ABSTRACT: An effort was recently undertaken by PEO STRI to examine simulation-based training system data model representations across a number of separate Army LVC programs. The objective of this examination was to obtain and analyze data model representations within these programs, and to develop the genesis of a common data model for Army LVC training systems. An LVC common data model will provide the Army with a unified training simulation data model for representing common data state across a number of separate Army LVC domain programs, and better support interoperability across those programs. A second goal was to establish an Army LVC roadmap, describing recommended steps needed for moving from the current state of LVC-supported training to a future where a common data model is applied. This paper provides a description of this effort and presents summary results and conclusions.

1. Background

The purpose of this study effort was to examine data model and interoperability standardization across Army and Joint Live, Virtual, and Constructive (LVC) domains and takes steps towards achieving the promise of integrated LVC training for the Warfighter.

A significant challenge in establishing a common Army LVC integrated training architecture is to determine how run-time simulation data is to be shared / aligned across the disparate LVC domains, as well as how simulation objects in the various domains will interact, both syntactically and semantically. Significant efforts have been underway focused on data models for Simulation-to-C4I systems interoperability (C2IEDM and related variants; SIMCI, and others). These efforts are primarily focused on interoperability between simulation and C4I domains and not necessarily the interoperability between C4I domains and the underlying simulation domains. For example, C2IEDM is a data model with almost two decades of development history, but remains very much an entity-oriented data model, and while it has been postulated it could serve as a common reference information exchange model, C2IEDM essentially requires the interfacing systems to share a similar resolution [1]. This limitation underscores a seminal characteristic of the LVC interoperability problem – varying levels of resolution and incongruous simulation representations.

The set of programs envisioned as partners in an Army LVC training environment use a collection of
separately developed, often incompatible data models and protocols, including the High Level Architecture (HLA) and many different HLA Federation Object Models (FOMs); Distributed Interactive Simulation (DIS); Common Training Instrumentation Architecture (CTIA); Test and Training Enabling Architecture (TENA), and others. Each domain’s execution data model and protocol have matured and evolved separately as appropriate solutions within their respective domains. However, these separate evolutions have resulted in different methods for representing what is often similar information or phenomena. Since these solutions have evolved separately and are successful solutions for each domain, there is little incentive for these programs to address the incompatibility of the Army’s LVC training environments. Without such an effort the incompatibility between Army simulation training domains will continue to grow.

Developing a Common LVC Training Data Model and Interoperability Specification collaboratively with CTIA, Synthetic Environment Core (SE Core), Joint Land Component Constructive Training Capability (JLCCTC) and other relevant Army Training Programs will provide these programs with important insight into how these domains can be integrated into an accurate and consistent training environment.

1.1 Related LVC Initiatives

LVC training has been accomplished for years, but there is constant desire to explore new and innovative ways to improve the effectiveness of such training, based on lessons learned from previous LVC training experiences [2]. Some past and on-going initiatives within the Army PEO STRI are described in [2] and include:

- Common standards, products, architecture, and/or repositories (CSPAR)
- I2 Maturity Model (I2MM)
- Data Models
- Object Models
- BCS System Interoperability
- Common CGF for Virtual domain
- Dual use tactical equipment

Looking wider, a “federated enterprise architecture” paradigm is DOD’s new approach for representing the “next generation” GIG Architecture, in which separate integrated architecture artifacts throughout the DOD are federated and employ a set of Enterprise Architecture Services for registering, discovering, and utilizing architecture data to support key DOD decision processes [3].

1.2 Scope of this effort

This effort focused primarily on examining the data models used by the principle programs’ in their portfolio of training systems for the purpose of developing an Army common LVC training data model; developing a roadmap for evolving the current state of LVC execution to an aligned future state; and development of LVC training interoperability standards. The programs included in this study span across the Army’s Live, Virtual and Constructive domains. The intent is to increase training system interoperability, supportability, and maintainability plus reduce the resources required to produce LVC exercises. This report discusses the analyses performed on the key LVC simulation programs and the findings that resulted thereof.

