BML and MSDL for multi-level simulations

José Ruiz
DGA/DS/CATOD, France
jose.ruiz@dga.defense.gouv.fr

David Désert
CASSIDIAN, France
david.desert@cassidian.com

Antony Hubervic
MASA Group, France
antony.hubervic@masagroup.net

Patrice Guillou
Thales Raytheon System, France
patrice.guillou@thalesraytheon-fr.com

Roger Jansen, Nico de Reus, Henk Henderson
TNO, Netherlands
roger.jansen@tno.nl, nico.dereus@tno.nl, henk.henderson@tno.nl

Kjell Magne Fauske
FFI, Norway
kjell-magne.fauske@ffi.no

Lennart Olsson
Pitch, Sweden
lennart.olsson@pitch.se

Keywords:
C-BML, MSDL, HLA, CAX, NATO

ABSTRACT: *Military training needs to reflect the complexity of real-world operations. The main training audience is currently often focused at a certain level (e.g. joint headquarters staff, component headquarters staff, platoon leader, individual combatant), but it will typically include some interactions with levels above and below the own organizational level. Federations of multiple modeling level simulations are now mature enough to provide a realistic and coherent synthetic environment. The challenge is now to translate military orders build in Command and Control systems to stimulate the target simulations (high level ones such as aggregated unit models or low level ones such as platform models), and in the opposite direction, to send back reports from the simulations that are customized for the recipient C2 system. The NATO Modeling and Simulation Group MSG-085 studies interoperation of C2 systems and simulators by standardizing the orders (C-BML: Coalition Battle Management Language) and the simulation initialization (MSDL: Military Scenario Definition Language). During 2012 and 2013, France, Netherlands, Norway and Sweden have worked together in the NATO MSG-106 (SPHINX: enhanced CAX architecture, design and methodology) to improve the process of multiple level Simulation-C2 initialization and information exchange. The different steps of this process are: the writing of military orders in C2 systems, the conversion of these orders into C-BML format, the conversion of these C-BML orders into High Level Architecture (HLA) objects and interactions for high level simulations and then the decomposition of these high level orders into low level ones suitable for lower level simulations. The focus of this work is in the HLA area where the MSG-106 experts defined a high level as well as a low level BML Federation Object Model (FOM) extension to the so-called 'NETN (NATO Education Training Network) reference FOM' that has been developed previously in MSG-068. Furthermore, this FOM extension has been tested with national simulations and support tools (C-BML/HLA gateway, HLA C2 agent…). This paper presents the basic design of the C-BML FOM modules, the extension of MSDL schema and the results of their experimentation.*

1. Background

1.1 Training context

Military training needs to reflect the complexity of real-world operations. The main training audience is currently often focused at a certain level (e.g. joint headquarters staff, component headquarters staff, platoon leader, individual combatant), but it will typically include some interactions with levels above and below the own organizational level.
For representation of military activity in a theater, federations of multiple modeling level simulations are now mature enough to provide a realistic and coherent synthetic environment. Thanks to this basis, the training staff now has to pay attention to the two following main challenges:

- The initialization of these simulations and sometimes C2 (Command and Control) systems with data from a common scenario repository.
- The translation of military orders build in C2 systems to stimulate the target simulations (high level ones such as aggregated unit models or low level ones such as platform models), and in the opposite direction, to send back reports from the simulations that are customized for the recipient C2 system.

To solve these challenges, several study groups under the umbrella of NATO and SISO are conducting M&S (Modeling and Simulation) research activities in order to provide enhancement and/or additional standards and reference models.

1.2 Interoperability standard limitations

The interoperability between simulations is covered by the well known standards DIS (Distributed Interactive Simulation) or HLA (High Level Architecture). In this domain, there are identified limitations, such as the interactions between heterogeneous levels of simulation models, but this part is not the purpose of this document.
With respect to HLA, under the framework of the NATO MSG-068 and its successor MSG-106, a reference modular FOM has been defined and named NETN (NATO Education Training Network) FOM.

