Effective and Efficient Training Capabilities through Next Generation Distributed Simulation Environments

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

robert.siegfried@aditerna.de

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

michaelbertschik@bundeswehr.org

Matthias Hahn
Planungsamt der Bundeswehr
Oberspreestr. 61L
12439 Berlin,
GERMANY

matthias3hahn@bundeswehr.org

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

guenter.herrmann@unibw.de

Johannes Lüthi
Fachhochschule Kufstein Tirol
Andreas Hofer-Str. 7
6330 Kufstein,
AUSTRIA

johannes.luethi@fh-kufstein.ac.at

Martin Rother
IABG mbH
Einsteinstr. 20
85521 Ottobrunn,
GERMANY

rother@iabg.de

ABSTRACT

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 a primary means to integrate real equipment and simulation assets for training purposes and system evaluation.

Current distributed simulation environments suffer from time- and cost-intensive development and initialization procedures. Furthermore, limited credibility resulting from unknown validity and ad-hoc processes is a serious problem.

To overcome these problems, MSG-086 "Simulation Interoperability" analyzed 46 currently prevailing issues that limit true interoperability and discussed possible solution approaches for these issues. Based on these findings and results from national research and development projects of the German Armed Forces, we present requirements and recommendations for next generation distributed simulation environments. The impact of these recommendations on providing effective and efficient training capabilities to NATO forces is illustrated and potential benefits are highlighted.

A key finding is the need to address simulation interoperability as a requirement for such distributed simulation environments not only on a technical level (as has been done in the past and up to now), but to address simulation interoperability more and more on higher levels of system interoperability. This includes agreements and standardization on semantic and pragmatic aspects to ensure fair-fight and to enable rapid provision of distributed simulation services.

The next generation simulation environments outlined in this paper directly address the objectives defined in the NATO M&S Master Plan (Version 2.0, September 2012) and detailed traceability is provided.

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 a primary means to integrate real equipment and simulation assets for training purposes and system evaluation.

Current distributed simulation environments suffer from time- and cost-intensive development and initialization procedures. Furthermore, limited credibility resulting from unknown validity and ad-hoc processes is a serious problem. A detailed description of current problems will be given in section 2.5.

1.1 MSG-086 "Simulation Interoperability"

In 2009 the Exploratory Team (ET) 027 of the NATO Modeling and Simulation Group (NMSG) identified various issues that severely limit simulation interoperability. Based on the findings of ET-027, MSG-086 "Simulation Interoperability" was initiated in 2010 and tasked to analyze these interoperability issues in order to recommend and prototype information products augmenting the Distributed Simulation Engineering and Execution Process (DSEEP) [5] to mitigate or obviate identified interoperability issues.

MSG-086 has the following objectives:

- Get common understanding of simulation system interoperability and of the structure of interoperability.
- Get common understanding of interoperability aspects related to the different levels of interoperability.
- Propose content and structure of required information products and determine the relation between these information products as described in the DSEEP to support interoperability at all levels.
- Get common understanding of the development of these information products by providing prototypes.

The three-year term of MSG-086 ends in November 2013. Until then MSG-086 "Simulation Interoperability" has achieved the following major results:

- Identification of 46 issues that limit simulation interoperability.
- Detailed description of all interoperability issues, including pointers to possible solution approaches.
- Development of a "Guideline on Scenario Development for (Distributed) Simulation Environments" that augments the DSEEP with regards to scenario development and scenario management.

- Proposal of recommendations on the way forward to achieving higher simulation interoperability.

Additionally, MSG-086 delivers the following secondary outcomes and deliverables:

- Recommendations for updating scenario-related aspects of AMSP-01 [10].
- Recommendations for updating the M&S gap list of the NMSG Military Operational Requirements Subgroup (MORS).
- Recommendations and change requests for SISO working groups (especially DSEEP and FEAT).

1.2 NATO M&S Master Plan

The NATO M&S Master Plan (NMSMP) [14] defines the M&S strategic plan for NATO and is binding on NATO organizations. The NMSMP 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. As the NATO M&S Master Plan is the strategic guidance document for M&S within NATO, this paper documents how the requirements on next generation simulation environments are related to the NMSMP objectives.

2.0 BACKGROUND INFORMATION AND CURRENT PROBLEM AREAS

2.1 What are "distributed simulation environments"?

The DSEEP defines a "simulation environment" as follows:

> „A simulation environment is 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." [5]

The adjective "distributed" refers to the fact that a simulation environment is geographically distributed (i.e., at least across multiple buildings) or logically distributed (e.g., across multiple computing nodes) but executes like a single overall model [2].

