The Johns Hopkins University
APPLIED PHYSICS LABORATORY

Enclosure (1) to: NSAD-L-2010-161
JHS01

**NSAD-R-2010-052**

# Live-Virtual-Constructive Architecture Roadmap Implementation, Common Capabilities—Federation Agreements Template Users' Guide

# Technical Report

**June 2010**

The Johns Hopkins University
APPLIED PHYSICS LABORATORY

NSAD-R-2010-052

# Live-Virtual-Constructive Architecture Roadmap Implementation, Common Capabilities—Federation Agreements Template Users' Guide

## Technical Report

June 2010

**FOR:**

**Joint Training Integration and Evaluation Center**
**12000 Research Parkway, Suite 300**
**Orlando, FL 32826**

**BY:**

**Johns Hopkins University Applied Physics Laboratory**
**11100 Johns Hopkins Road**
**Laurel, MD 20723**

**This page intentionally left blank.**

# JHU/APL Team

| Title | Name | Organization |
|---|---|---|
| Team Lead | Dr. Katherine L. Morse | JHU/APL |
| Program Manager | Dr. James E. Coolahan | JHU/APL |
| Team SME | Ryan Brunton | JHU/APL |
| Team SME | Jonathan Labin | JHU/APL |
| Team SME | Bill Riggs | JHU/APL |
| Team SME | Dannie Cutts | AEgis Technologies |

**This page intentionally left blank.**

# TABLE OF CONTENTS

EXECUTIVE SUMMARY ..... ES-1

1. INTRODUCTION ..... 1
   1.1 REUSE Of EXISTING SCHEMAS ..... 2
   1.2 DESIGN PATTERNS ..... 3
   1.3 EXTRA-AGREEMENT FEATURES ..... 4
      1.3.1 Federation vs. Event-Specific Agreements ..... 4
      1.3.2 Individual Agreement Status Indicators ..... 4
   1.4 DOCUMENT ORGANIZATION ..... 4

2. FEDERATION AGREEMENT DESCRIPTIONS ..... 5

APPENDIX A: REFERENCES ..... A-1

APPENDIX B: MAPPING TO THE DSEEP ..... B-1

APPENDIX C: ABBREVIATIONS AND ACRONYMS ..... C-1

## LIST OF FIGURES

Figure 1: Major Categories Of Federation Agreements ..... 5

Figure 2: Member Application Responsibilities Extract From Sample Federation Agreements Document ..... 12

Figure 3: Publication And Subscription Responsibilities Extract From Sample Federation Agreements Document ..... 13

## LIST OF TABLES

Table 1: Existing Schema Reuse..... 3

Table 2: Agreements Guidance..... 6

# EXECUTIVE SUMMARY

Federation agreements are critical to the successful design, execution, and reuse of federation assets. However, inconsistent formats and use across federations have made it difficult to capture and compare agreements between federations. This lack of a consistent approach to documenting federation agreements makes reuse and understanding more difficult. Lack of consistent format also prevents tool development and automation.

The federation agreements template is intended to provide a standardized format for recording federation agreements to increase their usability and reuse. The template is an eXtensible Markup Language (XML) schema from which compliant XML-based federation agreement documents can be created. XML was chosen for encoding agreements documents because it is both human and machine-readable and has wide tool support. Creating the template as an XML schema allows XML-enabled tools to both validate conformant documents, and edit and exchange agreements documents without introducing incompatibilities.

Wherever possible, the team that prepared the template leveraged existing, authoritative schemas for the representation of elements in this schema including:

- Modeling and Simulation (M&S) Community of Interest—Discovery Metadata Specification (MSC-DMS)
- XML Linking Language (XLink)
- XML Metadata Interchange (XMI)
- Common Platform Enumeration (CPE)
- Intelligence Community Information Security Marking (IC-ISM)
- eXtensible Configuration Checklist Description Format (XCCDF)
- Geography Markup Language (GML)

The federation agreements are decomposed into eight categories:

1. Metadata—Information about the federation agreements document itself.
2. Design—Agreements about the basic purpose and design of the federation.
3. Execution—Technical and process agreements affecting execution of the federation.
4. Management—Systems/software engineering and project management agreements.
5. Data—Agreements about structure, values, and semantics of data to be exchanged during federation execution.
6. Infrastructure—Technical agreements about hardware, software, network protocols, and processes for implementing the infrastructure to support federation execution.

7. Modeling—Agreements to be implemented in the member applications that semantically affect the current execution of the federation.
8. Variances—Exceptions to the federation agreements deemed necessary during integration and testing.

This document also provides brief examples of mapping an existing federation agreements document to a document conformant with the XML schema.

# 1. INTRODUCTION

Federation agreements are critical to the successful design, execution, and reuse of federation assets. However, inconsistent formats and use across federations have made it difficult to capture and compare agreements between federations. This lack of a consistent approach to documenting federation agreements makes reuse and understanding more difficult. Lack of consistent format also prevents tool development and automation.

