FUTURE OF LVC SIMULATION: EVOLVING TOWARDS THE MSaaS CONCEPT

Jose-Ramon Martinez-Salio
NADS
Madrid, Spain
jrmartinez@nads.es

Jose-Maria Lopez-Rodriguez
NADS
Madrid, Spain
jmlopez@nads.es

ABSTRACT

Live, Virtual, and Constructive (LVC) simulation has been one of the main topics of discussion in Modeling and Simulation (M&S) community in last decade. Reports like "Live Virtual Constructive Architecture Roadmap" (LVCAR) established a baseline to start planning next steps to improve LVC simulations. Based on LVCAR insights, new SISO workgroups, like LSA (Layered Simulation Architecture) or WebLVC have been born. While improvements in performance, usability and scalability of the LVC federations are still a hot topic for discussion a new demand is standing-out; the use of simulations assets as Services. Initiatives in US DoD (JLVC2020) and NATO -Modeling and Simulation as a Service (MSaaS) and Distributed Networked Battle Labs (DNBL)- are trying to change the paradigm of how simulations are developed, deployed and used, looking for a Cloud-based publisher-consumer service paradigm for the assets. This paper analyzes where we are in this quest, pinpointing gaps and main challenges we need to address to be able to do a fluent transition from LVC simulations to MSaaS clouds.

ABOUT THE AUTHORS

Jose-Ramon Martinez-Salio is International Presales Director at Simware, a company of NADS group. He received its M.Eng in Industrial Engineering from the Polytechnic University of Madrid, Spain. He is vice-chairman of SISO LSA group. In the last few years has been involved in NATO groups related with interoperability and distributed simulation.

Jose-Maria Lopez-Rodriguez is CEO at Simware, a company of NADS group. He received its M.Eng in Industrial Engineering from the Polytechnic University of Madrid, Spain. He is working with SISO and NATO groups to improve convergence of distributed simulation standards, enabling the use of simulations as services. He has more than 20 publications in international conferences, mainly related to M&S.

FUTURE OF LVC SIMULATION: EVOLVING TOWARDS THE MSaaS CONCEPT

Jose-Ramon Martinez-Salio
NADS
Madrid, Spain
jrmartinez@nads.es

Jose-Maria Lopez-Rodriguez
NADS
Madrid, Spain
jmlopez@nads.es

INTRODUCTION

The vision of Modeling and Simulation (M&S) as a service (MSaaS) is to offer users M&S solutions wherever they are and whenever they need them. [1] [2] Given a problem specification or a situation that needs to be confronted, a user searches and discovers what resources (systems and/or data) are available. The resources, once discovered, are evaluated with respect to their ability to contribute to the overarching solution, directly or by reuse, interoperation or composition. There might be several configurations that answer to the problem and all these possible candidates or a subset thereof is selected. These candidates are deconflicted in terms of data and/or functionality to ensure that information is not contradictory and that each candidate provides a unique functionality. The next step is to compose the resources such that the resulting composition can contribute a partial or complete solution to the problem. The execution of the resulting composition is orchestrated and results are collected and presented to the user. These results are fused when applicable to offer the user an integrated view of the information so as to facilitate and speed up the decision making process.

The current approach is to use the cloud [3] and offer M&S products as we would any other software product. Since a model is a purposeful abstraction of reality, M&S products cannot be treated as traditional software products that need to be offered as services. At each level of this process several algorithms, heuristics or methods must be used to allow the user to accomplish their goal. At a minimum, additional metadata is necessary in order to know 1) the referent being modeled including key assumptions and constraints in order to understand it 2) the purpose and context of validity in order to use it 3) the technical requirements for integration, 4) the syntactic and semantic requirements for data interoperability and 5) the pragmatic and dynamic requirements for composition. It is not clear whether this metadata should be offered as a standalone discovery service or be an integral part of every M&S service.

In this paper, we review the state of the art in MSaaS and propose a Layered Simulation Architecture (LSA) that takes into account the fact that simulations are simplifications of real world referents and most importantly that simulations are built to answer a modeling question.

The paper is organized as follows: Section 2 describes the requirements for MSaaS; Section 3 describes some potential candidates for MSaaS architecture; Section 4 centers on the alternative proposed by Layered Simulation Architecture and Section 5 discusses the challenges in implementing a fully functional LSA for MSaaS followed by conclusions and future work.

