LVC Aspects and Integration of Live Simulation

09F-SIW-090

Per M. Gustavsson
Ulf Björkman
Joakim Wemmergård
Objective

Present one approach to integrate Live Virtual and Constructive simulation from an operational need driven perspective.

Present Methods and Tools that are needed to allow heterogeneous architectures, data/information/object models and protocols to co-exist and still provide the simulation capacity needed for the warfighter.

This paper is based on findings from:
- The LVC-aspects group within MSG-068
- The Live Virtual Constructive Architecture Roadmap
- Model based Data Engineering
- Information Centric Integration
- …
Outline

- Background – situation
  - Snow Leopard – MSG-068
  - LVC
  - LVCAR an Outlook

- LVC Use Cases - MSG-068 LVC Aspects Group
- Identified IOP issues
- Traditional Integration
- Information Centric Integration
  - Benefits
  - Methods and Tools

- Summary
Snow Leopard → MSG-068 Snow Leopard

The Allied Command Transformation (ACT) has initiated the NATO Snow Leopard program

“to deliver to NATO and partners a persistent, distributed and joint training capability able to support training from operational to tactical level across the full spectrum of operations, while leveraging existing national expertise and capabilities”.

Snow Leopard four subprojects

- NATO Training Federation (NTF)
- NATO Live-Virtual and Constructive Federation (NLVC)
- Advanced Distributed Learning (ADL)
- Shared Scenarios.

MSG-068 - Education and Training Network (NETN) is a Research Task Group that will provide M&S recommendations to the Snow Leopard project.

the Federation Architecture and FOM design (FAFD) group has been formed to support NATO Training Federation (NTF) and NATO Live-Virtual-Constructive Federation (NLVC) subprojects in Snow Leopard.

LVC-Aspects group focus on LVC issues related to the Snow Leopard project (within FAFD)
LVC Definitions – once again

- **Live Domain** – The domain where live participants operate operational systems and platforms (including their full range of mobility) in the physical environment.

- **Virtual Domain** – The domain where live participants operate simulators/emulators/operational systems in a synthetic environment.

- **Constructive Domain** – The domain where live participants, typically command and staff trainees, conduct activities in an environment constituted by a large scale constructive synthetic environment (simulation). The trainees provide stimulus to simulated forces at all levels and act upon consequences generated by the simulation.

- **Live Simulation**: A simulation involving real people operating real systems with Simulated Weapon Effects.

LVCAR presentation [6] slide 15, the following definitions of LVC are presented. LVC-Aspects uses the following definition of Live simulation [7]:
Architecture Usage

- Test and Training Enabling Architecture (TENA) 16%
- High Level Architecture (HLA) 37%
- CTIA, Common Training Instrumentation Architecture (CTIA) 3%
- Distributed Interactive Simulation (DIS) 37%
- Other 7%
Simulation Architectures and Their LVC coverage

Live
Virtual
Constructive

DIS
ALSP
HLA
TENA
CTIA
VBS2

90 95 00 05 10 15
Figure should be seen in the context that the author’s of [10] (MSG-054) identified and provided a framework to move towards a harmonized LVC architecture.
LVCAR fundamental guidelines

**Do No Harm**
- … Rather, as the architectures converge, the users themselves should decide if and when to merge their architectures, based on both technical and business concerns. Any attempt by the DoD to force a convergence solution on an unwilling user base is likely to fail.

**Interoperability is NOT Free**
- … LVC interoperability is not free. It is not reasonable to expect that LVC interoperability goals can be met with little or no investment. The Roadmap will be designed to require only limited investment early in its implementation, with subsequent investments dependent on demonstrable progress…

**Start with Small Immediate Steps**
- … While architecture convergence could reduce or eliminate several of these problems, it is not practical to expect any significant degree of convergence to occur for many years; LVC users need near-term solutions. These solutions include actions such as improved gateways/bridges, common object models, and common development/execution processes. Many of these solutions can be implemented at low cost, and provide significant near- and midterm value to the LVC community.

**Provide Centralized Management**
- … a strong, centralized management team … needs to have considerable influence on the organizations that own the existing architectures, and must also have influence on funding decisions related to future LVC architecture development activities..
LVC Use Cases exemplifying integration of Live Simulation

- **Extended Air Defence simulation:**
  - Several types of real operational sensors (radar) airborne/ground-based Air Defence systems fed with a synthetic environment and with weapons like Patriot connected. Threats includes both aircraft, but also ballistic and cruise missiles.

- **UAV training:**
  - Often you are not allowed to fly UAVs over non-controlled areas. This can make it impossible to include UAVs in tactical ground training. Another problem is that UAV resources are not available for training. Therefore during Live training, e.g. tactical ground combat training, UAVs can alternatively be added through simulation. This will require high resolution 3D-terrain data and location status from moving objects to be able to create "simulated" video streams.