The federation agreements template is intended to provide a standardized format for recording federation agreements to increase their usability and reuse. The template is an eXtensible Markup Language (XML) schema from which compliant XML-based federation agreement documents can be created. XML was chosen for encoding agreements documents because it is both human and machine-readable and has wide tool support. Creating the template as an XML schema allows XML-enabled tools to both validate conformant documents, and edit and exchange agreements documents without introducing incompatibilities.

The individual agreements were identified via an extensive search of existing federation agreement documents, federation agreement templates, and literature identifying shortcomings of previous documents and templates. The resulting initial list of agreements was augmented with the development team's own federation engineering experience. The consolidated list was presented and reviewed at a workshop at The Johns Hopkins University Applied Physics Laboratory (JHU/APL) in November 2009 and later in a public workshop at the Spring Simulation Interoperability Workshop in Orlando, FL, in April 2010. Inputs and refinements from those two workshops were folded into the list of agreements.

The agreements were first decomposed into general categories. These groupings were selected based on a relationship to each other that included consideration of chronology in the federation development process, [e.g., the Federation Development and Execution Process (FEDEP) [Reference (a)][1] and the draft Distributed Simulation Engineering and Execution Process (DSEEP) [Reference (b)], and a set of use cases based on potential template users. Within each category, many individual agreements were drawn from a review of existing federation agreements and templates, the experiences of the development team, and a literature search. The final level of decomposition down to individual elements identifies specific data types as defined in the schema.

Because federations come in all sizes and levels of complexity, the template was designed for maximum flexibility. For example, early in the template's development, the development team considered and discarded the concept of mandatory and optional agreements after concluding that such decisions are federation specific and should be within the purview of federation managers. In the future, guidance on recommended sets of agreements based on federation type or use might be a useful addendum to the template itself.

[1] References may be found in Appendix A.

In preparation for the November 2009 workshop, the development team developed a list of federation roles relative to the agreements (i.e., roles that would require writing, reading, or using the federation agreements in any way throughout the federation life cycle). Workshop participants were divided into small teams representing the identified roles and instructed to review the agreements for completeness from the perspective of the team's role. This step was critical to ensuring that the resulting template would be an effective tool for all of its intended users.

The detail and structure of the template should aid all federation participants in effectively performing their roles by:

- Ensuring that federation and federate engineers accurately record all necessary agreements by providing a complete list of all potential agreements;
- Ensuring that agreements are recorded in a standardized, unambiguous manner;
- Enabling federation participants to locate agreements rapidly or determine that a required agreement has not been recorded;
- Enforcing internal consistency between related agreements; and
- Enabling the development of a new generation of federation engineering tools driven by the XML schema that reduce the cost, schedule, and risk of developing federations.

## 1.1 REUSE OF EXISTING SCHEMAS

Wherever possible, the development team leveraged existing, authoritative schemas for the representation of elements in this schema as detailed in Table 1.

**Table 1: Existing Schema Reuse**

| Existing Schema | Use |
|---|---|
| Modeling and Simulation (M&S) Community of Interest—Discovery Metadata Specification (MSC-DMS) [Reference (c)] | • Point of contact<br>• Remote document metadata (author, versioning, intellectual property rights, title, etc.)<br>• Notes<br>• Usage information<br>• Address<br>• Date |
| XML Linking Language (XLink) [Reference (d)] | • Hyperlinks |
| XML Metadata Interchange (XMI) [Reference (e)] | • Unified Modeling Language (UML) diagrams |
| Common Platform Enumeration (CPE) [Reference (f)] | • Definition of specific software platforms (e.g., operating system with major and minor version, tools, etc.) |
| Intelligence Community Information Security Marking (IC-ISM) [Reference (g)] | • Document security attributes, including classification and releasability information |
| eXtensible Configuration Checklist Description Format (XCCDF) [Reference (h)] | • Definition of information security procedures |
| Geography Markup Language (GML) [Reference (i)] | • All geography or rigorous timing requirements, including bounding boxes, coordinates, and timing constraints |

## 1.2 DESIGN PATTERNS

During decomposition of the categories, some recurring design patterns emerged:

- Because a federation may be large and complex, it may have entire documents for individual agreements. For example, if the federation is fairly small, the conceptual model might be captured in a few elements, including entities, event traces/sequence diagrams, and state transition diagrams/entity behaviors that leverage the first two elements. Adhering to the goal of maximum flexibility, the development team allowed for either inline capture of the conceptual model or a detailed reference to an external document. This design pattern appears in a majority of agreements.
- Where elements described processes or sequences within the federation, the team allowed for their representation in Unified Modeling Language (UML) that may be rendered in the schema as XMI and/or a graphic.
- Tools, member applications, hardware, and physical locations are cross-referenced in several agreements but are listed and described only once.