Components of a distributed simulation environment are so-called "member applications" [5] (or "federates" in usual parlance) and may be live, virtual, or constructive assets. Many simulation architectures have evolved in the past to interconnect member applications (federates) into a simulation environment (federation), of which the most popular are High Level Architecture (HLA) [4] and Distributed Interactive Simulation (DIS) [3].

2.2 Why do we need distributed simulation environments?

Although reasons for using distributed simulation environments are manifold, they may be broadly split into two categories: operational reasons and technical reasons. Most important are operational reasons. Operational reasons for distributed simulation environments include all use cases that are either impossible to realize otherwise or require more efforts otherwise. Typical operational reasons for distributed simulation environments include (but are not limited to):

- Training requirements that cannot be satisfied otherwise (e.g., combined training, joint training, training of critical conditions that are hard to achieve in reality).
- Test and evaluation of current and future systems (i.e., stimulation of live equipment), e.g., as part of the Distributed National Battle Labs (DNBL) initiative.
- Analysis on tactical and operational level (e.g., as part of a simulation-based acquisition process).

Besides purely operational reasons, there are also numerous technical and organizational reasons for distributed simulation environments, like:

- Member applications (e.g., expensive simulators) are only available in different locations.
- Member applications (e.g., simulation systems) are of different type or from different manufacturers and may not be connected otherwise. This situation implies also that member applications may have different age and functionalities.
- Reduction of travel efforts due to a distributed training audience (i.e., "move information, not people").
- Cost reductions compared to live exercises.

Distributed simulation environments are often complex and expensive activities. As many tasks may be easier solved without using a distributed simulation environment, good reasons are required for setting up a distributed simulation environment. Within the remainder of this paper, it is assumed that a distributed simulation environment is indeed required to satisfy specific user needs. Under this assumption, distributed simulation environments are a technical solution to satisfy specific user requirements.

2.3 Which steps are required for developing a distributed simulation environment?

Development of a distributed simulation environment usually involves many different persons, like the user/sponsor, subject matter experts, M&S experts, operator personnel, etc. The Distributed Simulation Engineering and Execution Process (DSEEP) [5] provides guidance on the activities that are required to prepare, setup, and execute a distributed simulation environment.

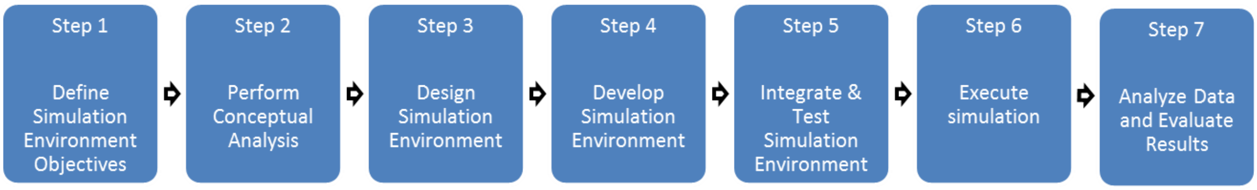

Figure 1: Top-level process as defined by the DSEEP [5].

Figure 1 illustrates the seven step top-level process as defined by the DSEEP. The DSEEP provides further details as each step is divided into distinct activities and inputs and outputs of each activity are identified. It is important to note that the involvement of the different stakeholders varies a lot during the process, e.g., the user/sponsor is mainly involved in step 1 for specifying the operational requirements but much less involved in steps 2 to 5.

2.4 What is meant by "Simulation Interoperability" and "Fair Fight"?

The term "simulation interoperability" (or just "interoperability") is frequently used with different meaning and intention. Within MSG-086 different approaches for capturing and structuring simulation interoperability have been investigated. Finally, the Levels of Conceptual Interoperability Model (LCIM) [21, 22, 23] was selected as preferred approach for capturing the notion of simulation interoperability.