PROBLEM STATEMENT

In this section we analyze the functional requirements of MSaaS. It is important to note that these requirements are for an ideal or final MSaaS model and that real implementations might not fulfill all of them at least initially.

Current requirements for MSaaS derive from Vision in NATO M&S Masterplan and from the work of MSG-086 "Simulation Interoperability" [4].

We can classify requirements at the following high levels: Usability requirements; Network, devices, and physical infrastructures requirements; Security requirements and; Interoperability and Composability requirements. Table 1 shows lower level requirements and their description followed by their high level classification

Table 1. Key Requirements of MSaaS

Requirement	Description	Type
Geographical availability	MSaaS has to be available in the entire NATO area. In some cases, they have to be available out of NATO area (e.g. in some interest area like Libya scenario).	Usability
Time Availability	MSaaS has to be available, ideally, on a twenty hours, seven days basis.	Usability
Multi-Platform	The services have to be running on devices ranging from traditional PCs and laptops to smart phones and tablets. This has powerful implications in terms of local resources that are needed, especially in processor and graphics cards.	Network, devices, and physical infrastructures
Heterogeneous networking	MSaaS have to be reached via any possible kind of network; Wi-Fi, WAN, LAN, etc. The kind of net and connection details have to be transparent to the final user.	Network, devices, and physical infrastructures
Disconnected, Intermittent, Limited (DIL) networks	Network quality cannot be guaranteed everywhere. MSaaS has to be tolerant with a great range of networks conditions.	Network, devices, and physical infrastructures
Low latency	Latencies have to be as low as possible in every device and geographical localization. Otherwise, resulting simulation won´t be useful.	Network, devices, and physical infrastructures
Classification of assets	Services provided have to be tagged in a way that guaranties the following: •Homogeneity of assets: The services provided have to be homogeneous for the same classification/tag. It is necessary to guarantee that the elements in the same level have the same quality (e.g. in terms of fidelity) •Interoperability of assets: it is necessary that the assets in the same level have to be able to run together without the user caring about limitations or interoperability problems. Correct tagging has to be guaranteed: the final user is not going to check if they are true, he will be free to combine any number of services with compatible tags in any possible way.	Interoperability and Composability
Service availability	The services must be available at all times with quality standards and interoperability committed. Since these services can have heterogeneous origins, this requirement has implications in terms of quality assurance, maintainability and compromise of the companies (that will potentially provide the services) involved.	Usability

Concurrency	MSaaS has to be able to serve a lot of users at the same time.	Usability
Legacy support	Integration of existing simulators and simulators assets should be included in MSaaS architecture. Current simulators and their components should be integrated seamlessly in the proposed MSaaS schema.	Interoperability and Composability
Transparence for the user	The entire schema has to be transparent to the final user. All the final user has to do is 1) connect to a generic server not caring about the server or its physical localization, 2) authenticate, 3) choose the combination of services that her device and level of security is able to provide OR continue with previous exercise and 4) run the services.	Usability
Easy and fast composability	User needs to have an easy and fast way to find assets, choose assets and run them. User doesn´t need to care about any composition or interoperation problem.	Interoperability and Composability
Interoperation between users	Users have to be able to build and run joint simulations with other users. This interoperation has to be transparent to the actual geographical localization of the users and of the devices been used.	Interoperability and Composability
Scenario change	The scenarios available have to cover, potentially, the entire world. As a bare minimum they have to cover the NATOs areas of operation or interest. It is necessary to have every possible scenario available and the change between the scenarios have to be easy and fast. Ideally, the user will only have to choose the scenario with a certain level of fidelity from a list to be able to use it.	Usability
Security	Security is one of the main challenges that have to be considered for MSaaS. Security requirements can be considered in a lot of different aspects [5]: • Access Security: it is necessary to consider access security, levels of security, etc. • Data network Security: Data have to travel in a secure way. • Cloud Security: The server side has to be secure. This may imply limiting and controlling servers, mirrors, etc. • Client side security: The devices that access to the services need to have security implemented both in the hardware and the software. E.g. the browsers or similar are a typical weak point in the security.	Security
Fair fight assurance	This requires an objective assessment whether a simulation environment built using services comply with the specified fair fight requirements [6].	Interoperability and Composability
Certification	The services that will be the body of MSaaS have to be certificated by a qualified certification authority. A service life-cycle has to be implemented	Interoperability and Composability