## 1.3 EXTRA-AGREEMENT FEATURES

There are a couple of features of the template that are not specifically agreements but that apply to all the agreements.

### 1.3.1 Federation vs. Event-Specific Agreements

Federation agreements may be reused between events with only minor changes. Individual agreements may be marked as either federation (persistent) or event specific.

### 1.3.2 Individual Agreement Status Indicators

Federation agreements are a work in progress throughout federation development and execution. Individual agreements may be in very different stages of completion, which may not be obvious just from looking at the agreements themselves. Therefore, individual agreements each have a status indicator that may have one of the following values:

- To Be Determined (TBD)
- Negotiated
- Designed
- Implemented
- Tested
- Validated

Normally, an individual agreement would progress sequentially through these states as the federation development progresses, although that development effort may cause rework, resulting in a regression in the state of some agreements.

## 1.4 DOCUMENT ORGANIZATION

Section 2 provides a breakdown and description of the categories and agreements, including guidance for completing them using the template schema. It also provides extracts from a sample federation agreements document using the schema. References are in Appendix A. Appendix B provides a mapping between the agreements and the steps of the DSEEP. This appendix should be particularly useful to new federation developers because it identifies the DSEEP step where each agreement should be addressed to support successful completion of subsequent DSEEP steps. Abbreviations and acronyms are in Appendix C.

## 2. FEDERATION AGREEMENT DESCRIPTIONS

This section provides an overview of the structure and basic guidance on the use of the top-level elements in the federation agreements template XML schema. Specific technical guidance on the syntax and semantics required to create a conformant XML-based federation agreements document can be found in the federation agreements schema and the associated Hypertext Markup Language (HTML) Programmer's Reference [Reference (j)]. Structurally, the federation agreements template schema consists of eight major categories of federation agreements as illustrated in Figure 1 and described in more detail in Table 2.

