Securing multiple simulation exercises in a national Defence Context

Doug Stapleton
dlca2576@bigpond.net.au

Abstract: There are several dimensions to securing a simulation scenario in a national Defence context that involves multiple players. This paper looks at how to secure individual exercises that share the same underlying infrastructure and keep players separate and able to view only their authorised exercises from several different perspectives:

- How an exercise scenario progresses from a 'Development' environment, through 'Test' and into 'Production';
- The need to allow an individual person to have different roles on different exercises;
- The need to take account of the individual person's security attributes such as clearance, briefings and nationality; to cater for ISM definitions such as AUSTEO, AGAO, AUS/USA EO, RELEASABLE TO during coalition exercises, and;
- The need to gradually refine the battle damage assessment, from a simple hit or miss during development to a refined precise engineering assessment during the production execution of the exercise.

1. INTRODUCTION

Modern military simulations can be run as a combination of Live with real people executing real commands on a battlefield, or Virtual where real people control virtual entities in the battlespace and Constructive where the entities operate according to predefined rules of engagement, as illustrated in Figure 1.

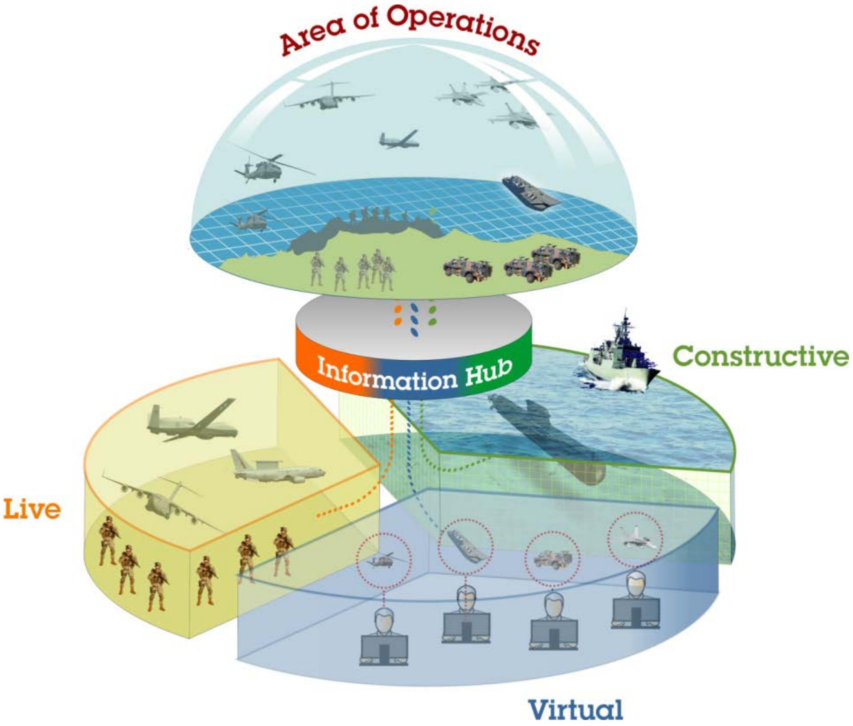

Figure 1: A simulation in an Area of Operations can be a combination of Live, Virtual and Constructive Simulations

Simulation exercises share a major proportion of the underlying infrastructure which makes security a challenge: if a person has access to the training environment, then how can their access be limited to particular exercises? Even for one exercise, the audience is different depending on the stage of the environmental lifecycle; development, test or production. This is further refined by the response to battle damage assessment routines which are simplistic in the development environments and can progress to refined engineering constructs in the production environment.

2. EXERCISE SECURITY

2.1 Security Administration in the Training Network

The training network is generally treated as one security domain, with a set of authorised users who are then further given access to particular exercise environments. The relationship between the training network and the Command and Control network and foreign simulation networks is generally a peer to peer relationship through gateways at the boundary of security control arrangements.