For initialization of a scenario used in a federation of simulations, it is necessary to start from a common reference in order to avoid misunderstandings and ambiguity during the data exchange in a federation. Another advantage of using a common reference is to avoid multiple data input in the different simulations and then to reduce the preparation time and avoid mistakes.
The MSDL (Military Scenario Definition Language) standard provides a relevant format to define the elements of a military scenario. MSDL has its origins in the USA OneSAF program, but was in 2008 formally approved as a SISO standard [1]. Since then MSDL has continued to expand and is now used by many nations and organizations as well as in commercially available simulation tools.

MSDL was designed to be simulation independent and is primarily geared towards the description of the ORBAT. As MSDL is the only available standard for specifying military scenarios for use in simulations and C2 systems, there has been an interest in the simulation community to extend MSDL in order to include more detailed information. This is especially relevant in a simulation federation where the different federates have different levels of detail requirements.
In a simulation federation, there is also a need for distributing the simulation responsibility of the units in the scenario among federates. Currently, there is no mechanism in MSDL for this.
Another limitation during the initialization phase is the lack of a standard support to share the scenario in a federation (data storage and transport).

MSDL and C-BML (Coalition Battle Management Language) are standards being studied in NATO MSG-048 and now MSG-085 to promote the interoperation of C2 systems and simulators.
C-BML captures orders and requests in a C2 language that typically addresses units at company level and above. This requires that a C-BML compliant simulation system models military doctrine for the level of the involved units. Such simulations are called high level simulations later in this document. Otherwise, it is mandatory to introduce men in the loop or automatic systems to split C-BML orders at the accurate level of the simulated units. Such simulations are called low level simulations later in this document.
Unfortunately, as C-BML is a recent standard, few simulation and C2 systems are natively compliant. Interfaces have to be developed to exchange data from a common C-BML server. For HLA federations, one solution to avoid multiple interfaces is to introduce a common gateway to insert the C-BML information into a dedicated FOM (Federation Object Model) module and the let the RTI (Run Time Infrastructure) transport it to the simulation interested in C-BML data (see figure 1). This way has been selected by the NATO MSG-106 for its experimentation and described later in this document.

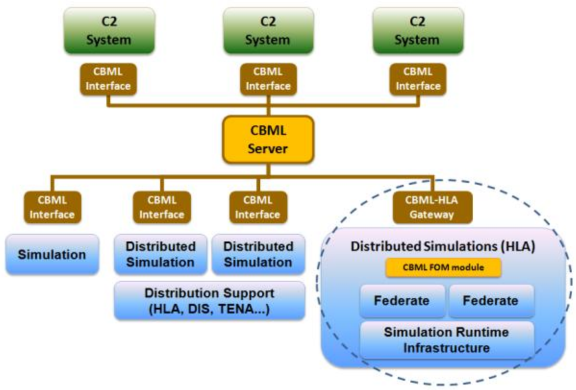

Figure 1 - C2-Simulation exchange using multiple interoperability standards

2. Improvements of existing standards

2.1 MSDL for initialization

The MSDL XML schema provides a support to define the units and their equipments (see figure 2). The types of unit/equipment should be defined in a way compatible with simulations which will manage them. The extensions proposed by MSG-106 to this MSDL XML schema, are used to better support initialization of distributed simulation systems.

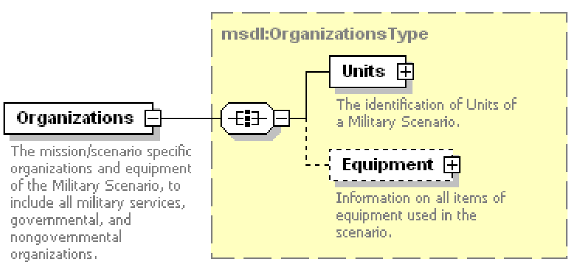

Figure 2 - Extract of C-BML XML schema

Initial allocation of modeling responsibilities

The initial responsibilities of modeling units and equipment in the ORBAT (Order of Battle) shall be allocated to the participating systems during the federation configuration. This deployment or allocation of ORBAT information is captured in an extension of the MSDL format which consists of a new "Deployment" tag element including multiple (one or more) "Federate" sub-elements. A "Federate" element shall contain information regarding which ORBAT elements have been allocated to the federate.