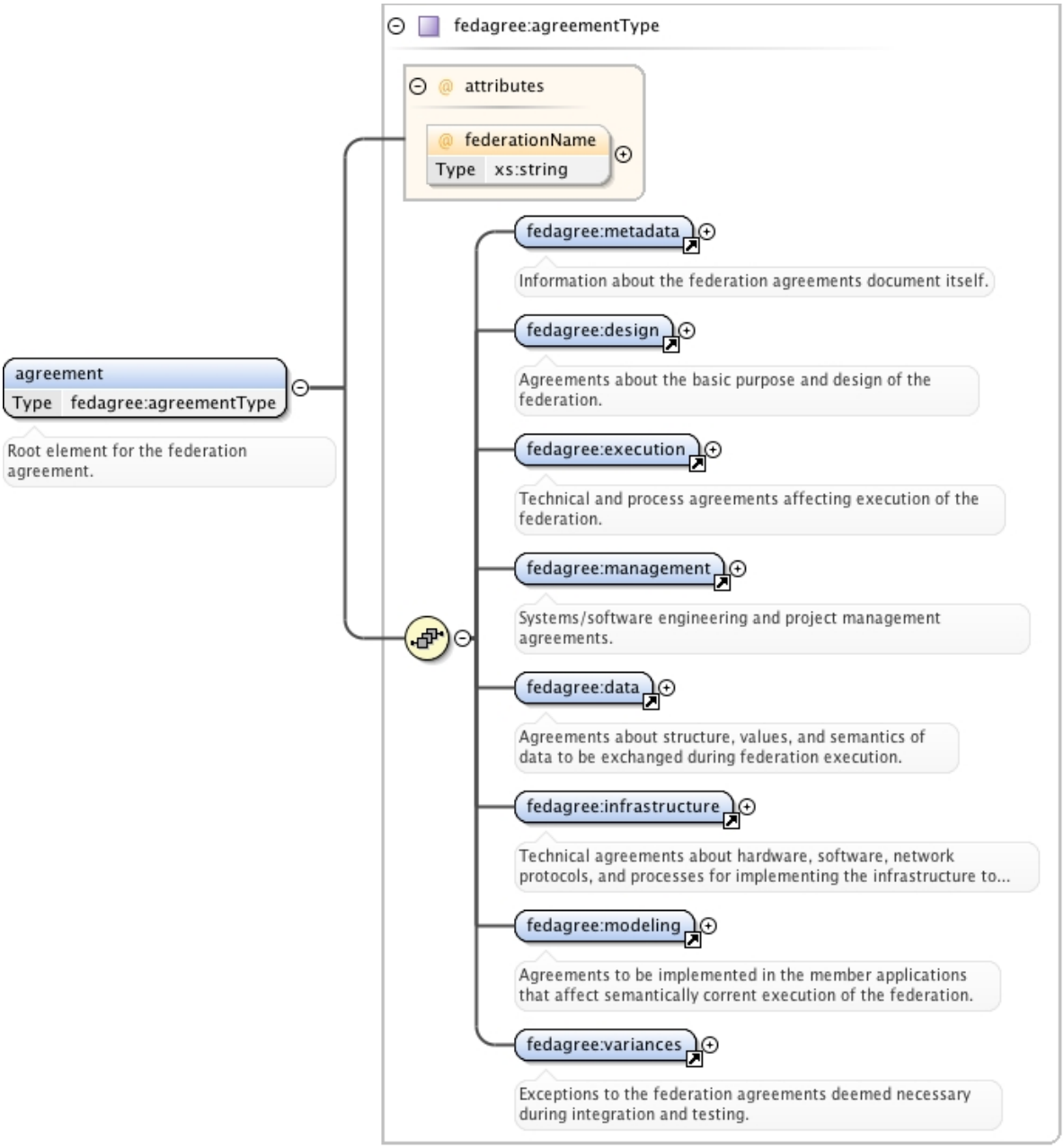


**Figure 1: Major Categories of Federation Agreements**

Table 2 provides an extract from the schema of the recommended guidance for each of the categories in Figure 1, as well as their children agreements. These guidance statements should help map federation agreements concepts in existing federation agreement documents into the structures required by the schema.

**Table 2: Agreements Guidance**

<table>
<tr><td>Metadata</td><td>Information about the federation agreements document itself.
<table>
<tr><th colspan="2">Agreements</th></tr>
<tr><td>Identification</td><td>Identifies this federation agreements document. Version number and name if applicable. Makes extensive use of the MSC-DMS metadata elements, making this the most important element relative to registering the federation agreements document in repositories based upon MSC-DMS markup.</td></tr>
<tr><td>Federation Executions</td><td>Defines the federation execution(s) to which the document applies.</td></tr>
<tr><td>Referenced Docs</td><td>Location and version of documents referenced in other agreements.</td></tr>
<tr><td>Previous Agreements</td><td>Prior agreements used for reference during simulation design and development. These may be reused or modified as necessary.</td></tr>
<tr><td>Glossary</td><td>Terminology, acronyms, and definitions.</td></tr>
</table></td></tr>
<tr><td>Design</td><td>Agreements about the basic purpose and design of the federation.
<table>
<tr><th colspan="2">Agreements</th></tr>
<tr><td>Conceptual Model</td><td>Element for inclusion of a conceptual model that includes entities within the domain of interest, static and dynamic relationships among entities, and entity behaviors. Can include 0 or more references to remote documents and 0 or more inline XMI-encoded models. Federations with detailed conceptual models can choose to include such models by reference using document number, title, version, and reference.</td></tr>
</table></td></tr>
</table>

| | | |
|---|---|---|
| | INFOSEC | Information security and information assurance procedures and policies applicable to the simulation environment; level at which the federation will be executed; compliance with Defense Information Technology Standards and Profile Registry (DISR) standards. Intelligence Community Information Security Marking (IC-ISM) attributes are used for inline reuse restriction information. Links to external documents containing the Information Security (INFOSEC) information and procedures may be substituted for or supplement any or all elements and attributes. For federations with detailed security requirements, a reference to a separate security plan or policy document is appropriate here. Alternatively, unclassified federations may choose to document applicable information technology security standards and processes here. |
| | Terrain | Playbox, format, resolution, correlation/transposition. GML is used to encode geographical bounding boxes and conversion and transformation options. |
| | Objectives and Requirements | Measurable objectives for the federation. Requirements, constraints, preferences, assumptions, and limitations. |
| | Environment | Synthetic Natural Environment (SNE) and weather descriptions, including supporting initialization databases, as well as algorithms and temporal database updates during simulation environment execution. For federations with detailed environment representation requirements, a reference to a separate document is appropriate here. |
| | Scenario | Scenario description, including initial laydown. Reference to scenario files. Scenario file format. |
| | Architecture | Describes the top-level architecture of the federation. For federations with detailed federation architectures, a reference to a separate document is appropriate here. Simpler federations may just insert a graphic or UML diagram of the architecture. |
| | Member Apps | Type, name, version, description, responsibilities of each member application. For federations with detailed federation architectures, a reference to a separate document is appropriate here, possibly the same document as the federation architecture. Simpler federations may just identify the member applications here. |
| | Tools | Tools used in the design and support of the simulation. This section should include any relevant operating systems, database engines, etc., explicitly required by the federation in addition to simulation-specific tools. |

| | | |
|---|---|---|
| | Domain Info | Information traceable to authoritative sources that is used to build supporting databases and algorithms for member applications and the simulation environment. |
| Execution | Technical and process agreements affecting execution of the federation. | |