Definition of service: The same definition of service is key to the architecture. Usually "software as a service" (SaaS) is considered part of the nomenclature of cloud computing. Cloud computing concept also includes infrastructure as a service (IaaS) and platform as a service (PaaS) [7]. Thus, the definition of service implies the use of cloud computing and of certain ways of access. Defined in that way, the term "service" is an open term; you can put into it what you want in any possible way. It is clear that the services have to be clearly delimited in the MSaaS architecture. Services included in this architecture have to be clearly classified and tagged around simulation features. The final list of such simulation features will determine admitted services. One example of such a list can be found in DNBL portal [8].

Use of the cloud: The term "simulation as a service" implies the use of cloud technologies to provide the substrate for providing the services. The National Institute of Standards and Technology (NITS) define the cloud as: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction Security Heterogeneity". The use of any kind of cloud has powerful implications in security.

Security: This is one of the main problems and potential stopper for MSaaS. Without being exhaustive, we can consider security from many points of view. Among others, we can consider.

- Device security issues: The devices that are going to connect to the MSaaS need to have security in:
    - Access to the device: A mere logon could not be enough. E.g. some bio-metric security could be implemented to guarantee that the user is habilitated [9].
    - White list of devices: Only a controlled list of devices should be allowed access to the cloud.
    - Auditory of the use of the services. It is necessary to register the use of services, devices that use them (IP, other), users, localization, duration, etc. This auditory has to be implemented in the server side.
    - Authentication of access to the services.
- Security in the "browser" or program that allows the access to the server. This is a problematic point. Here the MSaaS can use a commercial product or implement its own version of the software. Commercial products are full of security patches coming in every new version [10]. On the other hand, developing its own product need to address some of the security vulnerabilities already solved in commercial products and also implies upgrading and distributing updates of the program.
- Security in transition of data: The transition of data to the cloud and from the cloud has to be encrypted. The encryption algorithms have to be closely controlled. Use of encryption will sure affect performance. Also, even the protocols for secure transition, like SSL, can be problematic, like it was demonstrated recently by the Heartbleed Bug [11]
- Security in the cloud. The cloud itself has a lot of identified security issues [12] like distributed denial of service attacks and man in the middle attack.
- Loss of governance: by using cloud infrastructures, the client necessarily cedes control to the Cloud Provider on a number of issues which may affect security. These issues have to be covered by the cloud provider and usually are out of the control of the final user
- Security of the services deployed in the cloud: The services that are available in the cloud can be a potential security thread. E.g. a malicious service can create a trace or copy of every transaction. Later, this information can allow a third party to reconstruct the full simulation exposing its capabilities.
- Security of the services deployed locally: Some of the services will have to install information in the client's side (similar but not limited to cookies in browsers) creating a potential security issue

Heterogeneous technologies and standards: This is another important problem to be considered. As it has been stated in many documents and studies like LVCAR [13] simulation technologies and standards are impossible to connect and interoperate directly without the help of gateways. Moreover, the level of interoperation reached is very low. At the end, the use of different technologies and standards hamper the interoperation process and convert it in a "one time effort" that is made for a specific exercise and never in permanent basis or in a service orientation valid for MSaaS. The only permanent working reference is the US DMO that uses DIS as the unique simulation standard, thus avoiding the gateways. MSaaS has to consider the existence of many different potential services based on different standards like HLA1516, HLA evolved, DIS and DDS. MSaaS should try to use and connect all these assets.

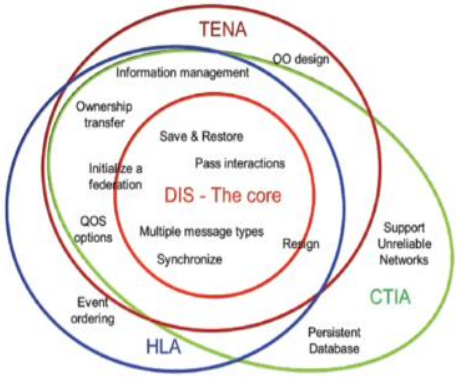

Figure 1: comparison of standards (from LVCAR study)

Architecture of the Simulation systems. MSaaS demands going to a system of systems architecture not only for the federation of simulations but also for the simulators themselves. Interoperability to a service level mandates the use of open architectures into the simulators, only in this way a simulator will be able to share simulation features and services with others simulators in the network. Examples of the required level of interoperability could be a virtual simulator sharing its state machine in order to be able to be controlled by a remote Instructor Station, or a live simulator that it is using simulated sensors provided by a simulation server located in the network. Any of these examples shows the potential of open architectures and MSaaS but also give us a good idea about how far we are from achieving this goal if we don't change the way in which we build a simulation system.

