Best of Both Worlds: DIS Identifiers in an HLA Federation

Engin Z. Altan, Matthew Wintercorn
Science Applications International Corporation
Orlando, FL
engin.z.altan@saic.com, matthew.l.wintercorn@saic.com

ABSTRACT

In today's distributed simulation world, there are several standards available for conducting real-time, platform-level wargaming across multiple, geographically dispersed locations. The most common standards used in military simulations are Distributed Interactive Simulation (DIS), High level Architecture (HLA) and Test and Training Enabling Architecture (TENA). The biggest challenge using different standards in the same wargame is to provide interoperability between the simulations to exchange data via a common set of exchange formats, called enumerations. However, each interoperability standard defines enumerations in its own way, according to its unique data structure. TRADOC's Battle Lab Collaborative Simulation Environment (BLCSE) federation simulation architecture consists of DIS and HLA based federates and utilizes OneSAF as a main entity driver in the federation. It has become apparent that the BLCSE federation needed a methodology to translate objects' definitions between the different standards used in the federation. Our engineering team has developed a methodology to utilize translation tables to provide interoperability between the various federates. Although our methodology successfully served the BLCSE federation for several years, it was labor-intensive and error-prone. Additionally, each new entity introduced into the federation required a FOM update to include related new enumeration. This year, our engineering team introduced a better methodology to reduce the time and man-power to obtain the target results and eliminate periodic FOM updates. They developed a method to utilize existing DIS enumeration tables to provide interoperability between the various HLA federates and OneSAF. It was proved that using only DIS enumerations throughout the federation would save time and effort. However this required some changes in the OneSAF and other federates' software. This paper will detail the procedural and technical implementation that combined legacy enumeration synchronization models and modern interoperability framework into one unified approach including sample OneSAF code.

ABOUT THE AUTHORS

Engin Z. Altan is a Simulation Section Manager in the Engineering and Analysis Operation unit in SAIC. She currently leads Federation Model Management Support (FMSS) Section to support TRADOC's Battle Lab Collaborative Simulation Environment (BLCSE) experiments' technical integration and OneSAF model management. She holds a Master of Science degree in Electrical Engineering with an emphasis in Meteorology from Technical University of Istanbul, Turkey. She received her CMSP in 2008.

Matthew Wintercorn is a software engineer supporting TRADOC's Battle Lab Collaborative Simulation Environment. He holds a Bachelor of Science degree in Computer Engineering from the University of Central Florida. Since 2007, he has been involved in the design and development of OneSAF constructive simulation software.

Best of Both Worlds: DIS Identifiers in an HLA Federation

Engin Z. Altan, Matthew Wintercorn
Science Applications International Corporation
Orlando, FL
engin.z.altan@saic.com, matthew.l.wintercorn@saic.com

INTEROPERABILITY

Every year Army's Training and Doctrine command (TRADOC)'s Army Capabilities Integration Center (ARCIC), conducts several simulation experiments to provide its user community with military training, concept and capabilities integration, developing and integrating Joint and Army concepts and architecture capabilities. In these multipurpose experiments, integrating diverse simulation systems and providing interoperability is still a significant challenge and the cost is not trivial. Integrating applications and models to run in such an environment is a daunting task. Providing meaningful interoperability is an equally challenging task, as well.

M&S Interoperability is defined by JSP 110 as "The ability of a model or simulation to provide services to and accept services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together". This definition's precise description formally depicts the interoperability's framework. However, it is proven that one of the challenges of developing such a framework is properly enumerating and describing all relevant simulation attributes and providing the community with this information. Developing a common set of enumerations and making sure that they are implemented in each simulation requires considerable patience and constant attention.

BLCSE ENVIRONMENT

Several years ago, ARCIC's Battle Lab Collaborative Simulation Environment (BLCSE) mandated a requirement to transition simulation platforms from DIS standards (IEEE 1278) to a common HLA (IEEE 1516) environment and introduced OneSAF as the primary entity driver federate. Thus, the engineering team designed and developed a middleware software layer to enable DIS based federates to communicate in the HLA environment. This middleware software, named the HLA Adaptor (HLAA), provided the capability to leverage existing DIS based simulations in the HLA environment with the HLAA providing the communication adaptations for DIS federates to participate in HLA environments.

