Outline of a Service-Based Reference Architecture for Effective and Efficient Use of Modelling and Simulation

Robert Siegfried
aditerna GmbH
Otto-Hahn-Str. 13 B
85521 Riemerling, Germany

robert.siegfried@aditerna.de

Andreas Diehl
Federal Office of Equipment, Information
Technology and In-Service Support
Ferdinand-Sauerbruch-Str. 1
56073 Koblenz, Germany

andreasdiehl@bundeswehr.org

Saikou Y. Diallo
Virgina Modeling Analysis and Simulation Center
1030 University Blvd.
Suffolk, VA 23435, USA

sdiallo@odu.edu

Michael Bertschik
Bundeswehr Technical Center
for Weapons and Ammunition (WTD 91)
49726 Meppen, Germany

michaelbertschik@bundeswehr.org

Günter Herrmann
ITIS GmbH
Werner-Heisenberg-Weg 39
88579 Neubiberg, Germany

guenter.herrmann@unibw.de

Martin Rother
IABG mbH
Einsteinstr. 20
85521 Ottobrunn, Germany

rother@iabg.de

ABSTRACT

Modelling & Simulation (M&S) is a widely used toolset within NATO and its Partners across many application domains. Most often associated with military training, M&S is also used for analysis, experimentation, test and evaluation (e.g., in the acquisition process). M&S products are therefore very valuable to NATO and military organizations and it is essential that M&S products, data and processes are conveniently accessible to a large number of users as often as possible. Therefore a new "M&S eco-system" is required where M&S products can be accessed simultaneously and spontaneously by a large number of users for their individual purposes. This environment has to support stand-alone use as well as integration of multiple simulation systems and real systems into a coherent (maybe distributed) simulation environment whenever the need arises.

For many reasons, service-based architectures are considered to be very promising for realizing these next generation M&S environments. The combination of M&S with service-based architectures and ideas taken from cloud computing is known as "Modeling & Simulation as a Service" (MSaaS).

NATO Modelling and Simulation Group 131 ("Modelling and Simulation as a Service: New concepts and Service Oriented Architectures") did a first survey on the topic of "M&S as a Service" and provided collective knowledge and experience of a dozen nations and NATO bodies to ACT. Based on this survey and extensive experiences from developing a service-oriented reference architecture for a distributed integrated test bed for the German Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw), this paper outlines a service-based reference architecture for effective and efficient use of M&S.

This paper shows how the requirements for the next generation of simulation environments are addressed *within the proposed reference architecture. The paper also shows the relations between the proposed reference architecture and the NATO M&S Master Plan and identifies missing but required standards. Finally it identifies potential follow-on activities and gives recommendations for possible standardization activities.*

1.0 INTRODUCTION

To a great extent, future military training capabilities will be provided by simulation systems (either stand-alone or via distributed simulation environments). This is a consequence of limited or decreasing budgets, restrictions due to security and safety regulations, and shorter response times as well as increasingly faster changing mission profiles and operational needs.

Whereas stand-alone simulation systems are regularly used for procedure training and education at the single warfighter level, distributed simulations show their strength at joint/combined training on tactical level, operational level, and above. Also, distributed simulation environments are commonly used to integrate real equipment and simulation assets for training purposes and system evaluation.

Many current simulation systems, whether stand-alone or distributed, suffer from time- and cost-intensive development and initialization procedures as well as from limited accessibility by users. Furthermore, limited credibility resulting from unknown validity and ad-hoc processes is a serious problem.

1.1 MSG-131 “M&S as a Service: New concepts and Service Oriented Architectures”

From August 2013 till August 2014, NATO Modeling and Simulation Group MSG-131 (“Modelling and Simulation as a Service: New concepts and Service Oriented Architectures”) has investigated the concept of M&S as a Service (MSaaS) and collected national perspectives and experiences regarding MSaaS. This survey provides an overview about the nations’ activities in this area and is input to “NATO M&S as a Service Concept” that is currently being developed by NATOs Allied Command Transformation (ACT).

The objectives of the NATO MSG-131 Specialist Team (ST) are summarized as follows:

- To agree on a common understanding of the terminology.
- To develop a primer of the NATO technical concept for MSaaS.
- To provide consolidated knowledge, informed by standards and technical documentation on MSaaS, which serves as a basis and permits development of a specific MSaaS concept and architecture to be used by NATO nations and bodies.
- To develop a draft Reference Services Oriented Architecture which will allow conducting improved training and exercises and other applications.

In accordance with its Technical Activity Description, MSG-131 recommends to investigate MSaaS in more detail. A Technical Activity Proposal for a follow-on research task group was developed by MSG-131 and endorsed in June 2014. The task group MSG-136 (“Modelling and Simulation (M&S) as a Service (MSaaS) – Rapid deployment of interoperable and credible simulation environments”) will start its 3-year term in November 2014.