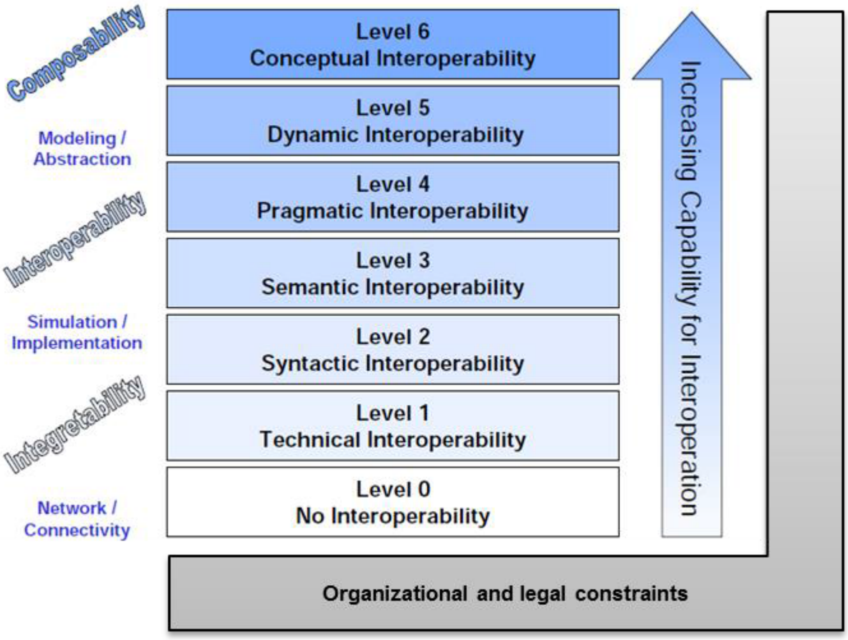

Figure 2: Levels of Conceptual Interoperability (Figure adapted from [22])

As shown in Figure 2 the LCIM defines six levels of increasing interoperability between simulation systems. Detailed descriptions are provided in [21, 22, 23]. The cross-cutting "organizational and legal constraints" were introduced by MSG-086 to reflect the fact that simulation interoperability may also be limited due to organizational or political reasons (e.g., due to intellectual property rights or security concerns).

One major conclusion of MSG-086 is that simulation interoperability needs to be addressed in a holistic way along the whole simulation environment engineering process (e.g., DSEEP) and on all levels of interoperability. It has to be stressed that simulation interoperability is not primarily a technical problem. Achieving simulation interoperability requires efforts and standardization on the technical, the syntactical, the semantic, and the pragmatic level.

Tightly related to simulation interoperability is the term "fair fight". The term "fair fight" is regularly used in context of distributed simulation environments and although everybody has an intuitive feeling about what is meant by this term, a good official definition is still lacking. The following definition was developed in national research projects and was proposed by the authors for inclusion in an upcoming version of the US DoD M&S Glossary [2]:

> "Fair Fight exists among two simulation systems if the differences in representing reality in the simulation models do not lead to a systematic and model immanent advantage and consequently unrealistic simulation results for one of the simulation systems." [19]

It is important to stress that fair fight may always only be defined for a specific simulation environment. There is no "absolute fair fight". Two (or more) simulation systems can only be in fair fight with regards to the objectives and constraints of a specific simulation environment. Therefore, fair fight problems result from differences in the representation of the real world that are not acceptable for a specific simulation environment.

Obviously, different simulation systems use a different representation of the real world. This is perfectly fine and will always be the case (as for example, simulation systems are developed for different purposes or by different developers). A different representation of the real world does not necessarily prohibit fair fight as long as these differences are compatible with the requirements for a specific simulation environment (e.g., if communication effects are not important for a specific simulation environment, fair fight is achieved even if communication between units is modeled differently by two simulation systems).

Achieving fair fight requires simulation interoperability on all levels (as described above) whereas simulation interoperability not automatically guarantees fair fight.

2.5 What are the current problems regarding simulation interoperability?

Current distributed simulation environments suffer from two major problems:

1. High efforts for preparation, initialization, execution, and analysis of a distributed simulation environment in terms of time, costs, and resources (mainly personnel).

2. High efforts for validation and verification of distributed simulation environments, i.e., achieving fair fight and credible simulation results require a lot of time and money (if achievable at all).

The underlying reasons for these problems are manifold and originate from all levels of interoperability. Detailed catalogues and descriptions of simulation interoperability problems are available from MSG-086 [13, 20] and from the NIAG Study Group 162 [15].

3.0 NEXT GENERATION DISTRIBUTED SIMULATION ENVIRONMENTS

3.1 What is the vision for next generation distributed simulation environments?

The vision for next generation distributed simulation environments can easily be derived from the NATO Modelling and Simulation Vision [14] and is defined by the authors as follows:

> "Exploit distributed simulation environments to their full potential across NATO and the Nations to enhance both operational and cost effectiveness while consistently achieving specified levels of fidelity and credibility."

This vision aims at distributed simulation environments that satisfy the users' needs with predictable and reliable quality.

3.2 Requirements on next generation distributed simulation environments

The vision as stated above needs to be operationalized and broken down into measurable requirements. Based on personal experiences of the authors and taking into account recent results of national and international research projects the high-level requirements on next generation distributed simulation environments are as follows:

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.

