The NATO MSG-136 Reference Architecture for M&S as a Service

Jo Erskine Hannay
Norwegian Defence Research Establishment (FFI)
P.O. Box 25, NO-2027 Kjeller, NORWAY
jo.hannay@ffi.no

Tom van den Berg
Netherlands Organisation for Applied Scientific Research (TNO)
P.O. Box 96864, NL-2509 JG The Hague, The Netherlands
tom.vandenberg@tno.nl

ABSTRACT

The Allied Framework for Modeling and Simulation (M&S) as a Service (MSaaS) is proposed by NATO MSG-136 as a permanent service- and cloud-based M&S ecosystem for use by NATO and partner nations. The framework is designed to aid stakeholders to utilize state-of-the-art service-oriented and cloud-based methodology and technology to achieve interoperability between participating simulation systems and ensuring credibility of results. This paper presents the reference architecture that is currently developed as part of the technical concept for the Allied Framework for MSaaS. The reference architecture is structured on The Open Group SOA Reference Architecture, while its contents is supplied via the NATO C3 Taxonomy in the form of architecture building blocks and architecture patterns. The MSaaS Reference Architecture is not a final product. It will change over time as new architecture building blocks and patterns are identified and added, and existing ones modified and improved.

1.0 INTRODUCTION

NATO and the nations use distributed simulation environments for various purposes, such as training, mission rehearsal and decision support in acquisition processes. Consequently, Modeling and Simulation (M&S) has become a critical technology for the coalition and its nations. However, achieving interoperability between participating simulation systems and ensuring credibility of results still requires large expenditures with regards to time, personnel and budget.

Recent technical development in cloud computing technology and service-oriented architecture (SOA) offers opportunities to utilize M&S capabilities better in order to satisfy NATO critical needs. A new concept that includes service orientation and the provision of M&S applications via the as-a-service model of cloud computing may enable composable simulation environments that can be deployed rapidly and on-demand. This new concept is known as M&S as a Service (MSaaS).

NATO MSG-136 *MSaaS – Rapid deployment of interoperable and credible simulation environments* investigates MSaaS with the aim of providing the technical and organizational foundations for a future permanent service-based M&S ecosystem within NATO and partner nations, called the *Allied Framework for MSaaS*. This paper presents the NMSG-136 MSaaS Reference Architecture, which is part of the technical concept for the Allied Framework for MSaaS.

2.0 OPERATIONAL CONTEXT

Imagine the following situation: A maritime task group has been deployed to the Persian Gulf region for a maritime interdiction mission. Surface track managers need to rehearse coordination in the task group and wish to use a simulation to prepare themselves for the mission. As the task group is already en route to their planned desitination, the track managers will use their on-board training facilities for the rehearseal.

A Simulation Operator is tasked to create the required simulation. The Simulation Operator uses a portal that offers online tools for composing a simulation at the conceptual level rapidly from pieces of specified simulation functionality offered as simulation services discovered in a cloud-based NATO M&S Repository. The composition is then assembled in terms of available and suitable service implementations located automatically by way of a cloud-based NATO M&S Registry. The assembly is deployed and executed in a secure Mission cloud that provides scalable resources at the appropriate time and place to enable the track managers to interoperate with the simulation from their vessels. The simulation is

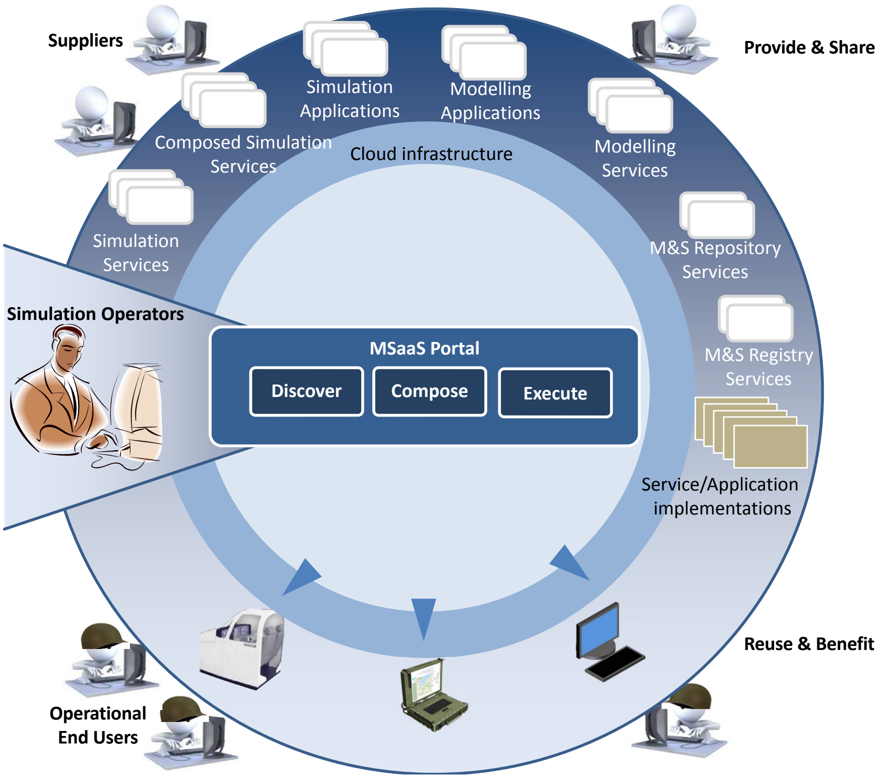

Figure 1: Allied Framework for MSaaS

presented through front-end apps uploaded on the vessels' command and control (C2) systems. Other services, such as map, terrain, weather and Automatic Identification System (AIS) services are called on demand and may be running form other cloud-based locations. The composition itself can be packaged as a composed simulation service and stored for later reuse or adaptation.

The Allied Framework for MSaaS (Figure 1) is envisioned to support the rapid demand for tailored M&S at varying places and time as illustrated above. The Allied Framework for MSaaS should directly support its *Suppliers* and *Simulation Operators*:

Suppliers are enterprise and solution architects, systems designers and developers who wish to

- develop simulation functionality to be offered through back-end *Simulation Services* or compositions of *Simulation Services* offered through back-end *Composed Simulation Services* (Figure 1);
- develop front-end functionality to be offered through loosely coupled, light-weight (including web-based and mobile device-based) *Simulation Applications* (Apps) (Figure 1) which can be combined to give users access to *Simulation Services* and *Composed Simulation Services* from operational systems or simulation front-ends;
- develop Framework functionality offered through back-end *Modelling Services* that support the MSaaS Portal's Discover, Compose and Execute activities (Figure 1);
- develop front-end functionality to be offered through loosely coupled, light-weight *Modelling Applications* (Apps) (Figure 1) which can be combined into front-ends of the MSaaS Portal to give users access to *Modelling Services*;
- share *Services* and *Applications* by deploying their metadata using the *M&S Repository Services* and by deploying their implementation binding information using the *M&S Registry Services* (Figure 1).

Simulation Operators are semi-technical personnel who wish to discover, compose and execute *Simulation Services* and *Composed Simulation Services* for the *Operational End-Users* by using the MSaaS Portal

(Figure 1). In the future, it is envisioned that higher levels of interoperability [1] will allow automatic discovery, composition and execution to larger extents, so that *Simulation Operators* may, in time, be non-technical operational personnel.

As is apparent from the above, both *Suppliers* and *Simulation Operators* compose simulations from *Simulation Services*. At present, *Simulation Operators* might hand over their compositions to the *Suppliers* who will form the composition as a "simulation as a service". In the future, *Simulation Operators* and even operational personnel might be able to do this without the aid of technical personnel.

The MSaaS Reference Architecture gives implementation-independent blueprints for developing and using the Allied Framework for MSaaS M&S capabilities; i.e., its services and applications as described above. In the next section, we will introduce architecture concepts necessary for defining the reference architecture.

3.0 ARCHITECTURE CONCEPTS

An *architecture* of a system or of a federation of systems is, according to [2]: "the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution". Thus, an architecture provides plans or blueprints for a system.

Architectures can be designed at various levels of abstraction. There is little consensus in general on the various levels of abstraction or on how to name them, but we declare the notions that are relevant for our discussion; see Figure 2. Two notions are central: *architecture building block* (ABB) and *architecture pattern* (AP) [3]. ABBs are the elements that constitute an architecture, and each ABB should have attributes that specify its function. APs are high-level suggestions for ways of combining ABBs.

Figure 2 shows several levels of abstraction for architectures. Following [4], at the very high level, an *architecture ontology* might declare *types* of ABB and AP. For example, [3] declares ABB types, such as '(business) process', 'service', 'repository', 'service container'; and AP types, such as 'consumer pattern', 'service invocation pattern', that are pertinent for any SOA.

Next, actual ABBs and APs of the various ABB types and AP types can be used for declaring a domain-specific *overarching architecture*. The manner in which ABBs and APs are specified might be standardized. For example an ABB representing a service would be of type 'service' and its specification may follow some standard for service specification.