1.2 NATO M&S Master Plan

The NATO M&S Master Plan (NMSMP) [10] defines the M&S strategic plan for NATO and is binding on NATO organizations. It defines five top-level objectives:

- I – Establish a Common Technical Framework to foster interoperability and reuse

- II – Provide Coordination & Common Services to increase cost-effectiveness
- III – Develop Models & Simulations
- IV – Employ Simulations to enhance NATO mission effectiveness
- V – Incorporate Technological Advances

All top-level objectives are further detailed into several sub-objectives. In the following sections it is explained how next generation simulation environments utilizing the outlined reference architecture are related to the NMSMP objectives.

2.0 NEXT GENERATION DISTRIBUTED SIMULATION ENVIRONMENTS

A detailed definition of requirements for next generation simulation environments and recommendations on their design are given in [14]. In the following, the main requirements and recommendations are summarized.

2.1 Requirements on next generation distributed simulation environments

NG-1	Improve development of effective simulation environments, i.e., ensure that a simulation environment satisfies the users' needs (related to DSEEP step 1). In terms of measurable requirements this requires that the users' needs (i.e., the requirements on a simulation environment) are completely known, consistent, and documented.
NG-2	Enable efficient preparation, development, and integration of distributed simulation environments. In terms of measurable requirements the time required for executing the activities defined in DSEEP steps 2 to 5 should be less than one month for average simulation environments.
NG-3	Enable efficient initialization and execution of distributed simulation environments (as specified by DSEEP step 6). In terms of measurable requirements this requires: (NG-3.1) Provide capability for centrally coordinated initialization of a simulation environment without manual interaction. (NG-3.2) Enable full initialization of a typical distributed simulation environment within 15 minutes.
NG-4	Enable distributed simulation environments that achieve fair fight. In terms of measurable requirements this requires an objective and automatic assessment whether a simulation environment and its member applications comply with the specified fair fight requirements.

NG-5	Enable distributed simulation environments that deliver credible simulation results. In terms of measurable requirements this requires: (NG-5.1) Provide traceable documentation of the simulation environment engineering process (requirements, assumptions, constraints, agreements, etc.). (NG-5.2) Provide automated control mechanisms for assessing the quality requirements of a distributed simulation environment during execution. (NG-5.3) Provide automated control mechanisms for assessing the quality requirements of a distributed simulation environment after its execution.
NG-6	Enable distributed simulation environments that consistently deliver identical simulation results when initialized with identical data and executed under identical conditions. In terms of measurable requirements this requires: (NG-6.1) Full documentation of a simulation environment (participating systems, software versions, configuration, etc.). (NG-6.2) Full documentation of initialization data and execution data (i.e., initial state, course of events, etc.) (NG-6.3) If required, long-term storage of configuration files, software applications, etc. The degree of reproducibility may vary greatly for different simulation environments (e.g., basic reproducibility may only require using the same data while full reproducibility may require using the exact same versions of participating systems) and may not always be fully achievable (e.g., in simulation environments with manual interaction). Depending on the required degree of reproducibility, the requirements defined above may need to be extended.

2.2 Non-functional requirements as drivers for next generation simulation environments

Non-functional requirements (e.g., regarding security or scalability) are regularly considered as major impact factors for system architecture and system design. The same is true with regards to simulation environments: While functional requirements (like NG-3.1) are comparatively easy to satisfy, non-functional requirements like NG-2 and NG-3.2 are considered to require substantially more efforts to be achieved.

The authors of this papers have been hesitant to specify actual objectives for non-functional requirements NG-2 (preparation time for a simulation environment should be less than one month) and NG-3.2 (full initialization of a simulation environment in less than 15 minutes) as simulation environments vary greatly in terms of size, complexity, and available resources. Nevertheless, due to the paramount importance of non-functional requirements on architecture and design of next generation simulation environments actual objectives are specified. Especially, the requirements NG-2 and NG-3.2 are considered as major drivers.

2.3 Recommendations for next generation simulation environments

Recommendation on system design	• SD-1: Design and document for interoperability • SD-2: Design and document for modularity and composability • SD-3: Favor open standards • SD-4: Design for securability
Recommendations on simulation environment infrastructure	• IN-1: Harmonize critical data and algorithms • IN-2: Establish permanent simulation infrastructure • IN-3: Establish member application compliance testing • IN-4: Establish simulation environment execution compliance testing
Recommendations on simulation environment engineering processes and organization	• PO-1: Enforce requirements specification • PO-2: Use a systems engineering process and document decisions • PO-3: Establish simulation repository
Recommendations on simulation environment data	• DA-1: Enforce "single source of truth" principle

3.0 OUTLINE OF A SERVICE-BASED REFERENCE ARCHITECTURE FOR M&S

3.1 Preliminary remarks about reference architectures

According to the NATO Architecture Framework (NAF) reference architectures are the linking element between overarching architectures and target architectures:

> *"Reference architectures reflect strategic decisions regarding system technologies, stakeholder issues, and product lines. They render user requirements, processes, and concepts in a high-level solution from which individual projects can be identified and initially programmed. Their primary focus is on services, processes and component functionality, and they provide the basis for the development of Target Architectures (TA)."* [11]

Therefore, reference architectures are considered to be generic blueprints that may be used as a basis for deriving specific architectures. In this sense, reference architectures are recommendations how to approach a certain architecture development task. [9]

The common assumption is that building target architectures for specific simulation systems or simulation environments on foundations from established reference architectures will increase not only the efficiency of work in time and budget, but also the quality of the results, and will lead to improved interoperability.