BLCSE FOM's BACKGROUND

During the BLCSE federation's transitioning from DIS standards to HLA, it was decided to take Army's Research, Development and Engineering Command's (RDECOM) developed Modeling Architecture for Technology, Research and eXperimentation (MATREX) FOM, Version 2.0, as a base model and build BLCSE required objects and interactions in it. This FOM called the BLCSE FOM. It was easy for the BLCSE federation to modify the BLCSE FOM during the experiments according to its unique objectives. However, the BLCSE FOM was accepted as an interim FOM since it had to be synchronized with MATREX FOM couple of times a year to keep the MATREX FOM as an official BLCSE federation's FOM.

The MATREX FOM provided users with two different data types that attributes and parameters utilize to determine an entity type. These data types are the PlatformType Enumerated Data Type (EDT) and the EntityTypeStruct Complex Data Type (CDT). For example, the Platform Object defined two different attributes that both can be used to determine the entity type: the "EntityType" attribute, which utilizes the PlatformTypeEDT, and the "DISEntityType", which utilizes the EntityTypeStructCDT (see Table 1).

Table 1. Snippet of MATREX FOM Attribute Table to Illustrate Two Different Enumerations' Availability in the FOM

Object	Attribute	DataType
Platform	.	
	.	
	EntityType	PlatformTypeEDT
	.	
	.	
	DISEntityType	EntityTypeStructCDT

In order to be interoperable with OneSAF, the team decided to use PlatformTypeEDT enumerations to represent entities. This was the official HLA enumeration used to determine entities in the BLCSE federation. On the other hand, the original DIS native BLCSE simulations (federates) in the federation were using DIS EntityType enumerations to represent their entities.

THE PROBLEM

Interoperability between DIS federates, OneSAF (HLA based OneSAF), and HLAA is made possible through utilizing translation mapping tables. Therefore, we needed to create "enumeration file mapping" process to overcome this challenge. The following enumeration file mapping strategy created the mapped relationship between entities in each standard (HLA and DIS) to OneSAF:

OneSAF↔HLA↔DIS

The translation table formats included comma separated values (CSV), eXtensible Markup Language (XML), and Lisp-style configuration files. Since the above mapping was necessary for each experiment, this required a lot of work. It had quickly become apparent that data translation was a burden on the BLCSE users.

- OneSAF translates internal designations for its platform composition models to external values via XML configuration files allowing it to communicate with external architectures.
- Likewise the HLAA software requires a CSV mapping from DIS entities to HLA enumerations. Manual correlation of these files represents a cumbersome, error prone and laborious task.
- The manipulation of these files requires intimate knowledge of the configuration of all federates.
- The development of mapping files required the person tasked to cross check all entries by hand.
- Required extensive knowledge about systems used for both blue and red forces.
- Some system models (entities) may not exist between simulations or no mapping possible between the entity types at all.
- The human cross check is prone to errors.
- Documentation for enumeration descriptions is difficult to obtain.

In addition to the above, the HLAA mapping files would occasionally map a single DIS enumeration to two different HLA enumerations. Therefore, if a DIS based federate sends out a DIS enumeration (enum), i.e., for a CH47 helicopter, because there was more than one CH47 HLA enumerations in the FOM, the HLAA code can map it to either HLA enumeration. By default, it chooses the biggest enumeration value. When it chooses the biggest enumeration value, everything would work fine because OneSAF's default has the bigger value mapped to an entity composition. However, if the DIS federate chooses to use the smaller enum value, which OneSAF does not have mapped, an error in proper entity translation would occur.