Then, a reference architecture is designed by composing ABBs guided by APs from the overarching architecture. In addition, an *architecture topology* (or several) should be designed at the reference architecture level to delineate intended systems boundaries and the boundaries in which interoperability standards are enforced. From a reference architecture, individual *solution architectures* (also called *target architectures*; see [5]) that specify solution implementations may be derived. There should be methods for refining architectures at one abstraction level to the next [6], [7], [8]. The spectrum of architecture abstraction levels and such methods are what we here refer to as an *architecture framework* (Figure 2).

Although, the notion of 'architecture framework' is not consistently defined, key points include that such frameworks are ontology based, open and extensible [9], and that they provide "conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders" [2]. Various frameworks also cover different aspects of architecting [10]; for example, the NATO Architecture Framework (NAF) [5] is a view-based description framework for architecture and The Open Group Architecture Framework (TOGAF) [8] emphasizes architecture governance and the transformation of one type of architecture into another.

It should also be noted that notions of 'overarching architecture', 'reference architecture' and 'target architecture' differ. For example, The Open Group SOA Reference Architecture [3] is a generic template with ABBs and APs that are pertinent for any SOA, and is, in our terminology, an architecture ontology providing ABB types and AP types, rather than a reference architecture.

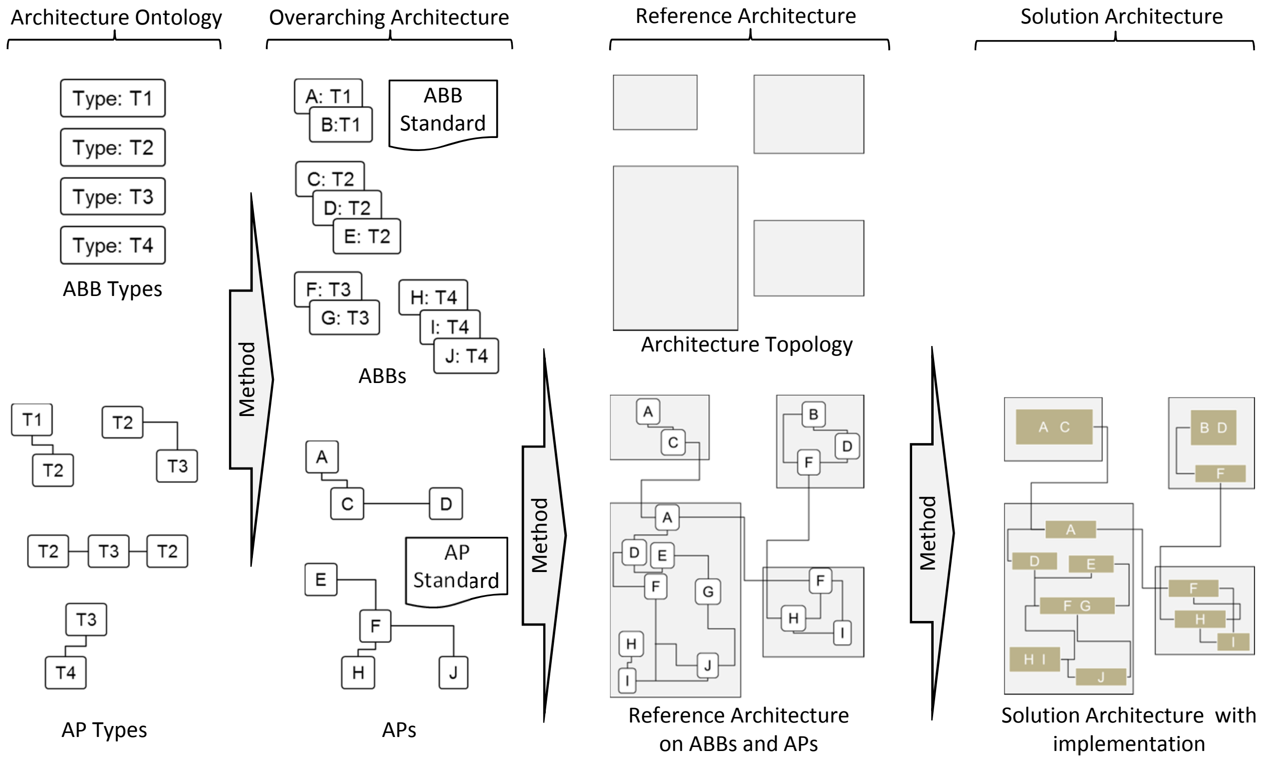

Figure 2: Architecture framework [4]. An architecture ontology provides types of architecture building block (ABB) and architecture pattern (AP). An overarching architecture consists of specific ABBs and APs of various types, with standards for specifying ABBs and APs. Various architecture topologies specifying system and interoperability boundaries aid in designing reference architectures using ABBs and APs. From this, solution architectures with implementation-specific systems (olive) can be designed.

4.0 ARCHITECTURE FRAMEWORK FOR MSAAS

In the following sections, we declare a MSaaS architecture framework in terms of the concepts in the previous section.

The architecture ontology will be The Open Group SOA Reference Architecture [3].

The overarching architecture is obtained by declaring M&S-particular ABBs of the relevant architecture ontology types. The M&S-particular ABBs will be declared in the NATO C3 Taxonomy [11], which is a dynamic library of NATO C3 capabilities.

While the C3 Taxonomy is structured from the viewpoint of C3 capabilities, The Open Group SOA Reference Architecture gives the structure of a SOA. Combining the two gives the necessary architectural structure and functional content for defining MSaaS.

Declaring the M&S-particular portions of this combination gives the MSaaS Reference Architecture together with an architecture topology that delineates the boundaries for the notion of composed simulation services. We now describe this in more detail.

4.1 Architecture Ontology: The Open Group SOA Reference Architecture

The Open Group SOA Reference Architecture (TOGSOARA) gives a structure and generic ABB types that are relevant for any SOA; see Figure 3. The MSaaS Reference Architecture inherits the entire structure but is specific only on ABB types which are pertinent to what is M&S-particular.

Starting from the bottom, the *Operational Systems Layer* gives ABB types concerning the operational

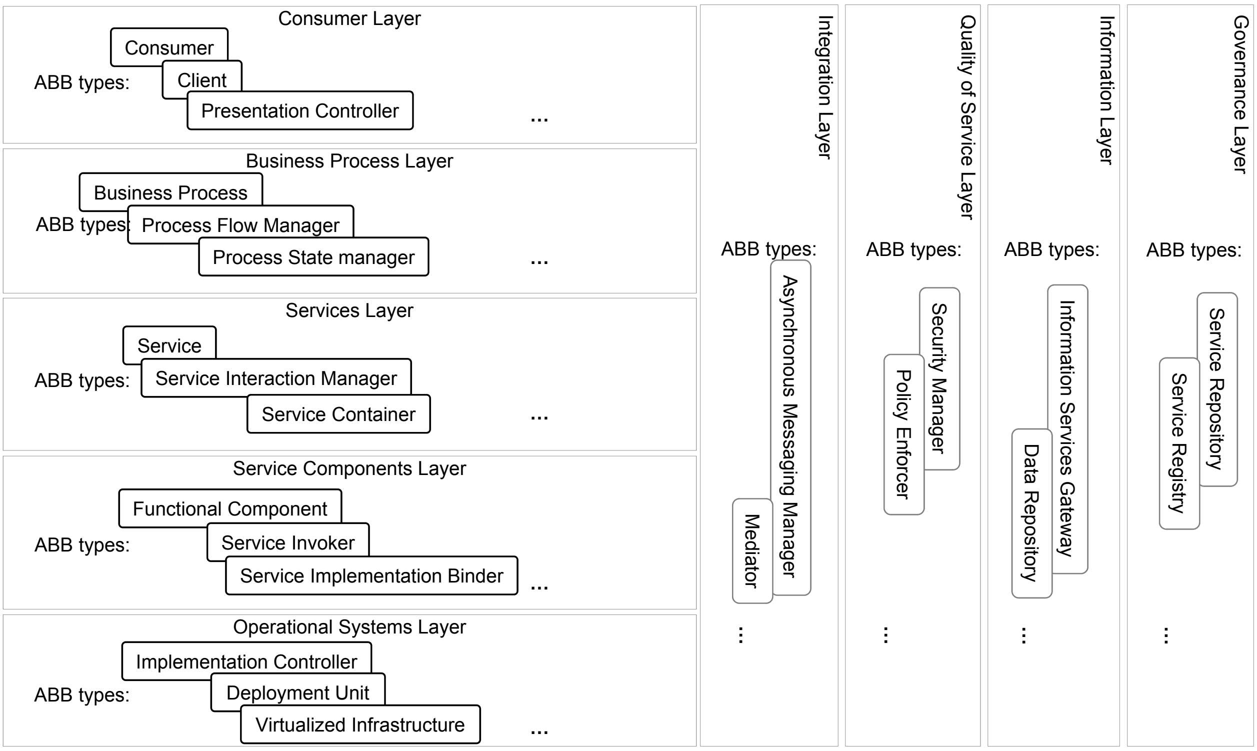

Figure 3: Architecture ontology: The Open Group SOA Rference Architecture [3] layers with Architetcure Building Block (ABB) examples.

run-time capabilities in a SOA (i.e., "operational" in a technical sense); such as Implementation Controller, Deployment Unit and Virtualized Infrastructure. The *Service Components Layer* gives ABB types concerning defining and running services; e.g., Functional Component, Service Invoker and Service Implementation Binder. Both of these layers hold ABB types that are general for any SOA, and at present no M&S-particular ABB types have been identified. Note however, that these layers are what enables the cloud infrastructure that is vital for MSaaS performance and scalability issues [12]. For the time being, it seems sufficient to use generic infrastructure, but M&S-particular ABBs may be deemed necessary here in the future.

