SISO-STD-014.1-2018

Standard for
Gateway Description Language

12 September 2018

Version 1.0

SAC Approved: 09/28/2018 **EXCOM Approved: 10/11/2018**

Prepared by:
Gateway Description and Configuration Languages (GDACL)
Product Development Group

P.O. Box 781238
Orlando, FL 32878-1238, USA

SISO Inc. Board of Directors
P.O. Box 781238
Orlando, FL 32878-1238, USA

Revision History

Version	Section	Date (MM/DD/YYYY)	Description
1.0	All	09/12/2018	Initial Version

Participants

At the time this product was submitted to the Standards Activity Committee (SAC) for approval, the Gateway Description and Configuration Languages Product Development Group (PDG) had the following membership and was assigned the following SAC Technical Area Director:

Product Development Group

Bob Lutz (Chair)
Dannie Cutts (Vice-Chair)
Kurt Lessmann (Vice-Chair)
David Drake (Editor)

Patrice Le Leydour (SAC Technical Area Director)

Jim Coolahan
Dannie Cutts
Lance Marrou
Angus (Thom) McLean
Michael O'Connor
Felix Rodriguez

The following individuals comprised the ballot group for this product.

Ballot Group

Jeffery Bergenthal
Curtis Blais
Dannie Cutts
David Drake
Michael Eady
Paul Gustavson
Scott Johnston
Michael Longworth
Robert Lutz
Lance Marrou
James McCall
Bjorn Moller
Roy Scrudder
Marcy Stutzman
Eugene Stoudermire
Tom van den Berg

When the Standards Activity Committee approved this product on 28 September 2018, it had the following membership:

Standards Activity Committee

Katherine L. Morse (Chair)
Jean-Louis Igarza (Vice Chair)
Lance Marrou (Secretary)

Grant Bailey
Brad Dillman
Kevin Gupton
Paul Gustavson (CC Chair)
Michael Heaphy
Aerial Kreiner
Patrice Le Leydour
Chris McGroarty
Thom McLean
John Stevens

1

When the Executive Committee approved this product on 11 October 2018, it had the following membership:

Executive Committee

Michael O'Connor (Chair)
Robert Lutz (Vice Chair)
Jeff Abbott (Secretary)

Paul Gustavson	Lana McGlynn	Randy Saunders (CC Chair)
Kenneth Konwin	Katherine Morse (SAC Chair)	Roy Scrudder
Chris Metevier	Stefan Sandberg	Robert Siegfried

Introduction

This document defines a standard for describing the capabilities that gateways can potentially offer to user organizations. The Gateway Description Language (GDL) defines a machine-readable Extensible Markup Language (XML) schema that facilitates automated methods of mapping gateway user requirements to gateway products that can meet those requirements. GDL was originally developed in response to U.S. Department of Defense needs expressed in the Live-Virtual-Constructive Architecture Roadmap (LVCAR) effort. However, the GDL specification provided in this document provides a common language to address the broader, more general need in the international Modeling and Simulation (M&S) community to better link producers and consumers in the gateway marketplace.

The data files associated with this SISO Standard may be downloaded from this URL: http://www.sisostds.org/Schemas.aspx.

Table of Contents

Standard for Gateway Description Language

1. Overview

Modeling and Simulation (M&S) is in wide use today within the international user community. While many M&S applications are standalone applications used in support of such areas as requirements development, concept exploration, and system design/development, the advent and increased maturity of technologies and standards for *distributed simulation* have led to much wider use of integrated simulation environments in support of major Test and Evaluation (T&E) and training events. While most distributed simulation events apply a single underlying simulation architecture, such as Distributed Interactive Simulation (DIS), High Level Architecture (HLA), or Test and Training Enabling Architecture (TENA), sponsor requirements sometimes necessitate the selection of simulations whose external interfaces support different simulation architectures. This is what is known as a *multi-architecture simulation environment*. When more than one simulation architecture must be used in the same environment, interoperability problems are compounded by the architectural differences [1, 7, 8]. 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.

