High Level Architecture and NATO Education and Training Network (NETN) Federation Object Model (FOM) Overview

Tom van den Berg
TNO Applied Physics Laboratory
THE NETHERLANDS

tom.vandenberg@tno.nl

ABSTRACT

NATO and the Nations regularly use distributed simulation based on the High Level Architecture (HLA) interoperability standard. The current release is IEEE 1516-2010. In the NATO context, several official documents are available to standardize the use of HLA across NATO Nations:

- *STANAG-4603 (IEEE 1516), describing the HLA standard.*
- *STANREC-4800 (AMSP-04), describing the NATO Education Training Network (NETN) Federation Object Model and the associated Federation Architecture and FOM Design (FAFD).*

The NATO Modelling and Simulation Group (NMSG) has a close relationship and a Technical Agreement with the Simulation Interoperability Standards Organization (SISO) with respect to Simulation Interoperability Standards and has provided significant input to SISO standards development over the last years. E.g., NMSG Task Groups MSG-068 and MSG-106 provided significant input on the modularization of RPR-FOM v2.0, and other Task Groups have provided significant input on C-BML, MSDL, C2SIM, and UCATT SISO standards. NATO is also a user of SISO developed standards including IEEE 1516 (HLA) series of standards which is covered in STANAG 4603.

SISO and NATO standards have successfully been applied in developing Federated Simulations to support Education, Training, Decision Support, Exercise and Evaluation amongst others. Other NMSG Task Groups apply the SISO and NATO standards to support different aspects of M&S, e.g., Task Groups MSG-147/191 are developing NETN FAFD FOM modules for Crisis Management and Disaster Response, and MSG-164/195 are developing the concept of Modeling & Simulation as a Service (MSaaS) to manage Federated Simulation etc.

There is a continued need for NATO to experiment with, update and further evolve NATO Standards for Federated Simulation to meet new and evolving simulation interoperability requirements and to harmonize with new and evolving SISO standards.

This paper provides an overview of the High Level Architecture, the NETN FOM, and how these standards can be applied in federated simulation.

1.0 INTRODUCTION

Modelling and Simulation (M&S) is a key enabler for several activities within NATO, including support to education, training, decision support, exercises, and test & evaluation of the alliance's units and systems. The use of M&S provides the ability to represent complex and challenging environments, large scenarios, and new concepts in distributed and federated systems. National and NATO M&S assets can be provided (as services) to create joint and combined synthetic environments.

Efficient and effective use of NATO and national M&S assets require standards for connecting and integrating M&S components. A strategic view on the development, evolution and implementation of these standards is necessary to exploit the full potential of M&S across the alliance. The NATO M&S Master Plan [1] identifies five (5) high-level objectives:

1) Establish a Common Technical Framework
2) Provide Coordination & Common Services
3) Develop Models & Simulations
4) Employ Simulations
5) Incorporate Technical Advances

The NATO Modelling and Simulation Group (NMSG) is a specific group within the NATO Science & Technology Organization [2] focused on the development and evolution of NATO M&S including the implementation of the NATO M&S Master Plan. Interoperability, Reuse, Affordability and Synergy are the principles that guide the NMSG in its implementation of the master plan objectives.

One of the fundamental standards related to the Common Technical Framework objective is STANAG 4603, which mandates the use of the IEEE 1516 High-Level Architecture standard for federated simulation. The same STANAG also recommends the use of IEEE 1730 DSEEP and requires alliance members to utilize a NATO Federate Certification Service for testing compliance with the HLA standard. STANAG 4603 is ratified by approximately 20 NATO nations, is supervised by NMSG, and is under the custodianship of the US DoD M&S Coordination Office.

Another fundamental standard for federated simulation is AMSP-04 NATO Education and Training Network Federation Architecture and FOM Design (NETN FAFD), covered by STANREC 4800. The NETN FAFD is a reference federation agreement document based on the use of STANAG 4603 HLA and the SISO-STD-001 Real-Time Platform Reference FOM. Input from national M&S programs to support Computer Assisted eXercises (CAX) influenced the development of a NETN FOM, which extends the RPR-FOM with additional concepts and patterns for simulation interoperability. Since its first release in 2011 the NETN FOM has evolved based on operational use by NATO and nations to its current version NETN FOM v3.0 published in 2021.

The following sections of this paper provide an overview of the HLA, discuss the NETN FOM, and briefly touch on the topic of simulation engineering. That is, IEEE 1730 DSEEP and SISO-STD-012 Federation Engineering Agreements Template.

2.0 OVERVIEW OF THE HIGH LEVEL ARCHITECTURE

The High Level Architecture (HLA) is an international standard for the development of distributed simulation environments. In the terminology of the HLA, individual simulation applications are known as federates. Federates may be simulation models, data collectors, simulators, computer generated forces or passive viewers. The collection of federates brought together to form a synthetic environment is known as a federation. It is the common interpretation of a shared data model, called the Federation Object Model (FOM), that allows federates to interact within a single synthetic environment. A federation execution refers to the process of conducting a distributed simulation. Federates interact via a Runtime Infrastructure (RTI). The RTI provides a number of Application Programming Interface (API) service groups that are used by a federate to interact with the underlying communication layer.

Figure 1 provides an example of an HLA federation, where simulators, support tools, and live participants interact through a common Run Time Infrastructure.

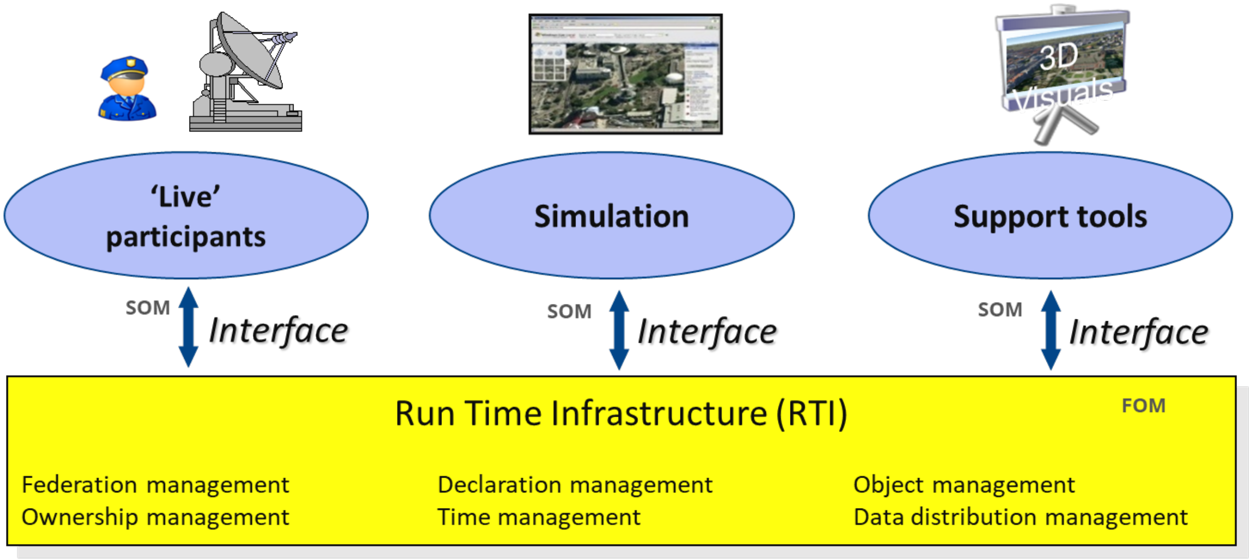

Figure 1: A graphical view of the HLA: federates operate together through a common Run Time Infrastructure (RTI).

The HLA is focused on interoperability between various types of simulations, and to promote reuse of simulations and their components. The HLA follows two general design principles:

- ***Modularity***: simulation components (federates) are composed into larger systems (federations) to obtain a specific functional behavior;
- ***Separation of concerns***: the functional behavior of the components (federates) is separated from the supporting communication infrastructure (RTI) via a well-defined interface.

The HLA was originally developed for defence applications but there is a growing non-defence user base of the HLA. Numerous publications on HLA applications can be found via google scholar. A search on the publications from 2010 with keywords "HLA RTI" yields over 2700 hits and shows a variety of topics such as warfare simulation, distributed-parallel computer simulations, cyber physical simulation, aircraft flight simulation, railway simulation, off-shore maritime simulation, engineering design analysis simulation, engine simulation, and lunar landing simulation.

The HLA is an international standard, developed and maintained by the Simulation Interoperability Standards Organization (SISO) and published by IEEE. The first complete version of the standard was published in 1998. It was known as "HLA 1.3". HLA became an IEEE standard (IEEE 1516) in 2000. The IEEE 1516 standard has been updated in 2010 and is known as "HLA Evolved". Currently the next edition of the standard is in preparation, named "HLA 4" and expected to be released in 2024. In the remainder of this paper the term HLA refers to HLA Evolved, unless indicated otherwise.

The HLA standard is composed of three parts: the HLA Framework and Rules, the HLA Interface Specification, and the HLA Object Model Template (OMT) Specification:

- IEEE 1516-2010. HLA Framework and Rules: ten rules describing the responsibilities of federations and federates and their relationship with the RTI [3];
- IEEE 1516.1-2010. HLA Interface Specification: identifies how federates interact within the federation. In fact, it specifies the API (Application Programmer's Interface) of the HLA Run Time Infrastructure (HLA-RTI) [4];

- IEEE 1516.2-2010. HLA Object Model Template (OMT) Specification: provides a common format for describing all HLA objects and interactions, and establishes the syntax and format of the Federation Object Model (FOM) and Simulation Object Model (SOM) [5].

These parts are discussed in the following sections.

2.1 Framework and Rules

The HLA Framework and Rules [3] mandate a certain structure for federates and federations to ensure that the models are re-usable across applications.

There are 10 rules.

The rules for federations are in summary:

- Federations shall have an HLA FOM, documented in accordance with the HLA OMT;
- In a federation, all simulation-associated object instance representation shall be in the federates, not in the RTI;
- During a federation execution, all exchange of FOM data among joined federates shall occur via the RTI;
- During a federation execution, joined federates shall interact with the RTI in accordance with the HLA interface specification;
- During a federation execution, an instance attribute shall be owned by at most one joined federate at any given time.