The ideas and approaches for a service-based reference architecture that are presented in this paper are taken from many sources. A major source is the work done by MSG-131 [9], as well as work that has been done in MSG-068 [7] and is currently been done in MSG-106 and MSG-128.

3.2 Layered Architecture

To distinguish between different components of the outlined reference architecture and to establish clear responsibilities of each component, a layered approach is proposed.

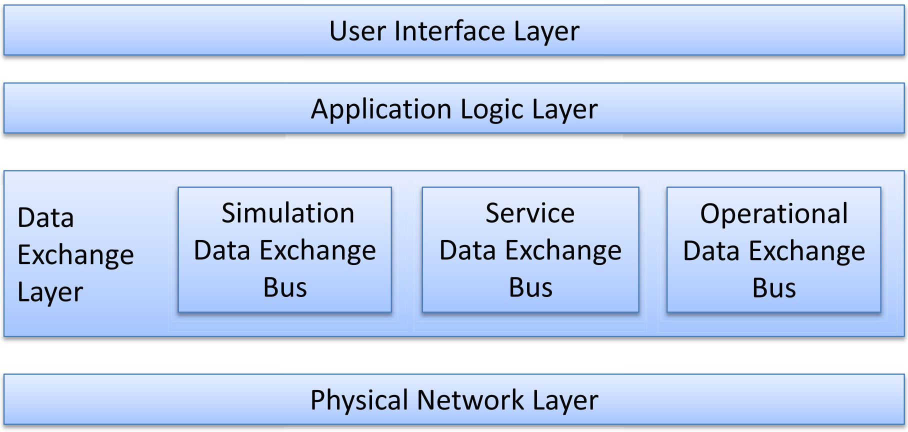

Figure 1: Layers of service-based reference architecture for M&S.

As illustrated in Figure 1, four layers are distinguished:

- Physical Network Layer: The physical network layer is concerned with physically interconnecting all member applications of a simulation environment. A typical choice for realizing physical interconnection is Ethernet, although other means may be used (e.g., radio connection to include specific operational systems).
- Data Exchange Layer: Building upon the physical network layer, the data exchange layer provides sophisticated means to exchange data via the underlying network. Depending on the actual requirements different approaches and protocols may be used to exchange specific data. The data exchange layers usually contains at least the following three distinct bus systems:
  - Simulation Data Exchange Bus: The simulation data exchange bus is used to exchange all simulation-related data and to control all simulation events. As demanded by STANAG 4603 [12] the recommendation is to use HLA Evolved [3] for realizing the simulation data exchange bus. This reference architecture does not mandate a specific FOM, as this may vary greatly depending on the application domain (e.g., CAX-oriented simulation environments may choose the NETN FOM whilst others might follow the recommendations of MSG-128). Specific services may require support for specific FOM modules.
  - Service Data Exchange Bus: The service data exchange bus is used for direct communication between member applications (e.g., simulation systems) and services as well as for direct communication between services (i.e., inter-service communication). Depending on the involved services and data exchange relations various protocols may be used (e.g., OGC WFS for communication and data exchange with a Synthetic Environment Service, JSON, custom XML messages via HTTP or SOAP, etc.). Two characteristics of the service data exchange bus require special attention:
    - Data exchange during simulation runtime and data exchange before/after simulation runtime may impose different constraints with regards to latency, response times, etc.
    - Data exchange may include large data sets (e.g., terrain data provided by a Synthetic Environment Service). Therefore, depending on the actual services and the actual usage pattern, the service data exchange bus may eventually be required to be able to handle such

large data sets.

  - Operational Data Exchange Bus: The operational data exchange bus is used for data exchange using either operational protocols or including operational systems that require data exchange using their native operational protocols. Examples include C2 systems (using e.g., MIP, C-BML, NFFI, TDL Link11/16, etc.) as well as providers/consumers of video feeds (e.g., reconnaissance UAVs/RPAs or advanced offensive units like JSF) using STANAG 4609.
- Application Logic Layer: The application logic layer collects all member applications of the simulation environment and all other applications (e.g., data logger, data recorder, monitoring tools, initialization service, etc.) that are not directly part of the simulation environment.
- User Interface Layer: The user interface layer provides functionalities to access and control the applications provided by the Application Logic Layer. The distinction between Application Logic Layer and User Interface Layer is especially important for all approaches towards providing M&S in a cloud-based way. The important aspect to note is that the Application Logic and the User Interface of an M&S application may reside in different places (e.g., application logic is executed in a central data center while the user interface is provided by a locally installed lightweight application).

As mandated by layered architectures, higher layers build upon the functionalities provided by lower layers.

