDY #1

The previous chapter described the overview of new CDA framework to share standard CDA and generate quantitative CDA data in a HLA/RTI federation. This chapter and next chapter present two kinds of case studies to test the new framework and show its application capabilities for the proper architecting. The case study chapter is divided into two parts because of technical limitation and budget limitation to build overall CDA module. Case study #1 will verify the process from step 1 to step 4. The step 5 of previous chapter will be verified by using the case study #2. The new CDA framework guidelines and recommendations are presented through these two case studies. These case studies were conducted in the Simulation Interoperability Laboratory (SIL) of University of Central Florida (UCF).

4.1 Case Study #1 Introduction

This case study tests the new CDA framework from step 1 to step 4 presented in the previous chapter by using Virtual and Constructive simulations. Two kinds of simulations are made up a HLA/RTI federation and CDA Module is connected to one simulation which has standard CDA logic. The method to remove entity of a non-standard federate by using CDA Module is presented in the next chapter 5.

This case study uses HLA 1516 version and MÄK RTI program. The MÄK RTI from VT MÄK company was officially verified by the US DoD as Fully Compliant with the HLA 1516 version of the HLA Standard (IEEE 1516.1-2000) in February, 2006.

4.2 Case Study #1 Design

Figure 4-1 is an overview of the hardware and software specifications in the case study.

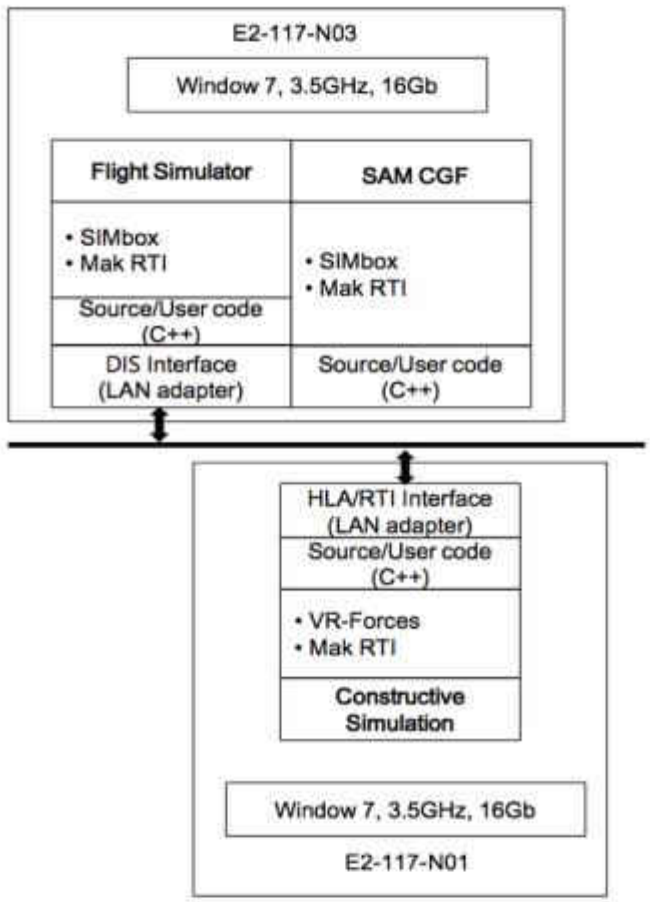

Figure 4-1. Simplified Software and Hardware Overview

Table 4-1 and 4-2 describes the operation environment of two simulations in the case studies.

Table 4-1. Operation environment for Virtual Simulation

Purpose	**Equipment**	**Description**
Virtual Simulation	Desktop Computer	• CPU: Intel® Core™ i7-4770K Processor 3.5GHz • HDD/RAM: 1TB/16GB • VGB: NVIDIA GeForce GTX 770 (2GB) • Monitor: 23inch LCD(1920x1080)
	O/S	• Window 7
	Operation	• SIMbox Knowbook + CDA module • MÄK RTI assistant
	Complier	• Microsoft Visual Studio 2010

Note. CPU=Central Processing Unit, HDD=Hard Disk Drive, VGA=Video Graphics Array, DVD=Digital Video, LCD=Liquid Crystal Display

Table 4-2. Operation environment for Constructive Simulation

Purpose	**Equipment**	**Description**
Virtual Simulation	Desktop Computer	• CPU: Intel® Core™ i7-4770K Processor 3.5GHz • HDD/RAM: 1TB/16GB • VGB: NVIDIA GeForce GTX 770 (2GB) • Monitor: 23inch LCD(1920x1080)
	O/S	• Window 7
	Operation	• MÄK VR-Forces • MÄK RTI
	Complier	• Microsoft Visual Studio 2010

Figure 4-2 describes the case study #1.

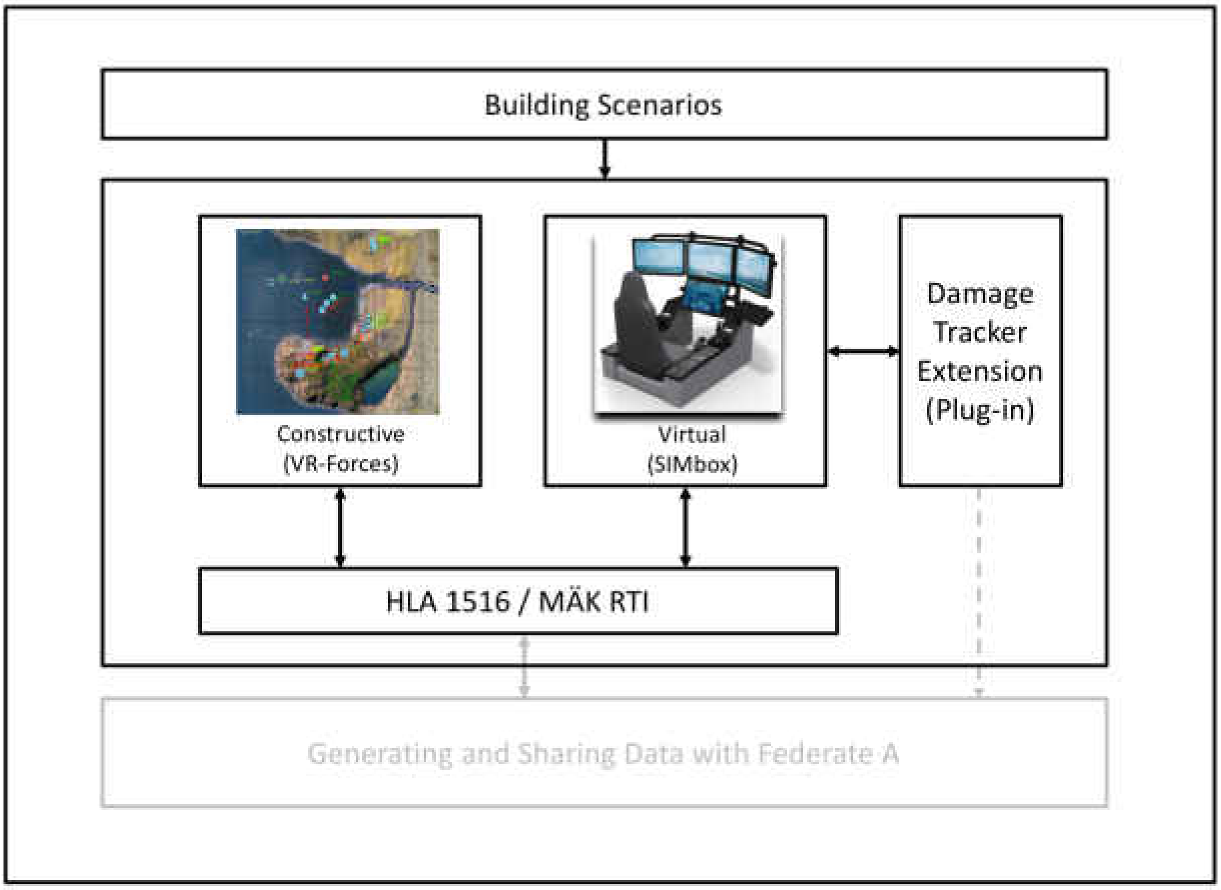

Figure 4-2. Case study #1 description

4.2.1 *Virtual simulation: SIMbox*

SIMbox is a simulation software platform for military and civilian applications and it provides a distributed simulation solution. Also, a solution software for creating contents, simulation, visualization and graphics modelling is provided from SIMbox. SIMbox is a HLA compliant, enabling combination with other components. Figure 4-3 shows the example of SIMbox HLA extension.