The actual establishment of a robust single common LVC data model in the time-frame and resource level of this effort was impossible. However, the emphasis and direction given this effort was to examine and understand the current state of the existing program data models, formulate some preliminary data model assessment and recommendations, and put together a roadmap for follow-on steps.

2. Data Model Sources

The data model analysis performed for this effort focused primarily on four programs considered representative for their portions of the LVC domain. These programs were Synthetic Environment Core (SECore), Common Training Instrumentation Architecture (CTIA), Joint Land Component Constructive Training Capability (JLCCTC), One Semi Automated Forces (OneSAF) and Future Combat Systems (FCS). Additional related data models from related programs or community standards were also included in this analysis including: Joint National Training Capability (JNTC), Realtime Platform Reference (RPR), Navy Aviation Simulation Master Plan (NASMP), and Modeling Architecture for Technology, Research, and Experimentation (MATREX).

2.1 Data Model Identification

Each of the identified program simulation systems is capable of sharing a subset of its internal data with other systems. The analysis described in this section was performed on the content of these data sets. The information used to examine and analyze the data models from across these programs was extracted from a variety of sources including: documents, XML files, data models, etc. provided by the respective programs. A summary of the data models and sources is provided in Table 1.
2.1 Data Model Discussion

The data model comparison began with an analysis and examination of the different data model products, leading to a decision about the "form" of the data object comparison. The capturing of the data models in a common format would make the comparison between the different systems’ data models more straightforward and less prone to error, as well as providing a consistent mechanism to compare and contrast the data model contents. In addition, there was an opportunity to apply an existing automated tool to assist in examining the data models.

Several options were considered for model taxonomies and formats. The options for the taxonomy included:

- using a DIS entity/interaction based structure,
- using a structure based on the SE Core specific object taxonomy,
- using a structure based on the CTIA specific entity structure, or
- using an HLA FOM-based representation.

Initially, the data models were all examined in their “native” format in order to better understand the data, and to determine the best method for comparing information across the data models. Following the initial examination, a more comprehensive and uniform method was established to better compare across the data models.

2.1.1 Initial Analysis

It was observed during the mapping and comparison process several ways in which the data models did not cleanly align. Most obviously, the different systems classified their respective entities at different levels of resolution. For example the space objects found in the SECore and CTIA both have three children for SpaceTrack: Satellite, Crewed Space Vehicle, and Space Station. OneSAF doesn’t separate different types of space tracks and treats them all the same. The JLCCTC MRF FOM includes space tracks and Space Satellites, while the JLCCTC ERF FOM only identifies basic SpaceCraft. There are many other instances in the data models where the different systems are represented at different granularities.

The differences in the object taxonomies between the systems imply the “organizational difference” is only one aspect where the data models were inconsistent. An early conclusion was that if the fundamental data model object hierarchies are different, then the underlying data model object attributes will also be divergent.

This initial data model comparison and analysis lead to the need to take a closer look at the data model object attributes to better understand the alignment, or otherwise, of the data values. The final data model analysis section next provided a more detailed and comprehensive look across the data models at the attribute and data type level.

2.1.2 Final Data Model Analysis

Because the majority of the data models included an HLA-based representation, the HLA FOM structure – specifically the Object Model Template (OMT) format – was selected for representing each data model, and to support a more uniform and comprehensive look at the data model information. For each program area, an HLA-based data model was identified or synthesized to support a data model comparison. Table 2 below provides a listing of the FOM-based data model source within each program.