<u>Example</u>:

<Deployment>
  <Federate>
    <ObjectHandle> *federate ID* </ObjectHandle>
    <Units>
      <Unit>
        <ObjectHandle> *unit ID* </ObjectHandle>
      </Unit>
      <Unit>
        <ObjectHandle> *unit ID* </ObjectHandle>
      </Unit>
    </Units>
  </Federate>
  <Federate>
    …

Unit and equipment type identification for simulation

The MSDL format uses the JC3IEDM object type values to identify the unit type and NATO Stock Number (NSN) codes to identify the equipment type. The proposed extensions to the MSDL XML schema provide additional fields for the type identifications of units and equipment based on SISO-REF-010 [7]. "EntityType" values are introduced in the existing "Model" attribute (from "UnitType" and "EquipmentItemType" sections) which already contains data for simulation.

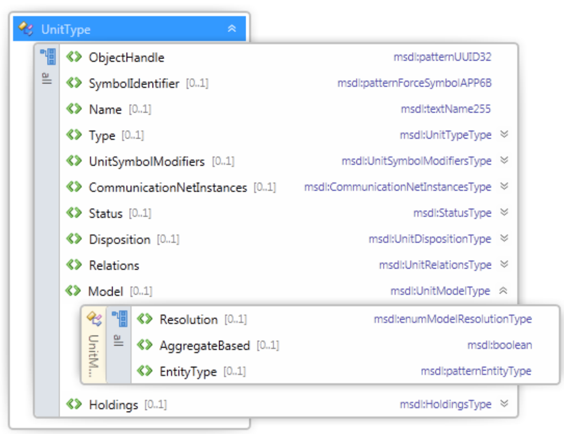

Figure 3 - Use of EntityType in the MSDL UnitType section

Representation of holdings

The MSDL standard does not impose a level when describing an ORBAT. A federation may use platoons as leaf elements of the ORBAT, whereas another may be designed for a more detailed ORBAT by using equipment as leaf elements.

In order to enable the exercise planner to design the federation with the appropriated modeling granularity (aggregated simulations, platform-level simulations or a mix of both modeling granularities), the description of the organization should be as detailed as possible.

The "Holding" concept is used to describe human, platform, equipment and resources that do not need to be individually defined in a simulation (extension of logistics concept). In this way, the MSDL is extended by introducing the "Holdings" attribute in the "UnitType" and "EquipmentItemType" sections.

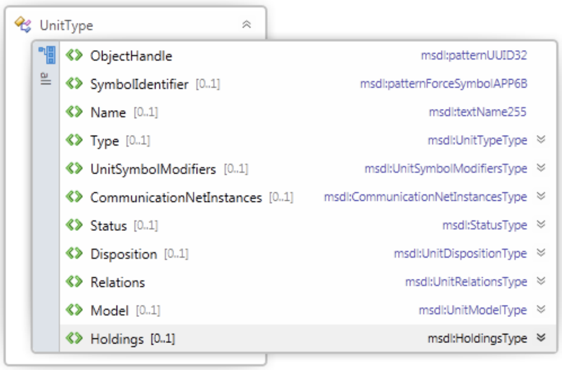

Figure 4 - Use of Holding in the MSDL UnitType section

Extended description of humans

To describe humans in the organization, more details should be provided. For example, it is necessary to make the distinction between officers and non-officers. The category of a human is also an important attribute required in the description. For example, the ORBAT should make the difference between a sniper and a driver.

This kind of information is already present in the JC3IEDM standard through the MILITARY-POST-TYPE concept which defines a rank code and a category code. These codes are used as new attributes in the "UnitTypeType", "EquimentItemType", "HoldingType".

In addition, when a human is described as a Equipment or as a Holding, the following NSN code should be used: **0500**. This will enable the representation of humans which do not need the full details of military post type in the simulation.

JC3IEDM currently has a limited set of defined military post types. It will therefore be necessary to extend the list of post types depending on the requirements of the different federates.