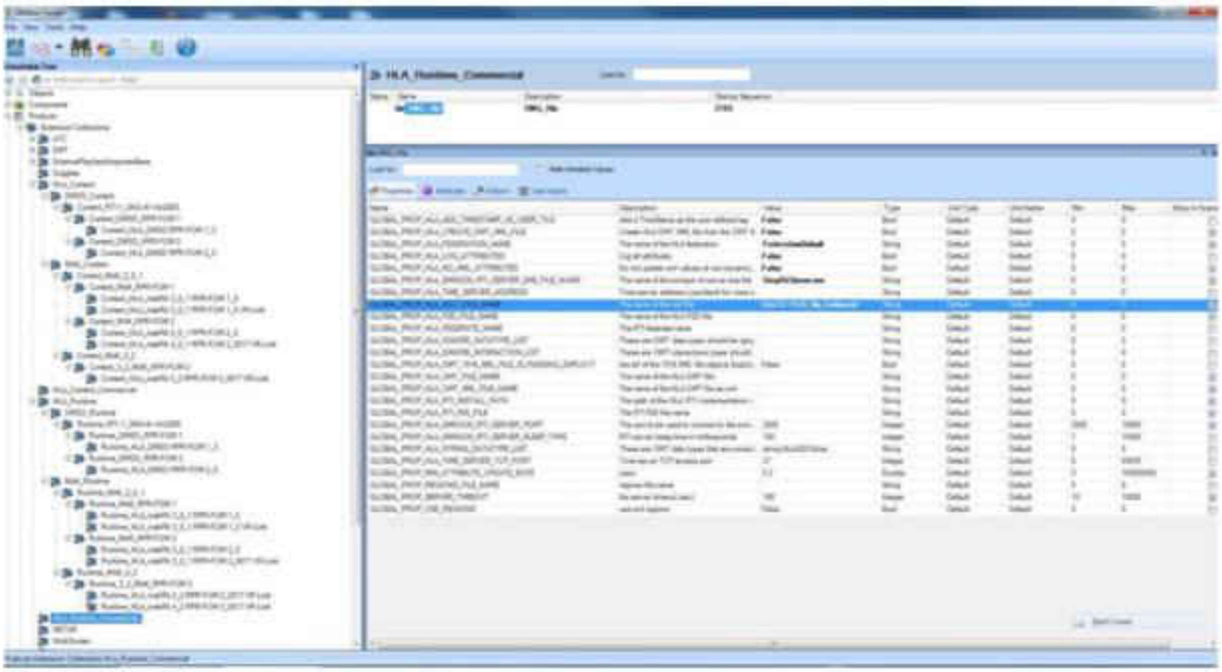

Figure 4-3. SIMbox HLA extension

4.2.2 *Constructive simulation: VR-Forces*

VR-Forces is a simulation software program and has a strong point for Computer Generated Forces(CGF) and Graphical User Interface(GUI) that helps non-experts to build scenarios. The entities in the VR-Forces interact with engage enemy forces, obstacles, communicate over simulated radios and terrain. “During scenario execution, VR-Forces vehicles and human entities interact with the terrain, follow roads, avoid obstacles, communicate over simulated radios, detect and engage enemy forces, and calculate damage VR-Forces comes with simulation models for a wide variety of battlefield entities and weapon systems" (MÄK).

VR-Forces satisfies to meet requirements of both the DIS and HLA simulation standards. VR-Forces also supports both the HLA 1.3, 1516 and 1516 evolved specifications and HLA PRP-FOM through the mapping feature. Figure 4-4 shows the simulation connection configuration for VR-Forces GUI and Simulation engine.

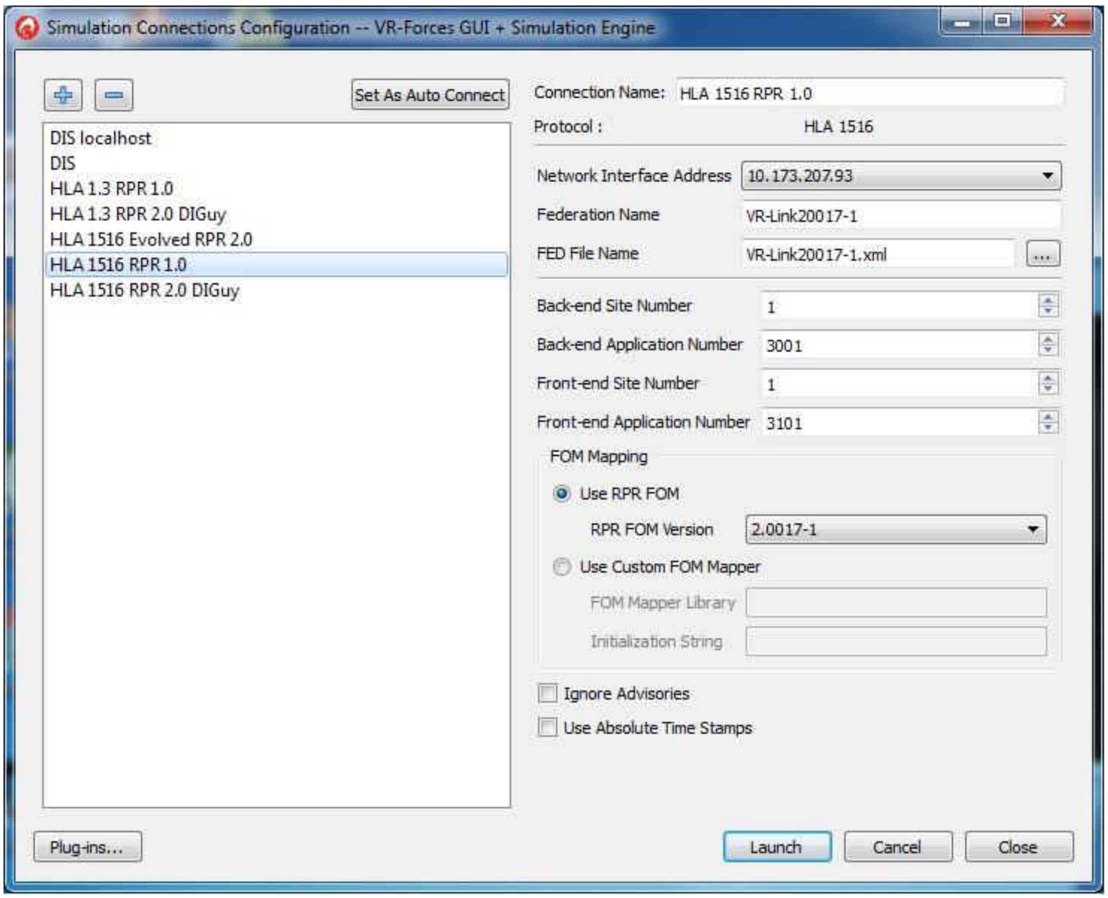

Figure 4-4. Simulation Connection Configuration of VR-Forces

4.3 Case Study #1 Implementation

In this case study #1, four steps of the new CDA framework are implemented. Step 1 is a conceptual phase to build military war fighting scenarios. Step 2 is a preparation phase to confirm the interoperability between federates of a HLA/RTI federation. In this case study, Virtual simulation and Constructive simulation are considered. Step 3 describes the development of a real-time CDA Module for a standard federate to control engagement factors, monitor combat results and generate combat damage assessment data. Step 4 is an essential phase to share the generated CDA data with other federates through the RTI.

4.3.1 *Step 1: Building a military war fighting scenarios*

This scenario consists of one Mig-29, three SA-8s and one target building for red team, and three F-16s for blue team. The main goal of the scenario is to protect the target building from blue team's attack. Two blue F-16s engages one red Mig-29 for virtual simulator that are circling at an altitude to protect the target building. The SA-8 is also located to protect the target building from the blue F-16s attack. The situation map is shown in Figure 4-5 and 4-6. This scenario is not related to any real military operations and the las vegas map is used for this scenario because only the area is available in both simulation programs.

Figure 4-5. The scenario map of SIMbox

Table 4-3 shows the extra information of this war fighting military scenario.

Table 4-3. Extra information of this scenario

Components	Description
Geographic Area	Las Vegas, Nevada
Climate	Normal daytime
Simulation end condition	F-16 fighters for blue team are all destroyed or returned to their base after destroying the target building