- **Composite Air Operations (COMAO):**
  - Large number of aircraft and air/ground defence to be able to attack certain targets. Live aircrafts and live sensors and AWACS with real pilots/real operators.

- **Close air support/ Indirect fire support:**
  - Training of the Forward Air Controller/ Forward part in a live environment. Includes live simulation using technologies such as augmented reality and high resolution terrain data.

- **Constructive wrap-around:**
  - A live simulation part (e.g. company) wrapped into a larger context (e.g. a brigade) of constructive or virtual simulation. The constructive part could be logistics, transportation of troops.

- **Indirect fire simulation:**
  - E.g. urban operations using simulated indirect fire (mortars) influencing soldiers/OPFOR on the tactical level.

- **Training in real C2-systems:**
  - In many of the above examples of training and simulation examples where real operational C2-systems are hooked to other live operational systems and with real people involved operating them. Processes on the top are the political values and objectives of the coalition.
Identified IOP issues

- **Availability of "state" information**
  - What type of information is needed, available and possible to communicate to and from live players (including both human and machine players) and what will the possible influence be on simulation interactions.

- **Inject simulated data into live operational systems**
  - When hooking into operational platforms, e.g. vehicle platform hardware and operational C2 systems. The simulation infrastructure and models need to adapt to the operational interfaces & technology. What data is available now on different platforms?

- **Automated Aggregation/disaggregation**
  - What information is possible to retrieve from operational as well as simulation systems over the communication links, e.g. what information is actually exchanged due to bandwidth considerations?

- **Unreliable communication links**
  - Intermittent loss of communication can be of a major issue, e.g. how to rejoin to the simulation and deal with differences if a "lost" live entity has been replaced by a simulated entity.

- **Gateways**
  - A need to adapt to existing operational interfaces and technologies, using data and interfaces that are available now on platforms.

- **Ownership transfer**
  - It is identified that the Ability to teleport into other entities e.g. in a synthetic wrap-around scenario, is a capability needed.

- **Non-real time issues**
  - Live is only real-time!
  - How to deal with reset, time jumps or faster than real time.
Traditional Integration of Systems

Application Centric

Protocol/Interface Centric

Shared Protocol
Integration Problems

**Complex Integration:**
- To integrate a Virtual or Constructive simulation element into a LVC simulation, it may be necessary to upgrade several existing applications. The more applications that are integrated, the more complex it becomes to integrate an additional application. Further, when upgrading an application, existing functionality may be affected, requiring even more work. This complexity makes it hard to adapt to new protocols.

**Long Time-to-market (customer-usage):**
- The time-to-market is how long it takes to create a new function by integrating a number of applications that together satisfies a new need or sudden requirement. Since integration is complex, the time-to-market will be long.

**Rigid Integration:**
- To change the way a number of applications are integrated may require re-integration of the applications all over again because of the interdependency between the applications. Integration is rigid, and inflexible.

**High Costs:**
- Each additional application that is integrated potentially requires more integration work than the last one. This makes integration costs increase exponentially. The integration becomes more and more complex for each application that is integrated. Further, since each additional application that is integrated may affect several other applications, life-cycle costs will also remain high.
Information Centric Integration
Information Centric Integration – benefits

- **Communication Autonomy**
  - ... each application only needs to communicate with the infrastructure. This communication is preferably made using the native protocol of each application thereby exposing the full data model of each application to the infrastructure.

- **Simpler Integration**
  - ... the information infrastructure needs to support its communication protocol. However, the other integrated applications can continue as before. The information infrastructure takes care of necessary translations. This makes it easier to adapt to new protocols.

- **Flexible Integration**
  - ... the only change is the configuration. There is no need to modify and thereby integrate the applications all over again.

- **Easier to Create New Capabilities**
  - it can be sufficient to combine information from existing systems in a new way. All that is needed to do is to create a new Configuration

- **Re-use Systems and Identify Gaps**
  - ... combining existing systems will provide the ability to identify missing parts.

- **Easier to Exchange Information**
  - Systems ... no longer need to be aware of each other. All a system needs to concern itself with, is what information to provide to the infrastructure, and what information it is interested in

- **Easier to Replace Systems**
  - Since systems are unaware of each other, it is easier to replace a system, add a new system, or remove an old system. As long as the relevant information is available via the information infrastructure, it does not matter which system supplies it.

- **Shorter Time-to-market**
  - ... operational procedures and methods can be evaluated, verified, trained early in the process.

- **Lower Costs**
  - To integrate an additional application requires roughly the same amount of work as the last one. The difference is if the infrastructure has to support a new communication protocol, or not. This makes integration costs increase linearly. Further, integrating an additional application will not affect other applications. Thus, life-cycle costs remain at a lower level.
Information Centric Integration – benefits

![Graph showing integration effort vs. integrated systems](image-url)
Supporting Methodology and Tools Model based Data Engineering

Data Administration
- Where are the data? In what format?
- How can the data be accessed?