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

In software engineering non-functional requirements (e.g., regarding security or scalability) are regularly considered as major impact factors for software architecture and software design. The same is true with regards to distributed simulation environments: While functional requirements (like NG-3.1) are relatively 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 on non-functional requirements on architecture and design of next generation distributed simulation environments actual objectives are specified. The requirements NG-2 and NG-3.2 are considered as major drivers and as such they are chosen ambitious.

4.0 RECOMMENDATIONS FOR NEXT GENERATION SIMULATION ENVIRONMENTS

4.1 Recommendations on system design

The following recommendations are concerned with simulation systems (and other member applications) that are likely to be part of distributed simulation environments. These recommendations target at easier integration of new or adapted systems into a next generation distributed simulation environment.

4.1.1 Recommendation SD-1: Design and document for interoperability

Probably the most important recommendation is to design and document a simulation system (or any other member application) for interoperability. ‘Interoperability’ as a requirement needs to be considered from the very beginning when developing or adapting a simulation system. To design for interoperability requires advocating for modularity and changing execution conditions. Hard coded algorithms, fixed configurations, and tacit assumptions need to be avoided.

Documentation for interoperability requires thoroughly documenting assumptions and limitations of simulation systems. Documenting assumptions and limitations is of great importance as this information is absolutely required for achieving interoperability on higher levels (i.e., on pragmatic to conceptual level). Furthermore, interoperability experiences made with a simulation system should be documented (e.g., Lessons Learned from past simulation environments and experiments) to avoid repeating work and allowing faster evaluation whether a specific integration of a simulation system into a simulation environment is feasible.

4.1.2 Recommendation SD-2: Design and document for modularity and composability

Distributed simulation environments are composed of multiple simulation systems (member applications), i.e., simulation systems are components of a simulation environment. Similarly, simulation systems themselves should be designed in a modular way and built from smaller components. Modularity and composability are two sides of a coin and need to be considered jointly. These two terms are often used synonymously and express the fact that a system is composed of other systems (modules) and that exchanging single modules is rather the rule than the exception.

To design for modularity and composability requires planning for exchangeability of algorithms, calculations, data, etc. Typical approaches towards modularity are object-oriented decomposition of a system (e.g., a cruise missile consists of guidance system, payload, and propulsion system) or functional decomposition of a system (e.g., a cruise missile has to execute functions for imaging, path finding, etc.). Although object-oriented approaches are more common, functional decomposition might be better in terms of modularity and reusability of modules [8, 9].

To document for modularity and composability requires documenting interfaces and relationships between modules. This documentation has to span all levels of interoperability. Furthermore, the decomposition strategy has to be documented to allow evaluating the pragmatic interoperability of simulation systems.

4.1.3 Recommendation SD-3: Favor open standards

Simulation systems that are intended to be used within distributed simulation environments should use open standards wherever possible. In general, compliance of a system with standards (not necessarily open standards) allows easier integration into a simulation environment. Also, it is more likely that additional tools (like e.g., gateways, data analyzer, etc.) are available for established standard protocols and formats (see also [10] for more reasons).

Open standards should be favored compared to closed or de facto standards (see [10] for details on terminology). The openness provides the additional benefit that a standard may be implemented more easily and avoids the danger of vendor lock-in that is immanent to closed standards. Furthermore, open standards are usually developed and maintained by a standards development organization (like SISO or OGC) that encourages participation in the development process and ensures long-term availability of a standard.

Examples for potential use of open standards are:

- Use open interface standards for initialization of systems like OGC WFS, OGC WMS, or MSDL.
- Use open data models like SEDRIS.
- Use reference data exchange models like RPR-FOM (open standard) or NETN FOM (potential future de facto standard [12]).

- Use open simulation execution control patterns as proposed by MSG-052 [11] (to be standardized).
- Use open control mechanisms like Distributed Debrief Control Architecture (DDCA) that is currently under development by SISO.

4.1.4 Recommendation SD-4: Design for securability

As distributed simulation environments are often faced with security concerns (e.g., due to different nations participating), simulation systems that are intended for use within next generation distributed simulation environments need to be designed for securability. In this context, securability is defined as the extent to which a simulation system (member application) is securable. This addresses especially the ability of a simulation system to interoperate on different security levels.

Many approaches are currently used when connecting differently classified systems, like “System high”, “Multiple single levels of security”, or “Multiple independent levels of security” [1]. As these approaches come with many drawbacks they are not recommended for next generation distributed simulation environments. Instead next generation simulation environments require an approach that enables a flexible combination of differently classified systems.