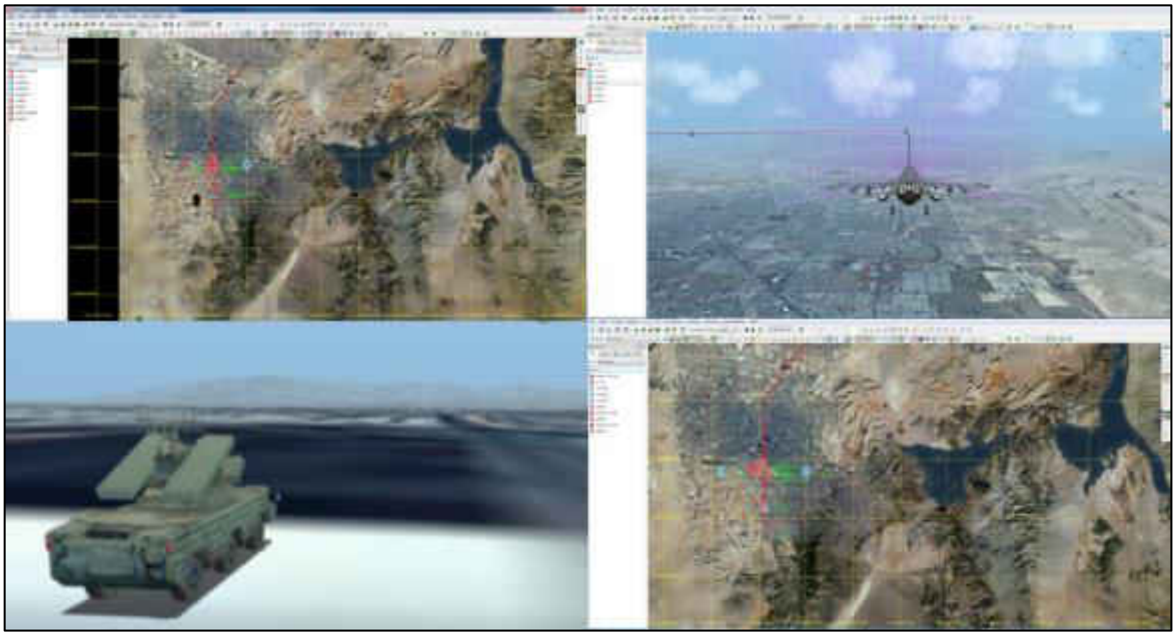

Figure 4-6. The scenario map of VR-Forces

4.3.2 *Step 2: Checking the interoperability of a HLA/RTI federation*

The HLA has features of reusability and interoperability for simulation systems “by setting rules for simulation system and participants, standardizing communication interface between participants and simulation infrastructure, and defining a template for Object Models that will be used for data exchange.” (Çelik, Gökdoğan, Öztürk, & Sarikaya, 2013).

In this case study, HLA 1516 version is adopted to interconnect with two simulations by MÄK RTI and RPR FOM was also adopted for them because it is a specified FOM for real-time virtual simulators. Figure 4-7 depicts the RPR FOM sample content data flow of SIMbox.

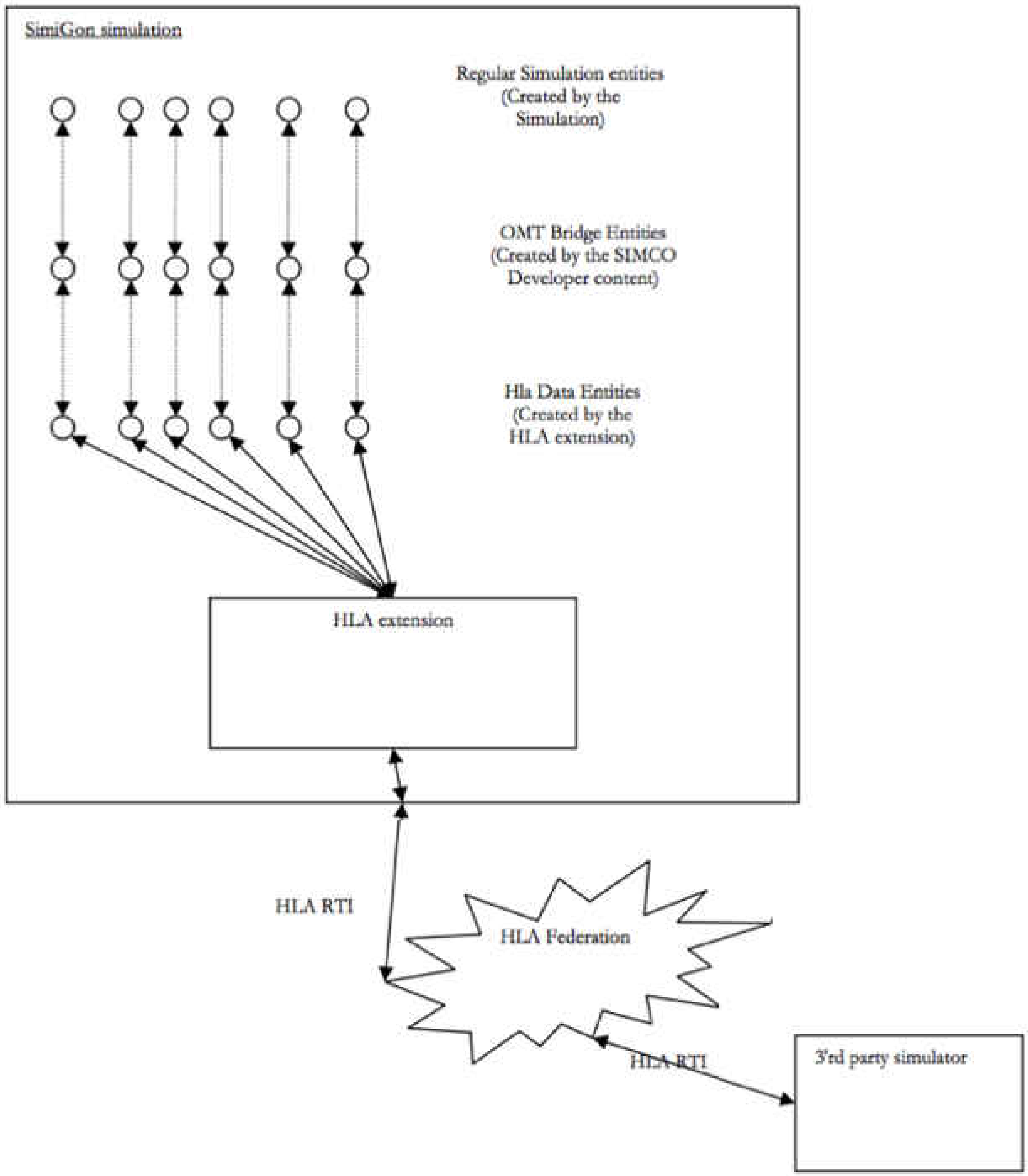

Figure 4-7. RPR FOM sample content data flow of SIMbox

Source: SIMbox Version 5.6.3 Release Notes

Interactions are the attempt to change or modify the status of another by one object. For example, direct shooting, logistics supply, and all communications are all interactions (Tolk, 2012). Specified mapping method is needed to interact between all federates in a HLA federation. The figure 4-8 shows the entities mapping in DisEntitiesMap.Xml.


```xml
<?xml version="1.0"?>
- <DisEntitiesMap>
  - <GeneralTypeMap>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeAircraft" simEntityTypeName="F-16C" hlaEntityTypeName="Aircraft"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="AIM-9" hlaEntityTypeName="Munition"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeShip" simEntityTypeName="Merchant_Vessel" hlaEntityTypeName="SurfaceVessel"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeBuilding" simEntityTypeName="Aircraft Shelter 1" hlaEntityTypeName="CulturalFeature"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeGround" simEntityTypeName="HLAMan-M16" hlaEntityTypeName="Human"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeGround" simEntityTypeName="Hummer" hlaEntityTypeName="GroundVehicle"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeGround" simEntityTypeName="Sam" hlaEntityTypeName="GroundVehicle"/>
      <GeneralDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="SA-8" hlaEntityTypename="Munition"/>
    </GeneralTypeMap>
  - <SpecificTypeMap>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeAircraft" simEntityTypeName="F-16C" hlaEntityTypeName="Aircraft" disExtra="0" disSpecific="0"
        disSubcategory="9" disCategory="1" disCountryCode="225" disDomain="2" disEntityKind="1"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="Hummer" hlaEntityTypeName="GroundVehicle" disExtra="21" disSpecific="7"
        disSubcategory="1" disCategory="6" disCountryCode="225" disDomain="1" disEntityKind="1"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeAircraft" simEntityTypeName="ApacheGunner" hlaEntityTypeName="Aircraft" disExtra="0" disSpecific="3"
        disSubcategory="1" disCategory="20" disCountryCode="225" disDomain="2" disEntityKind="1"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeAircraft" simEntityTypeName="F-16C" hlaEntityTypeName="Aircraft" disExtra="0" disSpecific="3"
        disSubcategory="3" disCategory="1" disCountryCode="225" disDomain="2" disEntityKind="1"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeAircraft" simEntityTypeName="Mig29" hlaEntityTypeName="Aircraft" disExtra="3" disSpecific="3"
        disSubcategory="3" disCategory="1" disCountryCode="225" disDomain="2" disEntityKind="1"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="AIM-9" hlaEntityTypeName="Munition" disExtra="0" disSpecific="1"
        disSubcategory="1" disCategory="1" disCountryCode="225" disDomain="1" disEntityKind="2"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="AIM-120" hlaEntityTypeName="Munition" disExtra="0" disSpecific="1"
        disSubcategory="2" disCategory="1" disCountryCode="225" disDomain="1" disEntityKind="2"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="AIM-7" hlaEntityTypeName="Munition" disExtra="0" disSpecific="1"
        disSubcategory="13" disCategory="1" disCountryCode="225" disDomain="1" disEntityKind="2"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="MK-82" hlaEntityTypeName="Munition" disExtra="0" disSpecific="1"
        disSubcategory="14" disCategory="1" disCountryCode="225" disDomain="9" disEntityKind="2"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="MK-83" hlaEntityTypeName="Munition" disExtra="0" disSpecific="1"
        disSubcategory="18" disCategory="1" disCountryCode="225" disDomain="9" disEntityKind="2"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeWeapon" simEntityTypeName="MK-84" hlaEntityTypeName="Munition" disExtra="0" disSpecific="1"
        disSubcategory="15" disCategory="1" disCountryCode="225" disDomain="9" disEntityKind="2"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeShip" simEntityTypeName="Merchant_Vessel" hlaEntityTypeName="SurfaceVessel" disExtra="0"
        disSpecific="1" disSubcategory="1" disCategory="17" disCountryCode="71" disDomain="3" disEntityKind="1"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeBuilding" simEntityTypeName="Bld_Office1" hlaEntityTypeName="CulturalFeature" disExtra="0"
        disSpecific="0" disSubcategory="4" disCategory="2" disCountryCode="0" disDomain="1" disEntityKind="5"/>
      <SpecificDisEntityMapping disBridgeSimEntityTypeName="DisBridgeGround" simEntityTypeName="HLAMan-M16" hlaEntityTypeName="Human" disExtra="0" disSpecific="1"
```