In a practical sense, it is the simulators and participants that can operate across Australia on the training network. This means that multiple simulators can be linked up between various locations using the training network. The peer level gateway allows traffic through to two other security domains for particular purposes:

1. The link to the Command and Control network allows for a particular exercise to reach out to a live platform and include them in the training exercise. An example may be on the bridge of a naval ship, that is responding to command and control requests from a particular training exercise. How are those instructions and the situational awareness communicated between the particular exercise on the training network and the real warship on the Command & Control network?
2. The link to Foreign Simulation allows for a particular exercise to reach out to foreign participants, who are authorised to join the exercise. This is the real time access method to allow Australia to participate in Coalition exercises. Thus the Australian pilots operating a virtual fleet of aircraft can use this mechanism to participate in one of the Coalition exercises.

In administering the training network, users are added to the security domain and their nationality, clearances and codeword briefing information will need to be accessible. Participating users first of all logon to the Training domain and then need to join an exercise if appropriately cleared.

2.2 Joining an Exercise

From the list of available active users in the Training domain, how do we give a limited subset of users access to a particular exercise environment? Access control must be implemented separately within the domain for the relationship between a user and the Exercise Environment they are authorised to access. An Exercise Environment is defined as the combination of the Exercise Name and the Environment Lifecycle Stage. An example may be Talisman Indigo 2016 and the environment may be Production; thus "Talisman Indigo 2016 - Production".

Security administration will setup the users in each exercise and their permitted environments. As a general statement, developers would normally have access to the Development environment and perhaps the Test environment, while exercise participants would have access to the Production environment and a few select personnel would have access to the Acceptance Test environment for final testing prior to the exercise commencement.

At a simplistic level, we can picture an exercise being the execution of a HLA federation (IEEE, 2010), with each participating simulator joining the federation as the exercise is instantiated. End users on their workstation will be connecting to their server instance of a simulator and the collection of simulators will operate as a federation.

With the collection of server instances for simulators being joined as an HLA federation, the technical point is that these virtual machines instances are linked together as part of a virtual private network (VPN). This makes each virtual machine instance invisible to other valid users on the Training domain, unless they have access to this particular VPN. So each VPN is a collection of virtual machines, making up the operating environment for the particular exercise environment. A user who logs on to the Training domain, will not automatically see all the exercise environments as they aren't connected to any particular VPN at the time of logging on. Thus the authorised users will need to "attach" their workstation to a particular exercise VPN to gain visibility of that particular collection of servers.

As a user makes a request to "attach" to a particular exercise, the security system will only display those Exercise Environments that they are able to join at that particular period in time. Thus a developer may see on their permissible list; "Talisman Indigo 2016 - Development" while a uniformed officer may see "Talisman Indigo 2016 - Production". The user selects their target exercise and the system will then complete its security checks (such as releasability) and if the user is authorised, the workstation will be attached to the VPN for that specific Exercise Environment and the user will then be able to see the participants and servers to commence their activity in the exercise. As workstation users we are familiar with networked drives such as H for the Home drive etc. We can think of this as an X drive for exercises. In principle, the user would attach to only one exercise at a time to prevent cross contamination via the workstation.

This can work if all the virtual simulators were in the one environment, but the question then becomes how does a remote simulator join a federation and keep up to date with the execution of an exercise. This might be a real warship on the Command & Control network or participants from another country who are trying to join the exercise.

The following diagram illustrates the challenge, how to operate a HLA federation of virtual simulators while crossing the boundaries between the Training network, the Command & Control network and a Coalition member.