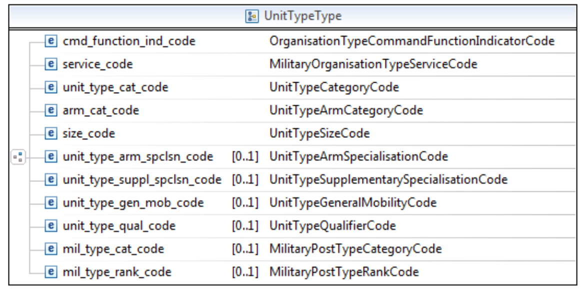

Figure 5 - Use of MilitaryPostCode in the MSDL UnitTypeType section

Embarkment status

This proposed extension to MSDL XML schema includes an element to specify that a unit or equipment is embarked on a hosting unit. This is modeled similar to a support relation in MSDL. The "SupportRoleType" information, in which an enumerated value describes the kind of provided support, is extended with a new enumerated value ("EMBARKMENT") to indicate that a unit or equipment is embarked on another unit.

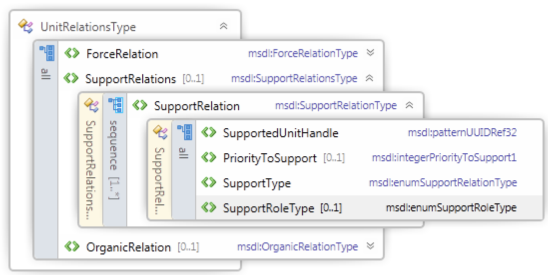

Figure 6 - Use of SupportRoleType in the MSDL UnitRelationType section

The MSDL format has to be used by both simulations and C2 systems to be sure that every system uses the same ORBAT data.

2.2 The concept of coupling BML and HLA standards

Figure 7 shows the architecture used within MSG-106 for connecting C2 systems to both high and low level simulations. C2 systems send C-BML orders to a C-BML server and a BML-HLA gateway encapsulates the messages received from the server into HLA objects. High level simulations can handle these high level orders, but low level simulations are dependent on a C2 Agent federate that transforms the “high level” orders into “low-level” BML tasks modeled as HLA interactions. Similarly the C2 Agent federate aggregates “low-level” BML reports into “high-level” BML reports. The BML-HLA gateway will send

these “high level” reports to the C-BML server to be received by the respective C2 systems.
In addition, some recent simulations compliant with C-BML technology are now able to send directly reports to the C-BML server without using HLA transport.

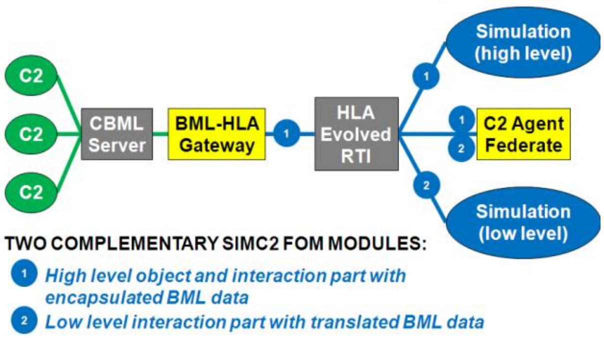

Figure 7 - C2-Simulation system architecture.

As can be seen in the figure, two modules is used, one for C-BML and one for what we call “Low Level” BML. These are explained in the following sections.

2.3 C-BML for High Level simulation

The C-BML module, within MSG-106 referred to as High Level BML module, is SIM-C2 oriented and focuses on transportation of C-BML messages:

- This FOM module has been assembled in order to facilitate simulations that can directly use the existing C-BML order schema defined by MSG-085.
- This module defines interactions or objects for the transfer of the whole information of ORDER and REQUEST without any change. In this case, there is no translation of BML message content into the HLA FOM module, except the extraction of heading information to transport it.
- This FOM module contains also interactions which design all the report (acknowledge, report).

This “High Level” C-BML FOM module was experimented in 2012 (see [8]) and enhanced during the MSG106 technical activity. It contains now three HLA object classes (see figure 8):

- *MessageBML*: global object class which describes a BML element,
- *OrderBML*: object class which describes a "C-BML Order" (subclass of MessageBML),
- *RequestBML*: object class which describes a "C-BML Request" (subclass of MessageBML).

