LVCAR Enhancements for Selecting Gateways

Kurt Lessmann
Trideum Corporation
4946 Research Drive
Huntsville, AL 35805
256-704-6116
klessmann@trideum.com

David Drake
Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723
240-228-4523
david.drake@jhuapl.edu

Dannie E. Cutts
The AEgis Technologies Group, Inc.
410 Jan Davis Drive
Huntsville, AL
256.799.1154
dcutts@aegistg.com

Michael J. O'Connor
ITT Corporation
6767 Old Madison Pike
Suite 160
Huntsville, AL 35806
256.964.1470
michael.oconnor@itt.com

Keywords:
LVCAR, Gateway , GPB, GDL

ABSTRACT: *The Live Virtual Constructive (LVC) Architecture Roadmap (LVCAR) report, sponsored by the Modeling and Simulation (M&S) Steering Committee, recommended several actions to promote the reuse of common M&S capabilities and to foster common formats and policy goals to promote LVC interoperability. One of these recommended actions pertained to the development of common gateway capabilities. The first phase of this effort reviewed the gateway needs of users and developers to develop strategies to support more effective LVC integration using gateways. The current LVCAR phase is focused on the implementation of the selected "Enhance" strategy, which emphasizes improvements to the native methodologies and practices LVC engineers follow to select and configure gateways for their specific applications. Two of the products support the selection of a gateway and two of the products support the use of gateways.*

The three products supporting gateway selection are the Gateway Capabilities List, the Gateway Performance Benchmarks (GPB) and the Gateway Description Language (GDL). The Gateway Capabilities List is the information that describes the required capabilities of a gateway from a user perspective, and documents the gateway capabilities from a gateway developer perspective. The GPB provides a structured approach to identify and collect gateway performance parameters. The GDL is intended to document gateway operational and functional capabilities as well as GPB-based performance metrics. The users may use the GDL to document their gateway needs for a particular multi-architecture event. The gateway developers may use the GDL to document the capabilities and performance details of their gateway solution. Having the providers and users adopt GDL as common specification enables more efficient and quantifiable gateway selection.

This paper describes how the three products support the efficient selection of the most appropriate gateway, how those products address gateway selection issues, and how they will facilitate effective application of these products in future LVC environments.

1. Introduction

In 2007, the Department of Defense (DoD) sponsored an initiative to examine the differences among the major simulation architectures from a technical, business, and standards perspective, and to develop a time-phased set of actions to improve interoperability within multi-architecture simulation environments in the future. This initiative is called the Live-Virtual-Constructive Architecture Roadmap (LVCAR). The development of the Roadmap began in the spring of 2007 and continued for approximately sixteen months. The result of this activity was a final report [1] and supporting documentation that collectively totaled over a thousand pages. The implementation of LVCAR recommendations began in the spring of 2009, and is currently in progress.

The core purpose of the LVCAR implementation effort is to improve the technical quality of multi-architecture simulation environments in the future while reducing costs and development time. Gateways are a key enabler of multi-architecture simulation environments; thus the LVCAR Common Gateways and Bridges Task was designed to provide the LVC user community with enhanced technologies and processes for gateway discovery, selection, configuration, and employment. These enhancements to the methodologies and practices that LVC engineers follow can be grouped into two primary categories: gateway selection and gateway use enhancements.

This paper presents the proposed enhancements to select an appropriate gateway that meet distributed multi-architecture event requirements. This paper presents the findings in 5 major sections: a) gateway selection problem statement; b) gateway selection technologies; c) gateway selection process; d) gateway selection future efforts and e) Summary

For additional information regarding the Common Gateways and Bridges Task, the reader is encouraged to read the following Spring 2011 Simulation Interoperability Workshop (SIW) papers "Gateway Concepts for Enhanced LVC Interoperability" (11S-SIW-024) and "LVCAR Enhancements for Using Gateways" (11S-SIW-025).