Figure 4-8. Entities Mapping in DisEntitiesMap.Xml

The DisEntitiesMap.Xml file contains essential information about entities' definition and interactions for a HLA/RTI federations. For example, if only one federate has a F-16 entity's information on it, the F-16 cannot interact in the federation with other federates' entities. Also, the specified number of all categories must be shared together for interoperability.

Figure 4-9 shows a HLA federation view by a MÄK RTI. All sharing information in a HLA federation can be checked by a MÄK RTI.

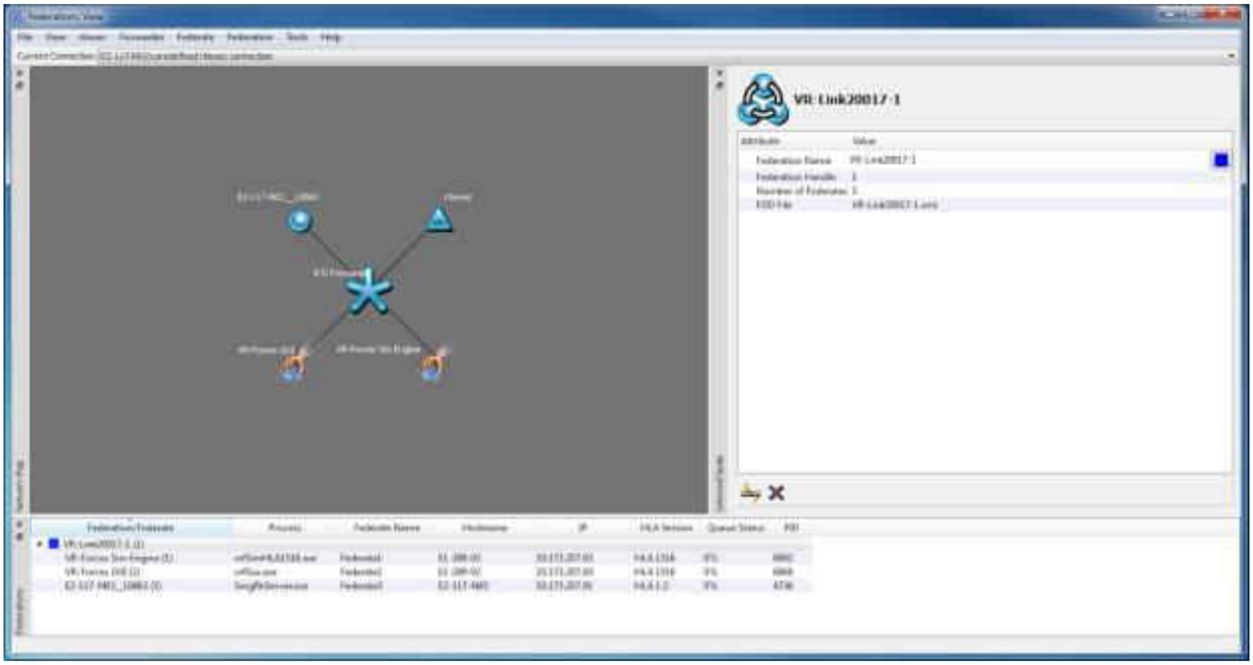

Figure 4-9. Federation view by MÄK RTI

4.3.3 *Step 3: Developing a real-time CDA Module for a standard federate*

The engage measurement module is developed as an extension application that interfaces with multiple entities properties and attributes in the SIMbox. Statistical analysis of the scenario, real-time modification of each entity's engagement factors, generating damage value and showing battle engagement results from the distributed simulation exercises are managed and presented through the engage measurement module. Figure 4-10 depicts the engage measurement module.

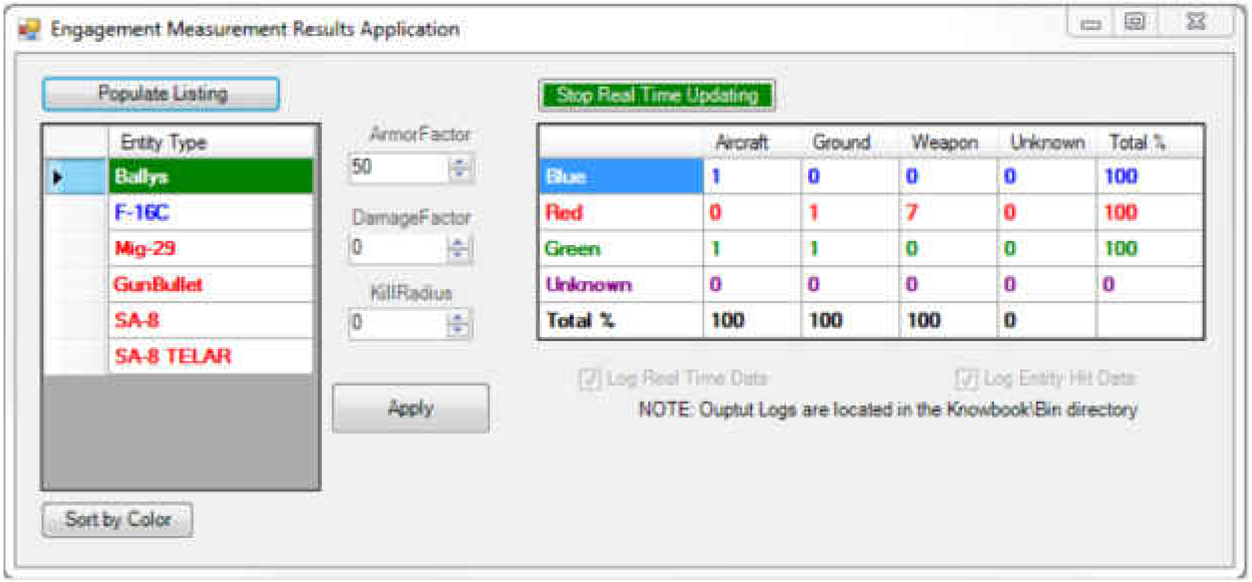

Figure 4-10. CDA module

The “Populate Listing” button should be used to activate and check the entity list of a HLA federation initially. Users can see the list of each of the different type of entities participating in a HLA federation. Then, the “Entity Type” table can be updated. Users can select and check the current entity’s Armor Factor, Damage Factor, and Kill Radius by numbers. Also, the “Sort by Color” button classify entities according to color.

Users can manage and modify the three kinds of attributes of entities in real time by using this CDA module. The three kinds of attributes are as listed below:

1. Armor Factor
2. Damage Factor
3. Kill Radius

These entity attributes are related to damage value calculation, so they affect the combat effectiveness and combat result. The figure 4-11 shows the damage value calculation.

```
float killRadius=0.0f;
GET_ENTITY_ATT(explodedEtt, EntityWorld::ATT_KILL_RADIUS, killRadius);

float distance = (explotionPos-myPos).norma();

float damageFactor=0.0f;
GET_ENTITY_ATT(explodedEtt, EntityWorld::ATT_DAMAGE_FACTOR, damageFactor);

float armorFactor = 0.f;
GET_ENTITY_ATT(_pOwnerEntity,EntityWorld::ATT_ARMOR_FACTOR, armorFactor);

int damage = ( fKillradius – distance ) * fDamageFactor / fKillradius * (( 100.0f – fArmorFactor ) /100.f );
```

Figure 4-11. Damage Value Calculation of SIMbox Simulation Engine

The “Start/Stop Real Time Updating” button will turn green color when “Start” was activated. It means that the module will update the combat result in real time on the module. Also, “Log Real Time Data” and “Log Entity Hit Data” checkboxes are available to generate engagement data in real time. The two functions of generating engagement data are key functions for this module and the proposed new framework in this paper.