The *Services Layer* gives ABB types concerning the topical services of the SOA; such as Service, Service Interaction Manager and Service Container. For MSaaS, ABBs that represent *Simulation Services* and *Modelling Services* (Section 2) are of type Service. To indicate this one might write, e.g., MTB Simulation Service:Service for a particular service ABB that offers a motor torpedo boat (MTB) simulation.

The *Business Processes Layer* holds ABB types which pertain to the functional processes that arise when combing the SOA's topical services; for example Business Process, Process Flow Manager and Process State Manager. Business processes are where the business value – the *raison d' être* of the SOA – manifests itself. For MSaaS, this amounts to combining *Simulation Services* into *Composed Simulation Services* (Section 2), and the business processes are then, in effect, simulations. For the sake of MSaaS, these simulations are themselves packaged as (composed) services. One might write, e.g., Maritime Interdiction Simulation:Business Process for a maritime interdiction simulation service ABB composed from a number of vessel simulation service ABBs.

The *Consumer Layer* holds ABB types pertaining to functionality that allows users (human and technical) to access the SOA, such as Consumer, Client, Presentation Controller and Cache. For MSaaS, the front-end, light-weight *Simulation Applications* and *Modelling Applications* (Section 2) are represented by ABBs of type Presentation Controller.

There are also four cross-cutting layers. The *Integration Layer* holds ABB types which concern integrating ABBs from various layers, the *Quality of Service Layer* concerns availability, reliability, security, and safety in the SOA, the *Information Layer* concerns the management of information shred in the SOA, the *Governance Layer* concerns authoritative rules and management of the SOA. The *M&S Repository Services* and *M&S Registry Services* (Section 2) are represented by ABBs of types, respectively, Service Registry and Service Repository in the Governance Layer.

TOGSOARA gives AP types in terms of the ABB types. These patterns are also generic for any SOA. For example, the Service Invocation pattern describes how the Service Interaction Manager invokes a Service in a Service Container while interacting with the Policy Enforcer (*Quality of Service Layer*), Access Controller (*Quality of Service Layer*), Policy Manager (*Governance Layer*) and Status Manager (*Quality of Service Layer*). This pattern applies to the MSaaS *Services*, but the MSaaS Reference Architecture simply inherits this and other generic SOA patterns, without specifying them in particular for M&S.

4.2 Overarching Architecture: The C3 Taxonomy

While the ABB types and the AP types are provided by TOGSOARA as the architecture ontology, the actual ABBs and APs of the MSaaS Reference Architecture are linked in from the NATO C3 Taxonomy [11]. The C3 Taxonomy is a library for NATO's Consultation, Command and Control (C3) capabilities; see Figure 4 for a high-level view.

The C3 Taxonomy's top-level capabilities are grouped into Missions and Operations, Operational Capabilities, User-Facing Capabilities, and Back-End Capabilities. Each group of capabilities is further decomposed into more refined and detailed levels of capabilities, such as Business Processes (here, in the sense of defence operational processes), User Applications, Community of Interest (COI) Services, Core Services and Communications Services, and so on. Thus, each category (oval box) represents a division into capabilities and is further divided into sub-categories; i.e., sub-capabilities. At the leafs of these capability trees, one finds individual operational processes (under Operational Capabilities), individual user applications (under User-Facing Capabilities) and individual services (under Back-End Capabilities). This capability structure can be viewed and modified through the C3 Taxonomy's Enterprise Management Wiki.

Thus, the C3 taxonomy's capabilities are C3-specific ABBs at various levels of detail. This enables architecture work at various levels of refinement and detail. Figure 5 shows a closer detail of the taxonomy for CIS capabilities; which will be our focus.

M&S is explicitly represented at the User Applications, COI-Specific Services and COI-Enabling Services levels. Here, User Applications are to be understood in the SOA sense as loosely coupled front-end apps that can be put together readily and rapidly for the purpose at hand. However, in the transition to true service orientation, C3 Taxonomy User Applications also include legacy or proprietary monolithic applications, such as legacy C2IS, Battle Management Systems (BMS) and, indeed, many simulation systems. The COI-Specific Services are back-end technical services that are specific to COIs, and the COI-Enabling Services are more generic cross-COI back-end technical services. The following are the M&S-particular capabilities:

- M&S Applications, which are user-facing capabilities containing *Simulation Applications* and *Modelling Applications* ABBs (Section 2) for accessing back-end M&S capabilities
- M&S Services, which are back-end capabilities containing the *Simulation Services*, *Composed Simulation Services* and *Modelling Services* ABBs (Section 2)
- M&S Enabling Services, which are back-end support capabilities pertaining to M&S containing, e.g., *Repository Services* and *Registry Services* ABBs (Section 2)

One might use the "subtype" relation <: to indicate which groups (classes) ABBs belong to: For example, one might write COI Specific Services:>M&S Services:>Simulation Services:>MTB Service:Service for the MTB simulation service exemplified above.

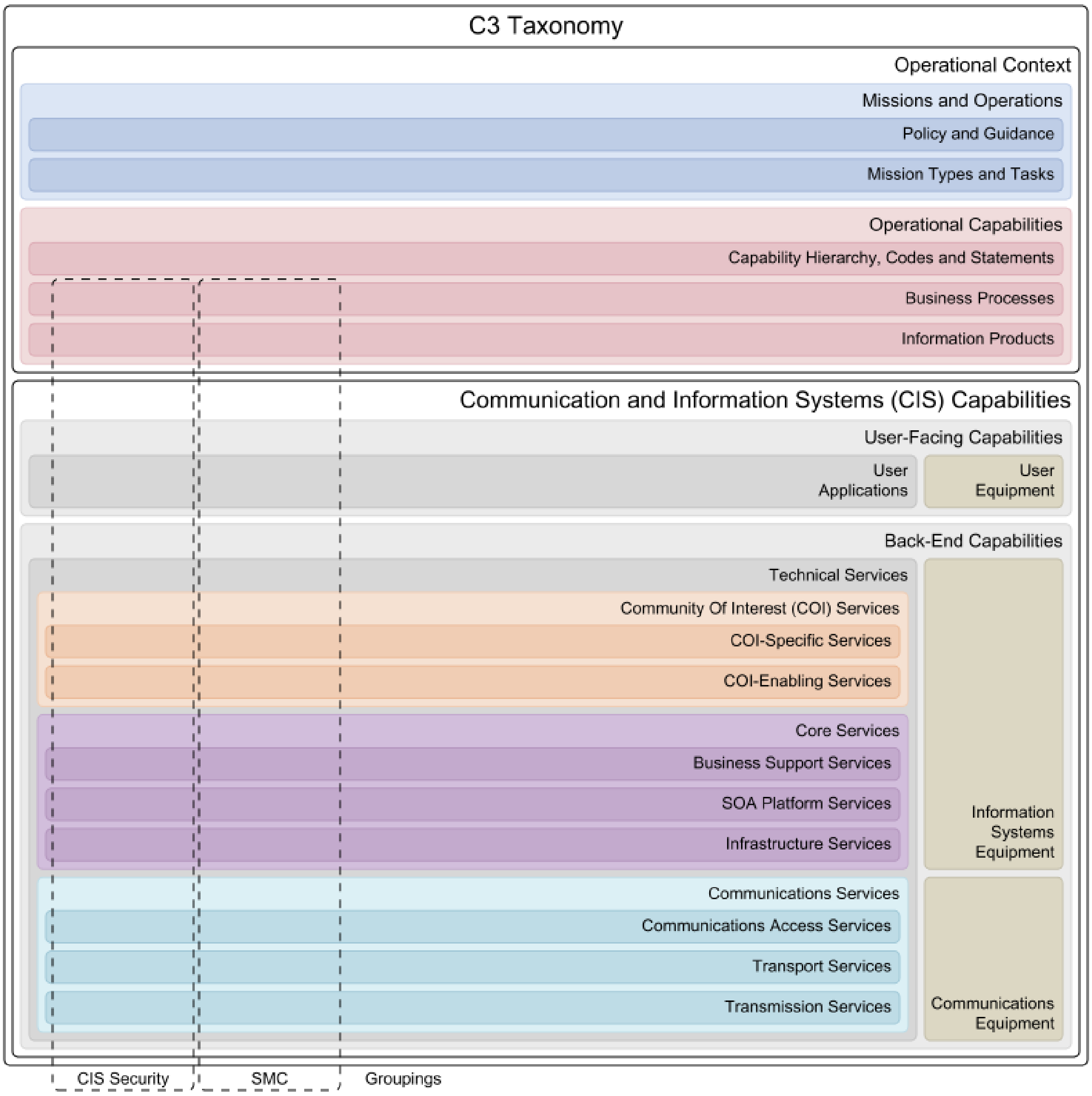

Figure 4: C3 Taxonomy – top-level view [11]

The ABBs are implementation independent, and can be realized in various implementations in software to be deployed at different places at different times. Such implementations are capability (service and application) *providers* in the technical sense, where a piece of software may provide one or more capabilities, and a capability may be provided by one or several pieces of software [13].