2. Problem Statement

The integration of Live, Virtual, and Constructive (LVC) Modeling and Simulation (M&S) assets into a unified distributed simulation environment is commonplace today. The need for such environments is widespread within the DoD, as applications like joint training exercises, joint experimentation, and distributed Test and Evaluation (T&E) events all require the ability to quickly compose new simulation environments from existing LVC assets. These environments can be very resource intensive to establish, and the time needed to design, develop, and test these environments can be excessive with respect to the schedules of operational users. Based on past experiences of integrating large-scale distributed simulation environments, it is generally preferable to adopt a single common simulation architecture when distributed simulation environments are designed.

A multi-architecture LVC simulation environment can be defined as an environment that is configured using simulations whose external interfaces are aligned with more than one simulation architecture. Common examples of these architectures include Distributed Interactive Simulation (DIS), High Level Architecture (HLA) and Test and Training Enabling Architecture (TENA). When more than one simulation architecture is used in the same environment, interoperability problems are compounded by the architectural differences. For instance, middleware incompatibilities, dissimilar metamodels for data exchange, and differences in the nature of the services that are provided by the architectures must all be reconciled for such environments to operate properly.

When distributed LVC simulation requirements dictate the use of multiple architectures, specialized applications called gateways are commonly used to enable interoperability between the systems using the disparate simulation architectures. A partial listing of requirements that event managers face that dictate the need for multi-architecture distributed simulations include: legacy simulations that cannot migrate to a common architecture, an enclave of simulations that are already interoperable using a common but different architecture than other simulations in the environment, and the desire to integrate common infrastructure resources like loggers, viewers, and test tools that utilize a different architecture. As a general rule, gateways are standalone applications that translate data from one simulation architecture to another.

If adopting a common LVC simulation architecture and a common Simulation Data Exchange Models (SDEM) across the simulation environment is not possible, the second recommended approach is to utilize a gateway to convert the simulation data from the various simulation architectures to a single common architecture for distribution across the environment. Having a common architecture for data distribution allows common tools and processes to be used at each node. This provides enormous benefits to the Event Manager by optimizing the verification of data transmission though the use of common test/visualization applications at each site, using common event management tools across the environment, enabling consistent logging capabilities and practices, elimination of logging discrepancies due to duplicative and inconsistent logging resources, elimination of duplicative data being sent over the wire when multiple architectures are used concurrently, and a host of other interoperability benefits.

As event managers are faced with the reality of integrating disparate architectures, it becomes a design challenge that necessitates the selection and use of appropriate gateways. The design challenges can be complex as the gateways must be selected and configured to provide specific functionality, capabilities and performance specifications that meet the interoperability requirements. The current process of identifying and selecting appropriate gateways can be considered ad-hoc at best. The Common Gateways and Bridges Task team addressed these design challenges by providing a structured systems engineering-based approach that define common technologies, processes, and tools. When combined, the processes and tools enhance the gateway selection process thereby reducing the risk associated with developing a multi-architecture environment.

The Common Gateways and Bridges Task team defined several technologies and processes to efficiently and effectively select an appropriate gateway(s) that meet the event requirements. The Common Gateways and Bridges Task technologies include the Gateway Performance Benchmark, the Gateway Capabilities List and the Gateway Description Language. These technologies, along with the processes and associated tool sets that employ them, will be further discussed in the sections below.

3. Gateway Selection Technologies

The Gateway Capabilities List, Gateway Performance Benchmark (GPB) and the Gateway Description Language (GDL) were developed to provide a common technology base to enhance the selection and use of gateways to help solve interoperability needs of multi-architecture simulation environments.

The Gateway Capabilities List

The Gateways Capabilities List is the core technology that defines a set of capabilities that a gateway might implement. This document describes the capabilities of a gateway from a user perspective, and documents the gateway capabilities from a gateway developer perspective. The list is grouped to support understanding of the capabilities. There are two main categories, Functional and Operational. The Functional capabilities tend to be concrete and testable. The Operational capabilities tend to be more descriptive of the gateway. These main categories are sub-divided into several more detailed subcategories. Each subcategory provides a definition for its associated capability.