<table>
<thead>
<tr>
<th>Program</th>
<th>Source Format</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTIA</td>
<td>CTIA Developers’ Documentation</td>
<td>CTIA Developer Documentation for CTIA Version 1.6, May 2007 - XML Dataset, XML Schema - CTIA Data Model Presentation</td>
<td>HTML-based documentation containing prepared presentations, reference files (IDL, XML, SQL formats) and mechanically generated documentation extracted from the source files.</td>
</tr>
<tr>
<td>SE Core</td>
<td>Virtual Simulation Architecture Distributed Interactive Simulation (V-DSS) for the Synthetic Environment Core Architecture and Integration (SE Core A&amp;I) Program Specification</td>
<td>Virtual Simulation Architecture Distributed Interactive Simulation (V-DSS)</td>
<td>Protocol Data Unit extensions to the IEEE-1378.1 DIS standards and SE Core-specific extensions to the SISO/IEEE/ANSI DIS Erflv2.1.0 standard DIS Description Document</td>
</tr>
<tr>
<td>JLCCTC</td>
<td>HLA FOM</td>
<td>JLCCTC HRF FOM V4.0 04/27/2007</td>
<td>Unique representations</td>
</tr>
<tr>
<td></td>
<td>HLA FOM</td>
<td>JNTC-Merged 04/25/2007</td>
<td>Similar to JNTC</td>
</tr>
<tr>
<td>OneSAF</td>
<td>Runtime data model represented as an HLA FOM</td>
<td>ODB_Test_FOM 04/25/2003</td>
<td>Runtime data model FOM dated 2003 is current</td>
</tr>
<tr>
<td>JNTC</td>
<td>HLA FOM</td>
<td>JNTC_DEV 160x107v2</td>
<td>Standard object model representation in support of the Joint National Training Capabilities and Joint Live Virtual and Constructive federation.</td>
</tr>
<tr>
<td>NASMP</td>
<td>HLA FOM</td>
<td>NASMP Candidate Final - Merged 1.4.7 01/26/2008</td>
<td>Supports HLA interoperability between U.S. Navy Aviation Training Systems</td>
</tr>
<tr>
<td>Standard</td>
<td>HLA FOM</td>
<td>Real-time Platform Reference FOM – Merged Draft 2 06/24/1999</td>
<td>Standard HLA FOM essentially equivalent to the DIS representation.</td>
</tr>
<tr>
<td>FCS / MATREX</td>
<td>HTML-extract files of draft FCS data model, and HLA FOM</td>
<td>MATREX fom_v4.0_26FEB2008</td>
<td>The FCS data model was obtained in an HTML-extract form.</td>
</tr>
<tr>
<td>Program</td>
<td>Source of HLA-based data model representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTIA</td>
<td>The data model information found in the CTIA XML documents was used to produce an HLA “omt” file representing the salient portions of the CTIA data model.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE Core</td>
<td>The RPR-FOM column can be considered the SE Core data model for comparison purposes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JLCCTC</td>
<td>Included a native HLA OMT file representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OneSAF</td>
<td>Included a native HLA OMT file representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JNTC</td>
<td>Included a native HLA OMT file representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASMP</td>
<td>Included a native HLA OMT file representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard (RPR-FOM)</td>
<td>Included a native HLA OMT file representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCS (MATREX)</td>
<td>The FCS program provided an elaborate, extensive (and evolving) data model covering a wide variety of items. FCS also involves the use of the MATREX FOM in their simulation virtual framework. The MATREX FOM was included for this data model comparison.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.1.3 FOM Comparison Process

The process used to compare the HLA FOM data models was to leverage an existing FOM analysis tool – FOMTool – to extract the data model information from the standard FOM OMT data file, and to import that data into a database [4]. After the data model information was populated in the database, queries, summarizations, and reports were created in order to facilitate a more consistent look across all of the data models, and to support the creation of a report whereby the data could be better compared and understood. Figure 1 provides the general process used to load and generate the extracted reports.

Execution of this process yielded two output reports. The first report provided a summary of the data model classes (both HLA objects and interactions) and the alignment of each data model class to each of the programs. The second report provided a comparison across the data models at the attribute and data type level.