SOLUTION

Even though the team successfully used the enumeration files mapping process, it was slow, cumbersome, expensive and error prone. In order to simplify this enormous process, we decided to eliminate the transitional process of mapping OneSAF enumerations to HLA enumerations and then to DIS. We would map OneSAF enumerations directly to DIS enumerations. We also knew that the MATREX FOM Version 2.2 was supporting this, where the platform object's DISEntityType attribute was represented by EntityTypeStructCDT. The EntityTypeStructCDT is a mimic of the DIS EntityType enumeration fields. All we needed to do is make a minor change in the OneSAF software to support this functionality and inform the other federates to populate the platform object's DISEntityType attribute in their simulation software. The HLA Adapter was already supporting DIS enumerations as were other major BLCSE federates, including Firesim and EADSIM. One of the biggest advantages of using DIS enumeration throughout the BLCSE federation was eliminating the frequent FOM updates. Occasionally, the requirements of individual simulation experiments required us to introduce new OneSAF entity types. Due to each new entity type in OneSAF requiring an update to the FOM, we frequently had to modify and release a new FOM version to community. Therefore, we had to request federates to download and use updated new FOMs in support of interoperability. This was not fair to all non-OneSAF BLCSE federates.

Moving Away from PlatformTypeEDT

The PlatformTypeEDT, which is used by HLA objects and interactions, lists the platforms that all federates utilize. A sample list of interactions that use the PlatformTypeEDT data type is shown in Table 2.

Table 2. Sample MATREX FOM Interactions Using PlatformTypeEDT Data Type

Interaction	Attribute	DataType
CheckIn	AircraftType	PlatformTypeEDT
MissileSensor Image	TargetEntity Type	PlatformTypeEDT
NodeSimID	TargetType	PlatformTypeEDT

FireMission	Firer	PlatformTypeEDT
SituationReport	EntityType	PlatformTypeEDT

Additionally, a sample of HLA objects that use the PlatformTypeEDT data type is shown in Table 3.

Table 3. Sample MATREX FOM Object Classes Using PlatformTypeEDT Data Type

Object	Attribute	DataType
Equipment	PlatformType	PlatformTypeEDT
Platform	EntityType	PlatformTypeEDT
SpottedObject	Guise	PlatformTypeEDT

One disadvantage of the PlatformTypeEDT enumerations is that they are not informative enough to help engineers and support staff quickly identify the entity type. This encountered during the distributed integration testing and debugging the problems that arose. Table 4 clearly depicts that the FOM enumeration values do not contain any detailed information that can be associated to the entity type it is representing.

Table 4. Enumerated Data Type Usage

Identifier	Enumeration Name	Enumeration Value
Tank	M1A2	79
Tank	Tank_T80U_Armor	3534
UAV	Shadow200	240
FWA	FA_18CD	568

The solution was to officially declare the use of DIS enumerations for entity type identification for all federates.

Benefit of DIS enumeration

A benefit of using DIS enumerations as identifiers is the prevalence of these enumerations in distributed simulations for over 20 years. They are structured, well defined, and have been used by many simulations over the past two decades. The Simulation Interoperability Standards Organization (SISO) released a publication of standard enumeration values to be used in distributed simulations. The document "*Enumerations for Simulation Interoperability"* (SISO, 2010) specifies the numerical values and associated definitions for those fields that are identified as enumerations in IEEE or SISO standards. The current document's lineage can be traced back to a draft standard by University of Central Florida's (UCF) Institute for Simulation and Training (IST) called *Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation* (IST, 1992). The draft was created as a standards proposal to the Army's Simulation, Training & Instrumentation Command (STRICOM) and the Department of Defense's Defense Modeling and Simulation Office (DMSO). The document is continually evolving by the introduction of new enumerations and the removal of old, deprecated enumerations.

THE CHANGE IN THE MATREX FOM

The Original MATREX FOM defined the DISEntityType attribute as an unsigned long. However, later on in their FOM Version 2.2, MATERX changed to replace DISEntityType attribute from unsigned long to EntityTypeStructCDT data type. This change helped the BLCSE federation to conveniently utilize the DISEntityType attribute as an entity enumeration.