Although the term “bus” is used, the above mentioned data exchanges are strictly speaking not necessarily bus systems (i.e., shared medium is not required). Point-to-point connections (without a “bus” in a strict technical sense) are also allowed. Depending on non-functional requirements and constraints like size of a simulation environment, number of services, etc. actual bus systems should be preferred (e.g., using an Enterprise Service Bus for realizing the service data exchange bus).

3.3 Components of a simulation environment

In general, this paper follows DSEEP [4] terminology. In the following, two key terms defined by IEEE 1730 (DSEEP) are reiterated:

- Member application: “A member application is an application that is serving some defined role within a simulation environment. This can include live, virtual, or constructive simulation assets, or can be supporting utility programs such as data loggers or visualization tools.” [4]
- Simulation environment: “A named set of member applications along with a common simulation data exchange model (SDEM) and set of agreements that are used as a whole to achieve some specific objective.” [4]

Figure 2 gives a high-level overview of the above described terms and their relations to each other. Fictitious systems are included as examples for illustration purposes.

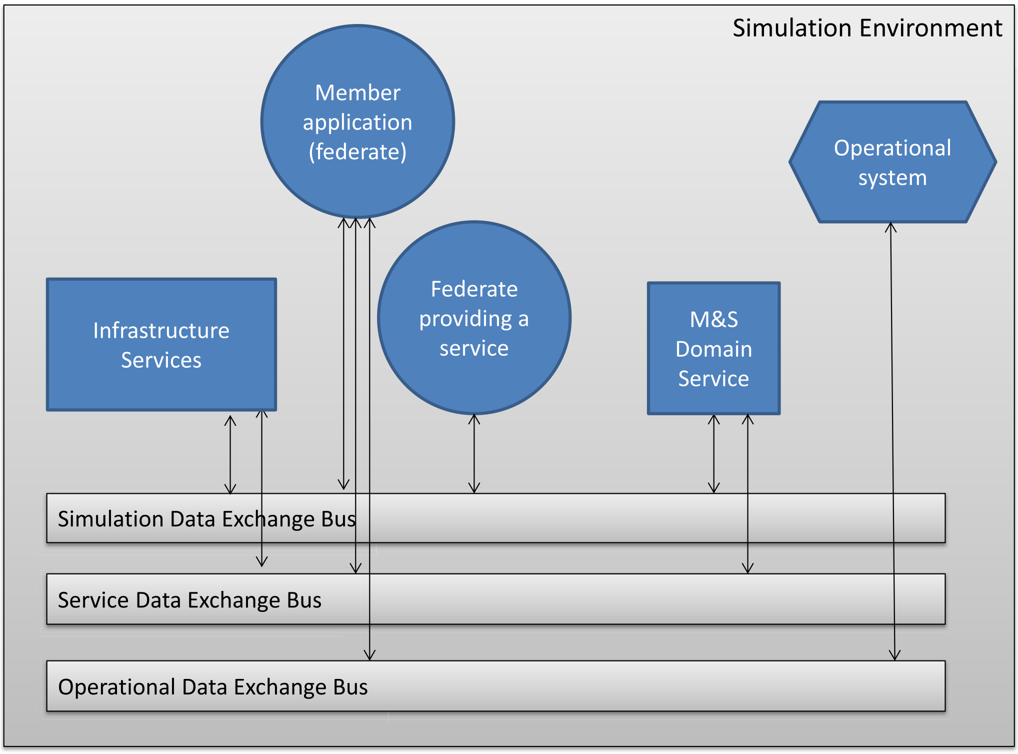

Figure 2: Illustration of components of a simulation environment.

3.4 Types of services

A service-oriented simulation environment may utilize very different types of services. Within this paper, it is proposed to distinguish between three different types of services (see Figure 3).

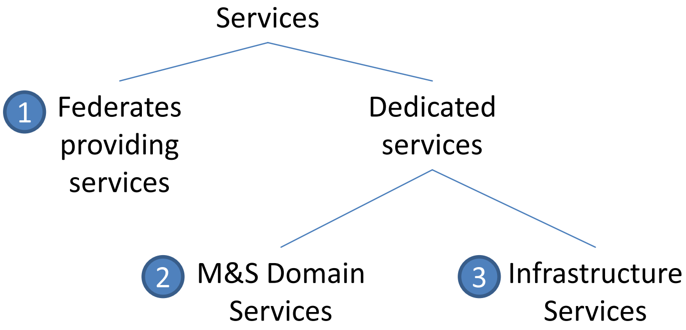

Figure 3: Types of services

Federates providing services refers to the notion of simulation participants (e.g., a simulated tanker aircraft) that provide specific services (e.g., supply of fuel) for other simulation participants (e.g., a simulated fighter aircraft). The services that are provided and consumed by the simulated entities are therefore part of the simulation environment and its conceptual model. MSG-068 and MSG-106 defined Consumer-Provider-Patterns for this type of service [7, 5].