| Agreements | |
|---|---|
| Initialization | Preconditions and procedures, including ordering, for initializing the federation. |
| Shutdown | Triggers and processes for orderly shutdown of the federation. |
| Save | Trigger conditions for save. Elements to be saved. Process for initializing save. |
| Restore | Trigger conditions for restore; elements to be restored. Process for initializing restore. |
| Synchronization | Synchronization triggers and procedures. Predetermined "points" or situations in which specific actions are to be taken by (some) members of the federation (e.g., pause, resume, take a break). |
| Time Management | How simulation time will be advanced. Particularly important if any part of the event is time. (Note: Simulation time is not necessarily scaled to wall clock time.) This is related to event ordering and repeatability. |
| Join and Resign | Preconditions and procedures, including ordering, for joining and resigning the federation under normal conditions. |
| Update Rates | Agreements on how often member applications agree to update states. May be static or dynamic. Might give upper/lower limits or set rates. Rates may be set for the entire federation, federates, object classes, or individual objects. |
| Performance Thresholds | Agreements on thresholds and how federates will respond. |
| Heartbeat | Agreements on whether and how often member applications will heartbeat. Also agree on response to non-heartbeating entities. Heartbeats can be decided for the federation, individual member applications, "object classes," or object instances. |
| Data Logging | Responsibilities for logging data, including uploading and configuration management. |
| Data Replay | Responsibilities and requirements for replaying data during an execution, including configuration management. |
| Monitoring | Tools and procedures for monitoring federation execution. |

| | | |
|---|---|---|
| | Exceptions | Procedures for recovering from crashes, faults, and exceptions. |
| | Middleware Agreements | Prescriptions about which middleware services may or may not be used, and under what circumstances. |
| Management | Systems/software engineering and project management agreements. | |
| | **Agreements** | |
| | Software Configuration Management (CM) | Tool identification and references to plan(s). Includes methodology and guidelines for establishing and managing CM baselines. Identification of standard software practices to support life-cycle development and reuse. Federations that use formal software management processes and procedures may reference these artifacts using document citation and/or reference. Alternatively, the federation agreements document may outline applicable processes and procedures in the body of the test with supporting graphics. |
| | Verification, Validation, and Accreditation (VV&A) | VV&A tool identification and references to plan(s). Description of verification and validation activities to support intended use of M&S in an event and formal accreditation where applicable. |
| | Schedules | Reference to programs schedule(s). Provides information about scheduled activities as part of a detailed event plan per DSEEP or applicable process/overlay. |
| | Test Plan | Test tool identification and references to plan(s). Test requirements and criteria. Planned activities to perform testing of M&S at increasingly higher levels of integration prior to Live-Virtual-Constructive (LVC) event execution. Test planning processes and artifacts may address M&S verification requirements to be addressed prior to event execution. Explicit traceability will be addressed as part of these agreements. |
| | Compliance | Agreement with respect to formal certification of compliance with simulation environment architecture specification and rules. Documentation of agreements pertaining to architecture compliance and artifacts demonstrating that compliance has been achieved. |
| Data | Agreements about structure, values, and semantics of data to be exchanged during federation execution. | |
| | **Agreements** | |
| | Encodings | Data encodings used in the federation, including acceptable and unacceptable representation of "no current value" by data type and endianness. |

| | | |
|---|---|---|
| | Data Exchange Models | Federation Object Model (FOM), Logical Range Object Model (LROM), and Protocol Data Unit (PDU) utilization and tailoring and direct interactions (e.g., remote methods). |
| | Naming Conventions | Naming conventions for objects, attributes/parameters, enumerations, and member applications. |
| | Publication and Subscription Responsibilities | Allocation of responsibility for data distribution among member applications, subject to the simulation environment architecture. |
| | Data Dictionary | Identification of standard data dictionary references used in the Simulation Data Exchange Model and associated Simulation Environment Agreements. |
| | Supporting Databases | Common and consistent data used across the simulation environment to guarantee interactions among all member applications. |
| Infrastructure | Technical agreements about hardware, software, network protocols, and processes for implementing the infrastructure to support federation execution. | |
| | **Agreements** | |
| | Networking | Networking layout, network protocols used, wide area network (WAN), local area network (LAN), virtual LAN (VLAN), ports used. |
| | Physical Security | Physical security requirements and procedures for the facilities in which execution takes place: visit requests, non-United States (US) restrictions. |
| | Middleware | Middleware type, version, and configuration parameters and switches; references to configuration files. Both of the child elements of this element allow a sequence of any valid XML elements as their children in order to support various vendor-specific XML-encoded configurations. |
| | Secondary Communication Channels | Non-data exchange model data, e.g., video or chat; live entity, command, control, communication, computers, and intelligence (C4I) protocols, and hardware-in-the-loop (HWIL) communication protocols. |
| | Asset Allocation | Allocation of simulation systems, member applications, and live systems to specific hardware. |
| | Hardware Configuration | Specifies hardware to which software (including middleware) must be deployed and hardware configuration, including operating system versions. |
| | Locations | Identification of sites where federation components will be instantiated. |