An ABB states requirements for development, but it also holds the metadata to be deployed in the Allied Framework for MSaaS's repository. An ABB in the MSaaS Reference Architecture therefore consists of

- an implementation-independent *description* of an M&S capability for the benefit of consumers of that capability, which consists of
  - an *interface* (for syntactic interoperability) [13], and
  - a *contract* (for a degree of semantic interoperability and a specification of contractual non-functional requirements) [13],
- a set of implementation-independent functional and non-functional requirements, for the benefit of developers who will realize the capability in software and hardware.

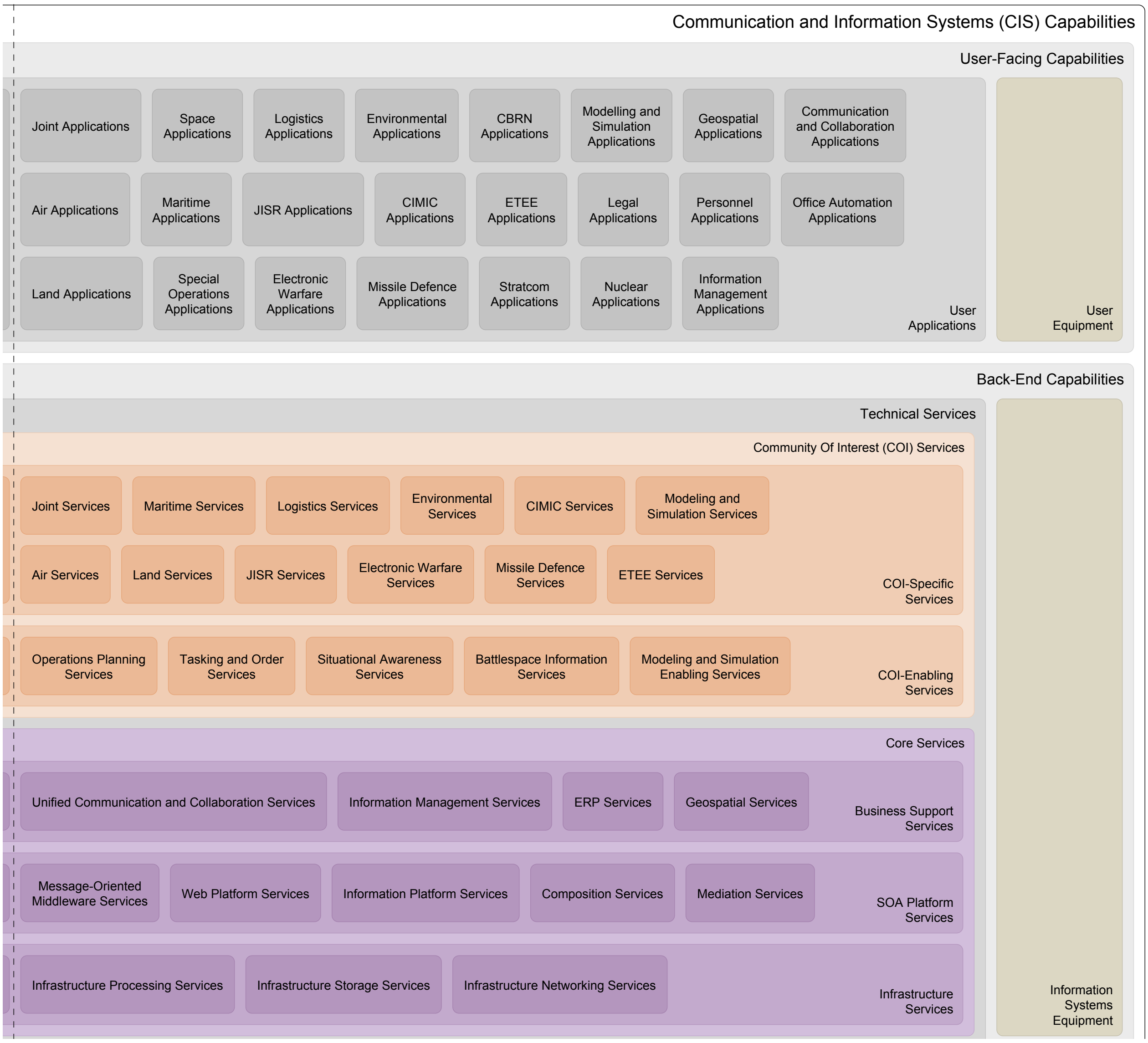

Figure 5: C3 Taxonomy (detail of CIS Capabilities) [11]

The C3 Taxonomy is under construction and will likely be in a state of constant flux to meet changing requirements. This is certainly the case for M&S. Work on the MSaaS Reference Architecture will therefore contribute to the C3 Taxonomy. Figure 6 shows the placement of high-level M&S-particular ABBs in the C3 Taxonomy as a result of the MSaaS architecture work. Several of these ABBs are currently not in the C3 Taxonomy. Using the C3 Taxonomy as a library for ABBs and APs means that the most dynamic part of the MSaaS Reference Architecture is factored out. This also brings the MSaaS Reference Architecture in context with the other C3 capabilities in NATO's systems portfolio. Figure 6 also shows which ABB types the C3 Taxonomy ABBs have, and illustrates the C3 Taxonomy as a library for the MSaaS Reference Architecture.

The C3 Taxonomy has its own layered structure, which has some resemblance to the layers of TOGSOARA, but are nevertheless substantially different. It is important to keep in mind that the MSaaS Reference Architecture retains the structure and layers of TOGSOARA, and simply links dynamically to relevant ABBs in the C3 Taxonomy, which has its own structure.

This document will elaborate on the three main categories above: M&S Applications, M&S Services and M&S