Table 5. MATREX FOM EntityTypeStructCDT Definition

Complex Data Type	Field Name	Data Type	Cardinality
EntityType StructCDT	Entity Kind	long	1
	Domain	long	1
	Country	long	1
	Category	long	1
	Subcategory	long	1
	Specific	long	1
	Extra	long	1

Each field of the EntityTypeStructCDT (Table 5) represented those of the DIS EntityType record fields. The following table (Table 6) depicts the usage of EntityTypeStructCDT.

Table 6. DIS Entity Type Enumeration Sample

Field #	Field Name	Possible Values
1	Entity Kind	Platform, Life Form, Munition, ...
2	Domain	Land, Air, Surface, ...
3	Country	USA, UK, Canada, ...
4	Category	Tank, Bomber, Carrier, ...
5	Subcategory	M1 Abrams, Leopard, T80, ...
6	Specific	M1A1D, Leopard A5, T-80B, ...
7	Extra	Kawasaki KLR 250

Through the use of the latest DIS enumeration standards and the knowledge of common field values, it can be possible to identify a platform, life form, or munition with reasonable accuracy. For example, it is possible for a DIS expert to quickly identify an entity as being a USA Tank just by the first four fields of the example identifier shown in Table 7.

Table 7. Example USA M1A2 Tank Entity Type Enumeration

Field #	Field Name	Field Value	Numerical Value
1	Entity Kind	Platform	1
2	Domain	Land	1
3	Country	USA	225
4	Category	Tank	1
5	Subcategory	M1 Abrams	1
6	Specific	M1A2	3
7	Extra	<EMPTY>	0

Thus, these identifiers have important information about the entity contained directly within the field values.

REQUIRED CHANGES TO ONESAF

The decision of utilizing the DIS style enumerations required some modifications in the OneSAF software and mapping files. The OneSAF software modules that provide HLA interoperability are called “converters”. These converters are bilateral, translation software modules, meaning that they translate internal OneSAF objects and messages to external HLA objects and interactions, and vice versa. OneSAF contains many converters, each with a unique purpose from handling weapon fire, to processing observation report messages, to supporting laser designation. Through the use of a configuration file, individual converters can be selected for use during runtime, depending upon the intended interoperability requirements for that experiment. The most common converter–and most critical–is the Platform Converter. The Platform Converter is responsible for creating internal OneSAF entities to represent external HLA entities and publishing entities from the local OneSAF simulation to the external HLA federation. The Platform Converter utilizes an XML-formatted entity composition mapping file to map the external PlatformTypeEDT enumeration to their respective internal OneSAF entity composition file. OneSAF uses this composition file to define the internal representation of that external entity (see Figure 1).

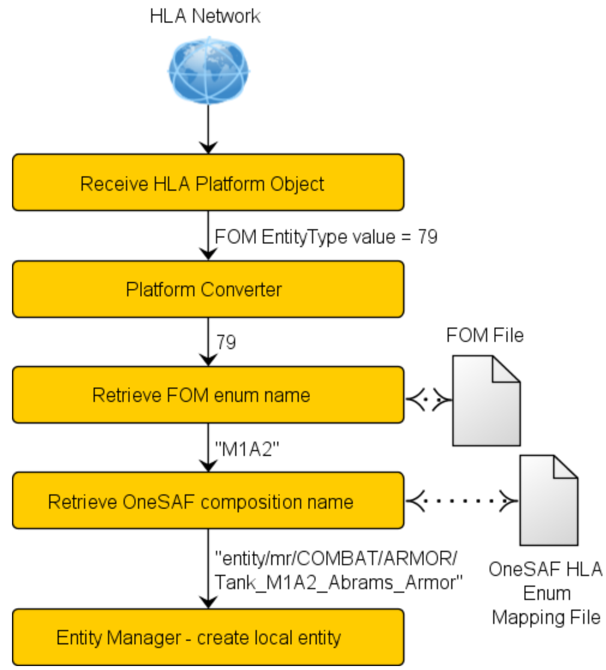

Figure 1. Flow of Incoming HLA Platform Object

Another mapping file that is used by various converters is the entity type mapping file. This is used when only the entity type information is necessary, like for observation reports or weapon fire. Both of these mapping files needed to be updated before any modifications to the converters could be made.