And the rules for federates are in summary:

- Federates shall have an HLA SOM, documented in accordance with the HLA OMT;
- Federates shall be able to update and/or reflect any instance attributes and send and/or receive interactions, as specified in their SOMs;
- Federates shall be able to transfer and/or accept ownership of instance attributes dynamically during a federation execution, as specified in their SOMs;
- Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of instance attributes, as specified in their SOMs;
- Federates shall be able to manage local time in a way that will allow them to coordinate data exchange with other members of a federation.

2.2 Interface Specification

The HLA Interface Specification [4] describes seven service groups which are used by the federate to interact with the underlying communication layer, called the Run Time Infrastructure (RTI). A service group is a term to refer to a collection of related interface calls to the RTI. All communications between the federates are processed through the RTI. The federates may give advice, or send requests to the RTI, and the RTI can respond asynchronously by invoking certain well-known call-back methods. A call-back is a user-defined piece of software code (with a given interface) that is invoked by the RTI when a certain event occurs. The call and call-back interaction between federate and RTI is illustrated in Figure 2.

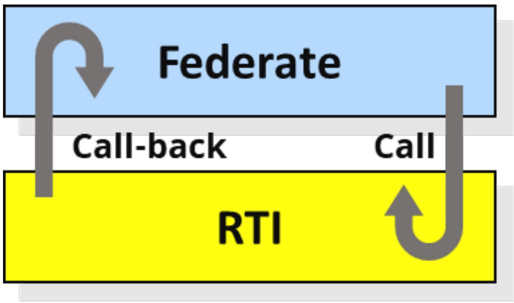

Figure 2: RTI call and call-back.

The seven service groups are in summary:

- **Federation Management.** These services allow for the coordination of federation-wide activities throughout the life of a federation execution. Such services include federation execution creation and destruction, federate application joining and resigning, federation synchronization points, and save and restore operations. This can for example be used to create ‘‘snapshots’’ of the simulation in order to resume execution at a later stage.
- **Declaration Management.** These services allow joined federates to specify the types of data they will supply to, or receive from, the federation execution. This process is done via a set of publication and subscription services along with some related services.
- **Object Management.** These services support the life-cycle activities of the objects and interactions used by the joined federates of a federation execution. These services provide for registering and discovering object instances, updating and reflecting the instance attributes associated with these object instances, deleting or removing object instances as well as sending and receiving interactions and other related services. (Note: Formal definitions for each of these terms can be found in the definitions clause of all three HLA specifications.)
- **Ownership Management.** These services are used to establish a specific joined federate’s privilege to provide values for an object instance attribute as well as to facilitate dynamic transfer of this privilege (ownership) to other joined federates during a federation execution.
- **Time Management.** These services allow joined federates to operate with a logical concept of time and to jointly maintain a distributed virtual clock. These services support discrete event simulations and assurance of causal ordering among events.
- **Data Distribution Management.** These services allow joined federates to further specify the distribution conditions (beyond those provided via Declaration Management services) for the specific data they send or ask to receive during a federation execution. The RTI uses this information to route data from producers to consumers in a more tailored manner, for example to receive only updates from objects that are in the geographical vicinity in the simulated world.
- **Support Services.** This group includes miscellaneous services utilized by joined federates for performing such actions as name-to-handle and handle-to-name transformations, the setting of advisory switches, region manipulations, and RTI start-up and shutdown.

The RTI services provide many ways to optimize the federation execution in terms of wall clock execution time and the amount of data exchanged. For example, via advanced time management schemes, object update rate reduction, data interest management, attribute ownership transfer, and data distribution management.

2.3 Object Model Template Specification

All possible data exchanged by federates in a federation is captured in an object model [5]. The object model defines the HLA object classes to describe the persistent state of entities, and the HLA interaction classes to describe transient events. The HLA-OMT provides a format for this object model.

HLA-OMT consists of several parts, such as:

- Object model identification: provides a general description of the object model, such as domain, purpose, description, use history.
- Object classes and attributes: describes the object classes, the object class attributes, and the object class hierarchy.
- Interaction classes and parameters: describes the interaction classes, the interaction class attributes, and the interaction class hierarchy.
- Synchronization points: describes the synchronization points used in the federation or by the federate.
- Datatypes: specifies the available datatypes (array, fixed record, variant record, simple) for the object class attributes and interaction class parameters.

There are three kinds of object models in the HLA framework, all using the HLA-OMT as format, namely:

- Simulation Object Model (SOM);
- Federation Object Model (FOM);
- Management Object Model (MOM).

An individual federate is described by its SOM. The SOM is an object model in the HLA-OMT format that provides details of the object attributes and interactions that this federate either provides or receives information about.

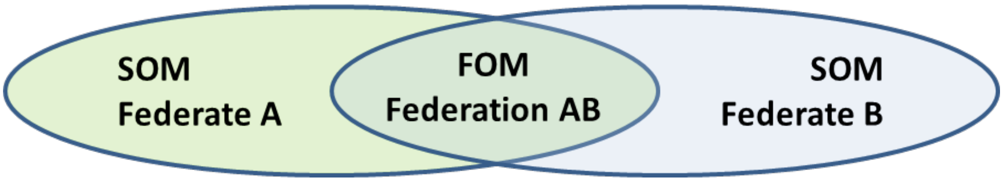

Figure 3: FOM and SOM.

All data that is potentially exchanged in a collection of federates (i.e., the federation) is described by the FOM. The FOM is also an object model in the HLA-OMT format that contains all objects and interactions that the federates may exchange. Since all information is available in the individual SOMs, the FOM can be constructed out of the SOMs. In addition, the FOM may contain some federation-wide information for efficient data distribution management.

The FOM and SOMs may be regarded as technical contracts that serve as interface specifications for the federate developers. A particular federate in a federation may be replaced by another version if it complies with the same SOM and federation agreements as the original federate.

A third object model is the MOM. The MOM is a group of predefined constructs that provide support for monitoring and controlling a federation execution. A predefined FOM module, called MOM and Initialization Module (MIM), contains predefined HLA constructs such as object and interaction roots, data types, transportation types, and dimensions.

The FOM may be developed from the individual SOMs, but the use of a reference FOM is often a good starting point. An example of a reference FOM is the SISO RPR-FOM (Real-time Platform-level Reference FOM) [6]. The RPR-FOM is a reference FOM that defines HLA classes, attributes and parameters that are appropriate for real-time, platform-level simulations in the military domain.

Lastly, a FOM, SOM, or MOM may consist of different modules.

- **FOM modules.** These are modules that contain FOM information and are combined together to form a valid FOM. A MOM Module is also required in a FOM.
- **SOM modules.** In much the same way as a FOM can be built by combining FOM modules, a SOM can be built from SOM modules.
- **MOM module.** This is a module that contains the MOM information, including predefined data types, transportation types, and dimensions. There is a standard MOM module that may be provided automatically, but it is also possible to provide a new MOM module that extends the standard MOM module.

FOM and SOM modules may be combined to create FOMs and SOMs for federations and federates. There are a number of advantages in using FOM and SOM modules, such as:

- Modularity: keep closely related classes together in a smaller coherent object model, that can be re-combined with other modules.
- Extendibility: keep extensions to an existing object model in a separate module.
- Reusability: reuse object models and related simulation environment agreements between different federations.

Figure 4 illustrates the FOM modules that make up the SISO RPR-FOM [6].

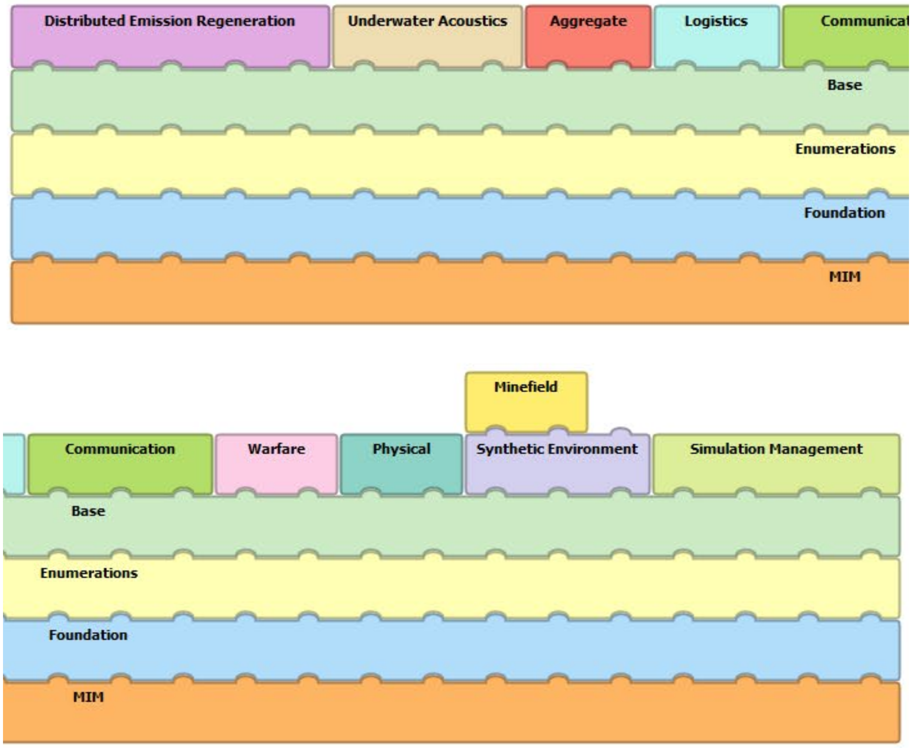

Figure 4: SISO-STD-001 RPR-FOM modules.

2.4 Putting Things Together

This section provides an overview of a typical usage of the RTI services.

The first service group that a federate will use is federation management. The federation management services enable federates to join the federation as depicted in Figure 5. Upon joining each federate provides a set of SOM or FOM modules that it will use for the exchange of data.

Next, federates will need to declare their interest in the data described in the SOM/FOM modules, and tell the RTI what data they provide and consume. For this the declaration management services are used. This is illustrated in Figure 6.