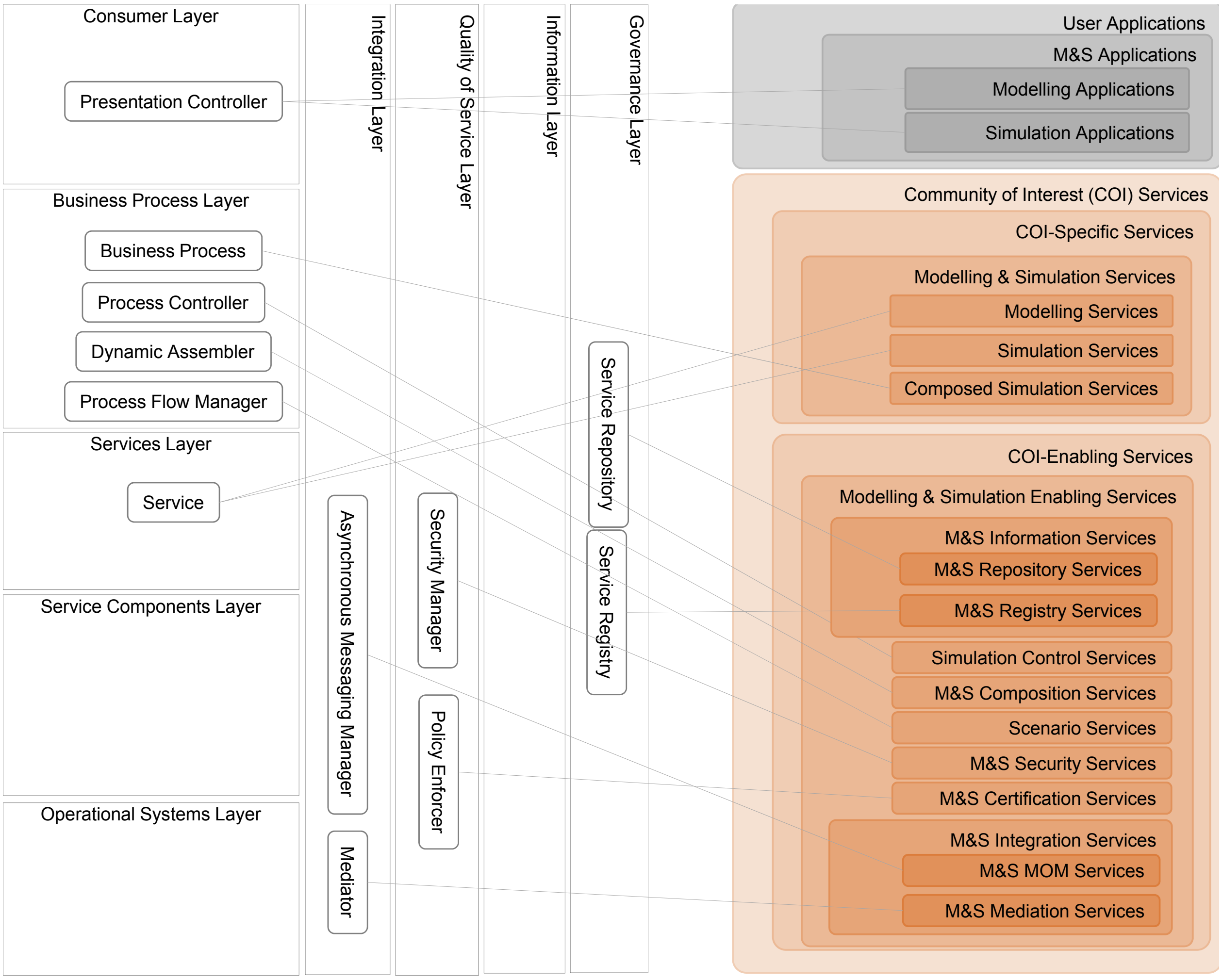

Figure 6: The NATO C3 Taxonomy as a dynamic library for the MSaaS Reference Architecture.

Enabling Services. The elaboration consists of declaring essential main sub categories, and declaring initial services and applications, but will not include a comprehensive list of services and applications, as this is left for the future continuous elaboration of content in the C3 Taxonomy.

4.3 Topology: Composed Simulation Services

The central topological issue for MSaaS is the notion of Composed Simulation Service; what its components are, what it does – and does not – encompass, and what the relevant boundaries of interoperability are.

At the level of ABBs, a Composed Simulation Service is the result of combining Simulation Services, where the combination of services adhere to a *Simulation Environment Agreement* and a *Simulation Data Exchange Model*, and where various simulation protocols may be adhered to in respective *Simulation Protocol Enclaves*. Figure 7 shows a topology that expresses this. Thus, a Composed Simulation Service is the ABB representation of a multi-architecture simulation environment in terms of the Distributed Simulation Engineering and Execution Process (DSEEP) [14] Multi-Architecture Overlay (DMAO) [15]. M&S Message Oriented Middleware (MOM) Services and M&S Mediation Services are used to integrate the Simulation Services.

At the implementation level, MSaaS implies an extension of the DMAO. At present the DMAO identifies enclaves for three simulation architectures: the High Level Architecture (HLA) [16], Distributed Interactive Simulation (DIS) [17] and Test and Training Enabling Architecture (TENA) [18]. MSaaS adds an enclave

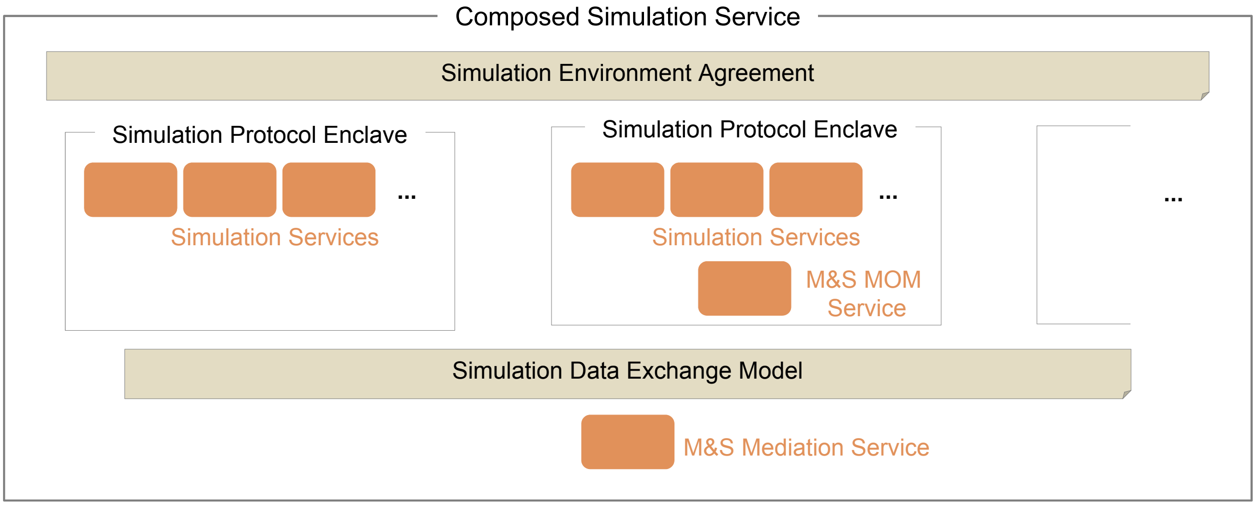

Figure 7: Composed Simulation Service.

for architecture that adheres to industry standard service-oriented technology (ISSOT), such as SOAP web services (WS*) [19], Representational State Transfer (RESTful Web Services) [20], Advanced Message Queuing Protocol (AMQP) [21], WebLVC [22], WebSocket [23], etc. This is illustrated in Figure 8. Member applications in a given enclave relate to that enclave's implementation of the common simulation data exchange model: a Protocol Data Unit (PDU) set for DIS, a Federation Object Model (FOM) for HLA, a Logical Range Object Model (LROM) for TENA, and a set of service interfaces for ISSOT. Thus a multi-architecture simulation environment may be a mix of, say, HLA federate applications and webservices, as long as they all adhere to the common simulation data exchange model and simulation environment agreement.

The MSaaS Reference Architecture implies that simulation environment member applications are to be seen as providers of Simulation Services. The viability of this hinges on several issues, e.g., independence of any particular simulation data exchange model [24] and semantic specification and composability [25], [26]; see [4] for pointers to a broader discussion. Nevertheless, at the level of implementation-independent ABBs, it is possible to write service descriptions for Simulation Services. Such services may then have various implementations according to particular simulation data exchange model and specific

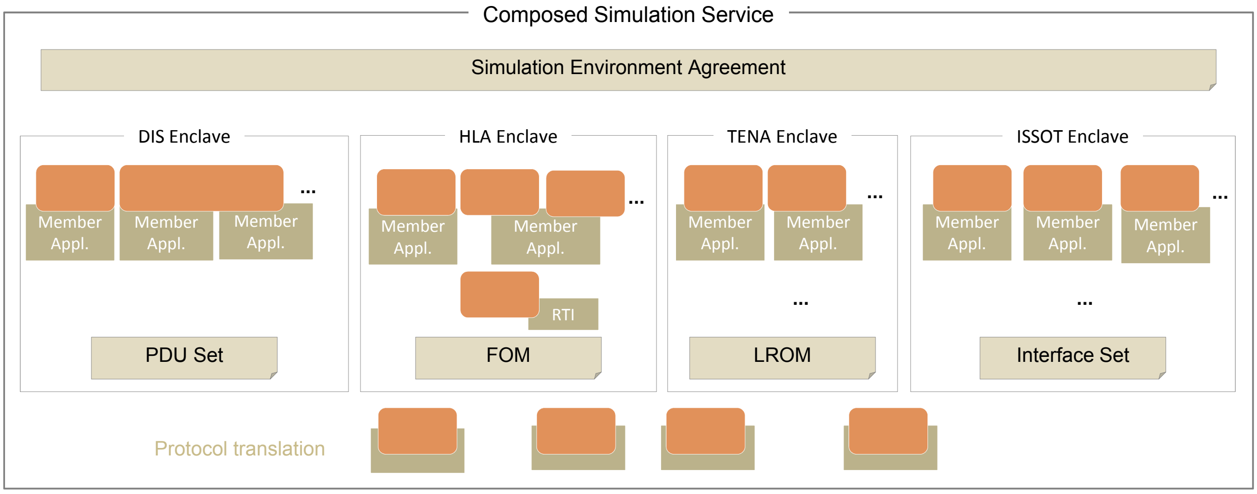

Figure 8: Simulation Enviroment. Concrete implementations in olive.

enclave technology. The MSaaS Reference Architecture embodies the vision that there is added value in designing simulation functionality in terms of Simulation Services, since this is instrumental for the rapid composability and deployment of simulations that the Allied Framework for MSaaS aims to support.

It is essential for packaging Composed Simulation Services that they be clearly delineated in terms of interface and contract. To provide this encapsulation, the MSaaS Reference Architecture states that the *Simulation Environment Agreement* encompasses exactly that which adheres to the *Simulation Data Exchange Model*. Wider agreements must be written when using simulations in conjunction with other systems. For example, in a C2-Simulation System (C2SIM) configuration [27], a semantic agreement (C2SIM federation agreement) might be written for that entire system.