Each capability has three elements: Capability Definition, Examples, and Levels of Implementation. The Capability Definition provides a concise definition of the capability. The Examples provide context using a real-world example with which most readers would be familiar. Many of the capabilities can be implemented to different levels. The Levels of Implementation provide a defined set of possible implementations. The number of levels varies based on the capability. Generally a level of 0 means that the capability is not implemented, and conversely, a level of 5 generally means that the capability is fully implemented. Generally a level of 3 represents a partial implementation. Some of the levels do not have defined definitions. These may be selected to represent implementation between the defined levels. Some capabilities are either implemented or not. For these, the levels are a binary “yes” or “no.”

Functional capabilities represent actions required by the gateway to meet SDEM, architecture, or exercise management needs. These are considered the core capabilities of a gateway. The functional capabilities usually determine if the gateway meets the user functional needs when interfacing between multi-architecture LVC resources [2,3]. There are six formal areas of gateway capabilities that are associated with “functional” translations. They are SDEM translations, SDEM behaviors, architectural translations, architectural behaviors, exercise management behaviors, and fault tolerance behaviors.

This definitive list of gateway capabilities will provide the necessary foundation for detailed side-by-side comparisons of candidate gateways. This product will support an informed selection decision as conducted by user programs in the future.

The Gateway Performance Benchmarks (GPB)

Distributed simulation is a strategy for combining multiple, interoperable LVC resources to model larger, more complex systems. In such composed systems, interoperability and performance have always been “holy grails.” Both interoperability and performance are dependent on the distributed simulation system components and gateways are a critical component of the overall LVC multi-architecture environment. However, a standard set of performance metrics for gateways does not currently exist. Thus, the GPB are intended to address this gap. This paper provides a brief overview of the GPB. Additional details of the product and processes will be made available to the community as the artifacts mature.

The purpose of the GPB is to better define measurable performance characteristics for distributed simulation gateways and to provide a set of test cases through which gateways can be evaluated against these characteristics.

A further-reaching goal for the GPB is that the developers will have the details of the performance metrics that are sought-after by the end-users, which will in turn drive the priority of improvements of future gateways and their upgrades.

The GPB is the formalization of performance benchmark metrics and processes that are specific to gateways. The GPB is defined so that developers, testers, and consumers of gateways have a consistent set of metrics to determine which gateway or gateways will best suit the needs of the end-user. The GPB defines a structured set of use cases that collectively define a range of "typical" application types (e.g., large virtual training event, small faster-than-real-time constructive event, or hardware-in-the-loop event) to which the benchmarks can be applied. Table 1 lists the proposed performance metric elements and means of measuring.

Performance Metric Element	Definition	Possible Means of Measure
Resource Utilization	Loading levels for system resources:	
	• *Memory*	Percent of available megabytes or number of pages input and output
	• *Central Processing Unit (CPU)*	Percentage used for both average and maximum, and number of instructions per second required
	• *Disk*	Percentage used, and number of access operations required
	• *Input / Output (I/O)*	Number of operations for both input and output
	• *Database*	Number of database accesses per second
	• *Network*	Percentage of bandwidth used
Speed / Response Time / Latency	Time required to process inputs	Input/output response time and queue lengths (#messages/tasks waiting)
Throughput	System processing capability	Processing rate for messages, data streams, or packets
Scalability	Ability for multiple system components to process data flow efficiently	Multiple system tested using parameterized filtering
Endurance / Robustness / Stability	System component reliability and uptime	Mean time between failures
Performance-Related Accuracy	Minimizing output errors that are due to performance characteristics	Percentage of correct output data

Table 1 Performance Metrics

Before a set of gateway-related use cases can be fully described, scenario and operational parameters must be defined. Scenario parameters describe characteristics of the simulation, while operational parameters describe the operating environment for the simulation hardware and software. Table 2 and Table 3 depict the scenario and operational parameters.