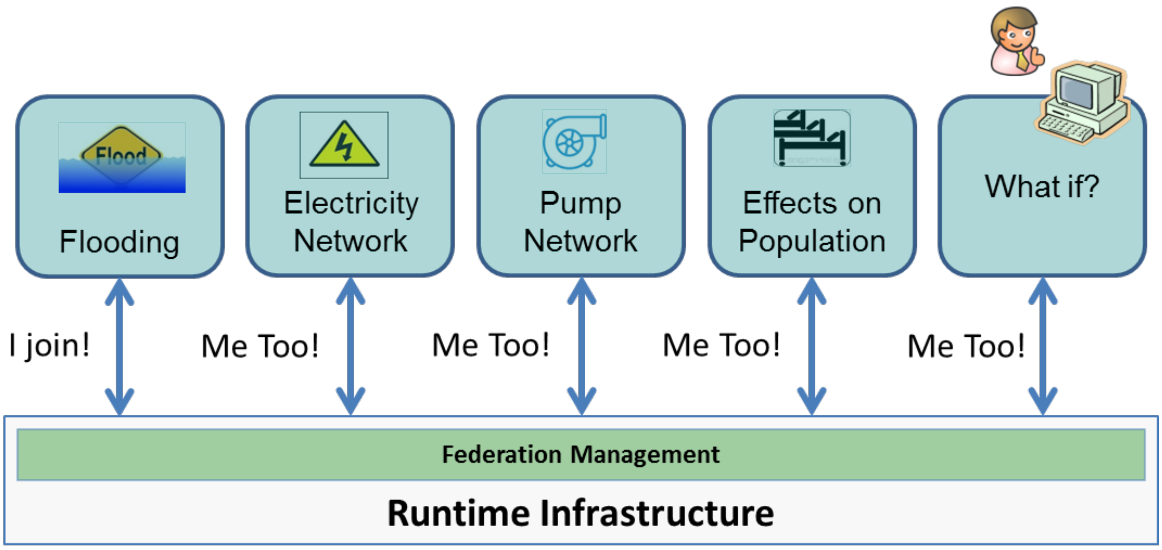

Figure 5: Federates joining a federation.

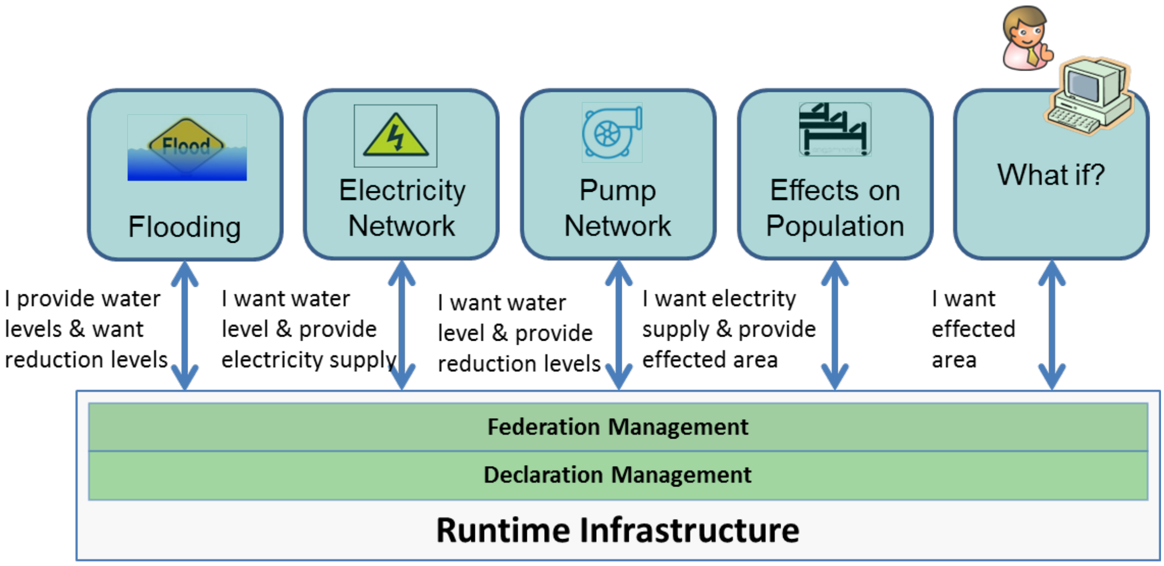

Figure 6: Federates need to describe what data they provide/consume.

To communicate with each other, federates use the object management services as depicted in Figure 7. The object management services deal with the registration, modification, and deletion of object instances and the sending and receipt of interactions.

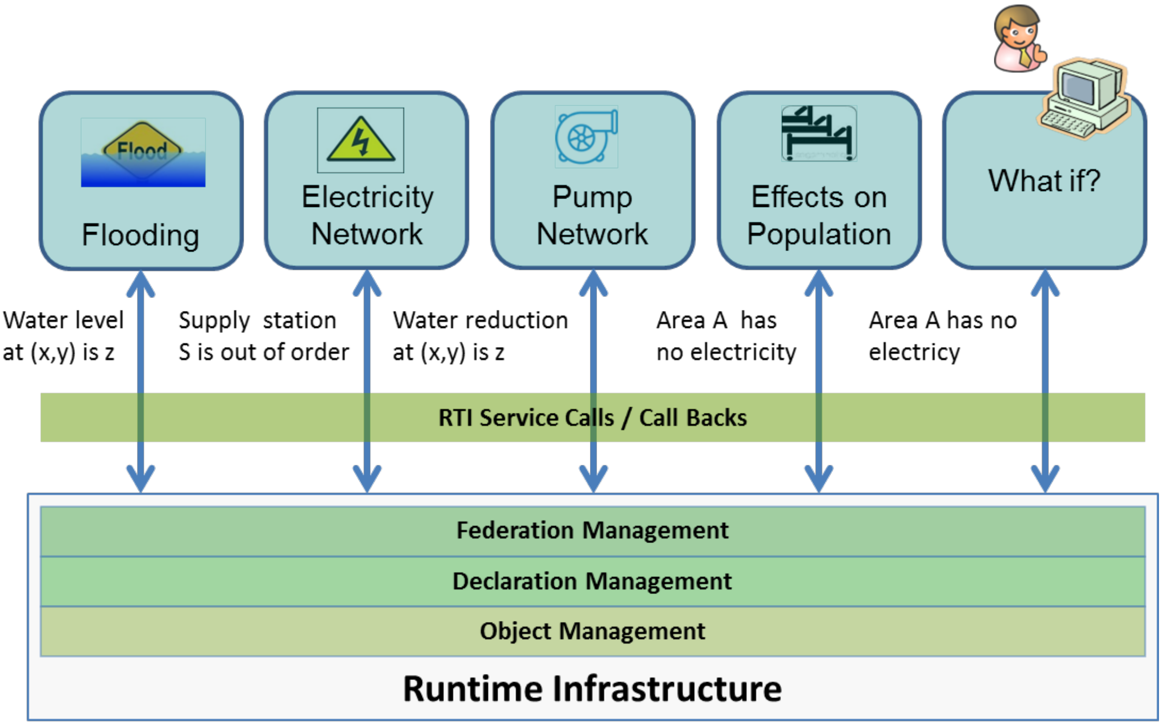

Figure 7: Federates need to exchange data and interactions.

In order to maintain causality between messages (object instance updates and interactions) that federates exchange, the RTI time management services must be used. A federate uses the RTI time management services to manage logical time and to ensure that the data that is exchanged with the object management services is delivered at the correct logical time at other federates. A federate can ask the RTI if it is allowed to proceed in time. The RTI checks whether all other federates are ready to proceed. If so, it tells the federates with which Δt they can progress. The time management services support different ways to progress logical time (e.g., time stepped, event driven) in order to optimize time advancement and concurrency in federation execution. The RTI is responsible for keeping the federates time synchronized.

Figure 8 provides an example what could happen if time is not synchronized; each federate progresses time at its own pace and the federates are all at a different logical time when they exchange data. Causality is not guaranteed.

To increase scalability of a federation and performance of federates, updating of information can be optimized. As depicted in Figure 9 a federate can instruct the RTI to forward only the information that is relevant for him. This mechanism reduces the workload on the federate: it doesn't have to process data that can be discarded anyway.

Federates can internally use different concepts than specified in the FOM of the federation it wants to join, such as units. The FOM may specify distance in kilometres, whereas the federate internally uses meter as unit. Mapping of FOM attribute values to internal values is the responsibility of the joining federate.

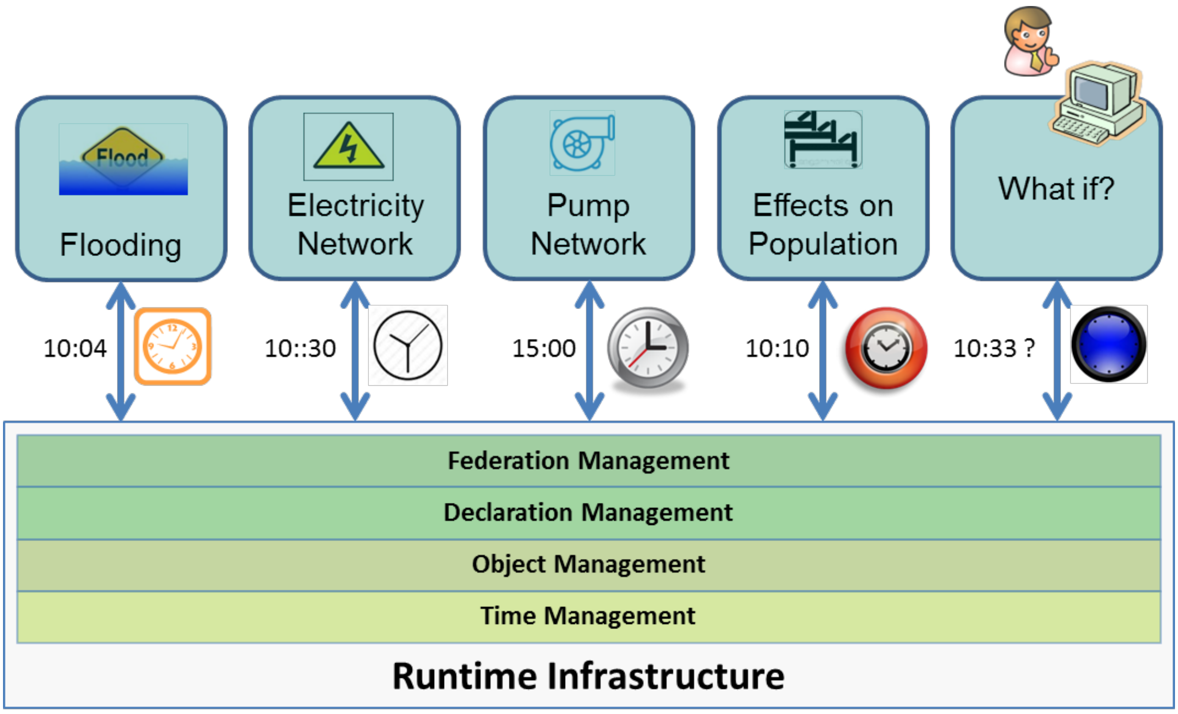

Figure 8: Federate simulation time need to be synchronized.