Designing for securability includes:

- Enabling a simulation system to use differently classified data (e.g., via different data sets that are provided by removable disk drives or usage of different service implementations) and differently classified algorithms. This recommendation is tightly related to SD-2 “Modularity”.
- Enabling a simulation system to connect to differently classified networks. This may also affect physical infrastructure issues (e.g., building etc.).

4.2 Recommendations on simulation environment infrastructure

The following recommendations are concerned with infrastructure issues regarding next generation distributed simulation environments. These recommendations target at faster setup processes and more credible simulation environments.

4.2.1 Recommendation IN-1: Harmonize critical data and algorithms

Currently, many problems are caused by incompatible data or algorithms of participating simulation systems (e.g., different visual representation of critical assets, different algorithms for computing weapons effects, etc.). To overcome these problems, critical data and algorithms have to be harmonized. Obviously, the decision which data and algorithms are critical depends on the application area of a simulation environment and cannot be generally determined. However, most military simulation environments have commonalities that are regularly considered critical (e.g., synthetic natural environment data, weapons effects calculation, communication effects calculation, etc.).

As a first step a harmonization of the identified critical data and algorithms is required. This may be achieved by providing (free-text) specifications for identified critical data and algorithms. Every simulation system may implement these critical components separately as long as the specifications are satisfied.

In a second step, dedicated components may be provided (e.g., as software libraries) such that redundant implementation efforts are reduced or eliminated. Taking this a step further, these components may be treated as services that are centrally deployed and utilized by many simulation systems (i.e., software as a service).

Finally, the components and services need to be standardized (on the technical, syntactical, semantic, and pragmatic level). This allows different realizations of components and services (e.g., a classified weapons effects service and a non-classified one) and goes hand-in-hand with recommendations SD-2 "Modularity" and SD-3 "Favor open standards".

4.2.2 Recommendation IN-2: Establish permanent simulation infrastructure

Significantly improving preparation and setup times requires the establishment of a permanent simulation infrastructure. This includes

- network connections (e.g., between different sites or nations),
- simulation environment control facilities,
- provision of shared components and services in a "Defense Cloud" (e.g., nation-wide or NATO-wide), and
- provision of an information management system that supports the whole simulation environment engineering process (e.g., with regards to documentation, execution planning, file sharing, etc.).

Depending on the actual requirements (especially NG-2 and NG-3) the amount of permanent simulation infrastructure may vary. As the permanent simulation infrastructure is an essential part of next generation simulation environments it has to be documented thoroughly (see PO-2 "Use a systems engineering process and document decisions").

Experiences from many national simulation environments have shown that a permanent simulation infrastructure is a key to achieving significantly faster and more reliable setup processes. It has to be stressed that establishment of a permanent simulation infrastructure does not only concern technical issues, but also establishment of a permanent support organization with skilled and experienced personnel.

4.2.3 Recommendation IN-3: Establish member application compliance testing

Fast and reliable development processes for next generation distributed simulation environments require automated compliance testing of participating simulation systems and other member applications (e.g., command and control systems). The automated compliance testing has to include test cases on all interoperability levels:

- Technical level: test compliance with TCP/IP, HLA interfaces, etc.
- Syntactical level: test compliance with interface syntax specifications.
- Semantic level: test compliance with data exchange model (e.g., with a specific FOM in HLA-based simulation environments).
- Pragmatic level: test compliance with conceptual models (e.g., with the service consumer-provider pattern [6]).

Regarding federates for HLA-based simulation environments this topic is currently investigated by ET-35 ("HLA Federation Compliance Test Tool"). In national research projects the experimental tools FACTS (Federation Agreements Conformance Test Service) and FIERS (Federation Integration and Experimentation Rehearsal Surrogate) are used to verify a specific subset of federation agreements and to easily provide mock-up federates for test purposes. A similar approach for testing simulation gateways is described in [7].

Configuration of such a compliance testing tool should be via a standardized data format (ideally defined in an open standard). Reuse of parts of this configuration is required (e.g., in form of "configuration modules") for efficient handling of recurring test cases (e.g., testing compliance with RPR FOM).

4.2.4 Recommendation IN-4: Establish simulation environment execution compliance testing

Achieving reliable and credible simulation results requires continuous monitoring of a simulation environment execution. Deviations from specified behavior, errors, etc. need to be detected and assessed whether they influence the simulation execution and simulation results.

Due to the distributed nature and the manifold data exchange between participating simulation systems and other member applications such a monitoring and assessment cannot be done manually but has to be done in an automated fashion. Execution compliance testing is similar to member application compliance testing as described in IN-3 but takes this one step further. Especially for ensuring credibility in simulation results continuous monitoring and assessment of a simulation environment execution is required.