2.1.3 Class Name Mapping

In many of the data models, specific object model class names varied slightly (and sometimes significantly), but appeared to be referencing the same kind of data model object. Because of this, a “mapping” process was created to help align the data model elements. A general class naming convention was created to align these disparate data model class names, and used to support a more comprehensive report across the data model classes. For example, the three classes:

BaseEntity, PhysicalEntity, Platform.Aircraft
Platform.Aircraft
Entity.Participant, Platform.Aircraft

were each “mapped” to a single class for Aircraft.

2.1.4 Comparison Reports

The output reports were used to understand, at the data element and data type level, which program data models use particular data elements within a data class. An example from such a report is provided in Figure 2.

The data model comparison reports provided an extremely large volume of information about the specific data model classes, attributes, and parameters across all eight program data models. The analysis of this information included both an overview examination of the alignment of the data model data class names and attributes, as well as review of the detailed reports and a summary of the observations from that detailed review.

2.1.5 Summary Analysis

Across the eight data models, there were a total of 625 unique specific object classes defined (essentially a union of all the defined object classes collected across the data models). Comparing across these classes, and using the class name mapping, over half (230 / 55%) of the object classes appeared in only one data model. By comparison, 45 / 11% of the object classes matched...
across two data models. Similar alignment data exists for matches of three, four, and a maximum of five data models, as identified in the chart at Figure 3. This graph provides the count and percentage of “object model matches” across all eight data models, when using the summary object class names.

![Figure 3 Summary Object Class Alignment](image)

The general conclusion from this object class alignment analysis is that, at the object class naming level, a majority (55%) of the object model contents are unique for one particular program. At best, if one data model was identified as “the LVC data model”, we would be only able to align objects with a maximum of 45% of the object data exchanged via any of the other program data models.

A similar process was used to study and analyze interaction information, and resulted in a similar view when comparing across interaction classes as shown in Figure 4.

![Figure 4 Summary Interaction Class Alignment](image)

### 3. Observations

Based on an examination of the summarized data model information, as well as examining the detailed report data, a number of observations were identified concerning the data models.

*No single data model offers an inclusive “super-set”.*

While there are many areas of the data models where there is overlap and similar (if not identical) usage of data elements, each data model also includes aspects unique to that program’s data model, or in conflict with one or more of the other data models. It is impossible to select one data model and have it meet all the data exchange needs found in the other data models.

**A subset of the data models have RPR-FOM origins and have some alignment for that RPR-FOM subset.**

When looking at the class and data type attribute comparison, it is clear where these similarities are present: there are similar usages of the data model classes, attributes, and data types across the data models. The report reveals good alignment for basic physical entities (ground, aircraft, surface, subsurface, emitter and transmitter systems) across these program data models: RPR FOM, ERF, JNTC, and NASMP. For many classes, ONESAF also has some alignment, although there are many more exceptions for ONESAF than with the others. The remaining data models do have some of these basic physical entities, but their representations are very different. Note that even with this alignment, there still remain attributes appearing in one data model and not another, or items can be aligned, but they have slightly different class, attribute, or parameter names.

*There is little consistency in how aggregate unit information is presented.*

Although the RPR-FOM data model (and the other RPR-FOM related data models) include an AggregateUnit class that is more or less aligned, other data models known to include extensive aggregate information (ONESAF, MRF) have their own unique representations. The result is there appear to be three major aggregate representations: RPR FOM (and the related variants) ; MRF, and OneSAF

**There is some consistency in representation of atmospheric data model classes across some of the data models, and is mostly absent from the others**

The ERF, JNTC, RPR data models, and MATREX all include consistent atmospheric data model information. MATREX include some variations – many of which are simply different data type names for the same attribute. MATREX is also missing some data elements found in the others (meaning it appears to be a subset).

**There is some consistency in representation of C3 Contact data across some of the data models, and is absent from the others**

The ERF, JNTC, and RPR data models all include consistent C3 Contact data model information. There are a few specific attributes missing from any one of
these models (see the detailed attribute report for the specifics).