A *gateway* is an intelligent translator designed to link simulation enclaves that use dissimilar architectures. A *bridge* is very similar to a gateway but links enclaves using the same underlying architecture such as two HLA federations. Gateways and bridges are ubiquitous in the simulation community today and continue to represent one of the most widely used means of addressing interoperability concerns in multi-architecture simulation environments. Throughout this standard, when the word “gateway” is used, it is understood to mean both gateways and bridges.

Just as there are multiple types of distributed simulation architectures, there are multiple types of gateways. Some gateways are general purpose and can be configured or programmatically extended to interface with not only different distributed simulation architectures, but also dissimilar metamodels. At the other extreme, some gateways are very specific in their capabilities with limitations on the type of distributed simulation and metamodels with which they can interface. The other dimension of gateways is their capability to translate and filter metamodel communications [2, 3, 4, 5, 6]. This situation creates a need for a standard way to describe the capabilities of a gateway in a consistent manner.

The Gateway Description Language (GDL) provides a common language for capturing both gateway user requirements and the capabilities that individual gateways can offer to users. Thus, GDL supports direct mappings between gateway requirements and capabilities, facilitating more informed user gateway selections.

In addition to the translation capabilities defined by GDL, GDL also includes the specification of gateway performance requirements and capabilities. This is in recognition that some users have low-latency requirements for their simulation environment, and thus, performance is a critical factor for gateway selection.

1.1. Scope

This specification defines the requirements for the GDL schema along with a detailed description of the formal language that addresses these requirements. The description includes graphical representations of GDL schema elements to illustrate the types of description capabilities supported. A full Extensible Markup Language (XML) schema for the GDL and an example are provided as annexes. This GDL applies to both gateways and bridges.

1.2. Purpose

The purpose of this Gateway Description Language specification is to specify a formal XML schema for identifying the capabilities that gateways can offer to users. GDL provides a common format and syntax for gateway developers to describe the capabilities of their products to potential users. Since GDL is machine-readable, users can search on these GDL descriptions to align their application requirements with the gateways that best suit their needs. The GDL, in conjunction with tools and repository capabilities that support GDL, streamlines the process of gateway selection in future exercises.

1.3. Objectives

The objective of GDL is to define a common mechanism for describing the capabilities of a gateway as well as the set of gateway requirements.

1.4. Intended Audience

The intended audience for this document includes:

- Exercise engineers who are not gateway experts can use this document to understand the kinds of capabilities a gateway may have and the scope of requirements that a user may request regarding the functional aspects of a gateway.
- Gateway developers can use this document to record the capabilities of a gateway in a machine- and human-readable form and to describe each capability to other distributed simulation professionals.
- Resource developers can use this document to create tools to facilitate gateway selection, integration, and usage in support of simulation event planning, preparation, and execution.

2. 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 Description, JHU/APL NSAD-R-2010-100, November 2010.

[3] Live-Virtual-Constructive Architecture Roadmap Implementation, Common Gateways and Bridges Task – Gateway Performance Benchmarks, JHU/APL NSAD-R-2011-016, January 2011.

[4] Live-Virtual-Constructive Architecture Roadmap Implementation, Common Gateways and Bridges – Gateways Configuration Model Report, JHU/APL NSAD-R-2011-034, April 2011.

[5] Live-Virtual-Constructive Architecture Roadmap Implementation, Common Gateways and Bridges Characterization Report, JHU/APL NSAD-R-2010-031, May 2010.

[6] Live-Virtual-Constructive Architecture Roadmap Implementation, SDEM Mapping Language (SML) Specification, JHU/APL NSAD-R-2011-220, August 2011.

[7] “IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP),” IEEE Standard 1730-2010, 24 January 2011.

[8] “IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO),” IEEE Standard 1730.1-2013, 23 August 2013.

3. Definitions, Acronyms, and Abbreviations

3.1. Definitions

bridge: A type of gateway where only a single simulation architecture is used. A bridge may translate between different SDEMs. Compare to *gateway*.

filtering: This is the elimination of simulation data transmitted from one Architecture/SDEM to another Architecture/SDEM, based on a criterion or criteria. Filtering is intended to reduce unneeded or unwanted data communication between architectures.

gateway: An application for translating objects from one simulation architecture/SDEM pair to one or more simulation architecture/SDEM pair(s). Compare to *bridge*.

multi-architecture simulation environment: A simulation environment where more than one simulation architecture is used.