A Composed Simulation Service may interoperate with ABBs from other parts of the C3 Taxonomy. For example, a Composed Simulation Service may serve a C2 system according to standards [28], [29], by way of a designated Simulation Service that exposes the Composed Simulation Service as a service. However, only services that adhere to the *Simulation Environment Agreement* and the *Simulation Data Exchange Model* are said to belong to the Composed Simulation Service; see Figure 9, where the C2 system is represented by the COI Application.

4.4 Scope of the MSaaS Reference Architecture

To summarize the scope of the MSaaS Reference Architecture, only M&S capabilities are in scope. Figure 9 illustrates this for the Composed Simulation Service, where the composed service itself is scope, as well as an M&S application that front-ends the Composed Simulation Service, but not an Application or a Service from a COI other than the M&S COI.

Further, only layers of TOGSOARA that are identified as pertinent for M&S-particular functionality are specified, and the MSaaS Reference Architecture only links in ABBs from the C3 Taxonomy's CIS Capabilities: The Consumer Layer links in ABBs from M&S Applications, the Business Process Layer links inn ABBs from M&S Services (and, of course, not from Business Processes in the C3 Taxonomy's Operational Capabilities), the Services Layer links in ABBs from M&S Services and M&S-Enabling Services and cross-cutting layers link in ABBs from M&S-Enabling Services.

Services and applications from other categories of the C3 Taxonomy are also relevant for M&S capabilities. For example, when composing simulation services into a simulation, a route planning service from the C3

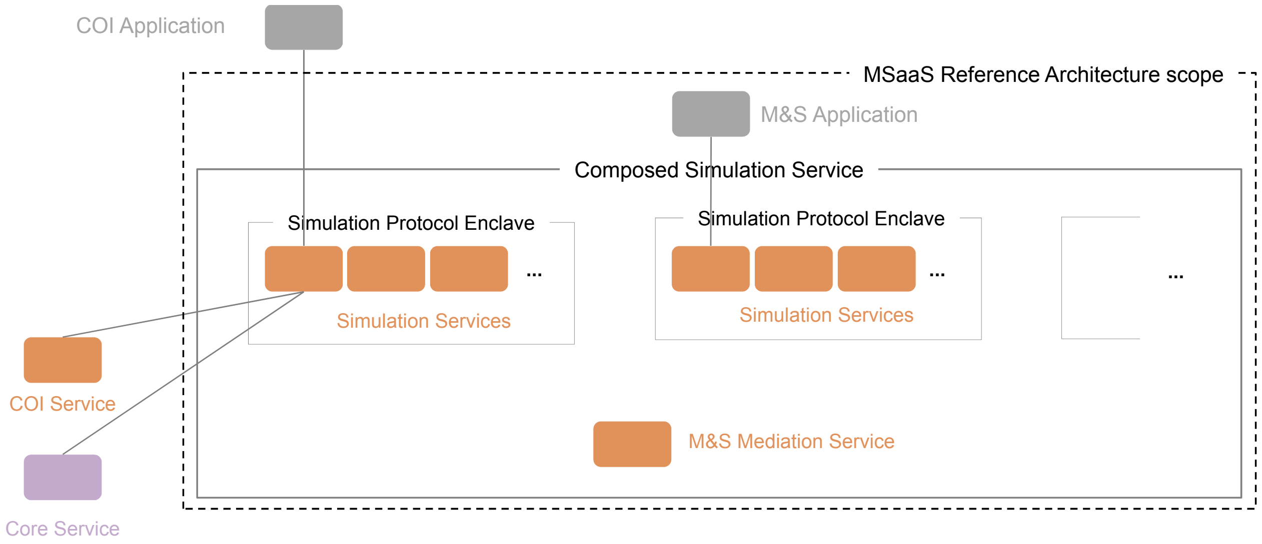

Figure 9: Boundary of Composed Simulation Service and scope of MSaaS Reference Architecture

Taxonomy's Land Services might provide routes for entities in the simulation, and Web Map Services from the C3 Taxonomy's Core Services may provide map data. However, this functionality is provided as services by other communities of interest which are outside the scope of the MSaaS Reference Architecture and the Allied Framework for MSaaS.

Thus, the MSaaS Reference Architecture does not specify non M&S-particular capabilities, such as Core Services (violet in Figure 4) or Communication Services (light blue), although these services may indeed be consumed by M&S capabilities. Whenever one identifies a need for an M&S-particular specialization of a capability from these other portions of the C3 Taxonomy, such a specialization is to be declared in the M&S capabilities. This has already been the case, and these are identifiable below with a "M&S" prefix; i.e. M&S Repository Services and M&S Registry Services.

5.0 MSAAS ARCHITECTURE BUILDING BLOCKS

Table 1 provides an overview of the MSaaS ABBs identified so far per layer and details Figure 6 above. For unpopulated layers, we indicate the assumed C3 Taxonomy categories, from which one might find it pertinent to declare M&S-particular ABBs in the future by square brackets in Table 1.

Once an ABB has been identified, it will be given an ABB type from the TOGSOARA architecture ontology. In some cases that type will be tentative. For example, the Simulation Control Services ABB has been assigned the type Process Controller. However, at present, Simulation Control Services – as it appears in the C3 Taxonomy – represents a variety of functionality that may be refined into several sub-ABBs; perhaps adhering to types such as Process Flow Manager and Process State Manager. In other cases, the assigned type will likely be set; for example, all Simulation Services will likely remain of type Service.

In this way, the TOGSOARA types structure the C3 Taxonomy ABBs – which focus on C3 functionality – in terms of the central roles in a SOA. This should also discipline ABB construction so as to delineate ABB scope more clearly.

Below, we will give examples of the current ABBs. We will only give informal descriptions here, but each ABB will have descriptions (interface and specification) as well as functional and non-functional requirements (Section 4.2) represented in standardized formats deployed in the C3 Taxonomy.

5.1 Modelling Applications

Modelling Applications are ABBs that specify front-end functionality for accessing the Modelling Services through the MSaaS Portal (Section 2). Recall that Applications are to be understood in the SOA sense as loosely coupled front-end apps that can be put together readily and rapidly for the purpose at hand.

5.2 Simulation Applications

Simulation Applications are ABBs that specify front-end functionality which enable users to interact with Simulation Services and Composed Simulation Services. Such apps can be combined to give simulation front-ends in operational systems or in dedicated simulation viewers.

5.3 Modelling Services

Modelling Services encompasses functionality that supports the modelling activities necessary to develop Simulation Services and Composed Simulation Services. These activities include the activities of the DSEEP. For MSaaS, this category contains the services that support the MSaaS portal's Discover, Compose and Execute activities.

Table 1
MSaaS ABBs sorted by layers:

Main layers:	
Consumer Layer	Modelling Applications:Presentation Controller
	Simulation Applications:Presentation Controller
Business Process Layer	Composed Simulation Services:Business Process
	Simulation Control Services:Process Controller
	M&S Composition Services:Dynamic Assembler
	Scenario Services:Process Flow Manager
Service Layer	Modelling Services:Service
	Simulation Services:Service
Service Components Layer	[Core Services:>SOA Platform Services]
Operational Systems Layer	[Core Services:>Infrastructure Services]
	[Communication Services]

Cross-cutting layers:	
Governance Layer	M&S Repository Services:Service Repository
	M&S Registry Services:Service Registry
Information Layer	[Core Services:>Business Support Services]
QoS Layer	M&S Security Services:Security Manager
	M&S Certification Services:Policy Enforcer
Integration Layer	M&S Message-Oriented Middleware Services:Asynchronous Message Manager
	M&S Mediation Services:Mediator

5.4 Simulation Services

Simulation Services are a set of capabilities for synthetic representation of (real-world) objects and events. Simulation Services are the service-oriented building blocks of simulations. Simulation Services are provided by simulation environment member applications that adhere to the simulation environment's simulation interoperability standard or protocol and that execute underlying models for simulation. Simulation Services can be consumed by other member applications and by consumers outside the simulation environment. Member applications that provide simulation services to consumers outside the simulation environment effectively expose the simulation as a service.

The following non-extensive list gives examples of simulation services. These should give rise to ABBs to be deposited in the C3 Taxonomy.

- Unmanned Aerial Vehicle Simulation Service
- Motor Torpedo Boat Simulation Service
- Tactical Data Link Simulation Service
- Simulation Report Generation Services
- Synthetic Environment Services
- Ballistic Weapons Effects Services
- Chemical, Biological, Radiological and Nuclear (CBRN) Weapons Effects Services
- Behavioural Simulation Services

5.5 Composed Simulation Services

Composed Simulation Services are the results of composing simulation services. They offer entire simulations as services. Examples are any number of composed simulations that are generic enough to be used across many situations and/or occasions; for example, Maritime interdiction simulations for mission rehearsal, wargaming simulations for use in operations planning [30], [31], simulations for Live, Virtual, Constructive training [32], etc.