Updates to Mapping Files

We had two options to modify the mapping files to accommodate DIS entity type enumerations usage: a) using the mapping files from existing DIS based OneSAF directories available within OneSAF or b) update our HLA mapping files by replacing the HLA enumerations with the corresponding DIS enumerations. At first glance, it seemed obvious to use the existing DIS mapping files. After further investigation, we discovered several BLCSE-specific platforms that were not listed in the existing mapping files. Foreseeing a tedious integration of BLCSE-specific platforms with the existing DIS mapping files and mitigating risk, we decided to perform option (b). We wrote a script to replace the HLA enumerations with the corresponding DIS enumerations for a specific Platform object. In order for the script to replace the HLA enumerations in the mapping file, it needed to know its corresponding DIS enumeration. We obtained another mapping file that was used with the BLCSE HLA Adapter. That mapping file maintained mappings between HLA enumerations (PlatformTypeEDT) and DIS enumerations (EntityTypeStructCDT). This allowed the script to quickly *and correctly* replace the

HLA enumerations with its corresponding DIS enumerations. The following shows an example from the entity type mapping file before and after the script replaced the HLA enumeration:

Original mapping:
```
<entry externalIdentifier="M1A2"
oosIdentifier="tankAbramsM1A2"
mapDirection="bidirectional"/>
```

(M1A2 is the HLA enumeration)

Updated mapping:
```
<entry
externalIdentifier="1.1.225.1.1.3.0"
oosIdentifier="tankAbramsM1A2"
mapDirection="bidirectional"/>
```

("1.1.225.1.1.3.0" is the DIS enumeration for "M1A2" HLA enumeration)

The same replacements were also performed to the entity composition mapping file. The following shows an example from the composition mapping file before and after the script replaced the HLA enumeration:

Original mapping:
```
<entry externalIdentifier="M1A2"
mapDirection="bidirectional"
oosIdentifier="entity/mr/COMBAT/ARMOR
/Tank_M1A2_Abrams_Armor" />
```

Updated mapping:
```
<entry
externalIdentifier="1.1.225.1.1.3.0"
mapDirection="bidirectional"
oosIdentifier="entity/mr/COMBAT/ARMOR
/Tank_M1A2_Abrams_Armor" />
```

Incoming HLA Platform Objects

Once the mapping files were updated, the Platform Converter needed to be updated to process the DISEntityType attribute of incoming HLA platform objects. A source code sample shows the converter collecting the seven individual attributes of the EntityTypeStructCDT (see Figure 2). The converter concatenates the seven values together separating each with a period to create a string representation of the DIS enumeration. It uses this enumeration to determine the corresponding OneSAF entity composition from the mapping file.

```java
// Decode DISEntityType attribute
XDRInputStream is =
    new XDRInputStream(new ByteArrayInputStream(encodedData));

try {

    if (is.readInt () == 1) { //If the DISEntityType Cardinality is 1
        kind = is.readInt();
        domain = is.readInt();
        countryCode = is.readInt();
        category = is.readInt();
        subcategory = is.readInt();
        specific = is.readInt();
        extra = is.readInt();
    }
} catch (Exception e) {
    e.printStackTrace();
}

mtxType = kind + "." + domain + "." + countryCode + "." + category
    + "." + subcategory + "." + specific + "." + extra;

compositionName = mapper.getOOSIdentifier(mtxType);
logger.debug ("composition name is " + compositionName);
```

Figure 2. Converter Code Sample - Incoming HLA Platform Object

This new process removes the step of decoding the PlatformTypeEDT value from the FOM. Instead, the new process is much simpler by retrieving the entity composition through the DIS enumeration to entity composition mapping file (see Figure 3).