Configuration of such a compliance testing tool should be via a standardized data format (ideally defined in an open standard). Reuse of parts of this configuration is necessary (e.g., in form of “configuration modules”) for efficient handling of recurring test cases (e.g., that only RPR-FOM compliant interaction messages are used, or that specific fields within a message are used correctly).

4.3 Recommendations on simulation environment engineering processes and organization

The following recommendations are concerned with organizational issues regarding next generation distributed simulation environments. These recommendations target at more reliable processes and more credible simulation environments.

4.3.1 Recommendation PO-1: Enforce requirements specification

Although seemingly obvious, the recommendation to enforce good requirements specifications is explicitly made. This includes all types of requirements (e.g., regarding desired terrain, required terrain fidelity, participating units, etc.) and includes also quality and fair fight requirements. The last two are often not specified explicitly but taken for granted or implicitly assumed. The resulting problem is: How to assess quality or fair fight if they are not specified and agreed upon?

Besides organizational measures (procedures, etc.) it is recommended to assist the user as much as possible. Good experiences were made using checklists for elicitation of typical quality or fair fight requirements. Also documentation templates have proven to be useful for ensuring more complete requirements specifications.

Talking about next generation simulation environments, dedicated information management systems should be established that further assist users during the requirements specification process and throughout the whole simulation environment engineering process (see IN-2).

4.3.2 Recommendation PO-2: Use a systems engineering process and document decisions

Setting up a distributed simulation environment is a complex task and requires professional management. Therefore, the recommendation is to use an appropriate systems engineering process to ensure that all persons involved in the process have a common understanding about ongoing activities and expected deliverables.

The DSEEP [5] is an obvious choice for such a systems engineering process. As it provides a generalized, high-level framework the DSEEP has to be adapted to the individual needs of an organization. The VEVA process model is an example of such an organization-specific adaptation of the DSEEP that is used by the German Armed Forces [16].

Besides the choice of a systems engineering process, a decision has also to be made regarding its documentation. Several approaches and standards may be used for documenting simulation environments (e.g., Base Object Models [18], or the NATO Architecture Framework [13]).

4.3.3 Recommendation PO-3: Establish simulation repository

In alignment with IN-2 (provision of an information management system that supports the whole simulation environment engineering process), it is recommended to establish a simulation repository.

Such a simulation repository (or information management system) should support the user throughout the whole simulation environment engineering process (e.g., using the DSEEP) and should provide a central repository for all kind of documentation as well as file storage.

4.4 Recommendations on simulation environment data

The following recommendations are concerned with all types of data that are likely to be part of distributed simulation environments. These recommendations target at faster setup processes of next generation distributed simulation environments and more credible simulation environments.

4.4.1 Recommendation DA-1: Enforce "single source of truth" principle

Many data-related interoperability problems are caused by different simulation systems using different data that is uncorrelated (although it is expected to represent the same facts). A typical example is a simulation environment with two simulators that use individually manufactured terrain databases of the same area. Although the same area is represented by the terrain databases, differences between the terrain databases are often large enough to cause severe interoperability problems and fair fight violations.

The "single source of truth" principle requires that each source data item is stored only once and that all application-specific data items or data formats are derived from this source data item. Figure 3 illustrates the "single source of truth" principle for terrain data.

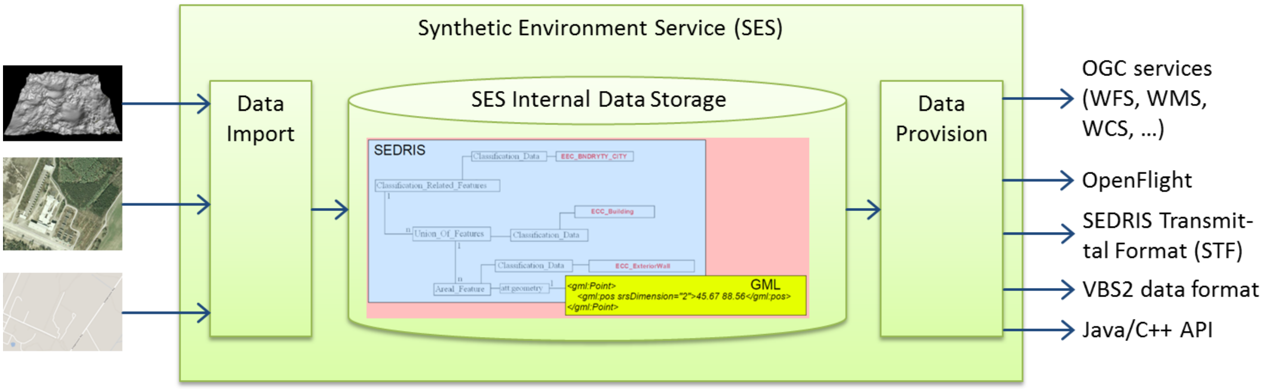

Figure 3: Basic idea of a Synthetic Environment Service (SES) that realizes the "single source of truth" principle for terrain data.