4.3.4 *Step 4: Generating data from developed CDA Module*

Two types of engagement data can be generated by the developed CDA module in real time. The two types of engagement data are the following:

- Engage Measurement Results
- Entity Hit Result

The engagement data can be generated when the respective checkbox is checked in the CDA module. The two types of engagement data are created when they are checked and stored in the c:\\ProgramFiles\Knowbook\bin in Window OS in real time.

4.3.4.1 Log Real Time Data

Table 4-4 depicts all parameters and explanation of engagement data generated from CDA module, when "Log Real Time Data" checkbox was selected.

Table 4-4. Parameters in Engage Measurement Results Log

Parameter	**Description**
SystemTime	Number of seconds since simulation started
UniqueName	Unique name of the entity within the scenario
EntityID	Unique ID used to identify the entity within our simulation
DamageValue	Amount of damage the entity has taken so far
WhichSide	Force indicator of entity
ArmorFactor	Armor factor of the entity
DamageFactor	Damage factor of the entity
KillRadius	Kill Radius of the entity
IsAircraft	Is the entity an Aircraft?
IsGround	Is the entity a Ground entity?
IsWeapon	Is the entity a Weapon?

The figure 4-12 shows the generated engagement data from CDA module, when “Log Real Time Data” checkbox was selected.

SystemTime	UniqueName	EntitiyID	DamageValue	WhichSide	ArmorFactor	DamageFactor	KillRadius	IsAircraft	IsGround	IsWeapon
5.616	Sam 2_SA-84_1	39	0	Red	50	100	100	FALSE	FALSE	TRUE
5.616	Mig-29 2	103	0	Red	50	0	0	TRUE	FALSE	FALSE
5.616	Mig-29 4_AIM-75_1	16	0	Red	10	100	100	FALSE	FALSE	TRUE
5.616	F-16C 4_AIM-99_1	48	0	Blue	10	100	100	FALSE	FALSE	TRUE
5.616	Sam 1_SA-84_1	25	0	Blue	50	100	100	FALSE	FALSE	TRUE
5.616	F-16C 2_AIM-93_1	57	0	Blue	10	100	100	FALSE	FALSE	TRUE
5.616	F-16C 1_AIM-911_1	89	0	Blue	10	100	100	FALSE	FALSE	TRUE
5.616	Sam 2	34	0	Red	20	200	0	FALSE	TRUE	FALSE
5.616	Mig-29 1_AIM-76_1	66	0	Red	10	100	100	FALSE	FALSE	TRUE
5.616	F-16C 3_AIM-92_1	98	0	Blue	10	100	100	FALSE	FALSE	TRUE
5.616	Mig-29 3_AIM-72_1	75	0	Red	10	100	100	FALSE	FALSE	TRUE
5.616	Mig-29 2_AIM-75_1	107	0	Red	10	100	100	FALSE	FALSE	TRUE
5.616	Sam 1	20	0	Blue	20	200	0	FALSE	TRUE	FALSE
5.616	F-16C 1_AIM-91_1	84	0	Blue	10	100	100	FALSE	FALSE	TRUE
5.616	Mig-29 1	61	0	Red	50	0	0	TRUE	FALSE	FALSE

Figure 4-12. Log Real Time Data Example

Only default data can be generated and stored according to the event time from “Log Real Time Data”. By using this generated data, users can check the default data in real time. This data can be used for verification by changing entities’ attributes in a HLA federation in real time. It means that the data provide to enable users to compare how the simulation entity is affected to each of the engagements with changing attributes.

4.3.4.2 Entity Hit Result Log

Table 4-5 depicts all parameters and explanation of engagement data generated from CDA module, when "Entity Hit Result Log" checkbox was selected.

Table 4-5. Parameters in Entity Hit Results Log

Parameter	Description
SystemTime	Number of seconds since simulation started
UniqueName	Unique name of the entity within the scenario
EntityID	Unique ID used to identify the entity taking damage within our simulation
HittingEntityID	Unique ID used to identify the entity doing damage within our simulation
ISDestroyed	Did the entity taking damage become destroyed?
DamageValue	Amount of damage the entity taking damage took

The figure 4-13 shows the generated engagement data from CDA module, when "Entity Hit Result Log" checkbox was selected.

SystemTime	UniqueName	EntitiyID	HittingEntityID	IsDestroyed	DamageValue
28.543	F-16C 3_AIM-92_1	98	37	FALSE	65
28.543	F-16C 3_AIM-911_1	102	37	FALSE	68
28.543	F-16C 3_AIM-91_1	97	37	FALSE	64
28.543	F-16C 3_AIM-93_1	99	37	FALSE	66
28.543	Sam 2_SA-82_1	37	37	TRUE	100
28.543	F-16C 3_AIM-910_1	101	37	FALSE	68
28.543	F-16C 3	96	37	FALSE	38
28.543	F-16C 3_AIM-99_1	100	37	FALSE	68

Figure 4-13. Entity Hit Results Log Example

The "Damage Value" is not default value in this generated data. The value is cumulative in this case and the maximum value is 100. Therefore, if the value reaches the maximum point or over the point, the entity will be destroyed and the status of “IsDestroyed” will be changed to “TRUE”. This damage value will be used to decide the termination of each entity when there are engagements between entities.

4.4 Verification & Validation(V&V)

This CDA Module is developed and verified only for SIMbox by developers from SimiGon company, so additional processes are needed to verify the CDA Module in a HLA/RTI federation. The module is verified under three assumptions as follow:

- All entities in a HLA/RTI federation should be presented on the real-time.
- CDA Module and engagement data should be generated in real-time.
- The engagement factors of all entities can be controlled by using CDA Module in real-time.

4.4.1 *Verification #1: Showing real-time engagement result*

The first verification process is to check whether the CDA Module can show real-time engagement result or not. It was implemented by using the military war fighting scenario in step 1. Figure 4-14 and Figure 4-15 compares the changed engagement result after the “destroyed” event. The total aircraft number of blue team was changed from 3 to 2 after “destroyed” event.

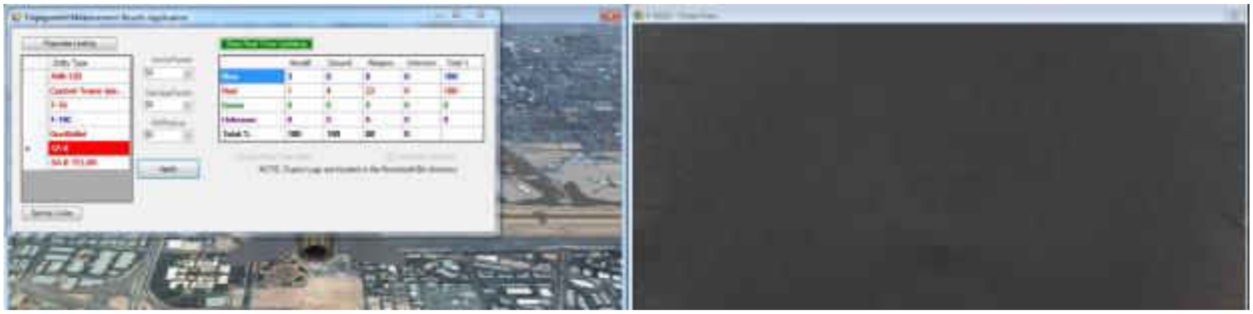

Figure 4-14. Real-time engagement result after “damaged” event

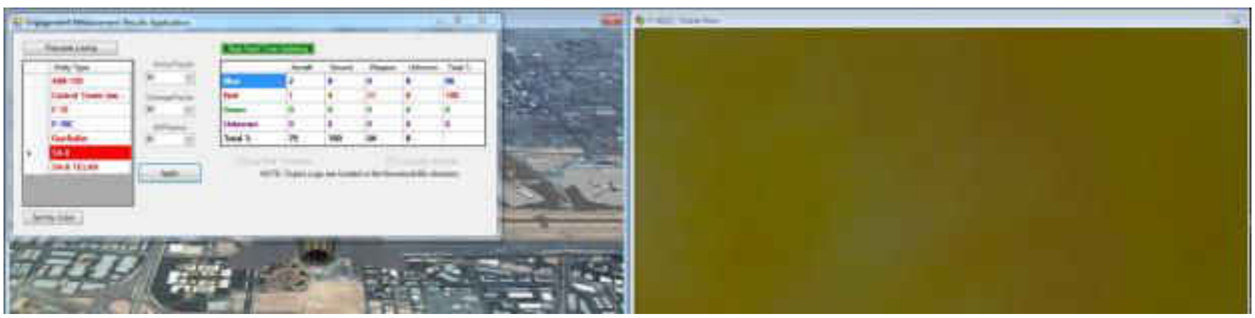

Figure 4-15. Real-time engagement result after “destroyed” event

4.4.2 *Verification #2: Generating real-time engagement data*

The second verification process is to check whether the CDA Module can generate real-time engagement data or not. It was also implemented by using the military war fighting scenario in step 1.

The two kinds of real-time log files were generated in the designated folder in real-time and the file was only read-only until the end of the scenario.