5.6 M&S Repository Services

The M&S Repository Services provides the design-time capabilities to store, retrieve and manage metadata about M&S Services, M&S Enabling Services and M&S Applications, including descriptions of the interface and contract, information about QoS policies and security and versioning information.

5.7 M&S Registry Services

The M&S Registry Services provides the capabilities to store, manage and retrieve references to authoritative information required for the execution of M&S Services, M&S-Enabling Services and M&S Applications. The M&S Registry Services enable advertising and discovery of these available M&S capabilities and supports their runtime binding and virtualisation.

5.8 Simulation Control Services

The Simulation Control Services provide the capabilities to provide input to a simulation execution, control the simulation execution, and collect output from the simulation execution. For example, one service may manage simulation environment execution states and trigger state transitions, and another service may support the orderly management of start-up, execution, and shutdown of distributed Simulation Services. Such control services may be implemented and provided by a member application using State Chart XML (SCXML) [33] for the description of execution states and triggers; see [34], [35], [36], [37].

5.9 M&S Composition Services

The M&S Composition Services provide the automated composition and run-time coordination of Composed Simulation Services in terms of Simulation Services. The services support two common design patterns for composition; namely *orchestration*, where the interaction between Simulation Services is coordinated centrally, and *choreography*, where Simulation Services interact in a non-directed way in a collaborative effort toward a common goal [13].

5.10 Scenario Services

The Scenario Services provide the technical capabilities to manage and load scenarios and direct simulations according to scenarios during execution.

5.11 M&S Security Services

M&S Security Services provide the capabilities to implement and enforce security policies for M&S Services. This includes measures to label, guard, and release information that is exchanged between M&S Services, or exchanged between M&S Services and User Applications. M&S Security Services enable information exchange across security domains. M&S pose particular challenges to security, in that simulations of multiple levels of security must interoperate, possibly together with live systems, with performance issues related to real-time or faster-than-real-time simulation.

Note that lower level security measures (such as authorization, single sign on, identity and access control) are provided by the *Operational Systems Layer* and *Service Components Layer* of the MSaaS Reference Architecture, perhaps relying on C3 Taxonomy Core Services.

5.12 M&S Certification Services

Certification of M&S capabilities is increasingly important for defence applications of M&S. Certification involves processes with manual activities performed by humans, supported by user applications and technical services to verify compliance. The M&S Certification Services are technical services that provide capabilities to verify compliance of Simulation Services, Composed Simulation Services and M&S Applications with NATO interoperability standards for M&S.

The M&S Certification Services must support the particular challenges to verification and validation [38], [39] of Composed Simulation Services posed by MSaaS' service-oriented paradigm of greater degrees of loose coupling and modularization into smaller units of Simulation Services.

5.13 M&S Message-Oriented Middleware Services

The M&S Message-Oriented Middleware (MOM) Services provide the capabilities for an efficient and time coherent exchange of messages between producing and consuming Simulation Services, independent of the message format and message content.

The primary enabling technology for M&S MOM Services is the HLA-RTI [16]. However, the number of options for and capabilities provided by M&S MOM Services technology is growing. Different MOM technology have different functionality, performance characteristics, and protocols [40]. Architectural patterns can over time support the architect in the identification of required capabilities for M&S MOM Services, and design patterns, e.g., for orchestration and choreography (see above), can help the architect in the selection of suitable enabling technology that fulfils these capability requirements.

5.14 M&S Mediation Services

The M&S Mediation Services provide broker and gateway services between incompatible producers and consumers of simulation-pertinent information. M&S Mediation Services receive data from information producers and transform it into a representation that is understood by the consumer. In doing so, M&S Mediation Services bridge the gap between both parties, enabling interaction that was not possible beforehand.

Examples are services that offer simulation gateway functionality (e.g. DIS-HLA), gateway functionality between disparate simulation data exchange models [41], [42], [43], C2-Simulation gateway functionality [44], and inter-simulation system coordination management offered as services [45], [46]. Such gateways should include transformation and translation functionality. Central to this is Model-Based Data Engineering, which is a technique to ensure that data exchanged between services is not only syntactically correct, but also semantically correct as interpreted by the interacting systems [47].

6.0 MSAAS ARCHITECTURE PATTERNS

The MSaaS Reference Architecture gives a number of APs which show how ABBs in the MSaaS Reference Architecture may be combined, how they interact with each other, and what information is exchanged. The APs serve as implementation-independent references for designs and design patterns for target architectures and are given primarily as sequence diagrams. Central APs are, for example:

Patterns focusing on M&S Enabling Services:

- Simulation Service Composition Pattern which shows how the Simulation Composition Services (front-ended by a Simulation Composer Application) interacts with M&S Repository Services for composing Composed Simulation Services from Simulation Services.
- Distributed Simulation State Data Pattern which shows how the M&S MOM Services can be used as a state-deferral mechanism [48] for relieving Simulation Services of holding the shared state of a Composed Simulation Service.

Patterns focusing on M&S Services:

- Simulation Stateful Pattern which shows how a Simulation Service may be consumed assuming the service has access to necessary parts of the shared state of a Composed Simulation Service.
- Simulation Stateless Pattern which shows how a Simulation Service may be consumed assuming the service does not have access to the shared state of a Composed Simulation Service.
- Shared Simulation Environment Services Pattern which shows how Synthetic Environment Services are consulted by Simulation Services in a Composed Simulation Service for common initial environment models and dynamic updates to these during simulation execution, to ensure credibility and fair fight in environmental factors.
- Shared Simulation Event Effects Services Pattern which shows how Simulation Event Effects Services are consulted by Simulation Services in a Composed Simulation Service for standardized computation of events (such as damage entailed by a Ballistic Weapons Simulation Service), to ensure credibility and fair fight in event effects.
- Shared Behavioural Simulation Services Pattern which shows how Behavioural Simulation Services are consulted by Simulation Services in a Composed Simulation Service for standardized modelling of behaviour (such as targeting and rules of engagement), to ensure credibility and fair fight in behaviour.
- Shared Presentation Applications Pattern which shows how Simulation Services and Composed Simulation Services use Simulation Applicationsto render and present entities and events in a standardized manner during execution, to ensure credibility and fair fight in presentation.

Patterns focusing on non-M&S-particular COI capabilities using the M&S capabilities of MSaaS:

- C2SSim Pattern which shows how a C2IS Application may interoperate with Composed Simulation Services via C2SIM Mediation Services
- Sim and COI/Core Services Pattern which shows how a Simulation Service may be interoperate with another COI Service or Core Service

7.0 CONCLUDING REMARKS

We have outlined the MSaaS Reference Architecture as part of an architecture framework for MSaaS. The purpose of the MSaaS Reference Architecture is to support the technical users of the Allied Framework for MSaaS. The architecture framework for MSaaS provides the means to continuously build and refine the MSaaS Reference Architecture as the demand for M&S capabilities increases and shifts.

The framework in this paper lays the ground for the substantial and hard tasks of determining how functionality should be divided into loosely coupled M&S capabilities (services and applications), for writing the corresponding M&S capability descriptions (interfaces and contracts) and functional and non-functional requirements, and for developing architecture patterns.

The success of MSaaS relies on solving foundational challenges in M&S; e.g., within conceptual modelling, computation and model reuse, composition and adaptation [49]. Although the MSaaS Reference Architecture defines architecture building blocks and architecture patterns to address several issues – such as verification and validation of the composition of simulation services, multilevel security issues particular to M&S, performance issues and scalability issues – these issues cannot be resolved without substantial advancements in research and development in M&S in general. Since MSaaS offers services to other communities of interest, the success of MSaaS also relies on solving challenges in those domains. The MSaaS Reference Architecture – with its link to the C3 Taxonomy – can help maintain an updated view on which building blocks and patterns are present and which are lacking in relation to the current state of research and development in NATOs C3 systems portfolio, and can help to identify where further technology or standards development should take place.

The concept of the Allied Framework for MSaaS and its reference architecture are visionary. The concept is under test in solutions that use container technology [50], [51], [52], where the functionality and the workflow of the MSaaS Portal are viably demonstrated. However, operational use will be the true test and the driver of further development. The MSaaS Reference Architecture is designed to support this development and to adapt accordingly.

8.0 REFERENCES

[1] A. Tolk, Terms and application domains, in: A. Tolk (Ed.), Engineering Principles of Combat Modeling and Distributed Simulation, Wiley, 2012, Ch. 4, pp. 55–78.

[2] International Organization for Standardization, ISO/IEC 42010:2007 Systems and software engineering – Architecture description (2011).

[3] The Open Group, SOA Reference Architecture Technical Standard, doc. no. C119 (2011).

[4] J. E. Hannay, K. Brathen, O. M. Mevassvik, A hybrid architecture framework for simulations in a service-oriented environment, Systems Engineering 20 (3) (2017) 235–256.

[5] North Atlantic Treaty Organization, NATO Architecture Framework v4.0 Documentation (draft) (2016).