Figure 8 – High-Level BML objects

This module contains also six HLA interactions (see figure 9):

- *Ack*: acknowledge the reception of the message and the capacity to execute a task in the Order,
- *Report*: global interaction,
- *TaskReport*: progress of the Task (subclass of Report),
- *StatusReport*: global interaction (subclass of Report),
- *SituationReport*: perceived view or BFT (Blue Force Tracking) information (subclass of StatusReport),
- *LogReport*: logistics report (subclass of StatusReport).

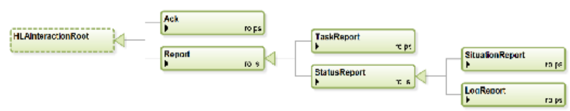

Figure 9 – High-Level BML interactions

2.4 C-BML for Low Level simulation

The concepts of Low-level BML and a Multi-Agent System (MAS) have been described in 0. The C2 Agent federate used in MSG-106 is a MAS that transforms (high level) C-BML orders to low-level tasks and low-level reports to (high level) C-BML reports, as shown in figure 10.

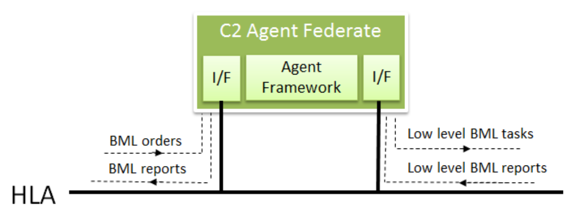

Figure 10 - C2 Agent federate

Low-level BML focuses on tasks and reports with a much finer granularity than C-BML orders and reports:

- It helps the simulation in understanding the C-BML message and it introduces also some concepts about "Command & Control" that simulations could use between themselves (for example during disaggregation process).

- It contains compact low-level tasks and commands that easily can be interpreted and executed by CGF (Computer Generated Force) tools. This is in contrast to C-BML orders that typically require more advanced processing to allow planning and collaboration according to doctrine.
- It reflects the capabilities commonly found in COTS (Commercial Off-The-Shelf) CGF tools, but it is independent of one specific COTS CGF tool and one specific agent framework or agent modeling paradigm.
- It is independent of any specific doctrine or tactics.
- It defines status reports needed for the agent decision making and for producing C-BML reports.

Figure 11 shows a sequence diagram of how a task in a C-BML order might be decomposed into multiple Low-level BML tasks. Note that the sequence diagram is simplified, because each task is not sent once to the CGF tool but to each involved lower level unit simulated by the CGF tool and also the reports are not depicted.

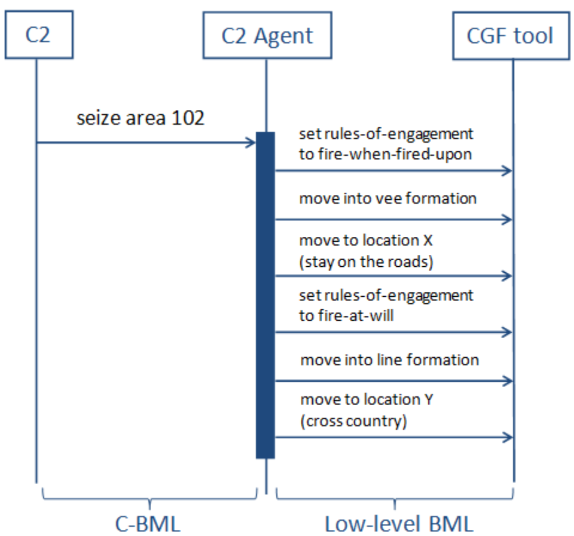

Figure 11 - Differences in granularities between C-BML orders and Low-level BML tasks

Figures 12 and 13 show most of the interaction classes of the near final draft of the Low-level BML FOM module for NETN. The abstract root interaction class is named *LBMLMessage* and has three subclasses:

- *LBMLTask*: the root class of low level BML tasks. Several move tasks, fire tasks, task adjustments, and more are provided.
- *LBMLTaskManagement*: the root class of task management interactions. This is currently only used to cancel tasks, but it may be extended to reschedule tasks, etc.
- *LBMLReport*: the root class of low level BML reports. Own units can report their status in several status reports, where the TaskStatusReport is related directly to the task to be executed. Spot reports are used to report intelligence about detected units.