Figure 4-16. Two kinds of real-time engagement log files

4.4.3 *Verification #3: Controlling engagement factors*

The third verification process is to check whether the CDA Module can control engagement factors of all entities in a HLA/RTI federation related to real-time engagement data. It was also implemented by using the military war fighting scenario in step 1.

Two variables were changed in this experiment to verify the function of controlling engagement factors of all entities. The variables are the armor factor of F-16 fighters and the damage factor of SA-8 SAM. Table 4-6, Table 4-7, and Table 4-8 shows the initial damage values of each aircraft damaged from SA-8 SAM. Each experiment was replicated same scenario 31 times.

Table 4-6. Damage values at Damage factor (50) for SA-8 & Armor factor (50) for F-16

	F-16(1)	F-16(2)	F-16(3)	Average
1	17	17	18	17.333
2	19	18	18	18.333
3	17	19	17	17.667
4	17	17	19	17.667
5	18	17	18	17.667
6	17	17	18	17.333
7	17	18	17	17.333
8	18	17	18	17.667
9	17	18	17	17.333
10	18	18	19	18.333
11	18	18	19	18.333
12	17	18	17	17.333
13	17	18	17	17.333
14	17	18	17	17.333
15	18	18	18	18.000
16	18	18	18	18.000
17	17	18	19	18.000
18	19	18	17	18.000
19	17	17	19	17.667
20	18	18	18	18.000
21	17	18	18	17.667
22	19	18	18	18.333
23	17	18	18	17.667
24	19	18	18	18.333
25	17	18	18	17.667
26	17	19	19	18.333
27	17	17	18	17.333
28	19	19	17	18.333
29	17	18	18	17.667
30	17	18	17	17.333
31	17	19	18	18.000
Total	**17.548**	**17.903**	**17.903**	**17.785**

Table 4-7. Damage values at Damage factor (50) for SA-8 & Armor factor (40) for F-16

	F-16(1)	F-16(2)	F-16(3)	Average
1	21	21	23	21.667
2	21	21	21	21.000
3	21	22	23	22.000
4	23	23	22	22.667
5	21	22	23	22.000
6	21	21	23	21.667
7	22	22	21	21.667
8	21	21	22	21.333
9	22	21	22	21.667
10	21	22	21	21.333
11	21	21	22	21.333
12	22	22	21	21.667
13	21	22	23	22.000
14	21	22	21	21.333
15	22	21	22	21.667
16	21	22	22	21.667
17	21	22	22	21.667
18	21	21	23	21.667
19	21	21	21	21.000
20	23	22	21	22.000
21	21	22	21	21.333
22	23	21	22	22.000
23	21	22	23	22.000
24	21	21	22	21.333
25	21	21	21	21.000
26	21	22	21	21.333
27	23	21	22	22.000
28	21	22	22	21.667
29	21	21	21	21.000
30	21	21	22	21.333
31	21	21	21	21.000
Total	**21.387**	**21.516**	**21.839**	**21.581**

Table 4-8. Damage values at Damage factor (80) for SA-8 & Armor factor (50) for F-16

	F-16(1)	**F-16(2)**	**F-16(3)**	**Average**
1	28	28	28	28.000
2	28	29	31	29.333
3	28	30	29	29.000
4	29	29	28	28.667
5	28	29	28	28.333
6	28	30	28	28.667
7	28	28	30	28.667
8	28	29	29	28.667
9	28	29	31	29.333
10	29	28	31	29.333
11	28	28	29	28.333
12	28	29	30	29.000
13	28	29	29	28.667
14	28	29	28	28.333
15	28	29	30	29.000
16	28	29	28	28.333
17	29	29	30	29.333
18	30	28	29	29.000
19	30	28	29	29.000
20	28	28	30	28.667
21	28	28	28	28.000
22	29	28	28	28.333
23	29	28	30	29.000
24	28	29	29	28.667
25	28	30	29	29.000
26	30	30	28	29.333
27	30	28	30	29.333
28	28	30	30	29.333
29	28	30	30	29.333
30	30	30	29	29.667
31	30	28	31	29.667
Total	**28.548**	**28.839**	**29.258**	**28.882**

The average damage values of F-16s are 17.785, 21.581, and 28.882 according to the damage factor and the armor factor. The damage values were calculated when the engagement factors were changed by CDA Module and it is obvious that the three values are significantly not to be on the same level.

4.4.4 *Validation*

The main purpose of this program is to generate reliable data after sharing CDA data with non-standard federate, so generated quantitative data from this CDA Module should be analysed to validate. Three kinds of validation techniques that are Animation, Event Validity, and Variability-Sensitivity Analysis can be adopted to this new CDA framework validation.

Table 4-9. Three kinds of validation technique

Technique	**Description**
Animation	"The model's operational behavior is displayed graphically as the model moves through time" (Sargent, 2005)
Event Validity	"The events of occurrences of the simulation model are compared to those of the real system to determine if they are similar" (Sargent, 1984)
Variability-Sensitivity	"This technique consists of changing the values of the input and internal parameters of a model to determine the effect upon the model's behavior and its output" (Sargent, 1984)

4.5 Case Study #1 Summary

The case study #1 implemented the new CDA framework from step 1 to step 4 presented in the previous chapter. Two kinds of simulations were made up a HLA/RTI federation and CDA Module was connected to one simulation which has standard CDA logic. This case study used HLA 1516 version and MÄK RTI program.

Through this case study, CDA Module was verified to use for the new CDA framework and several products of this case study were discovered additionally.

- Engagement factors related to damage values of entities in the non-standard federate can be controlled by using CDA Module of standard federate when the engagement factors are not shared by HLA/RTI.

- Consistent engagement data of two federates can be generated by using CDA Module of standard federate when other interoperability problems are not occurred.

The method to remove entities of non-standard federate by using CDA Module will be presented in the next chapter.

CHAPTER 5: CASE STUDY #2

5.1 Case Study #2 Introduction

This case study exercises the new CDA framework about step 5 presented in the previous chapter by using an example program (HLA bounce) of VT MÄK company. Two HLA bounce example programs can be made up a HLA/RTI federation in this case study. This case study also uses HLA 1516 version and MÄK RTI program. The HLA bounce program can show the subscription or un-subscription functions and ownership management process. Each HLA bounce shows one or more colorful balls. The color is changeable by users and users can know each ball's ownership by the ball's color. Figure 5-1 shows the HLA bounce (1516 version).