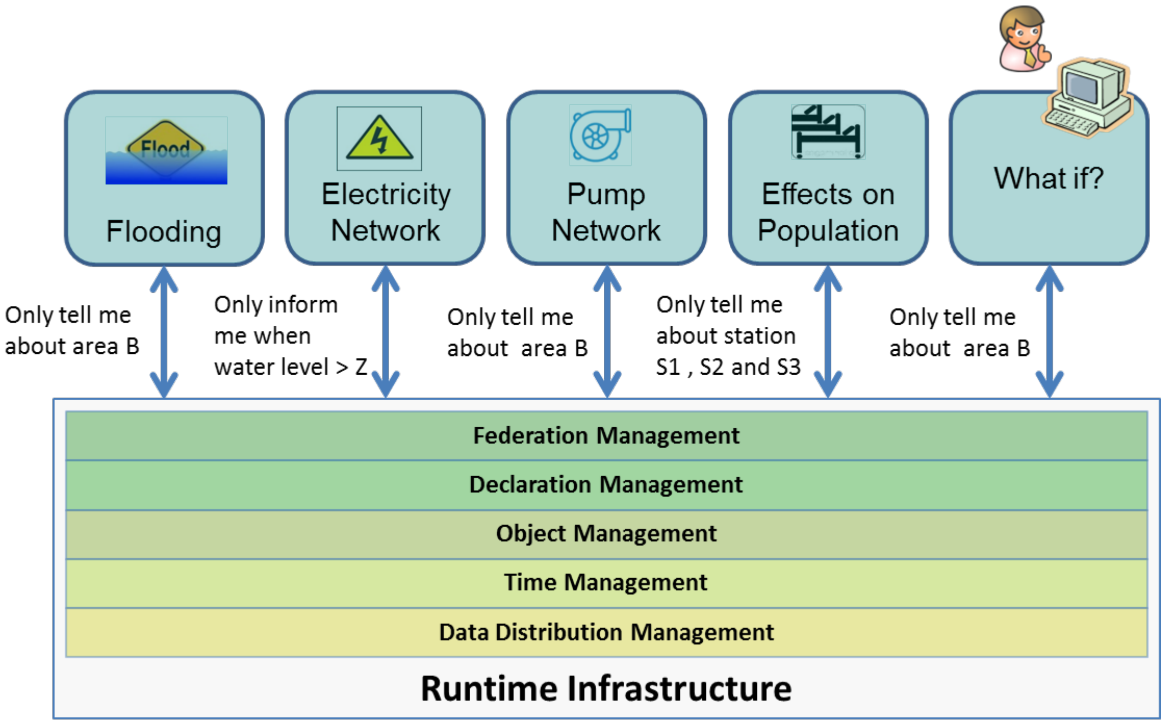

Figure 9: Updating of information can be optimized.

Finally, Figure 10 and Figure 11 show a high-level schema of the steps to create and execute a federated simulation. These are the typical steps performed in the lifecycle of a federation.

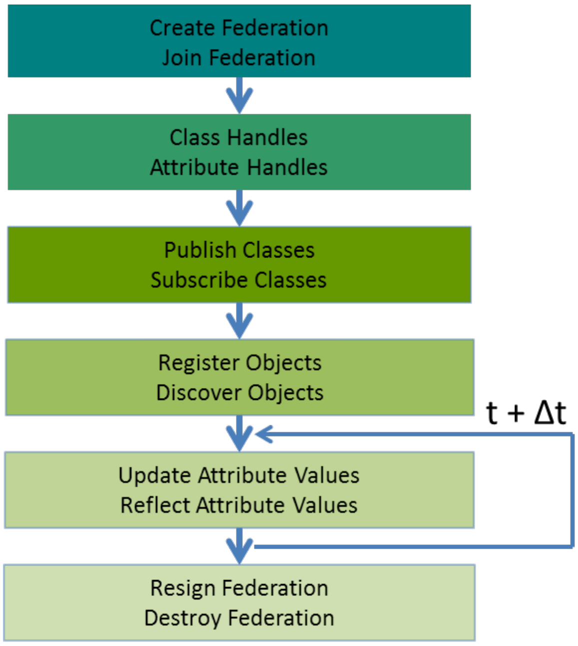

Figure 10: Schematized HLA program walkthrough: Lifecycle of a Federation.

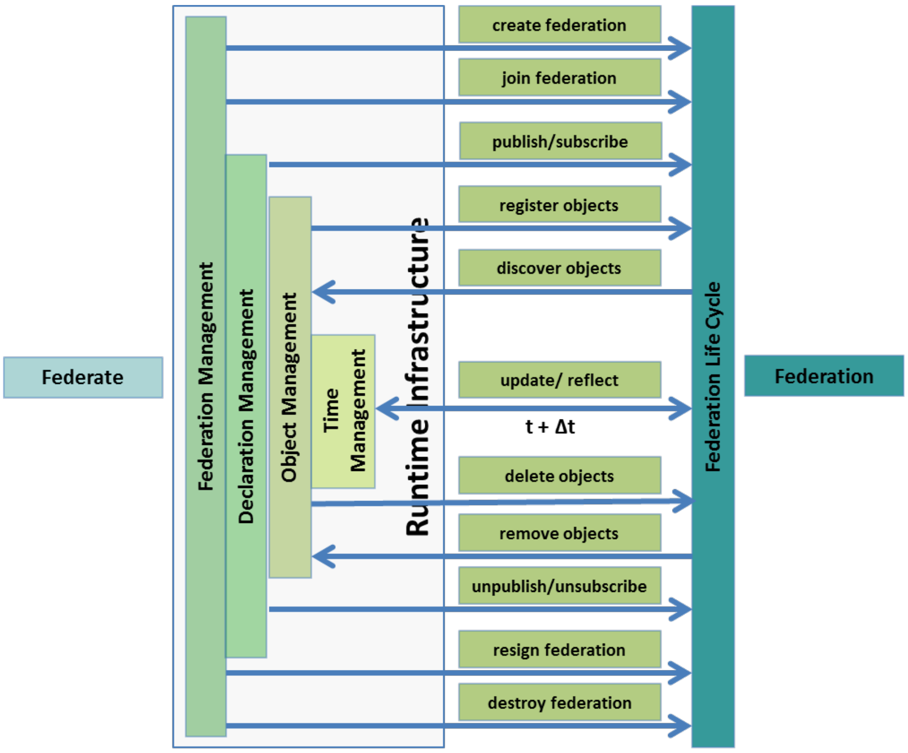

Figure 11: Program walkthrough schema and interactions: Lifecycle of a Federation.

3.0 NATO EDUCATION AND TRAINING NETWORK (NETN) FOM

NATO AMSP-01 “The NATO M&S Standards Profile” (NMSSP) identifies STANAG 4603 HLA and STANREC 4800 NETN FAFD as key standards to support federated simulation. NETN FAFD is a modular reference federation agreement that contains a set of HLA FOM Modules that extends and complements the SISO-STD-001 RPR-FOM v2.0.

The NETN FOM modules are recommended for use when implementing NATO AMSP-04 NETN FAFD compliant distributed simulation. These NETN FOM include reference to other standard FOM modules as well as NETN modules developed by NATO Modelling and Simulation Group (NMSG) Modelling and Simulation Standards Subgroup (MS3). The modules have inter-dependencies and have been designed to maximize re-use and interoperability with legacy systems using existing standards, and those having requirements for new patterns of simulation interoperability. The NETN FOM is the complete set of NETN modules and all other modules they depend on (e.g., SISO RPR-FOM modules).

The NETN FOM has evolved since it was first release in 2010 and the current version, NETN FOM v3.0, is currently under revision. Release of a draft NETN FOM v4.0 is expected in 2024 followed by a release by NSO as AMSP-04 Ed C covered by STANREC 4800. A timeline of the NETN FOM is provided in Figure 12.

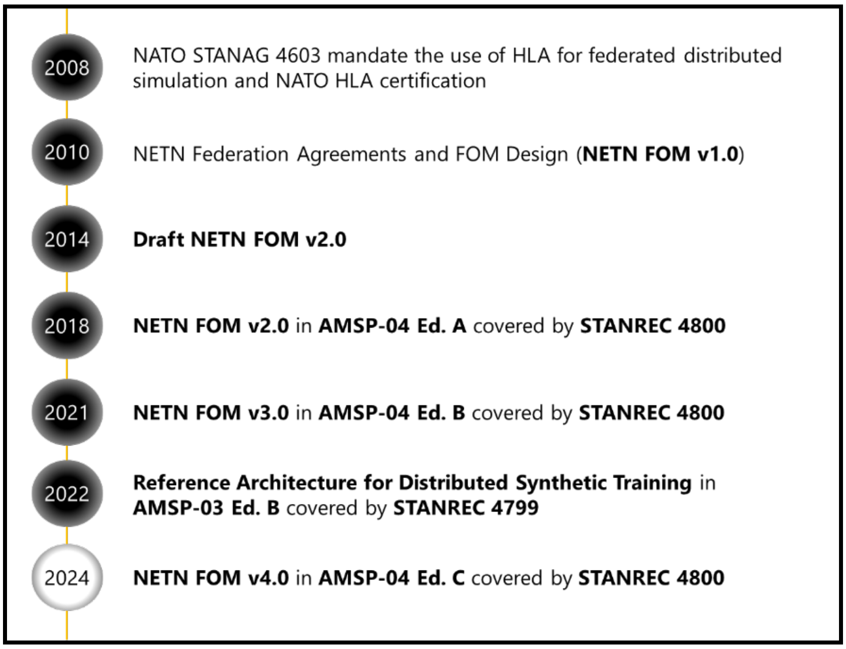

Figure 12: NETN evolution over time.