There is good consistency in representation of C3 Network Contact data across some of the data models, but has much different representation in others.

The ERF, JNTC, and RPR data models all include consistent C3.Network.ContactNet data model information (except for a few attributes). These specific attributes can be seen in the detailed attribute report. MATREX and CTIA include different or are not represented.

There is little consistency in how target information is presented. (ERF, JNTC) are similar; ONESAF, MATREX have much different representations.

There is little consistency in how LIFEFORM information is represented.

There are a few LIFEFORM data elements across ERF, JNTC, RPR, and NASMP that are consistent, but others that are not. CTIA, MATREX, and ONESAF all appear to have their own LIFEFORM representation. The result is there appear to be four major LIFEFORM representations: RPR FOM & equivalent, CTIA, MATREX, and OneSAF.

There are a few simulation / application specific elements in the data model information: CDMTS, H60, MH60, CWHS, OneSAF, and TEN

These are very specific to the particular simulation/application and have no counterparts in the other data models.

Alignment for interactions is very limited: some alignment is also misleading.

For example, a weapons detonation is represented in different data models as:

- MunitionDetonation (ERF, JNTC, NASMP, ONESAF, RPR)
- SimulationService.FireEngagement.MunitionDetonation (ERF, JNTC, NASMP)
- SOInteraction.RDMDetonate (ONESAF)
- StaticWeaponTargetDetonation (ERF, JNTC)

These interactions can be “aligned” if each is considered as a MunitionDetonation interaction. However looking at the parameters within each separate interaction, there is much more to determining an alignment.

Examination of these MunitionDetonation interaction parameters identified a characteristic found in most of the data model contents: the semantic meaning of the data values used with the objects and interactions varies by program. The reason the different programs evolved different sets of attributes or parameters for seemingly similar items is likely because each program assumes a particular set of semantics to the information. Different semantics translates to different data characteristics which yields different attributes or parameters.

There are a few instances where information is represented in one data model as a persistent object (a traditional HLA object), and in others as a quiescent object (a traditional HLA interaction).

Either approach can be entirely appropriate for the particular environment, but makes for a challenge to map between the environments.

There are many instances where the data model object is attempting to represent a real-world element at a particular level of fidelity. Other instances have the data model representation represent something more abstract, and particular to the specific simulation environment.

Alignment between these two approaches can be problematic, as the simulation abstraction may not be representing anything in real life, but is simply a means to meet a modeling objective.

Observations for the cited examples provided in the previous section can be found again and again in many areas across the examined data models. Because there is such a wide usage of varying parameters, attributes, and classes, it is not possible to simply pick object class / attribute, or interaction class / parameter “groups” from among these data models based on a “majority of usage” and expect the resulting data model to be useful.

The real key to being able to determine a comprehensive (and defendable) set of common LVC data model representations and exchanges will be to have a focused technical effort with program participation to help systematically identify, discuss, arbitrate, and decide on training objectives requiring common data representations and data exchanges between the LVC environments, and by using those identified training objectives, identify and prioritize the needed LVC data exchanges to meet those data exchange requirements. It would be a mistake to focus on engineering an alignment of data exchanges between LVC environments that do not address such training requirements. A similar observation and recommendation was identified in a paper comparing the Joint Common Database (JCDB) data model and the WARSIM Object Model [5].
4. Discussion
This analysis and study provided an in-depth review of a set of program data models. From this examination, a number of observations and conclusions were identified related to the establishment of a common LVC data model. Based on these observations and conclusions, a roadmap for moving the LVC state of implementation forward was established.

The actual establishment of a robust single common LVC data model in the time-frame and resource level of this effort was impossible. However, the emphasis and direction given this effort was to examine and understand the current state of the existing program data models, formulate some preliminary data model assessment and recommendations, and put together a roadmap for follow-on steps.