M&S domain services aim to harmonize specific characteristics of simulation models (e.g., providing terrain information, or calculating weapon effects) and to improve simulation interoperability on higher levels of interoperability, namely on the pragmatic and conceptual interoperability level (see [16, 17, 18] for a discussion of interoperability levels). Harmonization of critical algorithms and data increases fair fight with regard to a specific domain and improves simulation quality as well as credibility (assuming the M&S domain services themselves are properly developed, verified and validated). M&S domain services also significantly increase reusability and reduce verification and validation efforts.

Infrastructure services provide specific utility functionalities that are required additionally to the original simulation systems. Examples of infrastructure services include initialization services (e.g., distribution of initialization and configuration data), data logging/recording services, simulation data collection services (i.e., collecting simulation results from different remote simulation systems), and time services (like providing a precise time source for all member applications of a simulation environment).

3.5 Data Exchange Requirements

Table 1 gives an overview of data exchange requirements of the different types of services and other member applications of a simulation environment. Table 1 also identifies which bus systems are required for the data exchange.

Table 1: Data exchange requirements

Type of member application	**Simulation Data Exchange Bus**	**Service Data Exchange Bus**	**Operational Data Exchange Bus**
Simulation system (federate)	Connection to simulation data exchange bus required to exchange simulation data (i.e., regular HLA data exchange).	Connection to service data exchange bus required to use services.	Connection to operational data exchange bus required if data exchange with operational systems is required. Use of C2SimProxy-federate possible that translates from HLA (i.e., interaction classes) into a specific operational protocol (e.g., using specific messages).
M&S domain service	Connection to simulation data exchange bus required if M&S domain service needs to access simulation state.	Connection to service data exchange bus required if inter-service communication is necessary or communication with simulation systems is possible via the service data exchange bus only (i.e., M&S domain service does not need to access simulation state).	Usually no connection required.

Infrastructure service	Connection to simulation data exchange bus required if service needs to access simulation state.	Connection to service data exchange bus required if member applications and infrastructure services communicate via the service data exchange bus (and not via the simulation data exchange bus). This is the case when special protocols are used (e.g., OGC WFS, NTP, ...).	Usually no connection required.
Operational systems (e.g., C2 systems)	Usually no connection required as operational systems usually do not natively support data exchange via HLA.	Usually no connection required as operational systems usually do not natively support data exchange via service interfaces.	Connection required as this is the default mechanism of operational system for data exchange.

3.6 Example M&S Domain Service: Weapon Effects Service (WES)

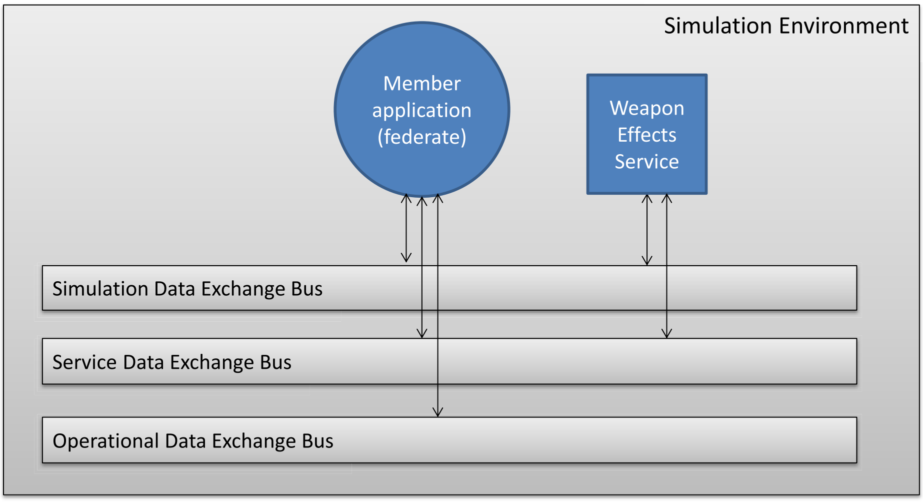

Figure 4: Basic idea of a Weapon Effects Service (WES).

Figure 4 illustrates the basic idea of a Weapon Effects Service (WES). The task of a WES is to calculate damage of a unit due to weapon effects.

WES characteristics:

- Type of service: M&S Domain Service (that harmonizes damage calculation across all member applications)
- Required data exchange:
    - Connection to Simulation Data Exchange Bus required as the Weapon Effects Service needs access to the simulation state (e.g., position of units).

  - Connection to Service Data Exchange Bus may be required for data exchange with other M&S services, e.g., an initialization service that properly configures a WES before simulation execution.

The following recommendations are addressed by a WES:

- IN-1: Harmonize critical data and algorithms
- DA-1: Enforce “single source of truth” principle

Prototype weapon effects servers have been developed by TNO [1], Spanish MoD and German Armed Forces [8].