Input to the revised NETN FOM comes from various national and NATO projects and programs using more or less of the FOM modules to support education, training, exercises and evaluation. e.g., the Swedish exercise series Viking, has used several of the current FOM modules in NETN to provide better support for automation of EXCON activities and Simulation-C2 interoperability. In addition, several other NMSG research Task Groups with participants from many NATO and partner nations have supported the development of NETN FOM Modules. The current NETN FOM v3.0 includes 11 FOM modules that complement and/or extend existing RPR-FOM v2.0 modules, see Figure 13.

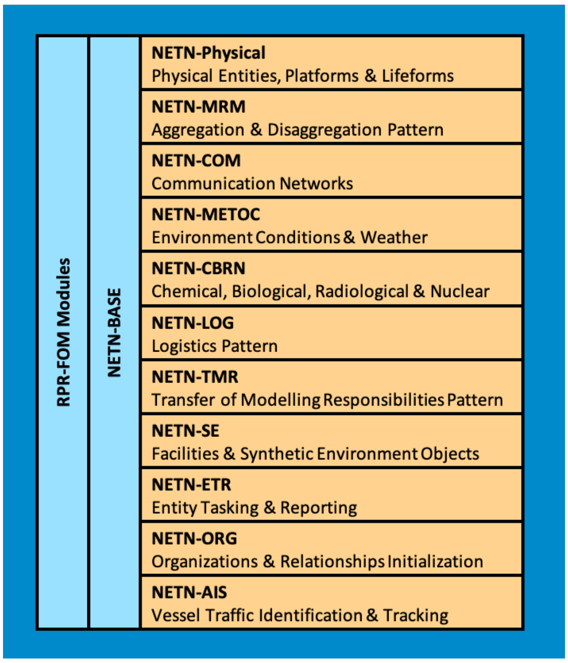

Figure 13: NETN FOM modules.

AMSP-04 Edition A was published in March 2018 covered by a NATO STANREC 4800 and defines the NETN FOM v2.0 FOM modules and associated documentation. The AMSP-04 including the NETN FOM modules are "Not Classified" and are releasable to the public. All NETN FOM modules are published in GitHub and are available to the general public for download and for providing comments/issues. The "development branch" of each FOM module contains the current "under development" version for the module. The version on the development branch can be accessed for early access and engagement. See [7].

A NETN federation design may extend NETN FOM modules (as new modules), include other FOM modules, and/or select to use only a subset of the NETN modules, all depending on the needs and requirements of the federation. When extending the FOM with additional modules, the naming of classes, datatypes and other identifiers must be de-conflicted and the basic FOM Module merging rules as defined in HLA Evolved are applied.

Registered objects and interactions are always discovered/received at the most specific subscribed class level. Extending a FOM Module with additional subclasses provides the possibility to add extra attributes/parameters at the more specific class level. Exchange of information using this more specific level can take place between federates publishing and subscribing to this level. However, to become compatible with and receive information from federates only publishing on the more general level, the receiving federate must subscribe to both class levels. Subscribers of the more general class will receive information from publishers of the more specific class level.

Example: A national extension to the NETN FOM Modules subclasses existing NETN object classes and defines additional attributes. National models aware of this extension can publish and subscribe to the more specific level defined in the national FOM module extensions. Other existing federates not aware of the extension can still discover the object and receive updates, but only on the level they subscribe to. In order for the national federates to discover and receive information from other federates they need to subscribe to the NETN class level as well as the national extension level. Note that the discovered object and attribute updates will be on the NETN level.

3.1 Representation of Organizations

Modelling of an organizational unit in a distributed simulation depends on its intra-organizational relationship with other units, e.g., superior, subordinate, and the relationship between organizations, e.g., friendly or hostile. This organizational information is normally provided to simulation systems as part of initialization based on the scenario.

The NETN Organization FOM Module (NETN-ORG) is a specification of how to represent organizations in a federated simulation and provides a common standard interface for the representation of the state of units including command structure and relationship between organizations. This representation can be used for setting the initial state of simulated entities, capturing subsequent snapshot states and for dynamic change of organizational relationships. An example federation design using MSDL to initialize the simulation and to generate snapshots for viewing or storage is provided in Figure 14.

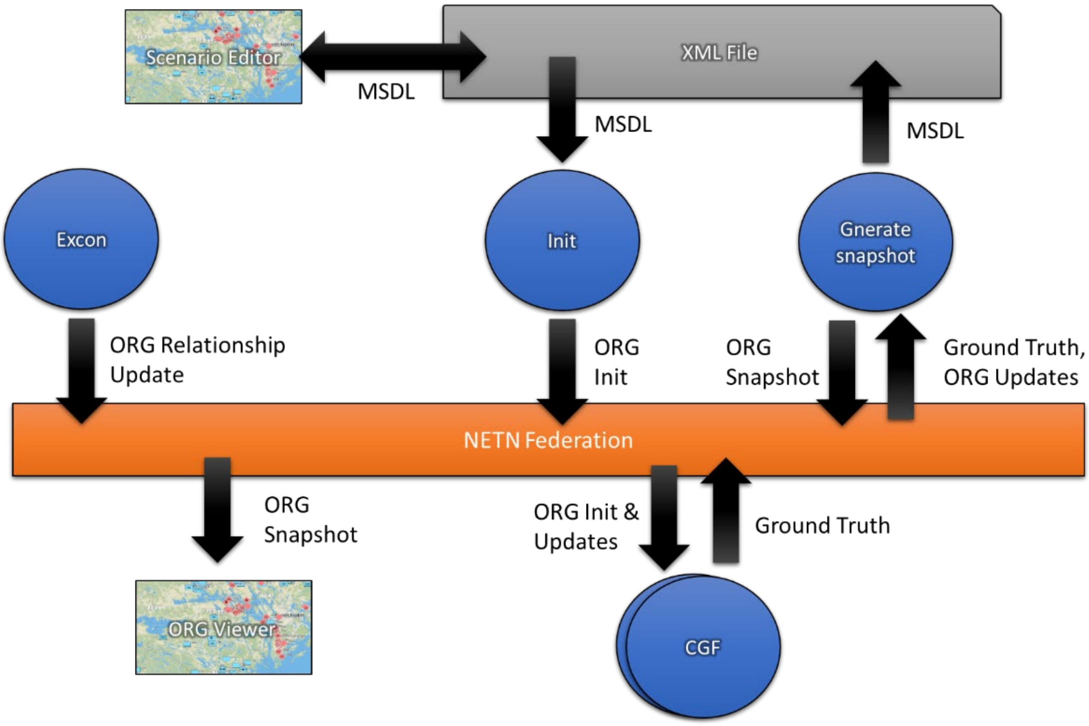

Figure 14: Use of MSDL to initialize the simulation and to generate snapshots for viewing or storage.

The SISO-STD-007-2008 Military Scenario Definition Language (MSDL) has been used as the foundation of NETN-ORG and the information modelled in the federation regarding organizations can easily be mapped to the corresponding MSDL representation in XML. This allows for easy generation of snapshots of current state in the federation to XML based MSDL and for initialization from MSDL files. Some optional extensions to MSDL developed by NATO MSG-106 are also supported including information about the initial allocation of modelling responsibility to federates.

The later developed SISO-STD-020-2020 C2SIM standard for Land Operations Extension (LOX) incorporates many concepts of MSDL and its **information model is compatible with MSDL. Also, the LOX information model may be used to initialize and save** the information modelled in the federation regarding organizations.

Both MSDL, LOX, and the NETN-ORG module use a Universally Unique Identifier (UUID) to identify the organizational elements. This provides a way to uniquely identify entities even between federation executions. To take full advantage of the UUID, the NETN FOM defines two FOM modules that extend the RPR-FOM representation of aggregate units and physical entities.

The NETN-MRM and NETN-Physical both extend the corresponding RPR-Aggregate and RPR-Physical modules respectively and adds a UUID attribute used to publish the unique identifier during federation execution. This extension allows NETN units and platforms to be represented in multiple executions, e.g., Monte-Carlo simulation, and later correlated for analysis.

3.2 Logistics Modelling and Simulation

Military logistics is the discipline of planning and carrying out the movement and maintenance of military forces including storage, distribution, maintenance and transportation of materiel. The NATO Education and Training Network Logistics Module (NETN LOG) is a specification of how to model logistics services in a federated simulation.

The NETN-LOG FOM Module provides a common standard interface for negotiation, delivery, and acceptance of logistics services between federates modelling different entities involved in the service transaction. E.g., simulation of the transport of a unit modelled in another simulator.

NETN-LOG covers the following services:

- Supply Service offered by a federate capable of simulating the transfer of supplies between consumer and provider.
- Transport Service offered by a federate capable of simulating loading, transport and/or unloading of non-consumable materiel.
- Repair Service offered by a federate capable of simulating repair of consumer provided non-consumable materiel, e.g., platforms.

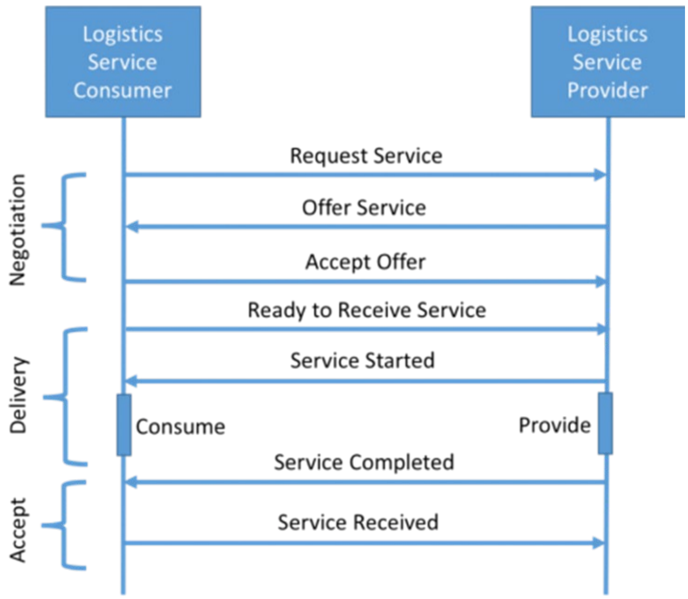

Figure 15: Logistics Service Pattern.