From the data model examination, some high-level conclusions include:

- A majority (over 55%) of the data model contents examined across the programs are unique.
- Although there are various areas of “localized” alignment between all the data models currently in-use, no single data model offers an all inclusive “super-set”.
- Some data model “name and attribute” alignment provide a false sense of true alignment: semantic meaning of data values differ by program
- True data model alignment “decisions” require more knowledge and guidance than is available in the typical program data model information.
- In order to proceed with establishing a common data model for Army LVC interoperability, there will need to be a focused technical effort with program participation to help systematically identify areas requiring data exchanges between the LVC environments.
- Identifying and prioritizing the needed LVC data exchanges will require identified and prioritized LVC training requirements.

4.1 Draft data model taxonomy
A draft common data model taxonomy was established as a result of the data model analysis efforts. This product was primarily a result of creating “summary classes” as a means to compare across data models. The taxonomy was very preliminary and was considered a work in progress and too draft to include in this discussion. This taxonomy will demand more efforts to continue to evolve to form the basis for a command data model standard.

4.2 Draft data model roadmap
“If you don’t know where you are going, any road will get you there”. [6].

This Lewis Carroll adage is equally true for the LVC interoperability environment. In order to build an LVC roadmap, there must be a target in mind. Figure 5 provides a depiction of a notional future LVC environment.

This LVC depiction recognizes our individual LVC environments have, and are likely to continue to have, individual standards for interoperability within their environments. However the end-state LVC common representation is realized, there is going to be a need for an interfacing level between a core LVC representation and others. From an architectural perspective, this is arguably a positive, as this allows both sides of such an interface to evolve at separate paces.

The LVC roadmap addresses steps along the way to establishing and building up to this common representation; the LVC interface to/from this common representation; and the interfacing necessary to allow the various LVC communities to utilize the environment.

4.3 Classification and the envisioned LVC environment
LVC operations often involve a mixture of environments in terms of classification levels. A potential solution path for this in the LVC world is either a) the LVC environment settles on a single classification level and the individual environments (L or V or C) have the responsibility to connect at that level, or b) the LVC environment offers a “family” (2 or more) classification levels and builds the necessary connection in-between. This is described as a notional
LVC security bridge and an example is depicted in Figure 6.

Figure 6: Notional LVC Security Bridge
This example shows the common LVC environment have both an unclassified and a classified side. In this example, the LIVE and VIRTUAL environments interact with the unclassified side, while the CONSTRUCTIVE side interacts with the classified side.

4.4 LVC Common Representation Candidates
The LVC common representation maybe realized by applying a number of different technologies. Candidate technology and representations include:

- HLA
  - RPR FOM
  - JNTC FOM
  - Unique LVC FOM
- DIS
- Web Services / SOA
  - Reflect data similar to DIS / RPR FOM
    - Entity state data
    - Interactions between entities
- Other Custom mechanism
  - CORBA / IDL
  - CTIA
  - TENA
- Inclusion of JC3IEDM for C2 information
  - Capitalize on this evolving standard for interfacing with BCCS
  - LVC components can chose to use LVC capabilities to interact with BCCS, or they can chose to use their own native mechanisms

4.5 LVC Common Interfacing Candidates
Interfacing between the LVC common representation and external representations will need to address the following concerns:

- Insulate separate LVC programs from core LVC design and development
  - Programs must continue to support their individual missions
  - Programs should be able to, over time, evolve their internal standards, or to evolve to the core LVC specifications
- LVC core efforts can develop assistance to using and testing to the common LVC interfaces
  - Interfacing components can be applied / removed as programs evolve
- Data model / interface stages
  - Based on current data exchanges in LVC, and on-going efforts in HLA-evolved to add Web services API to HLA implementation

An initial approach to the interfacing (and LVC common representation) standards, and the progression of those standards is provided in Table 3.