3.7 Example Infrastructure Service: Synthetic Environment Service (SES)

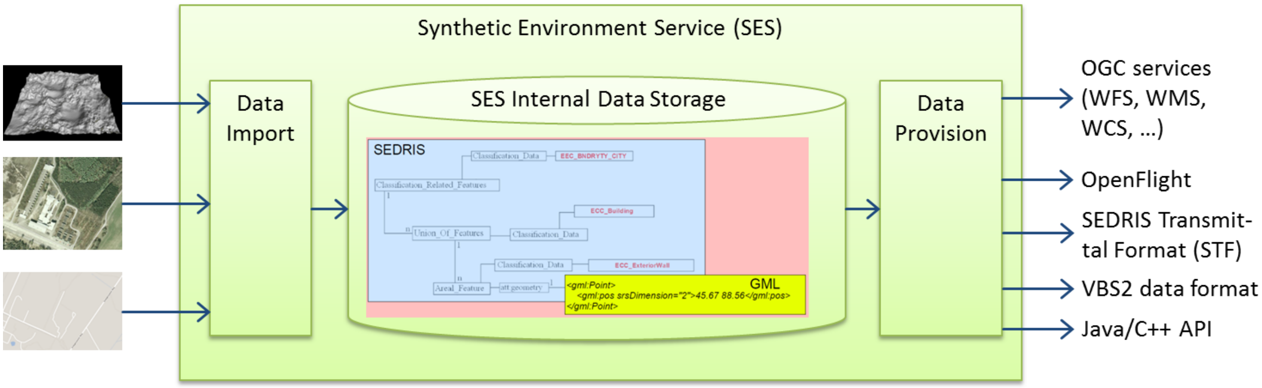

Figure 5: Basic idea of a Synthetic Environment Service

Figure 5 illustrates the basic idea of a Synthetic Environment Service (SES). An SES is used to initialize all member applications of a simulation environment with synthetic environment data before runtime, i.e., before simulation execution.

SES characteristics:

- Type of service: Infrastructure service
- Required data exchange:
  - Connection to service data exchange bus required for data exchange (request, response) with member applications that require synthetic environment data (e.g., simulation systems, M&S domain services, etc.).
  - As the SES is used to initialize member applications of a simulation environment before runtime, no connection to the simulation data exchange bus is required.

The following recommendations are addressed by an SES:

- IN-1: Harmonize critical data and algorithms
- DA-1: Enforce “single source of truth” principle
- SD-3: Favour open standards

A prototype SES has been developed by the German Armed Forces [6, 15].

4.0 DISCUSSION

4.1 Requirements on simulation systems

A major requirement for any kind of service-oriented simulation environment is that the member applications (i.e., especially the simulation systems) need to satisfy certain requirements to use M&S domain services:

- Use services instead of internal algorithms: Simulation systems need to be prepared to use external services (e.g., to use damage results due to weapon effects calculated elsewhere). If a simulation system is designed to work in a non-service environment and has built-in algorithms (e.g., for damage calculations) these must be switched off.
- Harmonized conceptual models: Simulation systems and M&S domain services need harmonized conceptual models (e.g., same number and understanding of different damage levels). Without a common understanding service usage may be possible on a technical level, but meaningful interoperability on higher levels (i.e., on pragmatic level) is not possible.

4.2 Efficient initialization of a simulation environment

Service-based architectures are well-suited to satisfy many of the requirements defined in [14] (see also Section 2.1). Especially, requirement NG-3 (reduction of initialization time of a simulation environment) may be satisfied by an Initialization-service that distributes initialization data (e.g., configuration files) to all member applications. Obviously, this requires that all member applications may be remotely initialized without operator interaction for initialization (see Section 4.1).

4.3 Credibility and Fair Fight

Requirements NG-4 and NG-5.2 are also addressable by service-oriented architectures (see also [13]). An excellent area for tool support and automation provides the Federation Object Model (FOM) which has to be defined for each HLA-based simulation environment. The FOM is available in a machine-readable format ever since; new possibilities for tool support and automation arise if also the federation agreements are made available in a machine-readable format instead of textual documentation used up to now. As illustrated in Figure 6, two possible applications are a compatibility check between the FOM and the federation agreements as well as automatic verification of compliance with federation agreements.

A compatibility check between the FOM and the federation agreements would allow finding subtle errors and inconsistencies between these two products.