Examples of use:

- Refuelling of aircraft at an airbase or in the air.
- Transport of supplies between facilities.
- Repair of damaged platforms in a facility or by unit.
- Transport of units, platforms, and humans by train, ship, or aircraft.

All NETN LOG services are based on a Logistics Service Pattern that includes negotiation, delivery, and acceptance of logistics services. Federates participating in the logistics service transaction are either a Service Consumer or a Service Provider.

3.3 Transfer of Modelling Responsibilities

In a federated simulation, the participating systems (federates) collectively model the synthetic environment. Allocation of modelling responsibilities are based on individual federate capabilities, federation design agreements, and initial scenario conditions. The responsibility of updating an attribute for a specific simulated entity is allocated to at most one federate. However, during execution, the modelling responsibility may change, and the ownership of attributes can be transferred.

Basic services for the divestiture and acquisition of attribute ownership are provided by HLA. A negotiated and coordinated transfer of modelling responsibilities requires agreements between federates before attribute ownership is transferred.

The NATO Education and Training Network Transfer of Modelling Responsibilities (NETN-TMR) FOM Module is a specification of how to perform a negotiated and coordinated transfer of attribute modelling responsibility between federates in a distributed simulation. It extends the HLA Ownership Management services by providing the means to:

- Negotiate the transfer of ownership.
- Initiate ownership transfer using a Trigger federate.

A transfer of modelling responsibility is performed during runtime, to dynamically change the responsibility to update specific attributes, to a more suitable federate.

For example:

- Transfer from a Live to a Virtual or Constructive simulation.
- Transfer between Virtual and Constructive simulations.
- Transfer between hi- and low-fidelity models.
- Transfer to allow backup, maintenance or load-balancing.
- Transfer of certain attributes to functional models such as movement, damage assessment etc.

NETN-TMR covers the following cases:

- Negotiated acquisition where a federate request to receive the modelling responsibility.
- Divestiture where a federate request another federate to take modelling responsibility.
- Acquisition without negotiation where a federate receives the modelling responsibility.
- Cancellation of transfer.

A negotiated transfer of modelling responsibility between two federates is shown in Figure 16.

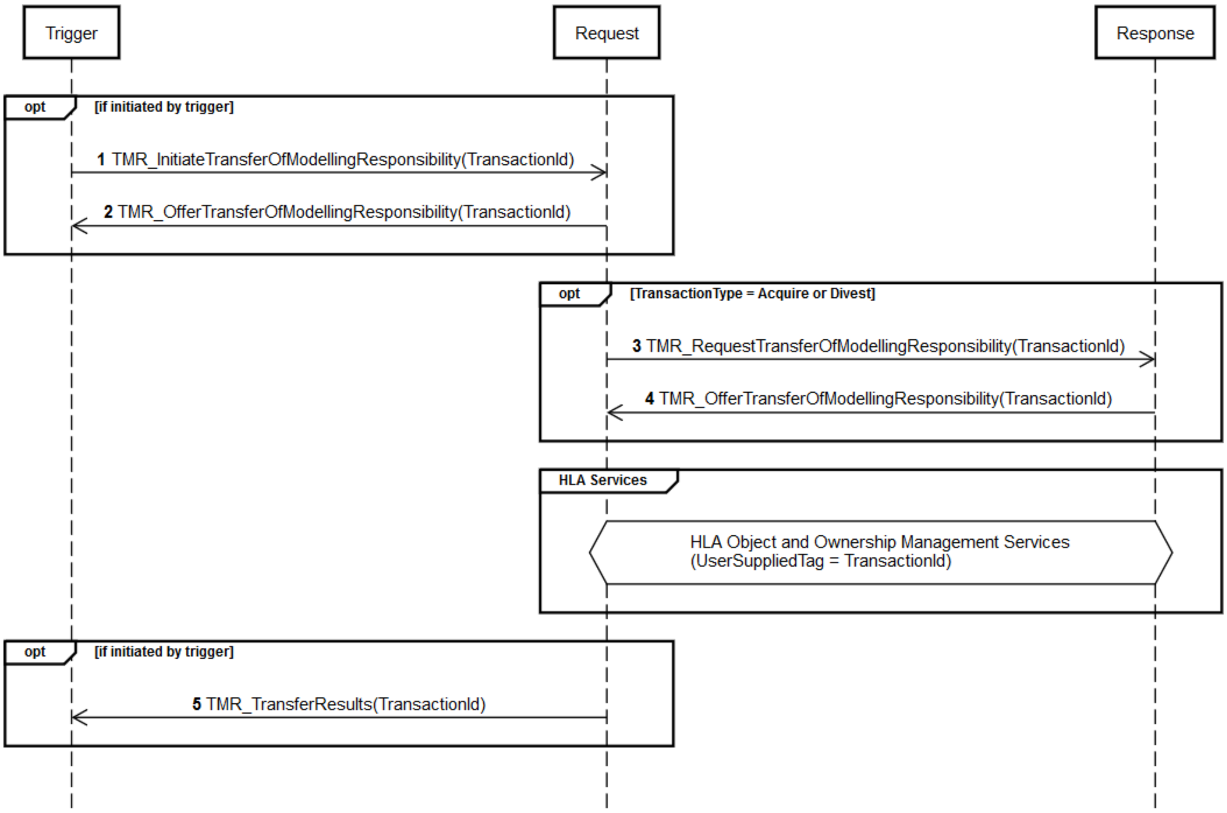

Figure 16: Negotiated Transfer of Modelling Responsibilities between federates.

NETN-TMR uses a combination of HLA interactions and HLA Ownership Management services to negotiate and perform a coordinated transfer of attribute ownership. The pattern includes a triggering interaction (optional) to initiate a transfer, and interactions for requesting, offering, cancelling and sending results of a completed transfer.

3.4 Multi-Resolution Modelling

Models of real-world objects, processes and phenomena are used to create a synthetic representation suitable for simulation. Depending on the purpose and requirements of the simulation, the models can have different levels of resolution and aggregation can be used to create representations of larger combined concepts.

The NATO Education and Training Network Multi-Resolution Modelling (NETN-MRM) FOM Module is a specification of how to perform negotiated and coordinated aggregation and disaggregation of models representing organizational units (aggregate level) and individual entities (entity level, e.g., platforms), in a federated simulation. See Figure 17.

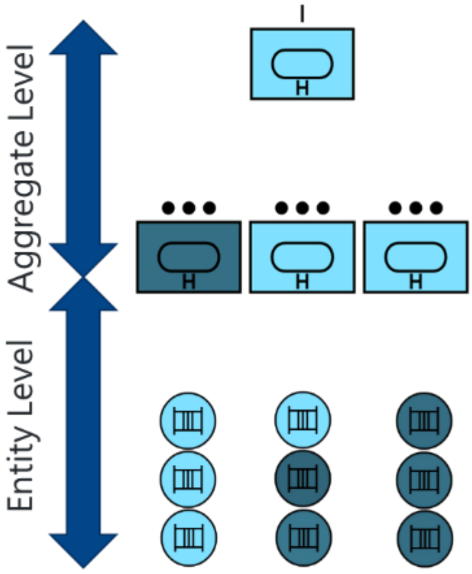

Figure 17: Aggregate and Entity level simulation.

The purpose of NETN-MRM is to support federations where models are represented at multiple levels of resolution and where the level of resolution can change dynamically during a simulation.

For example:

- Disaggregation of a Battalion represented as an Aggregate Entity into Company level Aggregate Entities.
- Disaggregation of a Company to individual platforms such as vehicles and individual soldiers represented at an entity level.
- Aggregation of platforms represented as individual entities to an attribute of an aggregate unit representing e.g., a Platoon.
- Triggering of Aggregation by user command.
- Triggering of Disaggregation based on geo-fencing.

NETN-MRM covers the following cases:

- Disaggregation of an Aggregate Entity into lower-level Aggregate Entities.
- Disaggregation of an Aggregate Entity into Platforms.
- Aggregation of Aggregate Entities into a higher-level Aggregate Entity.
- Aggregation of Platforms into an Aggregate Entity.
- Triggering of Disaggregation.
- Triggering of Aggregation.

3.5 Entity Tasking and Reporting

The NETN-ETR module is a specification of how to represent common low-level tasks that can easily be interpreted and executed by simulators that model the behaviour of entities. It also defines a set of reports to provide status information, including the status of the tasks being executed by simulated entities.

The NETN ETR FOM module is simulation oriented and focuses on tasks with a fine granularity:

- It enables the transformation of command and control messages into tasks that can be executed by a simulator.
- It defines status reports that can be used for producing command and control reports needed for decision making.
- It supports the modelling of simulated command and control interactions between federates in a distributed simulation, for example during an MRM disaggregation process.
- It contains a comprehensive set of tasks and reports that can easily be interpreted and executed by simulators.
- It reflects the capabilities commonly found in COTS Computer Generated Forces (CGF) tools, but it is independent of a specific COTS CGF tool, agent framework, or agent modelling paradigm.
- It is independent of any specific doctrine or tactics.

An entity in ETR can be either a physical entity (e.g., platform or lifeform) or an aggregate entity. If a task or report relates to only a physical entity or to only an aggregate entity, then this is specified in the definition of the task. In the definition of each task, it is not specified how an entity (physical or aggregate) will / should perform the task.

The main benefits from using NETN-ETR is the ability to separate applications that provide tasking from the actual simulation of the task by a federate currently having the modelling responsibility of the entity. Sending tasking information to the federation instead to a specific application allows separation of simulation user interfaces from the simulation application and potentially allowing multiple simulation systems to be controlled from a single common simulation GUI, see Figure 18.

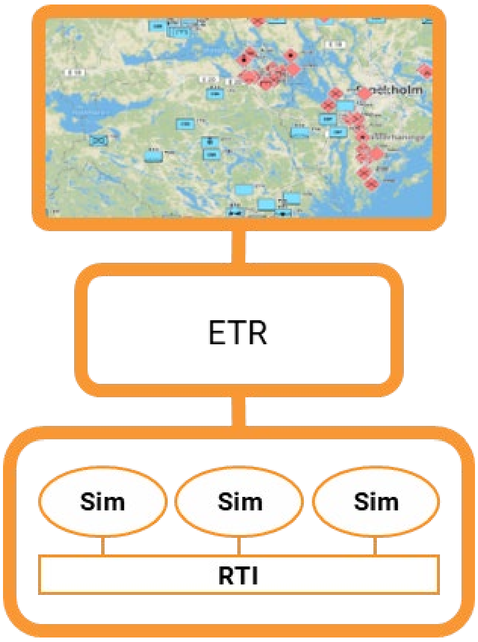

Figure 18: Simulation control via a common entity tasking and reporting.