<table>
<tr><td>Modeling</td><td>Agreements to be implemented in the member applications that semantically affect the current execution of the federation.
<table>
<tr><th colspan="2">Agreements</th></tr>
<tr><td>Effects Adjudication</td><td>Procedures for adjudicating effects such as damage and attrition.</td></tr>
<tr><td>Coordinate Systems</td><td>Reference to authoritative coordinate system representations. Included coordinate systems must be able to substitute for "gml:AbstractCoordinateSystem."</td></tr>
<tr><td>Dead Reckoning</td><td>Dead reckoning algorithms to be used.</td></tr>
<tr><td>Aggregation</td><td>Triggers and algorithms for aggregation and disaggregation.</td></tr>
<tr><td>Data Distribution Management (DDM)</td><td>Parameters and algorithms for interest management.</td></tr>
<tr><td>Control Transfers</td><td>Triggers, constraints, and algorithms for transferring control of entities between member applications.</td></tr>
<tr><td>Issues</td><td>Federation-specific modeling issues such as ground clamping, resupply/refuel.</td></tr>
<tr><td>Miscellaneous</td><td>Modeling agreements that cannot be easily categorized. As an example, notes would be included here about treating weekends and holidays different from weekdays or allowing for regular work hours during weekdays.</td></tr>
</table></td></tr>
<tr><td>Variances</td><td>Exceptions to the federation agreements deemed necessary during integration and testing.
<table>
<tr><th colspan="2">Agreements</th></tr>
<tr><td>Variance</td><td>Includes reference to agreement requiring variance, description of the variance, and the motivation/need for the variance. The variance element has two attributes: the variance identification (ID) and a reference to the ID of an agreement in the document. The description of the variance is in the body of the tag.</td></tr>
<tr><td>Remote Document</td><td>Zero or more references to remote documents containing detailed variance descriptions.</td></tr>
</table></td></tr>
</table>

As an example of how to use the schema to generate a conformant XML document, consider the following taken from a real federation agreements file [Reference (k)]. Figure 2 captures the set of responsibilities for a member application, in this case a federate modeling the effects of bioterrorism on a sample population. It also captures the publication and subscription requirements for this federate.

The initial information about the member application belongs in the "Member Apps" agreement and is illustrated below:

```
<fedagree:memberApps>
    <fedagree:memberApplication id="popModelFederate">                                  1
     <cpe-item xmlns=http://cpe.mitre.org/dictionary/2.0                                2
          name="cpe:/o:jhuapl:sparrows:1.0">
      <title>JHU APL Sparrows version 1.0</title>
     </cpe-item>
     <instance>                                                                         3
      <description>Population model for PACER Bioterrorism federation. Utilizes
                        the initial list of infected persons, population data, and TAZ
                data.</description>
      <responsibilities>
       <responsibility>Model the progression of smallpox in all infected
                              individuals through contagion,
                              death/recovery.</responsibility>
       <responsibility>Model the vaccination of individuals
                              stochastically</responsibility>
       <responsibility>Calculate fatalities</responsibility>
       <responsibility>Calculate number of people who recover with
                              vaccine</responsibility>
       <responsibility>Calculate number of people who recover without
                              vaccine</responsibility>
      </responsibilities>
     </instance>
    </fedagree:memberApplication
</fedagree:memberApps>
```

**Figure 2: Member Application Responsibilities Extract from Sample Federation Agreements Document**

1. A *memberApplication* element is created for the federate.
2. The software package implementing the federate is uniquely defined using the *cpe-item* tag.
3. A runtime instance with a description and various responsibilities is added. Note that any number of instance entries is supported for every *memberApplication* entry because a single implementation may have multiple instances at runtime with different responsibilities.

With the member application defined, it is now possible to list its publication and subscription responsibilities. This requires the use of the "Publication and Subscription Responsibilities" data agreement as illustrated in Figure 3.

```
<fedagree:pubSubResponsibilities>
    <appEntry appRef="popModelFederate">                                    1
      <publications>
        <publication architecture="HLA_1516">
          <demTargetEntity demName="DailyCommute"
                           xmlns:ns2="http://www.w3.org/1999/xlink"
                           ns2:href="file://./PACERFramework.xml#DailyCommute"/>   2
        </publication>
      </publications>
      <subscriptions>
        <subscription architecture="HLA_1516">
          <demTargetEntity demName="TraceInitiate"
                           xmlns:ns2="http://www.w3.org/1999/xlink"
                           ns2:href="file://./PACERFramework.xml#TraceInitiate"/>   2
        </subscription>
      </subscriptions>
    </appEntry>
</fedagree:pubSubResponsibilities>
```

**Figure 3: Publication and Subscription Responsibilities Extract from Sample Federation Agreements Document**

1. An *appEntry* element is added under *pubSubResponsibilites*, and the *appRef* attribute is given the value of the unique ID for the previously defined federate. If the ID has not yet been defined, the XML document will fail to validate against the schema.
2. Each publication and subscription is listed using both the unique name from the Data Exchange Model (DEM), and since the DEM is an High Level Architecture (HLA) 1516 FOM [Reference (l)], itself an XML file, an XLink *href* attribute points to the DEM file and the unique name of the element within the DEM that defines the object class or attribute being referenced.