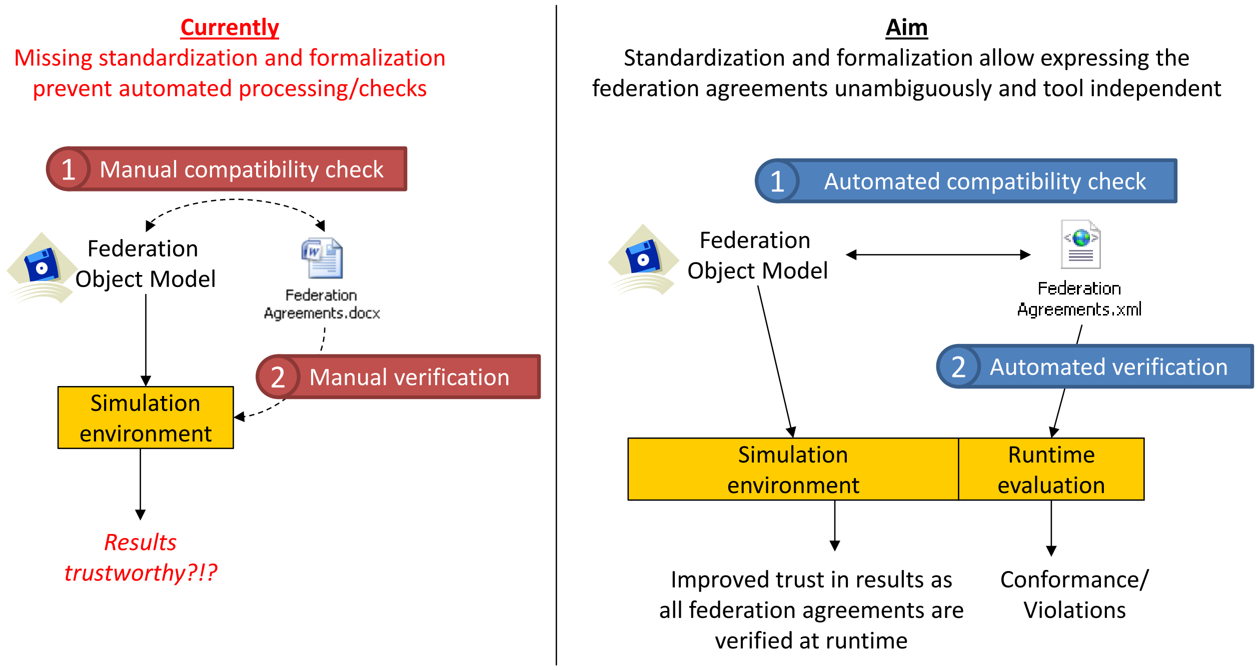

Figure 3: Possible tool support and automation possibilities regarding the Federation Object Model (FOM) and the federation agreements.

Once the simulation environment agreements are available in a machine-readable format and therefore accessible for support tools, a second use case is the runtime evaluation of simulation environment agreements. As illustrated in Figure 3, the simulation environment agreements would be made available to a conformance test service that evaluates during runtime of a simulation environment if any of the simulation environment agreements are violated.

Manual evaluation of a simulation environment whether simulation environment agreements are violated is usually not possible. Oftentimes violations of simulation environment agreements are only noticed because of obvious errors within the simulation execution (e.g., “jumping” objects, unrealistic behavior of objects, etc.). Being able to evaluate the conformance of a simulation environment (in a traceable and provable manner!) with a given set of simulation environment agreements allows time-efficient detection and correction of violations of simulation environment agreements. This increases efficiency of integration and test activities during setup of the simulation environment and finally significantly improves the users trust in the simulation results.

A prototype implementation of such a verification service has been developed by the German Armed Forces (FACTS = Federation Agreements Conformance Test Service). Similar work has been done by ET-35 (“Development of a High Level Architecture Integration, Verification and Compliance Test Tool”) and will be done by MSG-134 (“NATO Distributed Simulation Architecture & Design, Compliance Testing and Certification”). The main difference is that ET-35 and MSG-134 focus on compliance testing before executing a simulation environment while FACTS focuses on compliance testing during runtime. However, the approach is similar and results from MSG-134 should be transferable to compliance testing during runtime and vice versa.

5.0 OPEN ISSUES AND CHALLENGES

As identified and documented by MSG-131, service-based approaches are already utilized in NATO and the nations to a varying degree [9]. However, many issues are still open and need to be solved for a most efficient and effective use of service-based approaches for M&S.

5.1 Missing standards for M&S domain services

Currently no standards exist for describing M&S domain services and for integrating M&S domain services into a simulation environment (see issue CM-05 in [8]). Missing standardized interface specifications for M&S domain services lead to a missing exchangeability and interoperability of M&S domain services. Without standardized interface specifications it is not possible to use different M&S domain services for a specific purpose (e.g., a classified weapon effect service and a non-classified weapon effect service).

5.2 Harmonization of conceptual models

Conceptual models of services and consuming simulation systems must be harmonized (e.g., same damage states) or an appropriate mapping has to be defined. The alignment of conceptual models is crucial for achieving true interoperability and is the key to achieving high quality results (and fair fight conditions) in a simulation environment composed of different simulation systems and services.

5.3 Missing recommendations on integration of M&S domain services

Regarding HLA-based simulation environments, M&S domain services may be integrated into the simulation environment in different ways (e.g., as a special type of federate or as a web service). Not all possible ways of integrating M&S domain services into a simulation environment are conforming with the HLA-standard (e.g., if a communication effects service realizes communication or data exchange without involving the RTI). As part of its work to develop a service landscape, MSG-136 may contribute to this issue.