This recommendation is tightly related to SD-2 (Modularity) and SD-3 (Favor open standards). Realizing the "single source of truth" principle is ideally accompanied by establishing a permanent infrastructure for storage and maintenance of the original source data (see IN-2 "Establish permanent simulation infrastructure").

4.5 Relations of requirements, recommendations, and NATO M&S Master Plan objectives

Table 1 documents the relations of requirements (as defined in Section 3.2), recommendations (as described in this section) and objectives laid out by the NATO M&S Master Plan [14].

Table 1: Relations of requirements, recommendations, and NATO M&S Master Plan objectives.

Requirement	Related recommendations	Contribution to NATO M&S Master Plan Objectives
NG-1	PO-1	IV.1 (Plan employment)
NG-2	all	I.2 (Establish recommended standards pertaining to data interchange for M&S and C2 systems, promotion of true interoperability, pursue trust in M&S)
NG-3		
NG-3.1	SD-3, DA-1	IV.2 (Provide resources to operate simulations)
NG-3.2	IN-2	IV.2 (Provide resources to operate simulations) IV.3 (Provide databases)
NG-4	SD-1, IN-1, PO-1	II.1 (Develop common process and procedures to guide actions and decisions regarding M&S application)
NG-5		
NG-5.1	PO-1, PO-3	I.2 (Establish recommended standards pertaining to data interchange for M&S and C2 systems, promotion of true interoperability, pursue trust in M&S)
NG-5.2	IN-4	I.2 (Establish recommended standards pertaining to data interchange for M&S and C2 systems, promotion of true interoperability, pursue trust in M&S)
NG-5.3	(No recommendation yet, might be similar to IN-3 and IN-4)	I.2 (Establish recommended standards pertaining to data interchange for M&S and C2 systems, promotion of true interoperability, pursue trust in M&S)
NG-6		
NG-6.1	PO-2, PO-3	II.1 (Develop common process and procedures to guide actions and decisions regarding M&S application)
NG-6.2	PO-2, PO-3	II.1 (Develop common process and procedures to guide actions and decisions regarding M&S application)
NG-6.3	PO-3	II.4 (Promote the sharing of M&S resources through a knowledge management process and system)

5.0 WAY FORWARD

5.1 How to realize next generation distributed simulation environments

Realization of next generation distributed simulation environments as described in this paper will not happen at once. Due to financial implications and technical considerations a step-by-step approach has to be taken. The following steps are identified on the way to next generation distributed simulation environments:

1. Requirements analysis and identification of user needs.
   The starting point of all activities has to be a requirements analysis that thoroughly defines the operational and technical reasons for utilizing distributed simulation environments (see Section 2.2).
2. Identification of required simulation systems.
   Based on the requirements analysis the simulation systems have to be identified that will participate in future distributed simulation environments. This includes existing simulation systems as well as identification of missing simulation systems.
3. Prioritization and stepwise realization.
   The simulation systems have to be prioritized according to user needs, financial constraints, etc. and step-by-step newly developed or upgraded to comply with the recommendations described in this paper (see Section 4).

Obviously, all steps may be further subdivided into finer-grained activities. In context of this paper especially step 3 is of interest and deserves further attention. Stepwise realization refers to both: not all simulation systems are developed or upgraded at the same time, and not all recommendations are realized for a single simulation system at the same time.

Step 3 needs to be accompanied by international activities for coordination of efforts and necessary standardization activities. Standards required for next generation simulation environments that do not yet exist include (but are not limited to):

- Standards related to services (i.e., interfaces, semantic, etc.) (see IN-1)
- Federate conformance rules (see IN-3)
- Federation execution conformance rules (see IN-4)

Also, existing standards need to be maintained and, if necessary, updated (e.g., MSDL [17]). Ongoing exchange between users and developers about experiences, lessons learned, limitations, etc. is required throughout the whole process and considered extremely important.

5.2 Current and future activities

As described in this paper (and as experienced in many projects done by the Bundeswehr in the past), technical solutions alone are not sufficient. Instead, all levels of interoperability need to be addressed requiring technical, architectural and organizational measures [19].