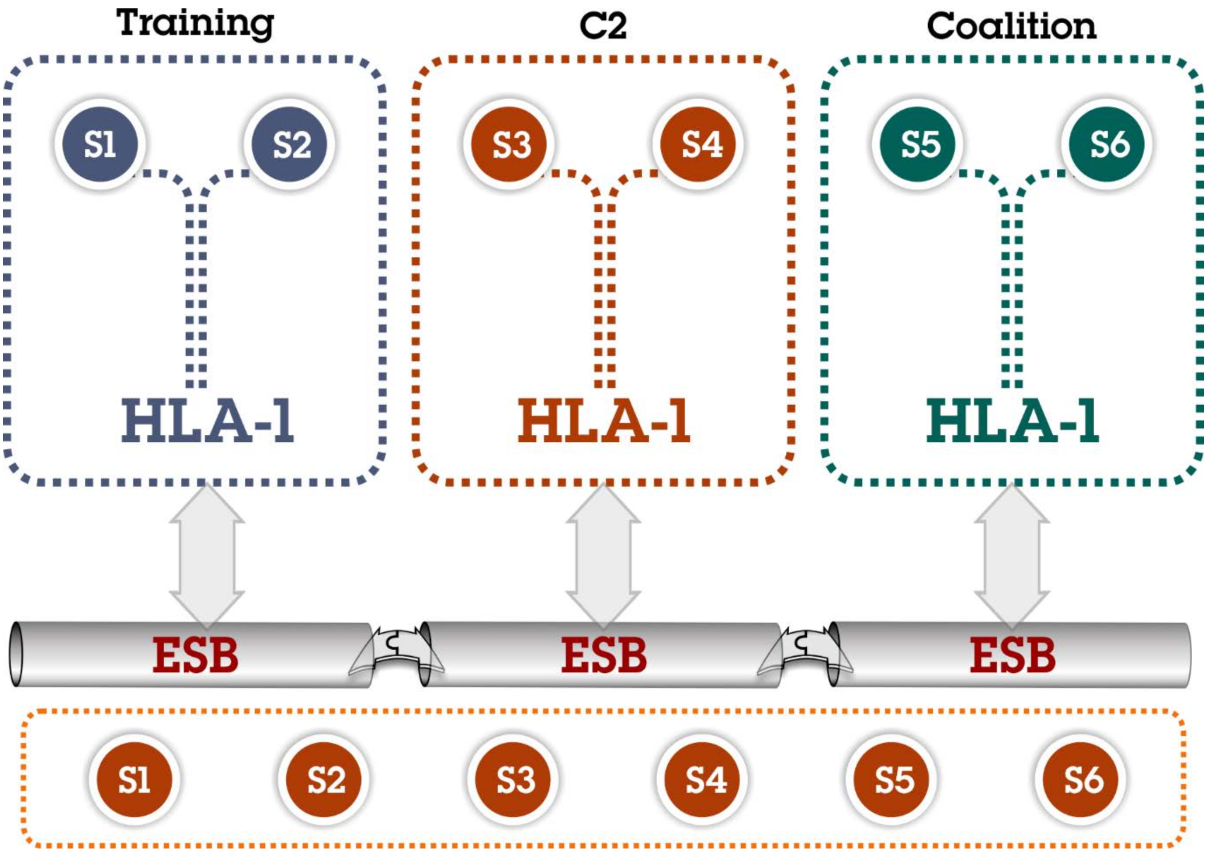

Figure 2: Illustrating a HLA federation across domain boundaries

Figure 2 above illustrates how virtual servers across domain boundaries can become part of the one virtual federation and thus see the actions of other entities. Within each domain, the simulator instances running on virtual machines as part of that exercise's VPN can see and become part of their local federation control using the local Run Time Infrastructure (RTI) which will reach out to its local Enterprise Service Bus (ESB) where necessary to share information. The protocol information is then packaged up and passed via the ESB to the local Run Time Infrastructure in another domain where it is then passed on to the specific simulator instance. The magic of this approach is that it makes all simulators look as if they are playing together in the one virtual federation as illustrated at the botton of the diagram. Regardless of the particular simulation protocol such as DIS (IEEE, 1995) or the use of HLA via a vendor specific RTI, the ESB can link each federation control together to create the impression of one unified exercise.

The challenge of connecting domain separated HLA run time infrastructures has been addressed by the HLA Evolved Web Services API (Möller et al, 2006; Tu et al, 2011; Pett, 2012). HLA specifically addresses the simulation requirements for exercise support components of shared state, time handling and synchronisation. Web Services addresses the more business level interoperability requirements across different operating systems and programming languages. HLA Evolved Web Services combines both approaches.

However the use of the modular approach of web services introduces a performance overhead in comparison to the direct API calls used in the Run Time Infrastructure within each domain. This is further complicated by the additional overheads of transferring SOA messages via an ESB across separate domains. One approach is to filter out unnecessary messages within the ESB so that a remote federate only subscribes to specific messages of relevance to avoid the overhead of delivering unneeded messages. Another approach is to subscribe to the federation messages at an appropriate rate, rather than delivering the full rate of messages to every federate. This approach is known as the HLA Evolved Smart Update Rate Reduction (Möller et al, 2005).