Parameter	Definition	Initial Estimated Parameter Values
Persistent Object Count	The number of persistent objects the gateway will have to address/translate.	The number of objects processed for the entire run of the benchmark: **Low**: 100 persistent objects **Moderate**: 1,000 persistent objects **High**: 10,000 persistent objects
Transient Object Count	The number of transient objects the gateway will have to address/translate.	The number of messages that are processed per second **Low**: 10 messages **Moderate**: 100 messages **High**: 1,000 messages
Update Rate for Persistent Objects	The update rate, based on a given unit of time, being sent through the gateway for each persistent object.	The number of persistent objects that are updated per second: **Low**: 20 persistent object **Moderate**: 200 persistent objects **High**: 2,000 persistent objects
Traffic Pattern	Traffic patterns can be generated in either a continuous or burst mode for persistent and transient objects.	The network traffic patterns for each of the sides of the gateway: **Continuous mode**: Each second of network traffic is plus or minus 10% from the last second. **Burst mode**: Network traffic includes bursts of 50% more packets than the average network traffic.
Complexity Of Data Exchange Models	Data Exchange Models (DEMs) for gateways range from simplistic, where every message has all the information regarding an object or interaction, to highly complex nesting and indirection, referencing stored arrays of object information.	Complexity level of data exchange models: **Low complexity DEM**: no more than three levels of inheritance with only simple data types. **Medium complexity DEM**: no more than five levels of inheritance with complex data types and enumerations, and only one containment restriction. **High complexity DEM**: would have more than five levels of inheritance with complex data types and enumerations, and multiple restrictions on search queries using containment.
Complexity Of Translation	The level of computational difficulty to translate between simulation data exchange models	Levels of translation complexity: **Low:** simple single algorithmic translation (e.g., feet to furlongs). **Medium:** moderate translation (e.g., translating complex data into separate data components). **High:** translations that require lookup tables and/or non-trivial computations (e.g., dead reckoning, converting a command and control message to a simulation message, etc.).
Object Creation	Object creation types are based on their creation time relative to the duration of the simulation exercise	Object creation types: **Static Creation**: creation of persistent object at the beginning of the simulation execution. **Dynamic Creation**: creation of objects during the simulation execution.

Table 2 Scenario Parameters

Parameter	Definition	Initial Standard Configuration Values for Gateway Performance Benchmarking
Number of Connected Simulation Architectures	Number of architectures connected to the gateway (both incoming and outgoing)	2 simulation architecture types
Number of Connected Simulation Components	Number of simulation components on each side of the gateway which are communicating to it	20 connected simulation components per simulation architecture
Central Processing Unit Processing Power	Number & speed of cores on the gateway machine	Two cores running 3 GHz
Computer Memory Capacity	Size/speed/configuration of Random Access Memory (RAM)	4 GBytes of RAM operating at 1000 MHz
Disk Performance	Size/speed of hard disk drives	500 GBytes Serial Advance Technology Attachment (SATA) disk drive operating at 3 Gigabit speed
Network Configuration and Speed	Communication configuration to the connected nodes	Two Network Interface Cards for full duplex, flow-controlled Ethernet, each connected through Category 5 cables to 100 Mbits/second routers

Table 3 Operation Parameters

The benchmarking parameters described above capture the significant differences in the structure and demands of distributed simulation scenarios in which gateways are employed. Table 4 lists a set of five use cases meant to capture a broad range of gateway employment scenarios is proposed using these parameters. Each use case has been given a short descriptive name for easy reference.