Rest of the paper will deal with different options we can consider for solving this problem. Proposed solutions that we will consider are mainly related with the architecture.

Some architectural solutions

The problem of the different standards and technologies and their incompatibilities can be solved in a number of different ways.

Standardizing gateways: The objective is setting and standardizing the needed gateways between technologies and standards in a way that these gateways will be easy to maintain understand and change. There are already some initiatives like "Gateway Description and Configuration Languages" in SISO [14] that follow this idea. Anyhow, the use of gateways deteriorates performance and is always a weak point, or even a bottleneck, in the final simulation.

In real projects, gateways are usually "ad hoc" solutions that are difficult to change, maintain and escalate. The standardizing of gateways is not incompatible with some other solutions we can consider.

Reducing the number of standards in use: The objective of this approach is reducing the number of standards in use to just one, usually HLA1516-2010. This is the approach of some experiments and experiences in NATO.

Big drawback is that this approach lefts a lot of existing simulation assets out of the MSaaS scope. One of the main ideas of both MSaaS initiative and DNBL services portal is the networking of NATO, national or industrial services. There are a lot of potential services in the industry and the nations, which are very interesting for MSaaS, which are not HLA-Evolved nor are planning to migrate to HLA-Evolved.

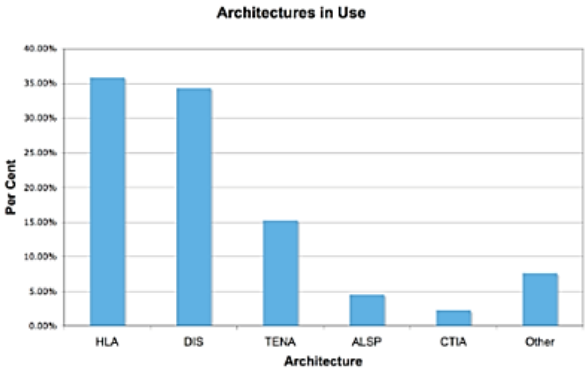

Figure 2: architectures in use (from LVCAR study 2008)

WebLVC: WebLVC is a study group of SISO [15]. As stated in SISO page, its rationale is "*In recent years, several new technologies and standards have been developed in the broad Web community that enables highly interactive, low-latency, real-time web-based applications written in JavaScript. These technologies include WebGL, WebSockets, HTML5, and a variety of JavaScript games engines, scene graphs, etc (…). But what is missing is a standard interoperability protocol for linking these new web-applications with each other, and with traditional M&S federations in a way that is: (…). Flexible enough to support interoperability regardless of the protocol being used in the target federation (e.g. DIS, HLA 1.3, HLA 1516, HLA Evolved, TENA, etc.)* ".The objective of the WebLVC protocol is defining a way of passing simulation data between a web-based client application and a server. This server can be deployed in the cloud.

WebLVC approach is well suited for MSaaS and guarantees transparency about the final standard used in the cloud side. The WebLVC approach can be used for any possible standard and architecture. On the other hand, this approach don´t address the problem of working, mixing together, different standards in the same exercise. Also it is not clear how current simulation services (HLA services) will be created and used in WebLVC.

LSA: LSA (Layered Simulation Architecture) is a proposed standard (now finishing study-group phase in SISO) for creating a common platform to build and interoperate simulators based on current simulation standards [16][17].

The rationale of LSA is based on the desire to apply current ideas on net-centric interoperability and open systems architecture to modeling and simulation. It has been inspired by recommendations of the Live, Virtual, Constructive Architecture Roadmap (LVCAR), and will draw upon advances made by other organizations such as the Network Centric Operations Industry Consortium (NCOIC), Object Management Group (OMG) and World Wide Web Consortium (W3C).