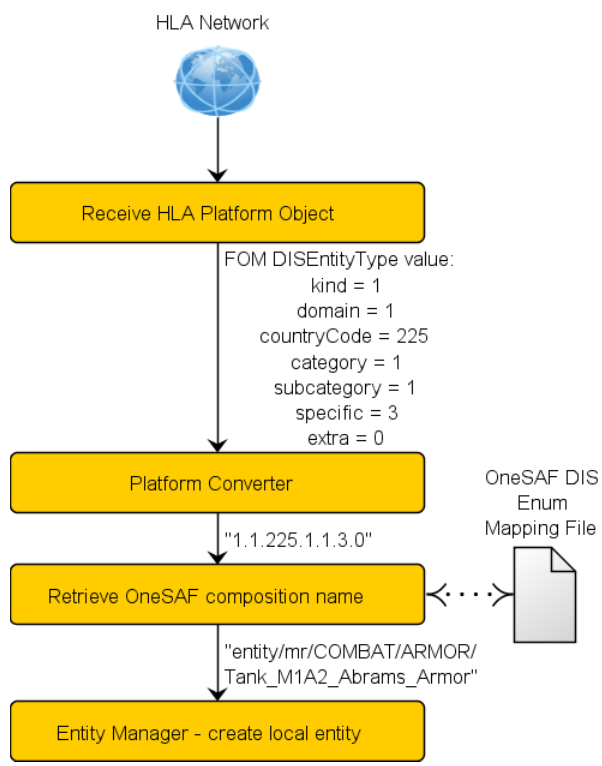

Figure 3. Modified Flow of Incoming HLA Platform Object

Outgoing OneSAF Entity Objects

The bilateral nature of the Platform Converter allows it to publish internal OneSAF entity objects to the external HLA federation. This process is very similar to the previous process but in a reverse order (see Figure 4).