**This page intentionally left blank.**

# APPENDIX A: REFERENCES

(a) Federation Development and Execution Process, IEEE 1516.3-2003, available at http://standards.ieee.org, 2003.

(b) Distributed Simulation Engineering and Execution Process, IEEE P1730, available in the future at http://standards.ieee.org.

(c) M&S Community of Interest Discovery Metadata Specification (MSC-DMS), available at http://metadata.dod.mil/mdr/ns/MSCDMS/1.2, July 2009.

(d) XML Linking Language (XLink), available at http://www.w3.org/1999/xlink, June 2000.

(e) XML Metadata Interchange (XMI), available at http://schema.omg.org/spec/XMI/2.1, September 2005.

(f) Common Platform Enumeration (CPE), available at http://cpe.mitre.org/language/2.0, March 2009.

(g) Intelligence Community Information Security Marking (IC-ISM), urn:us:gov:ic:ism:v2, April 2004.

(h) Xtensible Configuration Checklist Description Format (XCCDF), available at http://checklists.nist.gov/xccdf/1.1, January 2006.

(i) Geography Markup Language (GML), available at http://www.opengis.net/gml, February 2010.

(j) Federation Agreements Template Programmer's Reference, available at smb://aplfs01/Aplshare/NSAD/Software_Archive/JHS01A1/LVCFedAgreementsProgrammersReference, June 2010.

(k) Coolahan, J., Schloman, J., Kaisar, E., Kane, M., Kooullil, R., Morse, K. L., & Walsh, K. "Results of a Simulation of Response to a Bioterrorism Crisis," in Proceedings of the 2009 Interservice/Industry Training, Simulation and Education Conference, Orlando, FL, 30 November–3 December 2009.

(l) Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification, IEEE 1516.2-2000, available at http://standards.ieee.org, 2000.

**This page intentionally left blank.**

# APPENDIX B: MAPPING TO THE DSEEP

| Agreement | DSEEP Steps |
|---|---|
| **Design Agreements** | |
| Conceptual model | 2.2 Develop Conceptual Model |
| Information security | 1.1 Identify User/Sponsor Needs<br>1.2 Develop Objectives<br>1.3 Conduct Initial Planning<br>2.3 Develop Simulation Environment Requirements<br>3.2 Prepare Simulation Environment Design<br>3.3 Design Member Applications<br>4.2 Establish Simulation Environment Agreements |
| Terrain | 2.1 Develop Scenario<br>2.2 Develop Conceptual Model<br>2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment |
| Federation objectives/ Requirements | 1.2 Develop Objectives<br>2.3 Develop Simulation Environment Requirements |
| Environment | 2.1 Develop Scenario<br>2.2 Develop Conceptual Model<br>2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Scenario | 2.1 Develop Scenario |
| Federation architecture | 3.2 Design Simulation Environment |
| Member applications | 3.1 Select Member Applications<br>3.2 Design Simulation Environment<br>3.3 Design Member Applications<br>3.4 Prepare Detailed Plan<br>4.4 Implement Simulation Environment Infrastructure |
| Tools | 1.3 Conduct Initial Planning<br>3.4 Prepare Detailed Plan<br>4.3 Implement Member Application Designs |
| Authoritative domain information | 4.2 Establish Simulation Environment Agreements |
| **Execution Agreements** | |
| Initialization | 4.2 Establish Simulation Environment Agreements |
| Shutdown | 4.2 Establish Simulation Environment Agreements |
| Save | 4.2 Establish Simulation Environment Agreements |
| Restore | 4.2 Establish Simulation Environment Agreements |
| Synchronization | 4.2 Establish Simulation Environment Agreements |

| Agreement | DSEEP Steps |
|---|---|
| Time management | 4.2 Establish Simulation Environment Agreements |
| Join/Resign | 4.2 Establish Simulation Environment Agreements |
| Update rates | 4.2 Establish Simulation Environment Agreements |
| Performance thresholds | 4.2 Establish Simulation Environment Agreements |
| Heartbeat | 4.2 Establish Simulation Environment Agreements |
| Data logging | 2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>3.3 Design Member Applications<br>5.2 Integrate Federation |
| Data replay | |
| Monitoring/ Crash recovery/Fault tolerance/Exception handling (Anomalies) | 2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>3.3 Design Member Applications<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements<br>5.2 Integrate Federation |
| Middleware service agreements | 4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| **Management/Process Agreements** | |
| Software development/Configuration management/Version control processes | 1.3 Conduct Initial Planning<br>3.4 Prepare Detailed Plan |
| V&V | 3.4 Prepare Detailed Plan |
| Schedules | 3.4 Prepare Detailed Plan |
| Test plan | 3.4 Prepare Detailed Plan |
| Compliance certification | 4.3 Implement Member Application Designs |
| **Data Agreements** | |
| Data encoding | 4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Data exchange model | 4.1 Develop Simulation Data Exchange Model |
| Naming conventions | 4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Publication and subscription responsibilities | 4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Data dictionary elements | 4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |

| Agreement | DSEEP Steps |
|---|---|
| Supporting databases | 4.2 Establish Simulation Environment Agreements |
| **Infrastructure Agreements** | |
| Networking | 1.2 Develop Objectives<br>2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>3.4 Prepare Detailed Plan<br>4.2 Establish Simulation Environment Agreements<br>4.4 Implement Simulation Environment Infrastructure |
| Physical security (facility) | 1.1 Identify User/Sponsor Needs<br>1.2 Develop Objectives<br>1.3 Conduct Initial Planning<br>2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>3.3 Design Member Applications |
| Middleware | 1.2 Develop Objectives<br>2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>3.4 Prepare Detailed Plan<br>4.2 Establish Simulation Environment Agreements<br>4.4 Implement Simulation Environment Infrastructure |
| Secondary communication channels | 1.2 Develop Objectives<br>2.3 Develop Simulation Environment Requirements<br>3.2 Design Simulation Environment<br>3.4 Prepare Detailed Plan<br>4.2 Establish Simulation Environment Agreements<br>4.4 Implement Simulation Environment Infrastructure |
| Asset allocation | 3.1 Select Member Allocations<br>3.4 Prepare Detailed Plan<br>4.4 Implement Simulation Environment Infrastructure<br>5.2 Integrate Simulation Environment |
| Physical sites/Locations | 3.1 Select Member Allocations<br>3.4 Prepare Detailed Plan<br>4.4 Implement Simulation Environment Infrastructure<br>5.2 Integrate Simulation Environment |
| Hardware configuration | 3.1 Select Member Allocations<br>3.4 Prepare Detailed Plan<br>4.4 Implement Simulation Environment Infrastructure<br>5.2 Integrate Simulation Environment |
| **Modeling Agreements** | |
| Effects adjudication | 2.2 Develop Conceptual Model<br>3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Coordinate systems | 3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |

| Agreement | DSEEP Steps |
|---|---|
| Dead reckoning | 3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Aggregation/Disaggregation | 3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Data Distribution Management (DDM)/Interest management | 3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Transfer of control | 3.2 Design Simulation Environment<br>4.1 Develop Simulation Data Exchange Model<br>4.2 Establish Simulation Environment Agreements |
| Modeling issues | 3.2 Design Simulation Environment<br>4.2 Establish Simulation Environment Agreements |
| **Variances** | |
| Variances | N/A |
| **Document Metadata** | |
| Federation name | 1.3 Conduct Initial Planning<br>2.3 Develop Simulation Environment Requirements |
| Identification | N/A |
| Federation executions | N/A |
| Referenced documents | N/A |
| Previous agreements | 3.2 Design Simulation Environment |
| Glossary | N/A |

# APPENDIX C: ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| C4I | Command, Control, Communication, Computers, and Intelligence |
| CM | Configuration Management |
| CPE | Common Platform Enumeration |
| DDM | Data Distribution Management |
| DEM | Data Exchange Model |
| DISR | Defense Information Technology Standards and Profile Registry |
| DSEEP | Distributed Simulation Engineering and Execution Process |
| FEDEP | Federation Development and Execution Process |
| FOM | Federation Object Model |
| GML | Geography Markup Language |
| HLA | High Level Architecture |
| HTML | Hypertext Markup Language |
| HWIL | Hardware-in-the-loop |
| IC-ISM | Intelligence Community Information Security Marking |
| ID | Identification |
| INFOSEC | Information Security |
| JHU/APL | The Johns Hopkins University Applied Physics Laboratory |
| LAN | Local Area Network |
| LROM | Logical Range Object Model |
| LVC | Live-Virtual-Constructive |
| M&S | Modeling and Simulation |
| MSC-DMS | M&S Community of Interest – Discovery Metadata Specification |
| PDU | Protocol Data Unit |
| SNE | Synthetic Natural Environment |
| TBD | To be determined |
| UML | Unified Modeling Language |
| US | United States |
| VLAN | Virtual LAN |
| VV&A | Verification, Validation, and Accreditation |

| | |
|---|---|
| WAN | Wide Area Network |
| XCCDF | eXtensible Configuration Checklist Description Format |
| XLink | XML Linking Language |
| XMI | XML Metadata Interchange |
| XML | eXtensible Markup Language |

**NATIONAL SECURITY ANALYSIS DEPARTMENT**

THE JOHNS HOPKINS UNIVERSITY • APPLIED PHYSICS LABORATORY

Johns Hopkins Road, Laurel, Maryland 20723-6099