<table>
<thead>
<tr>
<th>Data exchange Stage</th>
<th>Technology</th>
<th>Data Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>HLA / DIS</td>
<td>RPR-FOM / DIS</td>
</tr>
<tr>
<td>Interim</td>
<td>HLA</td>
<td>Updated FOM (JNTC ?)</td>
</tr>
<tr>
<td>Initial target</td>
<td>SOA approach</td>
<td>Data exchanges based on the above FOM data exchanges</td>
</tr>
<tr>
<td></td>
<td>(Web services)</td>
<td></td>
</tr>
<tr>
<td>Final target</td>
<td>Web services</td>
<td>Data exchanges based on established LVC data exchanges</td>
</tr>
</tbody>
</table>

4.6 LVC Common Representation Progression
The progressions shown below assume an initial LVC representation of DIS, migrating to a common LVC representation and interfacing information.

Figure 7 provides a diagram of an example live environment progression with regards to an LVC common representation and interface.
CTIA is Army standard for live ranges and will continue to remain. I-HITS currently utilizes a TENA-DIS gateway. Migration will be from I-HITS to HITS, and from TENA usage to CTIA usage. Both the TENA-DIS and CTIA-DIS gateways can serve as the interface to Virtual systems as they do today. Eventually a DIS-LVC interface can be introduced, and subsequently a CTIA-LVC interface directly.

Figure 8 provides a diagram of an example virtual environment progression with regards to an LVC common representation and interface.

Virtual implies systems like AVCATT or CCTT – eventually SECore. Virtual may retain the internal usage of DIS (or DIS-like), or adopt some other internal mechanism.

Figure 9 provides a diagram of an example virtual environment progression with regards to an LVC common representation and interface.

This example is using an ERF federation participant (JCATS) as an example, and the current use of ERF-FOM and DIS in that environment. The progress would go thru a series of stages using and re-using bridges to allow the constructive environment continue to operate while they work towards interfacing to the LVC common representation. The final step showing the constructive system(s) directly interfacing to the LVC systems may not occur – it may be the best course of action to keep an interfacing layer between these environments.

Figure 10 provides an LVC roadmap for implementing the from the current LVC state to a future state similar to the earlier depictions. This chart depicts a spiral approach – each of the products produced in this roadmap would probably undergo a series of release, each building up more capability and systematically improving the LVC interoperability state.

4.6 Consolidated LVC Roadmap

Figure 10 provides an LVC roadmap for implementing the from the current LVC state to a future state similar to the earlier depictions. This chart depicts a spiral approach – each of the products produced in this roadmap would probably undergo a series of release, each building up more capability and systematically improving the LVC interoperability state.

The roadmap identifies elements for LVC Training Requirements; Engineering processes; common representations; Engineering assets ; and integration and testing. Table 4 further expands on these elements and their meaning.
Table 4: LVC Roadmap Elements and Attributes