Scenario Parameter	Use Case 1: Small Virtual Application	Use Case 2: Small LVC Event	Use Case 3: Firefight	Use Case 4: Large Constructive Application	Use Case 5: Large LVC Event
Persistent Object Count	Low	Medium	Medium	High	High
Transient Object Count	Low	Medium	High	Medium	High
Update Rate	High	High	Medium	Low	Medium
Traffic Pattern	Continuous	Continuous	Burst Mode	Burst Mode	Burst Mode
Complexity of Data Exchange Model	Low	Medium	High	Medium	High
Complexity of Translation	Low	Medium	Medium	High	High
Object Creation	Static	Static	Dynamic	Dynamic	Static

Table 4 Use Case Summary

The Gateway Description Language (GDL)

The Gateway Description Language (GDL) provides a mechanism to document user gateway requirements and capabilities of gateways provided by the developer. The GDL is based on the Gateways Capabilities List and Gateway Performance Benchmarks. The purpose of GDL is to communicate user needs and developer capabilities to support the selection of gateways.

GDL is implemented in XML to be both human and machine-readable. GDL contains the capabilities from the Gateway Capabilities List and levels of implementation as defined in the same document. GDL specifies the level of implementation for each capability. GDL can be used two ways; to specify user requirements and to define gateway capabilities. The same format addresses the needs for both types of use.

GDL has three major components: description, capabilities, and performance. Each GDL file has exactly one description. A GDL has one or more capabilities and zero or more performances. The following paragraphs describe the purpose of each component.

The description component defines who created the GDL file and for what purpose. A user would create a GDL file to document the requirements for a specific federation. This includes the name of the user (generally an organization) and the name of the federation. Some federations have requirements for more than one gateway. If this is the case there is an option to specify the use in the federation, then a gateway developer would create a GDL file to document the capabilities of their gateway. A developer generated GDL file includes the name of the gateway and version.

The next component of GDL is the capability. A user generated GDL file would capture the required capabilities for the gateway in a particular event. A gateway developer generated GDL file would identify the implemented capabilities for a particular gateway. Each capability entry has four components: capability identifier, capability description, implementation level, and priority level.

The final component of GDL is performance. Each performance entry is associated with a use case. This component includes a use case identifier and the values of the performance metrics. The performance metrics for a user created GDL represent the user's desired performance for each parameter. In a developer-created GDL, these values are the results from executing the Performance Benchmarks..

Figure 1.0 provides an example of what GDL may look like. The final GDL will likely have additional elements, but this example provides an initial draft for discussion.

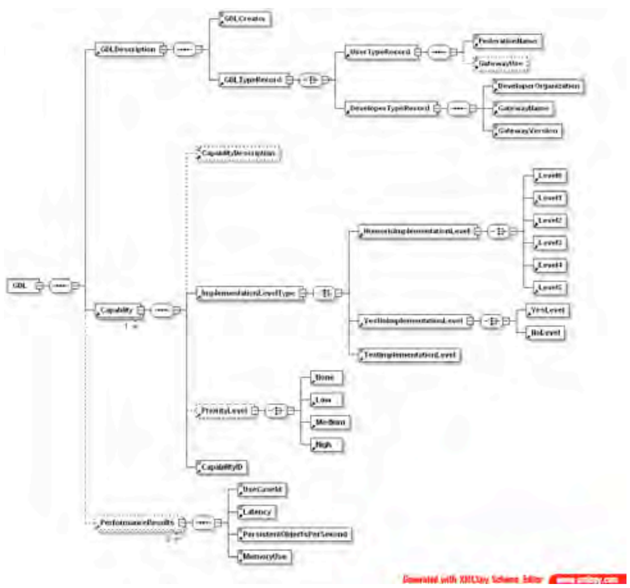


Figure 1 Draft GDL Schema

4. Gateway Selection Process

Typically, LVC simulation environments have stringent system-level performance requirements. In multi-architecture LVC simulation environments, gateways are frequently part of the overall system architecture. Therefore, an effective and sufficient process to select an appropriate gateway should be viewed as critical to support the requirements defined for the LVC environment. This section describes the process that utilizes the aforementioned technologies to enhance the gateway selection process

The event manager has two major gateway issues: 1) how to select a gateway; and 2) how to use the gateway [4].