[6] J. E. Hannay, K. Brathen, O. M. Mevassvik, Agile requirements handling in a service-oriented taxonomy of capabilities, Requirements Engineering 22 (2) (2017) 289–314.

[7] J. E. Hannay, Architectural work for modeling and simulation combining the NATO Architecture Framework and C3 Taxonomy, J. Defense Modeling and Simulation: Applications, Methodology, Technology 14 (2) (2017) 139–158.

[8] The Open Group, TOGAF Version 9.1 Enterprise Edition, doc. no. G116 (2011).

[9] R. Hilliard, The role of architecture frameworks: Lessons learned from ISO/IEC/IEEE 42010, unpublished (2013).

[10] J.-L. Garnier, L. Bischoff, M. André, B. Lavit, M. Peyrichon, J. Blanquart, N. Scuto, Architecture frameworks – a standard to unify terms, concepts, life-cycles and principles, in: Proc. NATO Information Systems Technology Panel Symp. on Architecture Definition and Evaluation (STO-MP-IST-115), 2013.

[11] NATO Communications and Information Agency, The C3 Taxonomy, accessed January 2016 (2016).

[12] E. Çayırcı, Modeling and simulation as a cloud service: A survey, in: Proc. 2013 IEEE Winter Simulation Conf., IEEE, 2013, pp. 389–400.

[13] The Open Group, SOA Ontology, Version 2.0 Open Group Standard, doc. no. C144 (2014).

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

[15] IEEE Standards Association, 1730.1-2013 – IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO) (2013).

[16] IEEE Standards Association, 1516-2010 – IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) (2010).

[17] IEEE Standards Association, 1278.2–2015 – IEEE Standard for Distributed Interactive Simulation (DIS) – Communication Services and Profiles (2015).

[18] Test Resource and Management Center, TENA The Test and Training Enabling Architecture – Architecture Reference Document, Version 2002 (2002).

[19] World Wide Web Consortium, Web Services Architecture – W3C Working Group Note (2004).

[20] R. T. Fielding, R. N. Taylor, Principled design of the modern web architecture, ACM Transactions on Internet Technology 2 (2) (2002) 115–150.

[21] Organization for the Advancement of Structured Information Standards, Advanced Message Queuing Protocol (AMQP) Version 1.0 (2012).

[22] L. Granowetter, The WebLVC protocol: Design and rationale, in: Proc. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2013, National Training and Simulation Association, 2013.

[23] I. Fette, A. Melnikov, The WebSocket Protocol—Internet Engineering Task Force (IETF) Request for Comments: 6455 (2011).

[24] P. Gustavson, T. Chase, L. Root, K. Crosson, Moving towards a Service-Oriented Architecture (SOA) for distributed component simulation environments, in: Proc. 2005 Spring Simulation Interoperability Workshop (SIW), Simulation Interoperability Standards Organization, 2005.

[25] V. Mojtahed, E.-O. Svee, J. Zdravkovic, Semantic enhancements when designing a BOM-based conceptual model repository, in: Proc. 2010 European Simulation Interoperability Workshop (SIW), Simulation Interoperability Standards Organization, 2010.

[26] I. Mahmood, R. Ayani, V. Vlassov, F. Moradi, Verifying dynamic semantic composability of BOM-based composed models using colored Petri nets, in: Proc. 2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation (PADS), IEEE Computer Society, 2012, pp. 250–257.

[27] Simulation Interoperability Standards Organization, The Command and Control Systems – Simulation Systems Interoperation (C2SIM) Product Development Group (PDG) and Product Support Group (PSG), accessed June 2015 (2014).

[28] Simulation Interoperability Standards Organization, SISO-STD-011-2014 – Standard for Coalition Battle Management Language (C-BML) Phase 1, Version 1.0 (2014).

[29] Simulation Interoperability Standards Organization, SISO-STD-007-2008 – Standard for Military Scenario Definition Language (MSDL) (2008).

[30] S. Bruvoll, J. E. Hannay, G. K. Svendsen, M. L. Asprusten, K. M. Fauske, V. B. Kvernelv, R. A. Løvlid, J. I. Hyndøy, Simulation-supported wargaming for analysis of plans, in: Proc. NATO Modelling and Simulation Group Symp. on M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence (STO-MP-MSG-133), 2015.

[31] E. Çayırcı, L. Özçakır, Modeling and simulation support to the defense planning process, J. Defense Modeling and Simulation: Applications, Methodology, Technology.

[32] J. E. Hannay, O. M. Mevassvik, A. Skjeltorp, K. Brathen, Live, Virtual, Constructive (LVC) simulation for land operations training: Concept development & experimentation (CD&E), in: Proc. NATO Modelling and Simulation Group Symp. on Integrating Modelling & Simulation in the Defence Acquisition Lifecycle and Military Training Curriculum (STO-MP-MSG-126), 2014.

[33] World Wide Web Consortium, State Chart XML (SCXML): State Machine Notation for Control Abstraction – W3C Recommendation (2015).

[34] T. W. van den Berg, R. E. J. Jansen, H. Ufer, Design patterns for and automation of federation state control, in: Proc. 2009 Spring Simulation Interoperability Workshop, no. 09S-SIW-009, 2009.

[35] S. Gallant, C. J. Metevier, C. Gaughan, Systems engineering an executable architecture for M&S, M&S Journal (2014) 16–24.

[36] C. Gaughan, C. J. Metevier, K. Athmer, S. Gallant, S. Murphy, K. Snively, Bringing next generation simulation into the land of practicality, in: Proc. 2013 Fall Simulation Interoperability Workshop, no. 13F-SIW-017, 2013.

[37] K. Snively, R. Leslie, C. Gaughan, Runtime execution management of distributed simulations, in: Proc. 2013 Fall Simulation Interoperability Workshop, no. 13F-SIW-019, 2013.

[38] A. Tolk, Verification and validation, in: A. Tolk (Ed.), Engineering Principles of Combat Modeling and Distributed Simulation, Wiley, 2012, Ch. 14, pp. 263–294.

[39] A. Tolk, S. Mittal, A necessary paradigm change to enable composable cloud-based M&S services, in: Proc. 2014 Winter Simulation Conference, 2014.

[40] J. Eriksson, Comparing message-oriented middleware for financial assets trading, Tech. Rep. 2016:63, KTH, Computer and Electronic Engineering (2016).

[41] G. W. Allen, L. Schroeder, Utilization of Service Oriented Architecture (SOA)-based commercial standards to address Live, Virtual, Constructive (LVC) interoperability challenges, in: Proc. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2011, National Training and Simulation Association, 2011.

[42] J. E. Coolahan, G. W. Allen, LVC Architecture Roadmap Implementation—results of the first two years, in: Proc. 2011 Fall Simulation Interoperability Workshop (SIW), no. 11F-SIW-025, Simulation Interoperability Standards Organization, 2011.

[43] G. W. Allen, R. Lutz, R. Richbourg, Live, virtual, constructive, architecture roadmap implementation and net-centric environment implications, ITEA Journal 31 (3) (2010) 355–364.

[44] J. M. Pullen, D. Corner, A. Brook, R. Wittman, O. M. Mevassvik, A. Alstad, MSDL and C-BML working together for NATO MSG-085, in: Proc. 2012 Spring Simulation Interoperability Workshop (SIW), Simulation Interoperability Standards Organization, 2012.

[45] D. L. Drake, K. L. Morse, Use of SOA for distributed simulation: A way forward, in: Proc. 2012 Spring Simulation Interoperability Workshop (SIW), no. 12S-SIW-060, Simulation Interoperability Standards Organization, 2012.

[46] D. L. Drake, I. X. Martins, R. A. Roca, F. Carr, Live-Virtual-Constructive Service-Oriented Architecture. Service-Oriented Architecture application to Live-Virtual-Constructive simulation: Approach, benefits, and barriers, Tech. Rep. NSAD-R-2011-025, National Security Analysis Department, The Johns Hopkins University, Applied Physics Laboratory (2011).

[47] A. Tolk, S. Y. Diallo, Model-Based Data Engineering for Web Services, Springer, 2008, Ch. 6, pp. 137–161.

[48] T. Erl, SOA principles of Service Design, Prentice Hall, 2007.

[49] R. Fujimoto, C. Bock, W. Chen, E. Page, J. Panchal (Eds.), Research Challenges in Modeling and Simulation for Engineering Complex Systems, Springer, 2016.

[50] T. W. van den Berg, B. Siegel, A. Cramp, Containerization of High Level Architecture-based simulations: A case study, J. Defense Modeling and Simulation: Applications, Methodology, Technology 14 (2) (2017) 115–138.

[51] T. W. van den Berg, A. Cramp, Container orchestration environments for M&S, in: Proc. 2017 Fall Simulation Innovation Workshop, no. 17F-SIW-006, 2017.

[52] T. W. van den Berg, B. Siegel, A. Cramp, Guidelines and best practices for using Docker in support of HLA federations, in: Proc. 2016 Fall Simulation Innovation Workshop, no. 16F-SIW-031, 2016.