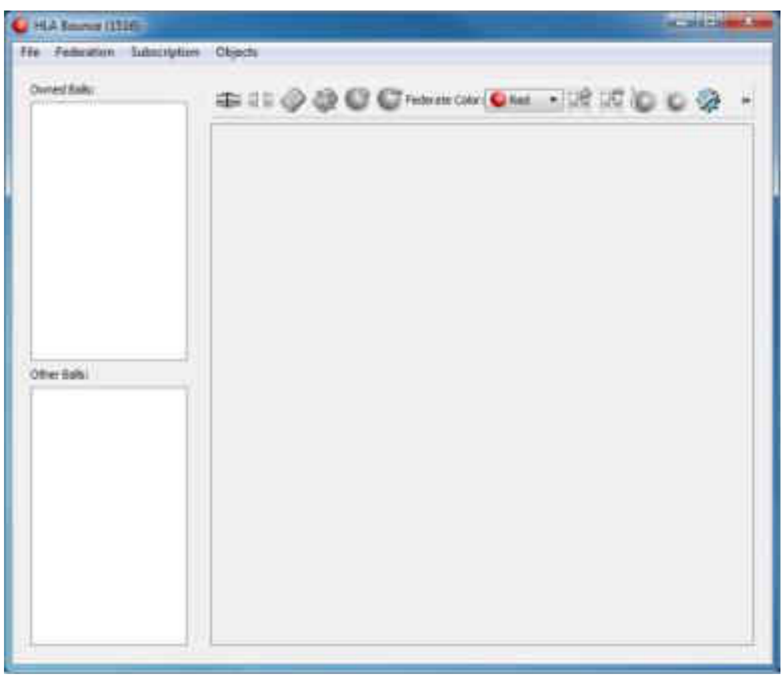

Figure 5-1. HLA bounce example program

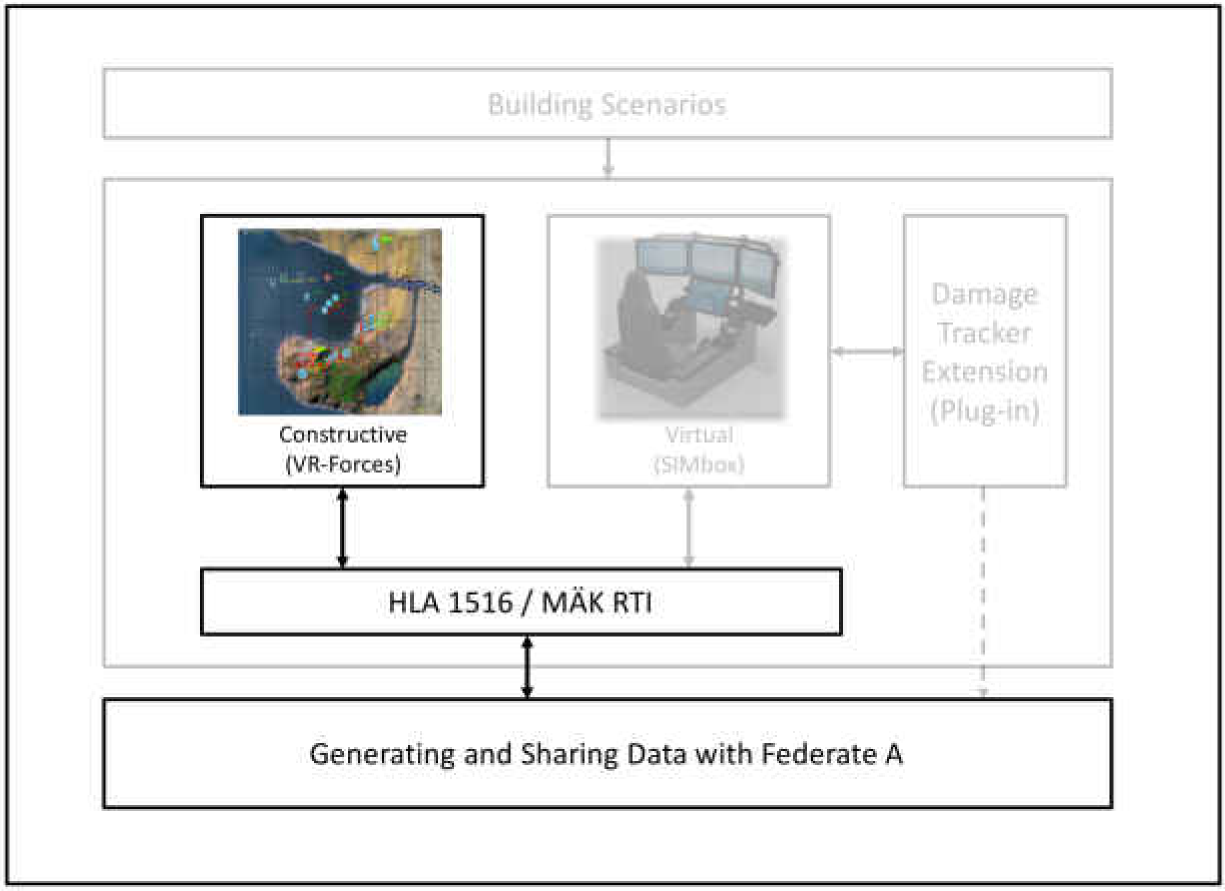

Figure 5-2. Case study #2 description

5.2 Case Study #2 Design

The main purpose of this chapter is to exercise the step 5 presented in the previous chapter 3, so this case study was designed to show how the entity can be removed after destroying by using a CDA Module federate. Figure 5-3 depicts the federation connection between two HLA bounce programs by using MÄK RTI.

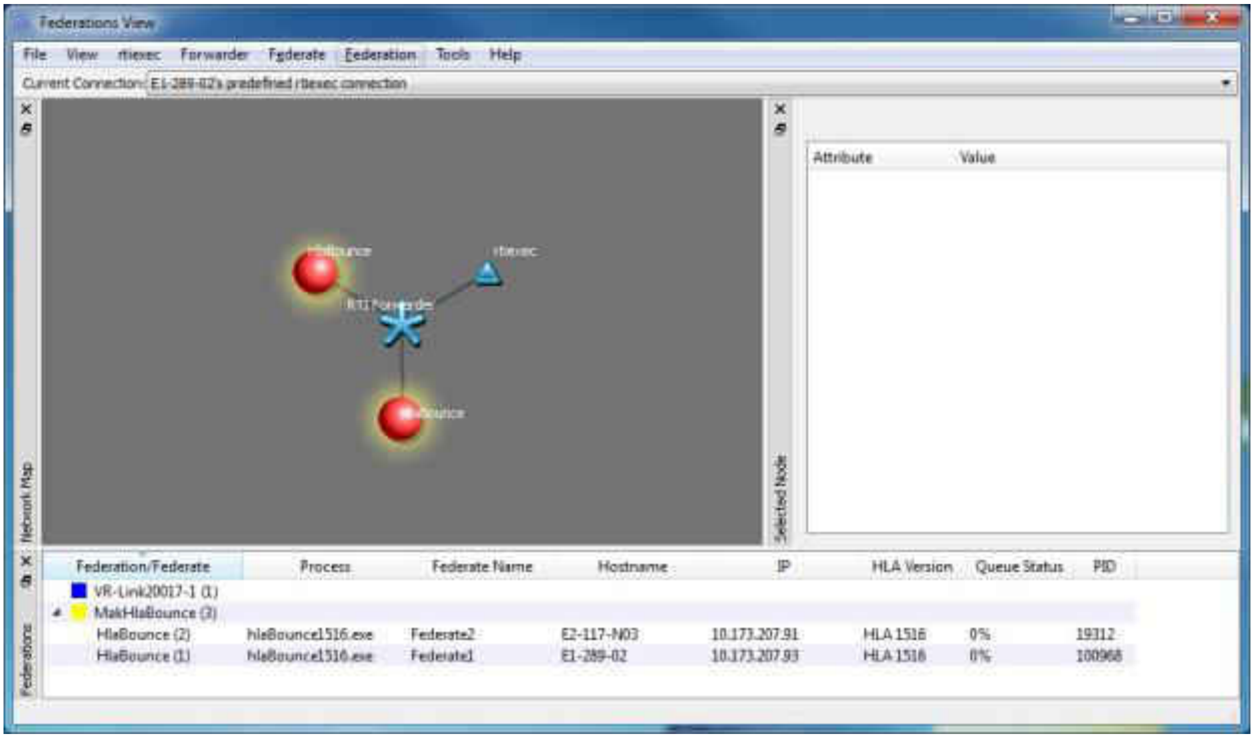

Figure 5-3. HLA bounce federation connection by using MÄK RTI

One instance of HLA bounce has a role of a non-standard federate and the other one has a role of a standard federate with CDA Module.

5.3 Case Study #2 Implementation

Each instance of HLA bounce added three balls from initial setting to implement this case study. Each ball represents an entity in each simulation. Figure 5-4 shows the initial setting after making up a HLA/RTI federation.

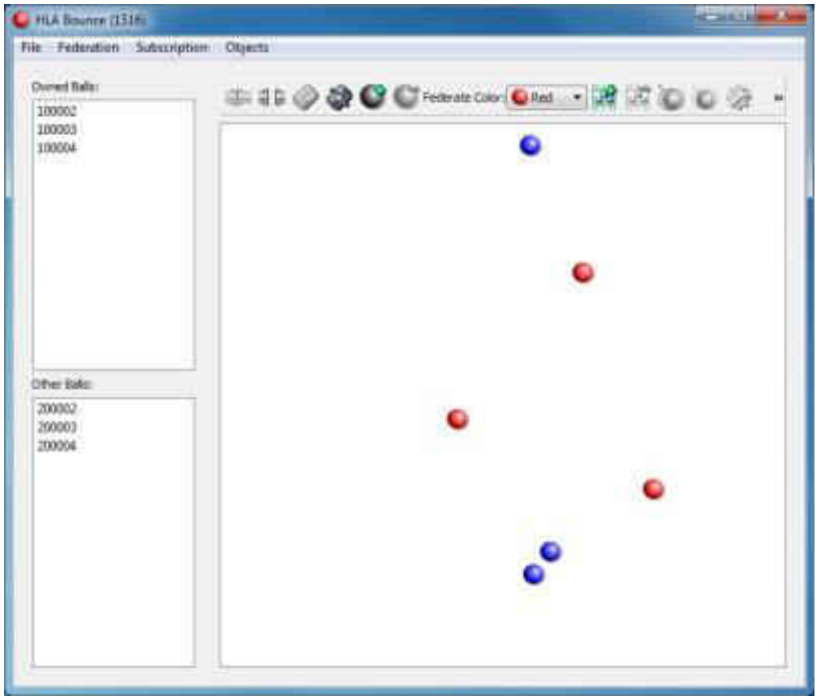

Figure 5-4. HLA bounce federation initial setting

In this federation, blue balls represent non-standard simulation's entities. If the blue ball (200002) was destroyed in a standard simulation, a CDA Module shall delete the blue ball (200002). The CDA Module can acquire the blue ball (200002) and remove it by using HLA/RTI like figure 5-5 and 5-6.