Central to the idea of LSA is the reuse of simulation standards "as is" without changing them via either 1.) the use of standardized gateways, or 2.) direct codification of specific layers of code into the standards. As common communication layer, the proposal is the use of DDS/DDSI (Data Distribution Service) [18] that is a real-time communication standard of the OMG (Object Management Group). Both the use of gateways and the direct

codification of DDS/DDSI as communication layer for different standards (e.g. in HLA1516-2000 and HLA1516-2010) have been proved to be feasible.

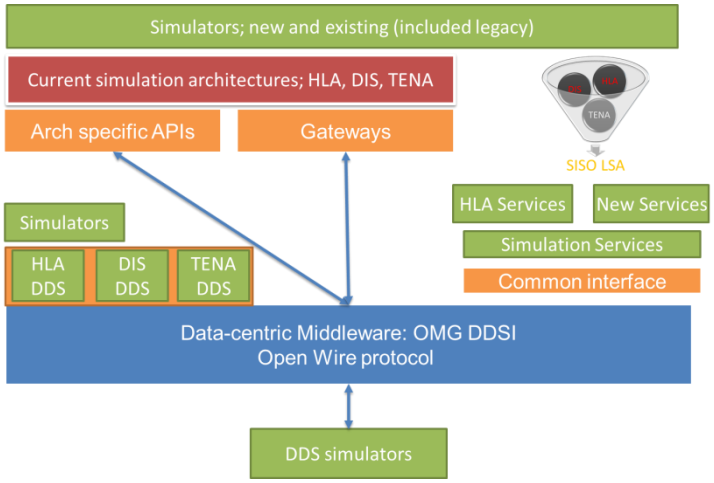

Figure 3: LSA proposed architecture

LSA architecture can include all the standards and architectures like HLA, DIS, DDS and TENA. Data model and services can be any available, without restrictions. LSA can also incorporate legacy assets like simulation services, old simulators, etc. Security (close to multilevel security) can be implemented on top of the DDS communication layer. Finally, it is possible to deploy this architecture on the cloud without any change.

About security, LSA proposal, for now, is the use of "DDS Security" [19] that is currently (July 2014) in Beta state in OMG portal. This DDS security is not yet equivalent to Multilevel Security, but can be used as a first approach to the final security level needed by MSaaS. This DDS security is built on top of DDS and will be provided by DDS vendors. Anyhow, we can still build our own layer of security, as an independent layer, later. The security requisites in MSaaS are not clearly stated yet.

SISO LSA can be applied to any level of aggregation. Because of that it can be used as System of Systems (SoS) simulation architecture, using LSA not only to interoperate simulators in the network but also as the foundation for the internal architecture of the simulator itself.

Our proposal is to use LSA as the common architecture for MSaaS.

LSA as common architecture for MSaaS model

In this section we are proposing the use of LSA as the base and common architecture for solving the use of different architectures and standards in MSaaS. LSA (Layered Simulation Architecture) is a proposed standard, now finishing study-group phase in SISO, for creating a common way of using simulators and simulation assets based on current simulation standards without any change.

These are the main advantages of the use of LSA for MSaaS:

- LSA defines a loosely coupled open architecture, with several layers. Compliance with LSA can be achieved by using only some selected layers and services. This flexibility allows to use it as a common

platform for every simulator in the network. Only in this way interoperability to a service level will be achieved.

- LSA can integrate new and previous existing simulators and services that use most popular simulation standards like HLA, DIS and TENA. Any HLA or DIS vendor or implementation can be integrated in this schema without any change. The only dependency to be addressed is the use of matching data models in the case of gateways or the same data model in the case of direct implementation.
- Service´s base architectures are transparent to the user. LSA treats all the incorporated simulators the same way and incorporates them to DDS common communication layer (in a way, it normalizes the services)
- Communication is guaranteed by the use of DDS over every possible kind of network (WAN, LAN, Wi-Fi) even in disconnected, intermittent and limited networks (DILs). Once the needed Quality of Services (QoS) are set, the behavior of the joint simulation in the same no matter the physical layout of the network. DDS acts as an independent layer keeping the communication alive in the established conditions.
- DDS can work over any device that can use TCP/IP communication from PCs to Android platforms [20]. E.g. web sockets allow the use of DDS in browsers.
- Part of needed security can be implemented by using DDS security standards (now in beta, pending final approval).
- DDS has a wire protocol called DDSI (DDS Interoperability Wire Protocol) that guarantees that every DDS vendor can communicate with the others [21]. That way, the use of DDS is not tied to any single vendor. All the DDS vendors can communicate between them without changing out-of-the-box products.
- Services, like HLA services or specific simulation services (e.g. a terrain server) can be integrated in the same architecture. From the beginning LSA has incorporated the definition of a common service interface in the proposed architecture.
- LSA original proposal can be easily expanded to incorporate other standards closely related with simulation. For example, an implementation of LSA architecture with C-BML has been built and probed. That way, the original concept of MSaaS can be also expanded to C2 systems.