NETN-ETR was originally developed by FFI, Norway and TNO, The Netherlands and have successfully been used to support exercises such as Viking.

3.6 CBRN Modelling and Simulation

CBRN is Chemical, Biological, Radiological and Nuclear materials that can be delivered intentionally as a weapon using conventional bombs, explosive materials and enhanced blast weapons (e.g., dirty bombs) or unintentionally caused by human error or natural or technological reasons, such as spills, accidental releases or leakages.

The NATO Education and Training Network CBRN Module (NETN-CBRN) is a specification of how to model CBRN related concepts in a federated simulation. NETN-CBRN provides a common standard interface for the representation of CBRN release, detection, effects, and protective measures in a federated simulation. E.g., the exposure effect on individual humans in a CBRN contaminated Hazard area where the human is represented in one simulation and the effect is modelled in another federate simulation.

The NETN-CBRN FOM module covers:

- CBRN Source release modelling.
- CBRN Detector modelling.
- CBRN Effects modelling.
- CBRN Protective measures modelling.
- Hazard area modelling.

Meteorological conditions and CBRN material properties for modelling the dispersion of CBRN material are not explicitly represented in the NETN-CBRN FOM Module. NETN-METOC FOM module can be used to model weather conditions that may impact the dispersion of CBRN materials and cause dynamic change to hazard areas.

3.7 Representation of Weather

The purpose of the NETN METOC module is to provide a standard way to exchange data related to weather conditions and primary effects of weather on terrain, on water surfaces, in the atmosphere and subsurface water conditions. The main objective is to provide a reference model that represents a core common subset of METOC related aspects and to allow extension of the module to incorporate additional detail if required. Therefore, the NETN METOC module shall be viewed as a reference FOM module where extensions are not only allowed but encouraged to fully meet federation specific requirements. However, any extension should also be considered as candidates for improving the NETN METOC module or candidates for new standard NETN modules.

Current weather conditions impact simulations such as platforms and sensors on the ground, on the sea, underwater and in the air. In a federated simulation a correlated representation of these conditions is key to meet interoperability and model requirements. Different simulations require different fidelity of weather conditions concerning data resolution and accuracy.

The NETN METOC focus on the representation of weather conditions related to surfaces and layers. The main difference is that a surface condition does not have a volume and only represents the conditions directly related to the surface of a piece of terrain or water. The layer conditions represent a volume of water or air and are specified with height/depth from surface and layer thickness. Both concepts are also geographically positioned by reference to other concepts shared in the federation such as the position of objects, areal objects or reference to terrain features such as roads etc.

Based on these concepts different levels of fidelity in representing weather conditions can be achieved. Global conditions can be expressed as well as highly detailed conditions e.g., surrounding a specific aircraft.

The aspects and attributes of weather conditions included in the scope of the NETN METOC module are based on input from several sources and are designed to cover the most common levels of representation required by a large set of existing simulators.

- **Atmospheric Layer Conditions cover the following aspects:** Temperature, Winds, Precipitation, Haze, Humidity, Barometric Pressure, Visibility Range and Clouds.
- **Water Layer Conditions cover the following aspects:** Temperature, Salinity and Currents.
- **Land Surface Conditions cover the following aspects:** Temperature, Winds, Precipitation, Haze, Humidity, Barometric Pressure, Visibility Range, Snow Condition, Moisture and Ice Condition.
- **Water Surface Conditions cover the following aspects:** Temperature, Winds, Precipitation, Haze, Humidity, Barometric Pressure, Visibility Range, Sea State, Salinity, Tide, Ice Conditions, Currents, Waves and Swell.

3.8 Vessel Identification and Data

The Automatic Identification System (AIS) is a world-wide automated tracking system used on vessels and by Vessel Traffic Services for identifying and locating vessels by electronically exchanging messages with other nearby vessels and VTS stations. The purpose of the NETN-AIS FOM module is to represent vessel traffic in a simulation using AIS messages, enable the exchange of AIS in both a real-time and non-real time platform level simulation, and allow HLA federate applications to use regular HLA interaction classes and parameters to represent vessel information rather than using the physical message format in ITU-R M.1371-5. However, the FOM is aligned well with this specification, enabling relatively easy mapping to/from this specification.

An example of an AIS message transmission is provided in Figure 19, where the CGF simulates the individual vessels and periodically transmits Entity State data, the AIS Transmitter simulates the vessel's AIS transmitter and generates AIS interactions, and where the AIS Base Station simulates a land-side AIS station forwarding AIS information as NMEA-0183 formatted messages to connected C2 systems. Obviously, a CGF may generate AIS interactions also directly, if it supports that. There may also be multiple AIS Base Stations, each covering a certain area in the simulation.

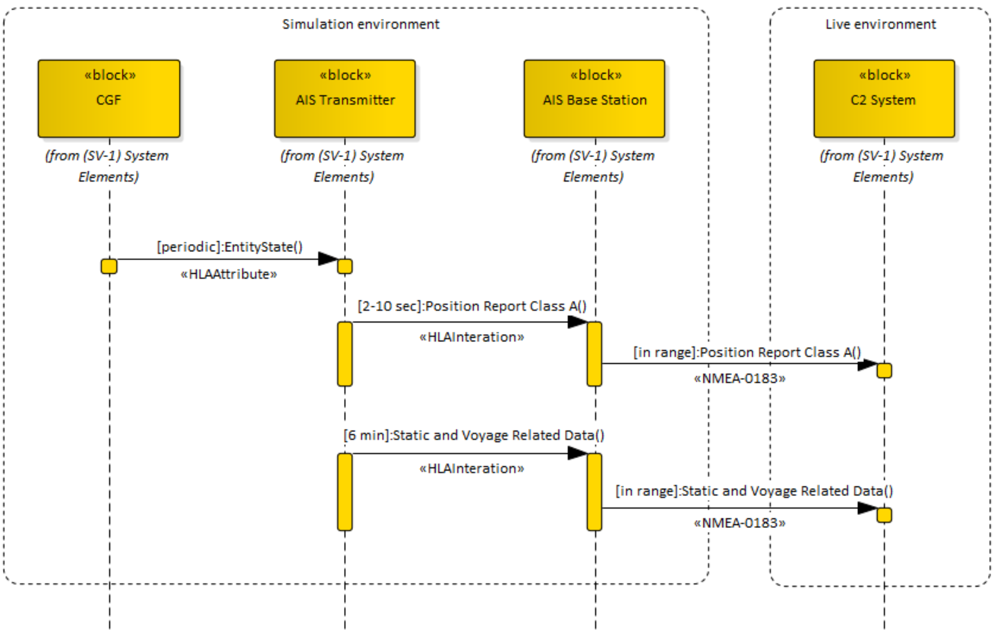

Figure 19: AIS message transmission event trace example.

4.0 DISTRIBUTED SIMULATION ENGINEERING AND EXECUTION PROCESS

As distributed simulations become more complex, and tend to be systems in their own right, a structured systems engineering approach is needed to develop and maintain them. Although traditional software development processes may be applied to the development of distributed simulation environments, these processes lack simulation specific steps and activities that are important for distributed simulation environments. For example, the development of a simulation conceptual model and simulation scenario, and the development of a simulation data exchange model with associated operating agreements between member applications. The only recognized industry standard process for distributed simulation environment development is described in IEEE 1730-2010 Distributed Simulation Engineering and Execution Process (DSEEP) [8]. This process is independent of a particular simulation environment architecture (for instance, HLA) and provides a consistent approach for objectives definition, conceptual analysis, design and development, integration and test, simulation execution, and finally data analysis.

The DSEEP was originally developed under the umbrella of the Simulation Interoperability Standards Organization (SISO) by a large community of (distributed) simulation practitioners, and became an IEEE standard in 2010. A top-level illustration of this process is provided in Figure 20.

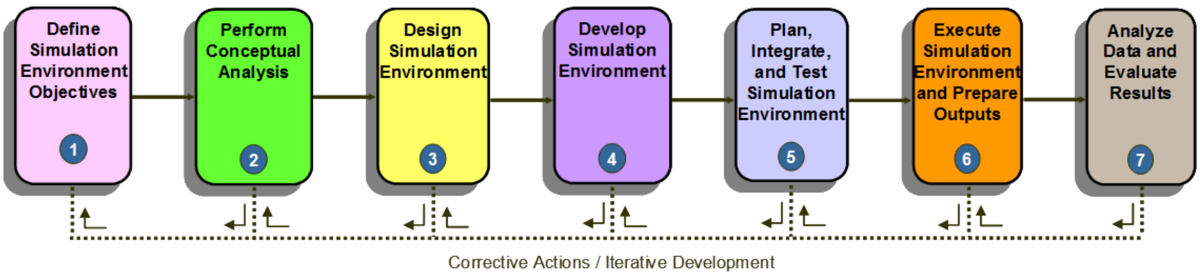

Figure 20: DSEEP process model.

The DSEEP identifies a sequence of seven basic steps with activities to design, develop, integrate, and test a distributed simulation environment of disparate simulation models. Each activity in the DSEEP is further broken down in tasks and work products. A brief summary of each step of the DSEEP is provided below. The guidance provided by the DSEEP is generally applicable to standalone simulations as well. For more information the reader is referred to the standard itself.

The DSEEP steps are:

- **Step 1: Define simulation environment objectives.** Define and document a set of needs that are to be addressed through the development and execution of a simulation environment and transform these needs into a more detailed list of specific objectives for that environment.
- **Step 2: Perform conceptual analysis.** Develop an appropriate representation of the real-world domain that applies to the defined problem space and develop the appropriate scenario. It is also in this step that the objectives for the simulation environment are transformed into a set of simulation environment requirements that will be used for simulation environment design, development, testing, execution, and evaluation.
- **Step 3: Design simulation environment.** Produce the design of the simulation environment that will be implemented in Step 4. This involves identifying member applications that will assume some defined role in the simulation environment (in HLA these are called federates) that are suitable for reuse, creating new member applications if required, allocating the required functionality to the member application representatives.