The event manager may decide to use an existing gateway, modfiy an existing gateway, or decide to build a new gateway. Currently, the event manager has no tools to assist with the gateway selection. The only documentation available is provided by the gateway developers. Some gateways have almost no easily obtainable documentation. Even when documentation does exist, the lack of standard capability definitions makes it difficult to compare gateways. The lack of well-defined terminology makes even direct conversations with gateway developers difficult. Harder still is determining the performance of the gateway. The required performance characteristics of gateways depend on the event. Until the effort by the Common Gateways and Bridges Task team, there was no existing benchmark standards for measuring and reporting gateway performance.

The Gateway Capabilities List is an integral product in the overall effort to improve selection and understanding of available gateway products. This product provides a common foundation for documenting the ability of a gateway to fulfill a particular set of requirements. It provides a means of completing standard functional testing on gateway applications, while the performance benchmarks report establishes a standard for performance testing. Taken together, these complementary products constitute a common framework enabling evaluation of gateways against both expected behaviors and desired performance levels.

As such, the Gateway Capabilities List and the GPB are the foundation for establishing gateway selection process based on a community standard for gateway description and assessment. The Gateway Capabilities List and GPB are the basis for the more formal GDL, which provides a standard way of describing gateway functionality in a human and machine-readable format. Once described by the GDL, each gateway can be assessed against the Gateway Capabilities List based on user requirements. Without the GDL, an equitable and efficient gateway analysis would not be possible [2,3].

The GDL allows the event support team to define the gateway capabilities and implementation level required for his event. Without GDL the event manager does not have structured way to document his requirements. The well-defined language removes misunderstandings from this process by providing a common set of definitions.

Once the user requirements have been documented in GDL, the event support personnel would acquire the GDL files from a repository populated with GDL files from the gateway vendors. Because the developer GDL files are in the same format as the user requirements, it becomes more efficient to determine the best match or matches. After comparing the requirements to existing gateway implementations, the user may determine more than one gateway is required.

Not only does the event manager have to translate between architecture pairs, the event may also require some advanced gateway features such as specialized filtering. Using the GDL, the event manager is able to efficiently select gateway(s) that meet all of the events needs. It also allows for the selection of multiple gateways that collectively fulfill all of the event requirements.

Figure 2 illustrates how the Common Gateways and Bridges Task products enhance the user experience for selecting and using a gateway when requirements dictate the need for multi-architecture interoperability in an LVC simulation environment. The left-hand side of this figure shows the “to be” view of the process of gateway selection. As users define their gateway requirements, a tool will capture these requirements and generate a GDL file. This tool is postulated to employ an interview process much like modern tax software, with the resulting requirements being based on Gateway Capabilities List and GPB content.