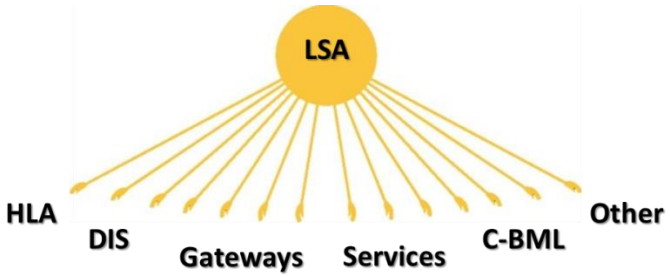

Figure 4: Some LSA standards and architectures

Nowadays, even when LSA is only in development, its architecture is already the foundation of important simulation projects, offering fresh solutions to Battle Labs, training centers and Test-beds [17][22][23]. All these solutions share a common feature: they are breaking the boundaries of every simulation systems going to an open distributed architecture based on the data-centric approach proposed in LSA.

A good proof of concept to demonstrate the use of LSA as the architecture for MSaaS would be the real deployment of a LSA-like architecture in the cloud. This proof is going on and it is, now, in early stages in a real project. Conclusions will be available during the first half of 2015. The aim of this project is related with cross-domains integration and one of its objectives is the deployment of a LSA-like architecture in the cloud (private cloud in this case) and the access to it by using thin web clients and Android platforms.

Challenges and pending decisions in MSaaS using LSA

Next steps in LSA go in the direction of MSaaS. LSA group is now working in the proposal of a PDG (Process Development Group) in SISO. The objective is converting LSA into a standard. LSA current approach is only centered in simulation standards and simulation services, but the study group is internally considering the proposal of adding new standards like C-BML to LSA original proposal.

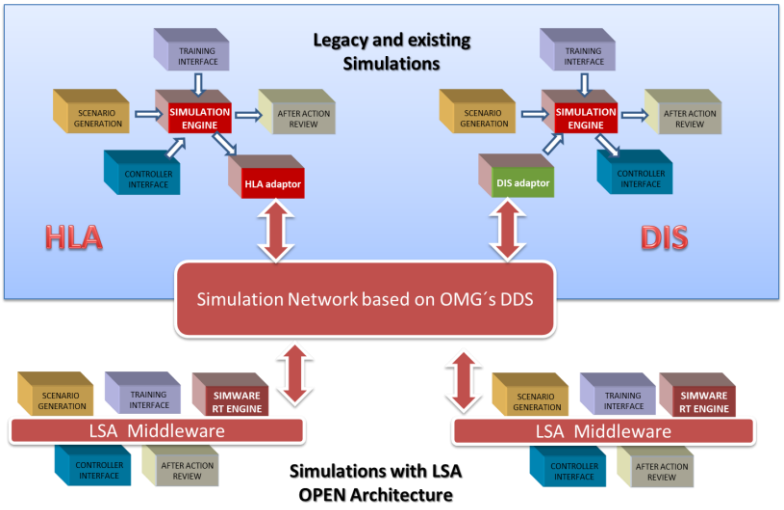

Figure 5: LSA proposed integration

In parallel, there is an experience going on, about the use of LSA for simulation in the cloud. This experience is part of a project that uses LSA architecture for interoperation with C2 (command and control) systems and robotic systems. The experience, tries to be a real proof of concept of the use of LSA architecture for simulation in the cloud and MSaaS challenges [22]

Finally, the LSA group is waiting for the standard of security layer of DDS to be finished. We are eager to start using it in real projects thus probing the security issues of MSaaS approach.