Simulation Data Exchange Model (SDEM): A specification defining the information exchanged at runtime to achieve a given set of simulation objectives. This includes class relationships, data structures, parameters, and other relevant information.

3.2. Acronyms and Abbreviations

CRS	Coordinate Reference System
DIS	Distributed Interactive Simulation
DoD	Department of Defense
GCD	Gateway Characterization Description (presented in reference [5])
GFL	Gateway Filtering Language
GML	Geography Markup Language
HLA	High Level Architecture
ID	Identification
JHU/APL	The Johns Hopkins University Applied Physics Laboratory
LVC	Live-Virtual-Constructive
M&S	Modeling and Simulation
SDEM	Simulation Data Exchange Model
T&E	Test and Evaluation
TENA	Test and Training Enabling Architecture
XML	Extensible Markup Language

4. GDL Definition

The requirements for GDL are as follows:

1. A formal mechanism shall exist for efficient means to compare application requirements to gateway capabilities.
2. A formal mechanism shall exist for an efficient, repeatable means to determine what gateways best meet the application requirements.

The objective of GDL is to define a common format, syntax, and content (i.e., language) for describing both gateway requirements and gateway capabilities. GDL addresses the first requirement (above) by facilitating a direct comparison of requirements to capabilities, removing the need for users to interpret (or guess) when the requirements and capability descriptions are not well aligned. Further efficiencies are achieved through the fact that GDL is machine-readable, allowing tools to be integrated into the gateway selection process for the first time.

As users execute the gateway selection process, several gateways may be discovered that meet all or most of the user requirements. The core differences among the candidate gateways may not be particularly obvious, and thus, user support is still needed to determine the best choice among multiple competing gateways. GDL addresses the second requirement by defining a standard description of all potential gateway translation capabilities (as defined in the Gateway Characterization Description [GCD] Report [5]). Therefore, for the first time, gateway developers have a formal language (and supporting tools) for characterizing (and advertising) the capabilities of their products. As users access these GDL descriptions, side-by-side comparisons are now possible due to the common underlying capabilities template provided by GDL. While users can do these comparisons themselves, the machine-readable nature of GDL also allows for the development of automated gateway sorting/ranking algorithms to assist the user in identifying the gateways that best align with their defined requirements. These algorithms can be designed to be repeatable (i.e., no random factors) so that, if desired, the factors and calculations that drive the matching process can be reconstructed in future applications. Significant efficiencies and improvements to the quality of gateway selections are accordingly introduced by the proper application of GDL and supporting tools and repositories.

GDL is implemented as an XML schema that allows users or developers to document their requirements or capabilities in a machine- and human-readable form. GDL is supported by two other schemas, *CapabilitiesDefinition* and *UseCases*. XML instantiations of these two schemas are required to create GDL documents.

The *CapabilitiesDefinition* schema allows for the machine-readable documentation of the GCD. This is the standard set of capabilities that a gateway may implement. The *UseCases* schema supports the documentation of use cases to support Gateways Performance Benchmarking. The GDL schema is composed of three sections: *GDLDescription*, *Capabilities*, and *PerformanceResults*.

A GDL file has one *GDLDescription* section, one or more *Capabilities* section, and zero or more *PerformanceResults* sections.

4.1. GDLDescription Section

The *GDLDescription* section provides information on the creator of the GDL file and the purpose for which it was created. There are two types of creators: users and developers. A user is someone that needs a gateway to meet the needs of their simulation environments. The user can specify the name of the simulation environment and the purpose of the gateway. A developer may specify the name of the organization that controls the gateway, the gateway's name, and version number.

The *GDLDescription* section also supports references. Each GDL depends on instantiations of the *CapabilitiesDefinition* schema and the *UseCases* schema. The references section allows the gateway user or developer to specify the names of the instantiations upon which they are basing their GDL file. The creator of the GDL file may also provide a link to an external reference for the GDL file.