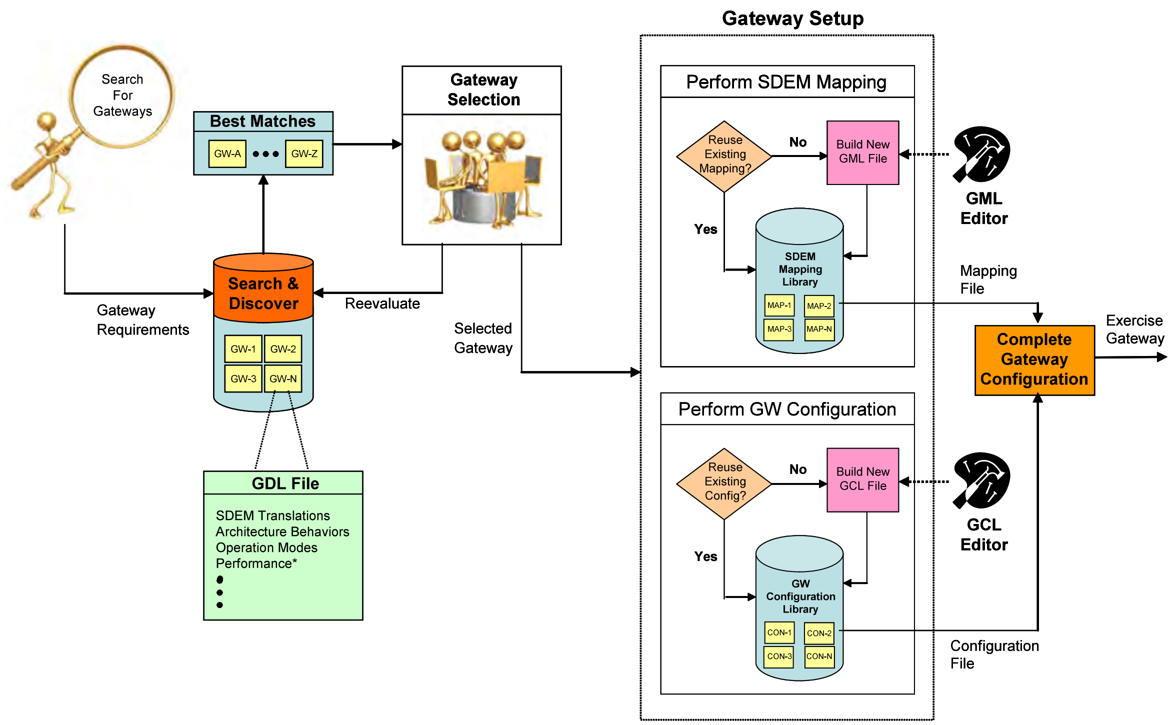
The ability to capture and document interoperability requirements is not only critical in supporting the gateway selection process, but also enhances the LVC simulation event design process. The interoperability and performance requirements for gateways are sometimes poorly documented which pushes simulation integration issues further in the event lifecycle. Oftentimes, this results in poor gateway performance that gets identified very late in the integration cycle. This creates unnecessary risk and increases the integration burdens at a time that has the least schedule and resource flexibility.

Gateway developers can use GDL to document the capabilities and performance of their gateway solutions. This would contribute significantly to the gateway community as it would allow gateway vendors to describe the various features of their gateway in a common format that is accessible to users. Having this information, and the documented event requirements in a common format would enable efficient gateway selection.

A repository or registry is envisioned which would archive GDL files from participating gateway developers, each describing the capabilities that their commercial or government-developed gateway product has to offer. An envisioned tool would then compare the user-defined gateway requirements with the capabilities offered by the various gateways, and would return “best matches.”

Users would then review the tool's mapping results of requirements to gateway capabilities and select among the best matches. Another option is for the user to reevaluate how their requirements were described, generate a new GDL file, and produce a new set of matches to choose from. The output of this part of the process is the gateway that best meets the needs of their immediate application.

The right-side of Figure 2 shows the "to be" view of gateway setup/employment. This process is presented in detail in the Spring 2011 SIW paper entitled "LVCAR Enhancements for Using Gateways" (11S-SIW-025).



* Based on *Performance Benchmarks Specification*

Figure 2 Gateway Selection, Setup, and Employment

5. Gateways – Future Efforts

The value of any tool or technology is in its use, and not in its design or development. If unused, the self-described "best tool ever developed", is of no use. It is for this reason that community involvement is critical in the success or failure of the gateway selection technologies and process. This paper is an initial effort to describe the activities and products to enhance the gateway selection process. There will be other opportunities for the community to provide feedback in the direction and implementation of these efforts. If the reader wishes to provide feedback outside these opportunities, the reader is requested to contact the author so that the feedback may be sufficiently documented.

6. Summary

The purpose of this paper has been to introduce emerging concepts for how gateway selection may be improved in the future. The tasks needed to implement these concepts are currently being worked under the LVCAR Implementation program managed by the Modeling and Simulation Coordination Office (M&S CO). Additional detail into how these concepts are being implemented by the LVCAR Implementation team can be found in related papers being presented at the Spring 2011 SIW.