5.4 Missing standards for verification criteria

Automatic assessment of a simulation environment and all its member applications to be compliant with a specific set of verification criteria requires formal specification of these criteria. Verification criteria may be derived from federation agreements (e.g., regarding naming conventions or interaction classes to be used for certain interactions) or fair fight criteria. With regards to compliance testing, basic work has been done by ET-35 and will be continued by MSG-134.

5.5 IT Security

This paper did not address the whole topic of IT security. In general, existing and proven techniques for ensuring IT security may also be used for the proposed service-based reference architecture. MSG-080 conducted an in-depth investigation of security concerns in distributed simulation environments and outlined potential ways forward [2]. Currently, MSG-128 is working on establishing permanent capabilities for mission training through distributed simulation (MTDS). As part of this work, MSG-128 conducts exercises to validate specific aspects of the devised reference architecture and collects encountered IT security issues and lessons learned on IT security in the MTDS context. Both the results of MSG-080 and initial results of MSG-128 indicate the requirement to investigate IT security in distributed, possibly service-based, simulation environments in more detail.

6.0 SUMMARY AND OUTLOOK

This paper presents an outline of a service-based reference architecture for Modeling & Simulation (M&S). For this purpose a layered architecture is presented and integration of different types of services into a simulation environment is discussed. Examples for different services are given to illustrate the proposed architecture and to underline that service-based approaches for M&S are already being used.

The most important next steps are to refine the outlined reference architecture with respect to protocols and standards to be used (e.g., for integrating a weapon effects service into a simulation environment) and to validate the reference architecture in actual simulation environments. These steps are both currently being worked on. While MSG-136 will refine the integration of services into a simulation environment and will provide a path forward with regards to service-specific standardization activities, MSG-106 and MSG-128 collect experiences using reference architectures in their respective application domain.

7.0 REFERENCES

[1] J.J. Boomgaardt, K.J. de Kraker, R.M. Smelik, "Achieving a Level Playing Field in Distributed Simulations", SISO 2007 Fall Simulation Interoperability Workshop, Orlando, FL, USA, 07F-SIW-009.

[2] Stella Croom-Johnson, Wim Huiskamp, Björn Möller, "Security in Simulation – A Step in the Right Direction", SISO 2013 Fall SIW, Orlando, FL, USA, 13F-SIW-009.

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

[4] IEEE: IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). 2011. – IEEE Std 1730-2010.

[5] Rachid Khayari, Karl-Heinz Neumann, Stephane Devaud, Jean-Pierre Faye, David Desert, José Ruiz, Björn Löfstrand, "NATO Education and Training Network (NETN): Logistics and Transfer of Control FOM Modules Harmonization", SISO 2011 Fall SIW, Orlando, FL, USA, 11F-SIW-012.

[6] Martin Krückhans, "ISO and OGC Compliant Database Technology for the Development of Simulation Object Databases", Proceedings of the 2012 Winter Simulation Conference, Berlin, 2012.

[7] NATO: Final Report of MSG-068 "NATO Education and Training Network", TR-MSG-068, February 2012.

[8] NATO: Final Report of MSG-086 "Simulation Interoperability", TR-MSG-086, 2013, available as pre-release.

[9] NATO: Final Report of MSG-131 "Modelling and Simulation as a Service: New concepts and Service Oriented Architectures", not published yet.

[10] NATO: NATO Modelling and Simulation Master Plan, Version 2.0, 14 September 2012, Document AC/323/NMSG(2012)-015.

[11] NATO Consultation, Command and Control Board (NC3B): NATO Architecture Framework. Version 3, AC/322-D(2007)0048-AS1. 2007.

[12] NATO Standardization Agency: "STANAG 4603 (Edition 1) – Modelling and Simulation Architecture Standards for Technical Interoperability: High Level Architecture (HLA)", NSA 0719(2008)-4603, 02 July 2008.

[13] Robert Siegfried, Johannes Lüthi, Günter Herrmann, Matthias Hahn, "How to ensure Fair Fight in LVC Simulations: Architectural and Procedural Approaches", NATO MSG Conference 2011, Bern, Switzerland.

[14] Robert Siegfried, Michael Bertschik, Matthias Hahn, Günter Herrmann, Johannes Lüthi, Martin

Rother, "Effective and Efficient Training Capabilities through Next Generation Distributed Simulation Environments", NATO MSG Multi-Workshop 2013, Sydney, Australia.

[15] Ralf Stüber, Martin Krückhans, "Increased sustainability of Simulation Object Databases using international norms and standards", NATO MSG Conference 2012, Stockholm, Sweden.

[16] Andreas Tolk, "The Levels of Conceptual Interoperability Model", SISO 2003 Fall SIW, Orlando, USA, 03F-SIW-007, September 2003.

[17] Andreas Tolk, Charles D. Turnitsa, Saikou Y. Diallo, Leslie S. Winters, "Composable M&S Web Services for Net-Centric Applications", The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, Vol. 3, No. 1, Pages 27-44, January 2006.

[18] Wenguang Wang, Andreas Tolk, Weiping Wang, "The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S", Proceedings of the 2009 Spring Simulation Multiconference, San Diego, USA, April 2009.