Here is one of the major aspects of security. Individual users are validated for their credentials at logon to the domain and then subsequently as they join a particular exercise. Interactions between simulator instances may track their authorised user from a logging point of view, but the security mechanism changes to one of enforcing the application level security. Thus it is the federation control software (i.e. the local RTI) that is allowed, as an application, to access the local ESB in the domain. The key aspect of the security model will be the ability to subscribe to the RTI at specific security levels, dictated both by the RTI requirements and the distribution requirements of the FOM messages passing between the participating systems. Between domains, security is enforced to ensure that the corresponding ESBs in each domain are talking to each other properly. These interactions will log the user on whose behalf the interaction is taking place but the end user has no direct access to the underlying components of the system, such as the ESB which can only be access by authorised application components.

3. BATTLE DAMAGE ASSESSMENTS

The general principle between entities within the simulation is that the entity receiving incoming fire makes an assessment based on the nature of the incoming ordnance, and the trajectory with the result being returned back to the source of fire. That result may be as simple as hit or miss. Typically for a development or test environment that would suffice and indeed the training audience may not be allowed to see refined engineering detail of the damage done. This is reminiscent of the "Battleship" game where hits and misses are shown in a binary way.

However the requirement becomes considerably more complicated and is a matrix of permissions based on the environment and the personnel who are accessing the simulation. For performance reasons, this is too complicated to resolve for each firing and consequential battle damage assessment, so the preferred approach is to provide a system level indicator as to the level of battle damage assessment (BDA) to be returned. The value of this system level indicator (*BDA System Level Indicator*) is available to all entity participants at all times during simulation execution and has a direct impact on the BDA returned. There are two broad levels possible for that indicator:

1. Hit or Miss - this value directs the entity receiving fire to return a binary assessment of hit or miss. In simplistic terms, the hit value indicates destruction of the entity receiving fire while the miss value indicates the entity can continue operating.
2. Engineering Precision - naturally in the real world of warfare, a hit can mean different degrees of damage, as can a miss which may partially immobilise a vehicle if the shot falls close enough. So instead of a binary answer, this value directs the entity receiving fire to undergo a more complex assessment based on the ordnance, incoming speed, angle, direction etc. The result may be given with more precision, such as the left track of a tank may be disabled but the gun is still operating. It would also go to the survivability of the crew in those circumstances.

For performance efficiency, the *BDA System Level Indicator* is only changed when the environment or participants change. The environment stays constant for the execution period of an exercise; that is to stay if an exercise commences in 'Development' the environment cannot change to 'Test' or 'Production' without a shutdown and restart of the whole exercise. So the initial state of the *BDA System Level Indicator* is set according to the following table:

Table 1: BDA System Level Indicator

Development	**"Hit or Miss"**
Test	"Hit or Miss"
Acceptance Test	"Engineering Precision"
Production	"Engineering Precision"

This initial setting is then further modified depending on the clearance level of the participants joining the exercise. The granting of access rights to the exercise is set externally based on the pool of available participants on the training network.

The exercise itself is set to a level of participation which affects the *BDA System Level Indicator* based on who joins the exercise. An example might be an exercise which is set to AUSTEO (Australian Government, 2015). For each participant joining the exercise, the security routine would need to ask "Does this person joining this exercise satisfy the AUSTEO requirements". This would be resolved by the security routines interacting with HR data to determine nationality attributes of the person and other attributes such as compartmented briefings (for access to caveated information) and to their rank and service arm etc.

The impact of a negative assessment will cause that participant to be denied access to the exercise. The practical impact of this is that an exercise as a whole can be set to AUSTEO in the production environment. Participants A B and C can be given access to that exercise. During the logon procedure, if any of participants does not meet AUSTEO requirements based on the HR data check, then they will be denied access to the exercise environment. Normally, at the time of granting access to an exercise, participants A B and C will need to have valid security credentials to be granted access to the exercise. A re-check is done at the time of logging on to the exercise which will catch situations where for example, participant C may have had valid security credentials at the time of being granted access to the exercise during security administration, but over time something changes and they no longer have access.

4. EXTENDED SECURITY IMPLICATIONS

4.1 Cross Domain Solutions

In general, simulations operate at the same classification level and therefore messages between domains will pass through a Gateway at the security boundary. In the case where the target is at a different classification level, either higher or lower, the messages will need to go via a Cross Domain Solution and have security metadata attached to enable agents to permit or reject access to the published messages (in part or in whole) by federates or other participants at differing levels of classification.