7. References

[1] Live, Virtual, Constructive Architecture Roadmap (LVCAR) Final Report, Institute for Defense Analyses, September 2008.

[2] Live-Virtual-Constructive Architecture Roadmap Implementation, Common Gateways and Bridges Task – Gateways Capabilities List, JHU/APL NSAD-R-2010-100, November 2010.

[3] Live-Virtual-Constructive Architecture Roadmap Implementation, Common Gateways and Bridges

Characterization Report, JHU/APL NSAD-R-2010-031, May 2010.

[4] Live-Virtual-Constructive Architecture Roadmap Implementation, Common Gateways and Bridges Execution Plan, JHU/APL NSAD-R-2010-049, June 2010.

Author Biographies

KURT LESSMANN is a Vice President at Trideum Corporation, specializing in providing solutions and services in the area of test and evaluation, modeling and simulation, systems analysis, and information technology. Mr. Lessmann has over 15 years experience in supporting Live, Virtual, and Constructive (LVC) integration activities for large-scale Joint Distributed Test activities. Mr. Lessmann also supports the test community by providing Object Model design, software and range system design and integration, and application of M&S techniques. He currently supports the DoD Test Reource Management Center (TRMC), the Central Test and Evaluation Investment Program (CTEIP) Test and Training Enabling Architecture (TENA) project and the Joint Mission Environment Test Capability (JMETC) Program as a senior Event Lead. Mr. Lessmann has a Bachelor of Aerospace Engineering from Auburn University.

DANNIE E. CUTTS is a Senior Computer Scientist with AEgis Technologies Group Inc. supporting the USJFCOM Joint Advanced Training Technologies Laboratory (JATTL) in Suffolk, VA. He has over 20 years experience in Modeling and Simulation for NASA and the DoD and has been involved with the High Level Architecture (HLA) since 1995, serving on the Interface Specification and Time Management Working Groups. He has provided HLA Training and Cadre support for DMSO, and currently serves on the IEEE Drafting Group for the HLA IEEE 1516 standard. Mr. Cutts is a Certified Modeling and Simulation Professional (CMSP). He has supported numerous federation development efforts, as well as projects bringing legacy and new simulations to HLA Compliance. At USJFCOM he is involved in efforts to improve interoperability between Live, Virtual and Constructive assets for Joint Training.

DAVID DRAKE is a Senior Technical Staff in the Global Engagement Department at Johns Hopkins University Applied Physics Laboratory performing research and development in the area of modeling and simulation supporting the U.S. government and military. Mr. Drake is currently the lead on the Live Virtual Constructive Service-Oriented Architecture Task, and supports the LVC Architecture Roadmap Implementation task for Common Gateways and Bridges. Mr. Drake has 8 years experience in modeling and simulation design, development, and standards, and 23 years as a computer security professional in computer security design, implementation and evaluation. Mr. Drake received his Bachelors in Mathematics from SUNY at Buffalo. He is published in the areas of simulation, service-oriented architecture, grid computing, security, and risk assessment and has a patent on the process for enterprise-wide intrusion detection.

MICHAEL O'CONNOR is a Senior Principal Engineer for ITT Corporation's Information Systems division. Mr. O'Connor has more than 20 years experience in modeling and simulation. He has been a key participant in the development of distributed modeling and simulation standards including IEEE 1278 and IEEE 1516. Mr. O'Connor was the editor for the SISO Real-time Platform Reference Federation Object Model (RPR FOM). He has held many positions in the community including Chairman of the SISO Standards Activities Committee and Vice-Chairman of the SISO Executive Committee. Mr. O'Connor oversees the development of ITT's Chemical, Biological, Radiological, and Nuclear Simulation Suite, which supports DIS, HLA, and TENA. He has a Bachelors of Computer Engineering from Auburn University and a Master's of Science in Computer Science from the University of Alabama in Huntsville.