- **Step 4: Develop simulation environment.** Define the information that will be exchanged at runtime during the execution of the simulation environment (for instance, the HLA FOM), establish interface agreements, modify existing or develop new member applications (for instance, HLA federates) if necessary, and prepare the simulation environment for integration and test.
- **Step 5: Integrate and test simulation environment.** Integration activities are performed, and testing is conducted to verify that interoperability requirements are being met.
- **Step 6: Execute simulation.** The simulation is executed and the output data from the execution is pre-processed.
- **Step 7: Analyze data and evaluate results.** The output data from the execution is analyzed and evaluated, and results are reported back to the user/sponsor.

The standard also includes a number of "overlays" for existing distributed simulation environment architectures such as DIS and HLA. Other overlays can be found in publications, such as a DoDAF/NAF overlay for simulation architecture development [9].

5.0 FEDERATION ENGINEERING AGREEMENTS TEMPLATE

The SISO-STD-012 Federation Engineering Agreements Template (FEAT) provides a template for capturing the results of the various steps within the DSEEP, such as objectives, scenario information, conceptual model, requirements, HLA FOMs and SOMs, and HLA RTI middleware versions. The template is defined as an XML schema (see Figure 21) with the idea to enable automated processing of FEAT compliant documents. In practice however textual formats (Word documents) are used to capture agreements, where possible in line with the content topics of the template.

The content topics in the template are decomposed into eight categories:

- **Metadata –** Information about the federation agreements document itself.
- **Design –** Agreements about the basic purpose and design of the federation.

  For instance, objectives, conceptual model and conceptual scenario, simulation environment requirements, identification of federates, allocation of modeling responsibilities to federates, security agreements.
- **Execution –** Technical and process agreements affecting execution of the federation.

  For instance, execution control and execution states, RPR Simulation Management messages, HLA time management, RTI join and resign agreements.
- **Management –** Systems/software engineering and project management agreements.
- **Data –** Agreements about structure, values, and semantics of data to be exchanged during federation execution.

  For instance, HLA FOM modules, publication and subscription agreements, update agreements, HLA object and federate naming conventions.
- **Infrastructure –** Technical agreements about hardware, software, network protocols, and processes for implementing the infrastructure to support federation execution.

  For instance, network sites and addresses, HLA federates per site, HLA-RTI middleware agreements.
- **Modeling –** Agreements to be implemented in the member applications that semantically affect the current execution of the federation.

  For instance, effects adjudication, coordinate systems, dead reckoning, aggregation/de-aggregation, transfer of control and modeling responsibility.

- **Variances –** Exceptions to the federation agreements deemed necessary during integration and testing.

  For instance, not supported message exchanges by certain federates.

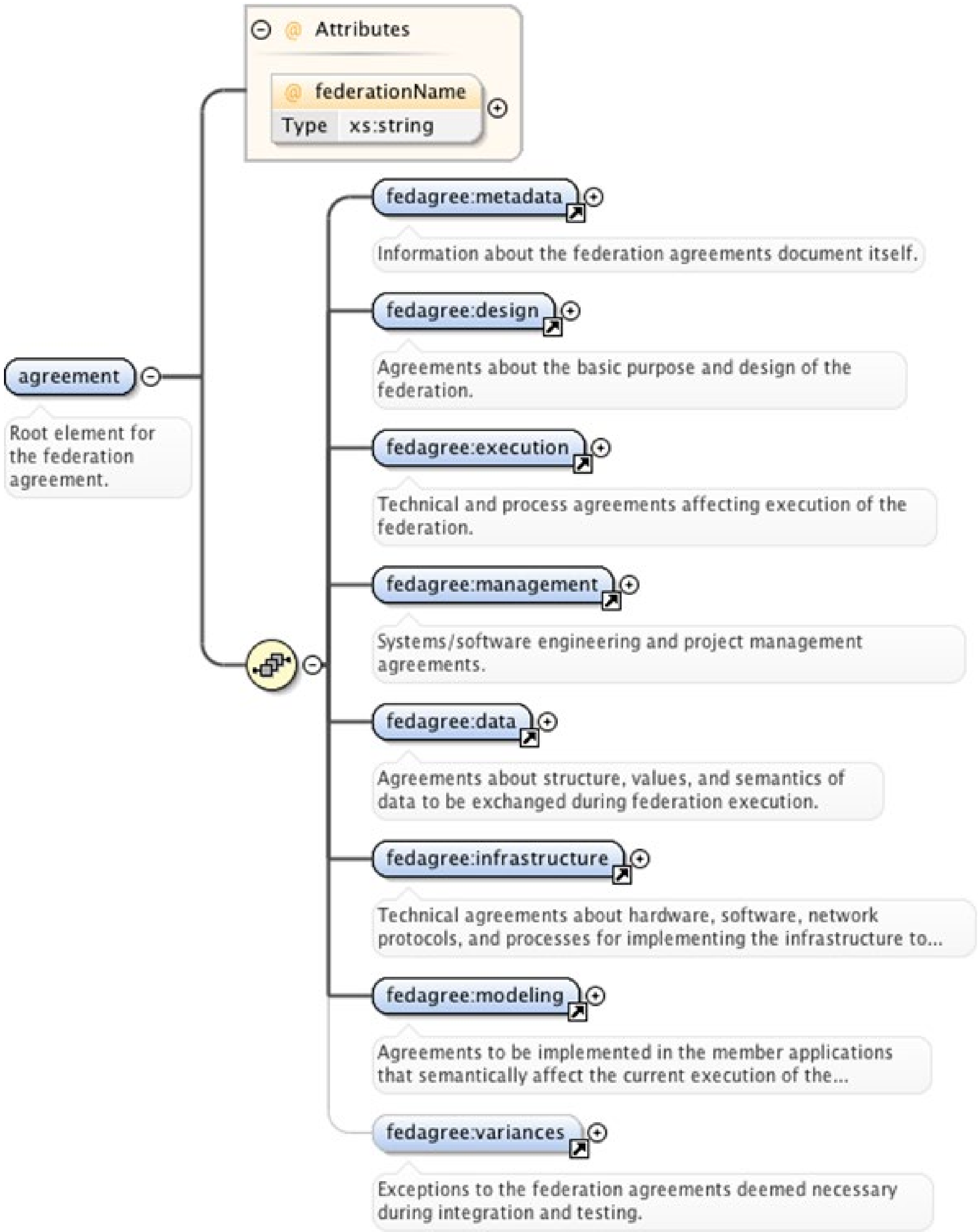

Figure 21: Top level FEAT XML schema.

6.0 SUMMARY

NATO Standards for Federated Simulation as well as processes and tools for simulation engineering are important to NATO in order to successfully connect and integrate multi-national M&S assets. The NATO Education and Training Network Federation Architecture and FOM Design (NETN FAFD) is a NATO M&S interoperability standard (STANREC 4800), based on the use of IEEE 1516 HLA and SISO-STD-001 Real-Time Platform Reference FOM. This standard was initially released in 2011 and has evolved based on operational use by NATO and nations to its current version NETN FOM v3.0. The NETN FOM modules are recommended for use when implementing a NETN FAFD compliant distributed simulation. Several NETN FOM modules of this standard have been used in exercises such as the Viking series and CWIX 2022/2023. The NETN FOM is managed as open-source project on GitHub, enabling the user community to engage the developers and providing early access to releases. The latest release of NETN FOM (3.0) is available on GitHub [7].

7.0 FURTHER READING

The provided references provide more information on the topics in these papers. Other directions worth to explore further are:

- AMSP-03 Edition B "NATO Reference Architecture for Distributed Synthetic Training" provides a common architectural approach for designing, developing and implementing a Distributed Synthetic Training (DST) environment. AMSP-03 defines architecture building blocks and patterns for such an environment, and references amongst others IEEE 1516 HLA and AMSP-04 NETN FAFD as two recommended standards for federating simulations. AMSP-03 thighs together several M&S standards and is also used as source of reference for the NATO FMN M&S activities. AMSP-03 is covered by STANREC 4799. For AMSP-03 see the NMSG website [10] or the NATO Standardization Office [11].
- NATO Research Task Group MSG-163 made significant contributions to AMSP-04 Edition A and NETN FOM version 2.0, updating these to AMSP-04 Edition B and NETN FOM version 3.0. Other contributions were to the NATO IVCT, and to the NATO HLA Certification Service CONOPS. More about these topics can be read in the MSG-163 final report [12], downloadable from [13] (search for MSG-163). AMSP-04 Edition B is updated to edition C by NATO Research Task Group MSG-191, expected to be completed in 2024.

ACKNOWLEDGEMENTS

The author would like to acknowledge the contribution of the authors of SISO paper "Evolution of NATO standards for federated simulation" [14] in relation to the NETN part of this paper.

8.0 REFERENCES

[1] "NATO Modelling and simulation Master Plan – Strategic Plan," AC/323/NMSG(2012)-015, 14 September 2012.

[2] "NATO Science and Technology Organisation," [Online]. Available: https://www.sto.nato.int

[3] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules (IEEE 1516-2010), IEEE, 2010.

[4] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification (IEEE 1516.1-2010), IEEE, 2010.

[5] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (IEEE 1516.2-2010), IEEE, 2010.

[6] SISO, Standard for Guidance, Rationale, and Interoperability Modalities (GRIM) for the Real-time Platform Reference Federation Object Model (RPR FOM), Version 2.0 (SISO-STD-001-2015), SISO, 2015.

[7] "NATO Distributed Simulation Federation Object Model (NETN FOM)," 2023. [Online]. Available: https://github.com/AMSP-04

[8] IEEE, IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP) (IEEE 1730-2010), IEEE, 2010.

[9] Berg, "Simulation environment architecture development using the DoDAF," in SISO 15F-SIW-019, 2015.

[10] "NMSG," [Online]. Available: https://nmsg.sto.nato.int/amsp/amsp03

[11] [Online]. Available: https://nso.nato.int

[12] "STO-TR-MSG-163 Evolution of NATO Standards for Federated Simulation," NATO STO, 2022.

[13] [Online]. Available: https://www.sto.nato.int/publications/Pages/default.aspx

[14] T. Berg, van den, B. Lofstrand, R. Herzog, T. Kuhn and H. Behner, "Evolution of NATO standards for federated simulation (SISO-20W-SIW-025)," in SISO Innovation Workshop, 2020.