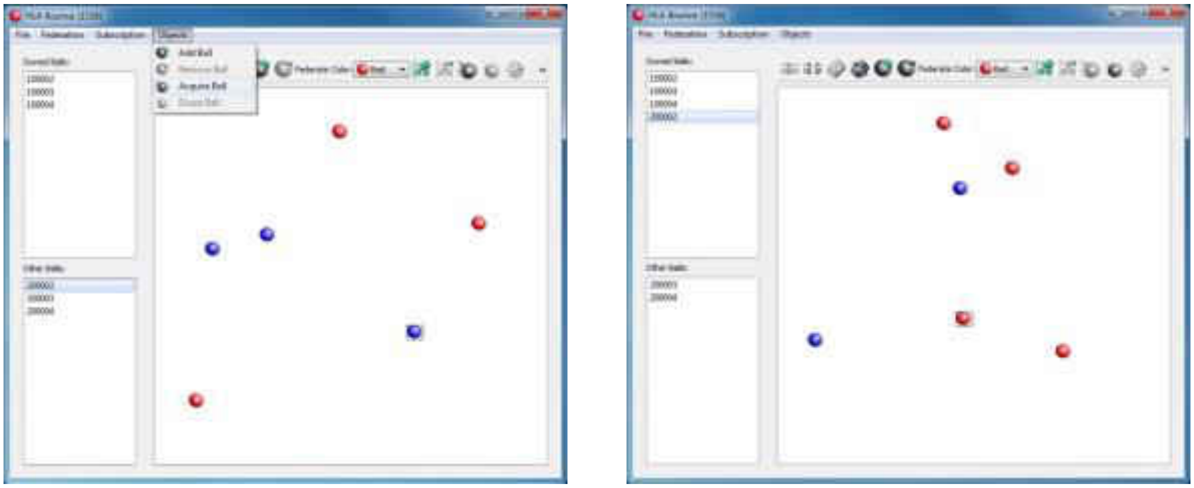

Figure 5-5. HLA bounce acquire ball process

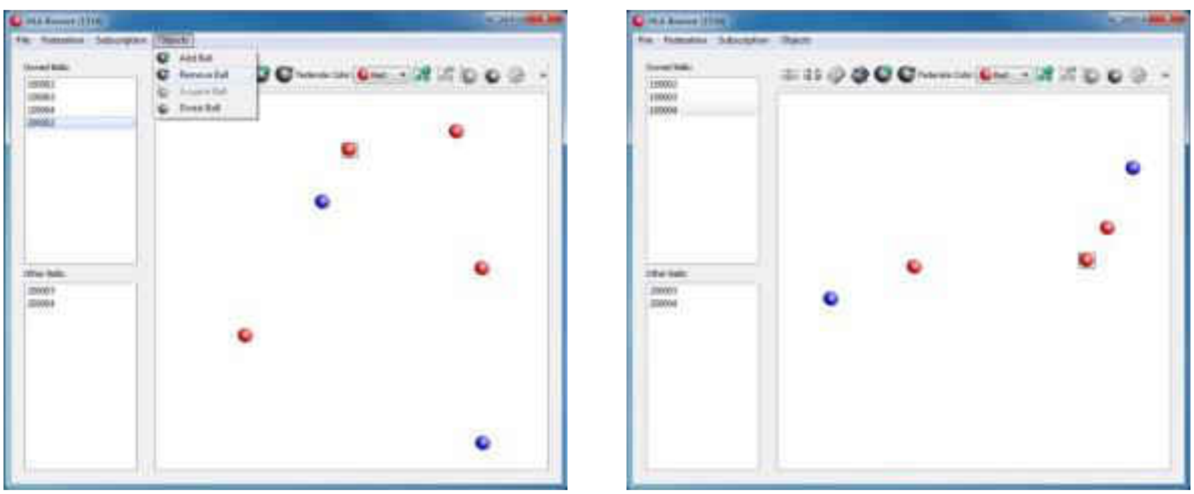

Figure 5-6. HLA bounce remove ball process

Like Figure 5-5 and 5-6, the CDA Module federate can acquire any entity in a non-standard federate and remove the entity by using HLA/RTI.

CHAPTER 6: CONCLISION AND RECOMMENDATION

6.1 Research Summary

The objectives of this research are:

- To develop in depth comprehension about HLA/RTI;
- To develop a new framework for sharing and generating CDA data; and
- To implement case studies for the new CDA framework of a HLA/RTI federation.

Finally, the research objectives are accomplished as below:

- Literature review chapter introduces and explains related HLA/RTI fundamentals for the new framework.
- The new framework for sharing and generating CDA data is suggested by using a diagram and a comparison with other methodology.
- Two case studies for the new CDA framework of a HLA/RTI federation are tested and verified.

In conclusion, this new framework can generate reliable quantitative engagement data from CDA module in a HLA/RTI federation and this unified engagement data can be helpful for virtual simulator.

6.2 Limitations

There are many challenges remaining to complete this framework with functions of step 5 presented in the Chapter 3 into the CDA module. Although the proposed framework was proved the feasibility of interoperability in a HLA/RTI federation through the two case studies, the overall software program is needed to realize it.

Also, additional function is needed to balance the damage effect for complex interaction systems. For example, if F-16 entity of the non-standard federate got a small damage from other entity of the standard federate, it makes some interoperability problems because F-16 can be moved by following its own damage effect logic. Therefore, additional Damage Balancing Module that can control the interactions is needed to solve this problem.

6.3 Future Works

While considerable research has been carried out to develop a framework for sharing and generating CDA in a HLA/RTI federation, additional work still need to be performed. The limitations explained in the previous section should be overcome. Also, the CDA framework should be verified and validated in various situations like Live, Virtual and Constructive(LVC) or two more federations.

The best way to make perfect solution for the interoperability problems about CDA is to develop one standard engagement logic for all simulation engine

REFERENCES

Aldrich, C. (2004). Simulations and the future of learning: An innovative (and perhaps revolutionary) approach to e-learning. San Francisco, CA: Pfeiffer.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Çelik, T., Gökdoğan, G. F., Öztürk, K., & Sarikaya, B. (2013). An HLA-based tactical environment application framework. *The Journal of Defense Modeling and Simulation*, *10*(3), 217-233.

Dahmann, Salisbury, M., Barry, P., & Blemberg, P. (1999). HLA and beyond: Interoperability challenges. Paper presented at the Simulation Interoperability Workshop.

DoD, D. M. (1995). Simulation (M&S) Master Plan. *Washington, DC, October*.

Fisher, Marcus, S. (2003). Software verification and validation : an engineering and scientific approach IN: Springer eBooks; New York ; London : Springer, c2007. 172 p. : ill. ; 24 cm. Language: English, Database: UCF Libraries Catalog

IEEE Standards Association. (2012). 1516–2010-IEEE Standard for modeling and simulation (M&S) High Level Architecture (HLA).

Kaushal, A., & Shukla, M. (2014). Comparative Analysis to Highlight Pros and Cons of Data Mining Techniques-Clustering, Neural Network and Decision Tree.

Kindley, R. (2002). The power of simulation-based e-Learning (SIMBEL). *The Elearning Developers Journal. The Elearning Guild, 17,* 1-8.

Lutz, R., & Drake, D. (2011). *Gateway Concepts for Enhanced LVC Interoperability.* Paper presented at the Proc., 2011 Spring Simulation Interoperability Workshop.

MÄK. VT MÄK. Retrieved 03/16/2016, 2016, from http://www.mak.com/

MÄK. (2013). *VR-Forces Users Guide*

Millett and Murray. (1986). Military Effectiveness, 2; Biddle, "Explaining Military Outcomes."

Maria, A. (1997, December). Introduction to modeling and simulation. In *Proceedings of the 29th conference on Winter simulation* (pp. 7-13). IEEE Computer Society.

Moon H. S., Kim H. S., Hwang M. S., Bae H. W., Lee D. K. (2011). The Study on Consistency of Simulation Logic about Close Combat Damage Assessment among Constructive Models : Based on Combined Arms Integrated Interoperability System. Korea National Defense Business Analysis Journal, 37-1.

Morris, E., Levine, L., Meyers, C., Place, P., & Plakosh, D. (2004). System of Systems Interoperability (SOSI): final report: DTIC Document.

Reigeluth, C. M. (Ed.). (1999). Instructional design theories and models: A new paradigm of instructional theory (Vol. II). Mahwah, NJ: Lawrence Erlbaum Associates.

Sargent, R. G. (1984). Simulation model validation. In *Simulation and model-based methodologies: an integrative view* (pp. 537-555). Springer Berlin Heidelberg.

Sargent, R. G. (2005, December). Verification and validation of simulation models. In *Proceedings of the 37th conference on Winter simulation* (pp. 130-143). winter simulation conference.

Schlesinger, et al. (1979). Terminology for Model Credibility, Simulation, 32, 3,, pp. 103-104.

SimiGon. SIMbox Version 5.6.3 Release Notes [Press release]. Retrieved from http://wiki.simigon.com/wiki/images/d/df/ReleaseNotes_563_Light.pdf

SISO. (2007). Reference for Guide: DIS Plain and Simple.

Symington, S., Morse, K. L., & Petty, K. (2001). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification (IEEE Std 1516-2000).

Tolk, A. (2012). *Engineering principles of combat modeling and distributed simulation*. John Wiley & Sons.

United States Marine Corps. (1996). Marine Corps Reference Publication (MCRP) 3-0A. Unit Training Management Guide. Washington, DC: Department of the Navy.