CONCLUSION

The vision of M&S as a service is to offer users services, wherever they are and whenever they need them, to create under-demand simulations. Given a problem specification or a situation that needs to be confronted, a user will search and discover what resources (systems and/or data) are available, will download them and will use them.

Final user won´t need to be aware of the technologies and standards involved and of the security issues. Services will be guaranteed in the features and quality of the features that they have tagged.

Different alternatives have been proposed for the architecture of MSaaS. In this paper, we present LSA as the main candidate for this architecture.

LSA stands out in the idea of having a common simulation platform, being able to be used to a System of Systems level. Convergence of different standards over a common wire protocol will improve interoperability a lot. Common simulation services will provide the capability to share features and capabilities between different simulators and remote simulation servers.

Finally, multilevel security can be also addressed by LSA, leveraging DDS security profile that it is now pending of approval in OMG.

Future work in LSA is adding standards closely related to simulation as C-BML and creating a proof of concept of LSA in the cloud. LSA is now beginning the process of becoming a standard in SISO.

REFERENCES

[1] Headquarters Supreme Allied Command Transformation (SACT) NATO. Modelling & Simulation Vision. December 2008

[2] Robert Siegfried. M&S as a Service: Perspectives, Expectations, and Challenges *NATO CAX Forum 2013, Rome, Italy*

[3] Erdal Cayirci. MODELING AND SIMULATION AS A CLOUD SERVICE: A SURVEY *Proceedings of the 2013 Winter Simulation Conference*

[4] Siegfried et al., Effective and Efficient Training Capabilities through Next Generation Distributed Simulation Environments, *NMSG Conference 2013, Sydney, Australia*

[5] Björn Möller, Peter Karlsson et al., Towards Multi-Level Security for NATO Collective Mission Training, *2011 Spring Simulation Interoperability Workshop, 4-8 April 2011, Boston, MA, USA*

[6] Rena Zhang et al., A "Fair Fight" Analysis Tool for Distributed Simulation, *SISO Spring 2003, Orlando, FL, USA*

[7] Peter Mell, Timothy Gance, NIST. The NIST Definition of Cloud Computing, *NIST Special Publication 800-145, September 2011*

[8] DNBL–Communities of Interest Service Exchange Operating Model Ed 2.3.

https://dnbl.ncia.nato.int/Reference%20Library/DNBL%20Operating%20Model%202013.pdf, September 2013

[9] Systems and Network Analysis Center, Information Assurance Directorate NSA. Biometrics Security Consideration, *http://www.nsa.gov/ia/_files/factsheets/i73-009r-007.pdf*

[10] *https://technet.microsoft.com/es-es/library/security/dn631937.aspx*

[11] *http://heartbleed.com/*

[12] Sean Carlin et al., University of Ulster, UK and Danish Jamil et al. Cloud Computing Security, *International Journal of Engineering Science and Technology (IJEST) 2011*

[13] Amy E. Henninger, Dannie Cutts et al. Live Virtual Constructive. Architecture Roadmap (LVCAR). *Final* Report. *Institute for Defense Analyses September 2008*

[14] *http://www.sisostds.org/StandardsActivities/DevelopmentGroups/GatewayDescriptionandConfigurationLanguages.aspx*

[15] *http://www.sisostds.org/StandardsActivities/StudyGroups/WebLVCSG.aspx*

[16] *http://www.sisostds.org/StandardsActivities/StudyGroups/LayeredSimulationArchitectureLSASG.aspx*

[17] Jose R Martinez, Dan Gregory. A New Approach for Converging LVC Simulation Architectures. *SISO Spring SIW 2013, Orlando, FL, USA*

[18] http://www.omg.org/hot-topics/dds.htm

[19] *http://portals.omg.org/dds/content/document/dds-security-extensions-rfp-proposal*

[20] Ronald Leung. DDS "Data Distribution Service" Android Integration. *http://blogs.rti.com/tag/dds-data-distribution-service-android-integration/ March 2011*

[21] *http://www.omg.org/spec/DDSI/2.0/*

[22] Jose R Martinez et al. Extending LSA philosophy to real world challenges. *SISO Fall SIW 2014, Orlando, FL, USA*

[23] Jose R. Martinez et al. Spanish MoD NOGESI Battle Lab. *SISO Session. ITEC 2014, Cologne, Germany.*