Based on experiences from German research projects service-based approaches are very promising for realizing next generation distributed simulation environments. Service-based approaches are well suited to satisfy many of the recommendations outlined in this paper directly (e.g., SD-2 “Modularity”, IN-1 “Harmonize critical data and algorithms”) and provide a good technical basis for satisfying recommendations like IN-2 “Establish permanent simulation infrastructure” and DA-1 “Enforce single source of truth principle”.

The combination of service-based approaches with ideas taken from cloud computing is also known as "Modeling & Simulation as a Service" (MSaaS). Currently, the Specialist Team MSG-131 ("Modelling and Simulation as a Service: New concepts and Service Oriented Architectures") is doing first steps towards MSaaS. For a more in-depth analysis a task group on MSaaS is proposed that should continue the work of MSG-131 (thus starting in fall 2014). MSG-131 and a potential follow-on task group directly address objective I.1 of the NMSMP ("Develop NATO standard interoperability architecture").

6.0 SUMMARY AND OUTLOOK

To a great extent, future military training capabilities will be provided by 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. Current distributed simulation environments suffer from time- and cost-intensive development and initialization procedures. Furthermore, limited credibility resulting from unknown validity and ad-hoc processes is a serious problem.

To overcome the problems of current distributed simulation environments this paper defines measurable requirements on next generation distributed simulation environments. Detailed recommendations are given how to satisfy these requirements when developing new simulation systems or updating existing ones.

Next generation simulation environments will heavily rely on open standards and service-based architectures. A roadmap is provided how to realize next generation distributed simulation environments step-by-step.

7.0 REFERENCES

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

[2] Department of Defense: Modeling and Simulation (M&S) Glossary, 1 October 2011.

[3] IEEE: IEEE Standard for Distributed Interactive Simulation. 2012. IEEE Std 1278.1-2012.

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

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

[6] 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, USA, 11F-SIW-012.

[7] Robert Lutz, David Drake, Ryan Brunton, Michael O'Connor, Kurt Lessmann, Dannie Cutts, "A LVCAR-I Retrospective: Tools, Processes, and Specifications for Gateway Selection and Configuration", SISO 2013 Fall SIW, Orlando, USA, 13F-SIW-011.

[8] Tony Moynihan, "An experimental comparison of object-orientation and functional-decomposition as paradigms for communicating system functionality to users", Journal of Systems and Software, Vol. 33, Issue 2, May 1996, Pages 163–169.

[9] Ph. Narbel, "Functional Programming at Work in Object-Oriented Programming", in Journal of Object Technology, Vol. 8, No. 6, September-October 2009, pp. 181-209, http://www.jot.fm/issues/issue_2009_09/article5/

[10] NATO: Allied Modeling & Simulation Publication (AMSP) 01 – M&S Standards Profile.

[11] NATO: Final Report of MSG-052 "Knowledge Network for Federation Architecture and Design", TR-MSG-052, November 2011.

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

[13] NATO: Final Report of MSG-086 "Simulation Interoperability", not published yet.

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

[15] NATO Industrial Advisory Group (NIAG) + NAFAG Aerospace Capability Group 6 (ACG 6): Final Report of NIAG SG 162 Study on "Distributed Simulation for Air and Joint Mission Training", 15 October 2012, Document NIAG-D(2012)0022 (PFP).

[16] Robert Siegfried, Günter Herrmann, Johannes Lüthi, Matthias Hahn, "VEVA as German approach towards operationalizing the DSEEP: Overview and first experiences", SISO 2011 Spring SIW, Boston, USA, 11S-SIW-044.

[17] Robert Siegfried, Alexander Laux, Martin Rother, Dieter Steinkamp, Günter Herrmann, Johannes Lüthi, Matthias Hahn, "Scenarios in military (distributed) simulation environments", SISO 2012 Spring SIW, Orlando, USA, 12S-SIW-014.

[18] Robert Siegfried, Alexander Laux, Martin Rother, Dieter Steinkamp, Günter Herrmann, Johannes Lüthi, Matthias Hahn, "Components and reuse in scenario development processes for distributed simulation environments", SISO 2012 Fall SIW, Orlando, USA, 12F-SIW-046.

[19] 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.

[20] Robert Siegfried, Johannes Lüthi, Martin Rother, Dieter Steinkamp, Matthias Hahn, "Limiting issues for true simulation interoperability: A survey done by MSG-086 and resulting recommendations", NATO MSG Conference 2012, Stockholm, Sweden.

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

[22] 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.

[23] 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.