The message flow can be seen in terms of a Request/Response paradigm and this is useful for the security accreditation approach. The overall business flow from one domain to another will need to be accredited in the normal flow for a cross domain solution. From the perspective of the Training Domain, the flows will be either High to Low, or Low to High depending on the classification of the target domain relative to the Training Domain. The ability for a lower classification domain to respond to a message from a higher classification domain is generally accepted provided that the returning message can be clearly identified as being a response to a previous request. That can be characterised as the higher domain "reaching out" to the lower domain with a structured query which can be answered. The flow of individual answers back to the higher domain when validated as being a Response to a previous Request can be accredited in the normal course.

So the more difficult case is when the Training Domain wants to send a message to a domain that is of a relatively higher classification. In generic terms, how does a lower domain make a request of a higher domain and receive answers in a way that will pass security accreditation? Part of the answer is in regard to the gateways and cross domain flows; under the general heading of a bi-directional SOA gateway between adjacent security domains of different classifications.

Given that there is security accredited infrastructure to deliver these bi-directional SOA messages, the issue then becomes one of content and whether the question can indeed be answered by the higher security classification. These are business issues and security judgements that need to be made. For example, a broad question such as list all the personnel on Ship XYZ would not be permitted from a lower to higher domain; as any response would be seen in the context of a data spill. So designing the message flow and creating specific questions that can be answered without compromise can be a challenging exercise. In this example, it may be permissible to ask "is Person X deployed on Ship Y"?

A special case is the evaluation of Battle Damage (BD) from a simulator that is running at a clearance level above those of other participants in the exercise.

4.2 ITAR Security

Another complication that will add a further dimension to exercise security is that of access to ITAR controlled software or simulators. ITAR constraints imply that within an exercise a company managing a platform/system A might not have access permission to connect to platform/system B. Further issues arise around access by government employee (ADF/APS) versus contractors. The security design can be come very complex when some data flows become subject to ITAR constraints and requirements.

4.3 Dissemination of After Action Reports

After action reports are shared among the participants depending on the bilateral, country to country arrangements. This further complicates the security model; a participant may be granted access to a particular exercise environment but that does not automatically presume they are entitled to see the after action reports. These reports are still subject to the releasability rule, expressed through the ISM definitions such as AUSTEO etc (Australian Government, 2015). Hence information release in documents relating to the conduct of the exercise will be governed separately to the execution phase of the exercise.

5. REFERENCES

Australian Government. (2015, April). *Information security management guidelines Australian Government security classification system.* Retrieved March 23, 2016, from Australian Government Attorney-General's Department Protective Security Policy Framework: https://www.protectivesecurity.gov.au/informationsecurity/Documents/INFOSECGuidelinesAustralianGovernmentSecurityClassificationSystem.pdf

IEEE. (1995). *1278.1-1995 - IEEE Standard for Distributed Interactive Simulation - Application Protocols*. Retrieved March 2016, from IEEE Standards Association: https://standards.ieee.org/findstds/standard/1278.1-1995.html

IEEE. (2010). *1516-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules*. Retrieved March 2016, from IEEE Standards Association: https://standards.ieee.org/findstds/standard/1516-2010.html

Möller, B., & Karlsson, M. (2005). Developing Well-Balanced Federations Using the HLA Evolved Smart Update Rate Reduction. *Proceedings of 2005 Interoperability Standards Organization.* Pitch Technologies.

Möller, B., & Löf, S. (2006). A Management Overview of the HLA Evolved Web Service API. *Simulation Interoperability Workshop.* Linköping: Pitch Technologies.

Pett, M. D., & Gustavson, P. (2012). Combat Modeling with the High Level Architecture and Base Object Models. In A. Tolk, & A. Tolk (Ed.), *Engineering Principles of Combat Modeling and Distributed Simulation* (pp. 413-448). New Jersey, NY, USA: John Wiley & Sons.

Tu, Z., Zacharewicz, G., Chen, D. (2011). Developing a Web-Enabled HLA Federate based on poRTIco RTI. In *Proceedings of the 2011 Winter Simulation Conference* S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.