Data Management
- What do the data mean?

Data Alignment
- Can all needed data be obtained?

Data Transformation
- “How to transform/mediate data?

A Common Reference Model (CRM) consists of:

Data Elements that are the basic containers for data as used in data models.

Value Domains comprise the allowed values for an associated data element.

Conceptual Domains define sets of categories where the categories represent the meaning of the permissible values in the associated value domains.

Data Element Concepts describe the contextual semantics. They comprise the contextual information on the conceptual level.
OSI implementation

- The business logic in the User Application only knows the information model loaded into the Datamanager.
- The Datamanager is responsible to keep the information model consistent and up to date.
- The driver only knows:
  - How to transfer changes in the information model to the native protocol.
  - How to update the information model according to the protocol.
The structure of a database is defined in an object model file.

Transformation between two object models is defined in an object model transformation file.

The connectivity project file defines the combination of drivers, object models and object model transformations.
Units are created and associated automatically by WISE using the order of battle provided at exercise setup.

System A is an aggregated-level simulation. System B is an entity-level simulation.

Unit relationships are resolved using the order of battle provided to WISE at exercise setup.

WISE

System A

System B

Entity A1

Entity A2

Entity A1

Entity A2

Entity A1

Entity A2

Unit X

Unit A

Unit X

Unit A
Disaggregation

The unit level is used to determine whether disaggregation is needed.

System A is an aggregated-level simulation.

System B is an entity-level simulation.

Unit relationships are resolved using the order of battle provided to WISE at exercise setup.

Entities are created by WISE when Unit A is created and updated whenever Unit A is updated.

System A

WISE

System B
Object models differ

Object Model A

Vehicle
- Name
- Category
- Position

Create if
Vehicle.Category="Car"

Object Model B

Car
- Name
- Position

Create if
Vehicle.Category="Motorcycle"

Motorcycle
- Name
- Position

Create if
Vehicle.Category <> "Car"
AND
Vehicle.Category <> "Motorcycle"

Other
- Name
- Position
WISE ENHANCED LVC & C2 INTEGRATIONS

- C2-Framework (NL)  L – C2 – C
- TaCCS (GBAD AU)  Sensor – C2 – Shooter
- GÜZ Interoperability (AU)  L – C – L
- MSV (FI)  L – V – C
- Common Shield (DE)  L – C2
- OneSAF – DITS (PL)  L – C – C2
- Saab CDC (SE)  C2 – L – V – C
- Saab TES – TENA (US)  L – V – C
- SEPO (Secure Port Operations, SE)  L – V – C
- QuickWin (SE, NATO)  L – C2
- Saab TES – OneSAF (US)  L – C2
- Saab TES – CTIA (US)  L – L
- Bluefor tracker NBG08 (EU)  L – C2
- BMS NBG 11 (EU)  LVC – C2
- WCU (SE)  L – C2
WISE CONNECTIVITY DRIVER SELECTION

- **Standards & Architectures**
  - DIS
  - TENA
  - CTIA
  - HLA (1.3 and 1516) with RTI’s from MÅK, Pitch NG Pro and others
  - MIP (C2IEDM, JC3IEDM)
  - Link16 (Socket-J, LinkPro)
  - C-BML
  - MSDL
  - SOA
  - HTTP
  - ODBC
  - NMEA0183

- **Saab Products**
  - Saab BMS (MIP)
  - Saab CAOC (Link16)
  - Saab TES
  - Saab Wargaming (BattleTek)
  - Saab Tactical Synthetic Environment
  - Saab 3D (Grape)
  - Saab Civil C2
  - Saab Anti-Air (RBS 70 Manpad)

- **Applications & Programs**
  - Google Earth
  - Mil/Mel Planning (Exonaut)
  - Saab Capability Development Center

- **2009 Releases**
  - RTPS
  - DDS
  - Mail (POP3, SMTP, MS Exchange)

- **Serious Gaming**
  - Steel Beasts 2
  - Microsoft ESP
  - VBS2
Cope With Change

Summary

No new protocol will solve all problems

How is it possible to achieve cost-effective training?
- By using a common training environment where different systems can work together regardless of protocols and internal object models. Following the ideas of Information Centric Integration ([28]).
- By focusing on the information that each system JCOM, the LCIM concept, Common Reference Model, C-BML etc.
- By integrating the common training environment with configuration mechanisms instead of software programming.
- By allowing the user to define the common information model. Following the MBDE approach and indicated in JCOM
- By integrating information instead of communication protocols. As described in [28][11] and defined in LVCAR [5][6][12], MSG-045[10], MSG-068 [7].

From a system theoretical view point such approach is System Information Ego-Centric and supported by agile Integration Architecture Frameworks that are based upon the underlying theory in Model Based Data Engineering.

The issue “What methods and tools are needed to allow heterogeneous architectures, data/information/object models and protocols to co-exist and still provide the simulation capacity needed for the war fighter?”