<table>
<thead>
<tr>
<th>Roadmap element</th>
<th>Purpose</th>
<th>Impact / Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Develop and Evolve specific LVC training requirements</strong></td>
<td>Working on alignment of data items across different programs may not be solving the data exchange needs for a particular training need. Very likely to focus on solving data alignment on areas less important.</td>
<td>If an articulated set of LVC training requirements is identified, and those requirements developed into a set of more specific data change needs between the LVC programs, it can then help to produce a prioritized set of data changes that need to be engineered (standardized) across the LVC environments.</td>
</tr>
<tr>
<td>▪ Prioritized LVC training requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ Prioritized LVC data exchange requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Establish LVC systems engineering processes and perspectives to guide LVC-related efforts</strong></td>
<td>Much of the LVC efforts fall in-between programs. A modest systems engineering effort would allow the pursuit of the required engineering artifacts.</td>
<td>Established processes will lend more formalism and structure to the development of LVC interoperability. An engineering group can investigate, prototype, and prove-out candidate technologies. The resulting process and procedures will guide the development of support assets (e.g., tools &amp; standards)</td>
</tr>
<tr>
<td>▪ Established LVC Architecture group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ Established LVC engineering group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ Establish LVC processes and procedures</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Establish LVC supporting assets &amp; lab Tools (testing, processes standards)</strong></td>
<td>Development of LVC standards and processes will require some investigative efforts. Establishing standards are less effective if they are not married with well defined processes and where possible, automated support to help implement and guide users of those standards and technologies.</td>
<td>An LVC integration lab will allow for LVC technical investigations and prototypes to execute in order to prove identified technical solutions. It will also provide a venue for programs to help to “LVC-ize” their products, or test their adherence to LVC standards. The LVC standards will provide the cohesion for the programs to help eliminate the stovepipe solutions. The testing tools will help guide the implementation to those standards.</td>
</tr>
<tr>
<td>▪ Established LVC integration lab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ Established LVC set of standards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ Established LVC set of tools to help adherence to standards</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Establish central LVC integration role and oversight</strong></td>
<td>The individual programs are not focused on the LVC integration task. This needs to occur in a venue other than the fielded sites.</td>
<td>Development of a scheme to bring together LVC products to form a system before delivering them to a Warfighter training is imperative to fielding an LVC capability that will function appropriately.</td>
</tr>
<tr>
<td>▪ Established process and rhythm for bringing together LVC components and testing prior to fielding</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Conclusion
As the LVC efforts continue, it is vital to ensure the separate LVC programs remain insulated from a core LVC design and development variations. Programs must continue to support their individual missions without disruption. As LVC core implementation evolves, interfacing components can be applied / removed as programs evolve. Programs should be able to, over time, evolve their internal standards to the established LVC standards. The core LVC efforts can develop assistance (processes, technology, and standards) to provide help to program efforts. The efforts described here are believed to represent a sound path forward for achieving better LVC operations in support of Army training objectives.

6. References


Author Biographies

JOHN TUFAROLO is the Technical Director of the Simulation and Training Group for Raytheon Virtual Technology Corporation, and has more than 22 years of experience in simulation system and simulation technology design, development, integration, and testing supporting development, testing, and application of military training and analysis simulations. Mr. Tufarolo served as Chair, Vice-Chair, and Information Director for the Association of Computing Machinery (ACM) Special Interest Group on Simulation (SIGSIM), and is a member of the ACM and SIGSIM. His professional interests include discrete event simulation, simulation systems development, and distributed system performance assessment.

PAUL PERKINSON serves as a Senior Technical Advisor and is one of Raytheon Virtual Technology Corporation's (Raytheon VTC) co-founders. Paul provides senior management and business development support and acts as a senior technical advisor for numerous distributed simulation efforts. Paul has extensive expertise in the design, development, integration, test, and operation of large-scale distributed simulation enterprises including live, virtual, and constructive domains.

HARRY SOTOMAYOR is the Chief Engineer for the Warrior Trainer Integration (WTI) Office at the U.S. Army PEO STRI responsible for the Live Virtual Constructive Integrating Architecture (LVC-IA). He has 20 years of acquisition experience working on Live, Virtual and Constructive training and simulation programs. His current interests include service oriented architecture (SOA) and LVC interoperability in support of Army training. He earned a Bachelor of Science in Electrical Engineering from the University of Puerto Rico in 1986, and a Master of Science in Operations Research (Stochastic Simulation) from The George Washington University in 1993.

LOWELL (TOM) BAKER is an electronics engineer with the US Army Program Executive Office for Simulation, Training, and Instrumentation (PEO STRI) located in Orlando, Florida, and is currently the Lead Systems Engineer for the Intelligence and Electronic Warfare Tactical Proficiency Trainer (IEWTPT). He has over eight years of experience in research, design, and acquisition of Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Systems for the US Army and US Navy. He has a Bachelors of Science degree in Electrical Engineering from The Citadel, The Military College of South Carolina, and a Masters of Science in Program Management from the Naval Postgraduate School.