4.2. Capabilities Section

The *Capabilities* section allows the users to specify their requirements or developers to document their capabilities. There is one entry for each capability either required by the user or provided by the developer. To support machine processing, an XML schema is used to document the capabilities that have been developed. This will be discussed in detail below. Each capability in the file is identified by a *CapabilityID*. This identifier is found in the XML file that instantiates the *CapabilitiesDefinition* schema. Each capability has one of the following values: an implementation level that is a numeric (0-5), yes/no, or text. The implementation level entered by gateway users specifies their requirement for the level that is numeric (0-5), yes/no, or text. The implementation level entered by gateway developers specifies the level at which their gateway implements the capability. Users may also specify a priority level that is an enumerated value of either "Low," "Medium", or "High" indicating how important this capability is to them. This is used by tools that determine the best fit of user requirements to developer-provided capabilities. The final field is a

comment. Both users and developers may exercise this field to provide additional information on their requirements or capabilities.

4.3. PerformanceResults Section

The *PerformanceResults* section provides information on the performance metrics for each use case the user is interested in, or for which the developer produced data. Each *PerformanceResults* has an identifier for the use case that is found in the XML file instantiation of the *UseCases* schema. The next two fields are the metrics based on the gateway's performance in the context of the specified use case. The user enters the desired performance, and the developer enters the measured performance. Gateway users may also assign a priority to the metric. Gateway users or developers may add a comment to the metric.

4.4. CapabilitiesDefinition Schema

The *CapabilitiesDefinition* schema is a machine-readable version of the GCD document. It contains all of the information provided in the document. The schema provides a field to specify the version of the GCD document used for GDL file creation. The schema allows for the specification of the primary and specific categories to which the capability belongs. The schema provides fields for the definition, example, and implementation levels. The implementation levels may be numeric (0-5), yes/no, or text. If the capability is numeric, the definition for each level specified is provided. Some numeric levels do not have definitions. For text levels, the type of information required may be listed.

4.5. UseCases Schema

The *UseCases* schema provides a machine-readable version of the use cases specified in the Gateways Performance Benchmarks document. Each use case has a unique identifier and name. The use case also has a description. The actual definition of the use case is in the *ScenarioParameters* section of the schema. The parameters are *Persistent Object Count*, *Transient Object Count*, *Update Rate*, *Traffic Pattern*, *Complexity of Translation*, and *Object Creation*. Each of these is an enumeration. The schema defines the enumeration values for each parameter. An XML instantiation of the *UseCases* schema is needed to match the parameters specified in the Gateways Performance Benchmarks document. However, users desiring developers to perform custom benchmark testing may use the *UseCases* schema to document their needs.

Annex A Gateway Description Language Schema (Normative)

The following files constitute the XML schema for Gateway Description Language (GDL). Readers should consult Section 4 (GDL Definition) for descriptions of all GDL components.

XML Schema	File name
GDL	*GDL.xsd*
CapabilitiesDefinition	*CapabilitiesDefinition.xsd*
ImplementationLevels	*ImplementationLevels.xsd*
UseCases	*UseCases.xsd*

These files are normative parts of the specification. The files associated with Annexes A, B, and C are stored in a Zip archive that can be downloaded from this URL: http://www.sisostds.org/Schemas.aspx.

Annex B Gateway Description Language Instance Documents (Informative)

In addition to the schemas, the following instance documents for use cases and capabilities were developed based on community input.

Instance document	File name
CapabilitiesDefinition	*GDL_CapabilitiesDefinition_instance.xml*
UseCases	*GDL_UseCases_instance.xml*

Those instance documents are informative parts of the specification. They are critical to the creation of compatible Gateway Description Language (GDL) files. The files associated with Annexes A, B, and C are stored in a Zip archive that can be downloaded from this URL: http://www.sisostds.org/Schemas.aspx.

Annex C Example of Gateway Description Language Document (Informative)

The *GDL_example.xml* file is a valid Gateway Description Language (GDL) XML document utilizing the schemas from Annex A and the instance documents from Annex B.

It is an informative part of the specification. The files associated with Annexes A, B, and C are stored in a Zip archive that can be downloaded from this URL: http://www.sisostds.org/Schemas.aspx.

Annex D Bibliography (Informative)

The following document was used for generating this standard.

[D1] The Open Geographic Information System Filter Encoding Standard (http://www.opengeospatial.org/standards/filter).