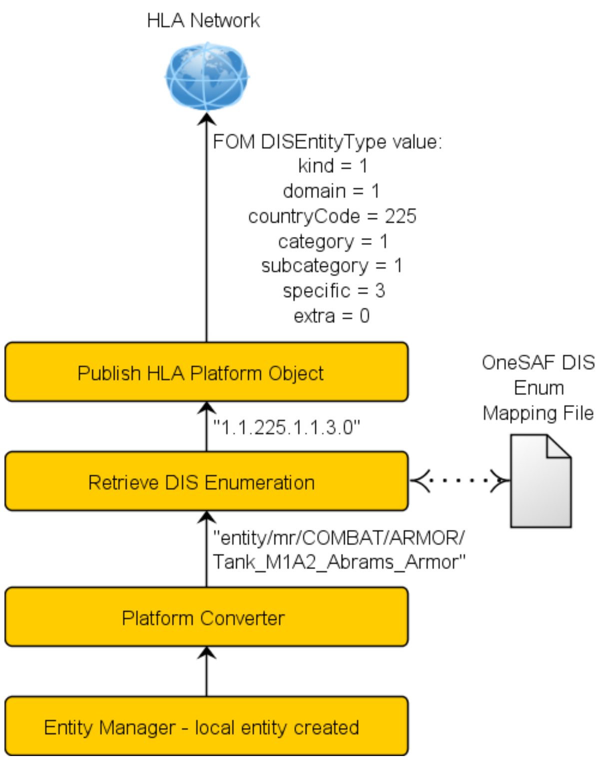

Figure 4. Modified Flow of OneSAF Entity Object Publication

The Platform Converter uses the OneSAF entity's composition to retrieve the DIS enumeration from the mapping file. Once it has this string value, the converter divides it up into its individual numerical values and sets the numerical values on the HLA platform object's EntityTypeStructCDT attribute. The OneSAF interoperability framework then completes the rest of the tasks to publish the HLA platform object with a DIS style identifier to the HLA federation. A source code sample of the converter translating the internal OneSAF entity to the outgoing HLA platform object is shown in Figure 5.

```
String compName =
    modelsLibrary.stripName(odmEntity.getCompositionName());
String externalIdentifier =
    mapper.getExternalIdentifier(compName);

// ...
// Warns if mapping does not exist for composition
// Uses default composition (M1A1, Apache, ICFullyLoaded)
// ...

// Split the DISEntityType external identifier
// and assign to individual elements
String[] eTypeElements = externalIdentifier.split("\\.");
if (eTypeElements != null && eTypeElements.length < 7) {
    logger.error("externalId is: " + externalIdentifier
        + " eTypeElements is of size: " + eTypeElements.length
        + " but should be of size 7");
}

outXDR.writeInt(1); // Set the DISEntityType
                    // attribute cardinality to 1
outXDR.writeInt(Integer.parseInt(eTypeElements[0])); // kind
outXDR.writeInt(Integer.parseInt(eTypeElements[1])); // domain
outXDR.writeInt(Integer.parseInt(eTypeElements[2])); // country
outXDR.writeInt(Integer.parseInt(eTypeElements[3])); // category
outXDR.writeInt(Integer.parseInt(eTypeElements[4])); // subcategory
outXDR.writeInt(Integer.parseInt(eTypeElements[5])); // specific
outXDR.writeInt(Integer.parseInt(eTypeElements[6])); // extra

setAttributeInClassHandle(classHandle, fomType,
        outByte.toByteArray(), outputData);
```

Figure 5. Converter Code Sample - Outgoing OneSAF Entity

RESULTS

The new method of using DIS enumerations as identifiers in the HLA federation reduced the number of unnecessary iterations of the FOM. Previously, every time we created a new entity, we were forced to add the entity's HLA enumeration into the FOM and release the new FOM to the community of practice. Subsequently, the changes in the new FOM would force the BLCSE federates to modify their simulation software accordingly. That is no longer necessary since all that is required is a change to a couple of text-based mapping files.

The decision to use DIS enumerations did not impact DIS-native federates and they did not have to make any software changes in their simulation software to accommodate it. Their architecture originally was designed to utilize DIS enumerations anyway. The software changes were primarily centered around OneSAF and supporting HLA tools and scripts. The entire process took less than a two months to transition the BLCSE HLA federates to use the DIS enumerations.

During the Army's Omni Fusion experiment of 2009 (OF 2009), there were 5 FOM changes and all of the changes were the result of creating new HLA enumerations. The following year's Omni Fusion (OF 2010) experiment, there were 7 FOM changes and 6 of them were related to the modification or creation of HLA enumerations. We first introduced the usage of

DIS enumerations as identifiers in a HLA federation during the Joint Forced Entry Warfighting Experiment 2011 (JFEWE 2011) experiment. The following table (Table 8) shows that during this latest experiment, we had no changes of the FOM due to enumeration related changes.

Table 8. FOM Changes During Recent BLCSE Experiments

Experiment Name	Total # of FOM Changes	Enumeration Related Changes
OF 2009	5	5
OF 2010	7	6
JFEWE 2011	1	-

In order to postulate an impact that this new method had on our BLCSE experiment, we must consider the amount of time not spent with the tasks spawned from a FOM modification, such as: federates' software modifications, coordination and distribution of the new FOM via the BLCSE online portal, insertion of the new FOM into the federates' systems, and the mistakes that may occur during this process. Considering these attributes, we can attribute approximately 90% savings in man hours due to this new method; this translates into cost savings for our government customer. This savings coincides with our expectations and was shown to work as planned during JFEWE 2011. The changes did not have any negative impact on the operators, event staff, or the outcome of the experiment. Therefore, it was decided that this method of defining entities and munitions will be our standard for all future BLCSE simulation experiments.

REFERENCES

Joint Services Glossary. Joint Service Publication JSP110

Simulation Interoperability Standards Organization (2011). *Enumerations for Simulation Interoperability*, SISO-REF-010-2011. Retrieved June 10, 2011, from http://www.sisostds.org/ProductsPublications/ReferenceDocuments.aspx

University of Central Florida's Institute for Simulation and Training (1992). *Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation*, IST-CR-92-12. Retrieved May 15, 2011, from http://digitalcollections.lib.ucf.edu/u?/IST,25091