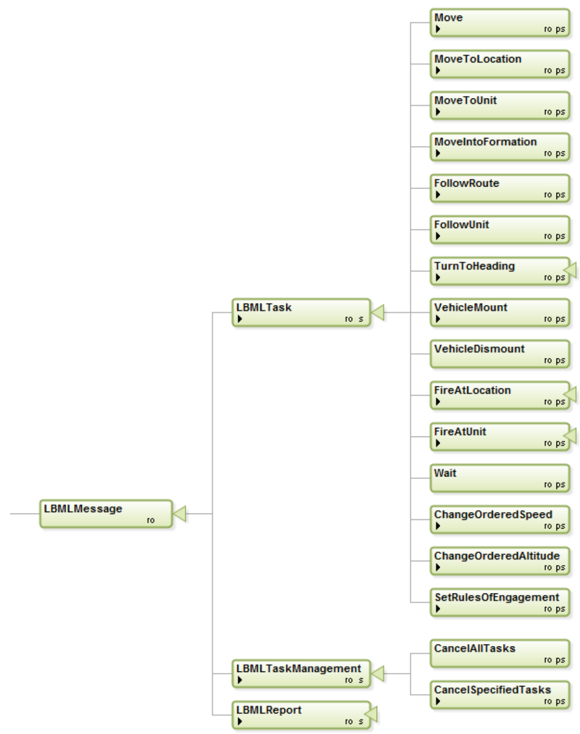

Figure 12 - Low-level BML task and task management interaction classes

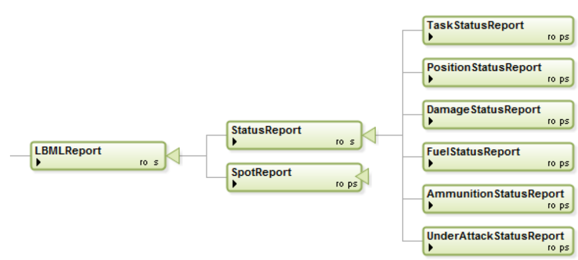

Figure 13 - Low-level BML report interaction classes

3. Experimentation

3.1 Testing architecture

Under the framework of the NATO MSG-106, an experimentation was conducted by France, Netherlands and Sweden during the first half of 2013 in order to test all standard extensions introduced in the previous section.
These experimentation events occur both during face-to-face meetings and distributed tests over Internet.

With respect to the C2 systems, the French C2 system was chosen to send orders and receive reports.

With respect to the simulations, two classes of simulation models were selected:

- "High Level" simulations: some simulations integrate high level models (aggregated unit) and automatic behaviors that allow them to process directly the C-BML orders. ORQUE (used by France), WAGRAM (used by France) and SWORD [9] (used by France) are examples of such simulations.
- "Low Level" simulations: there is a lot of existing CGF tools that are in general not capable of processing and simulating C-BML orders. MÄK VR-Forces 0 (used by the Netherlands) and Pitch Actors (used by Sweden) are examples of such simulations.

The technical tools used as C-BML support were the German FKIE C-BML Server and the French Stimulus (this tool is just used to publish and subscribe information in case of no relevant tools such as simulation or C2 systems).

The technical tools used for the HLA support were the BML-HLA gateway (France), the Pitch RTI (Sweden), the Pitch Booster (Sweden), the IVS (Sweden) and the C2 Agent Federate (Netherlands).

The figure 14 summarizes all tools used during the MSG-106 experimentation.

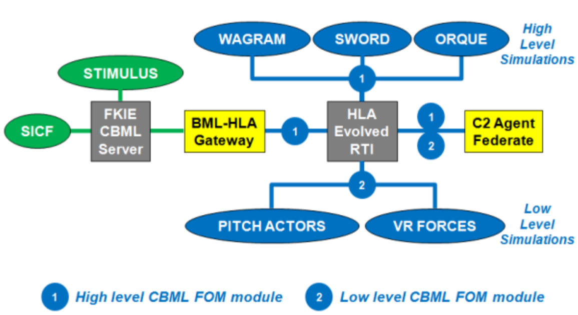

Figure 14 – Experimentation architecture

3.2 Testing support

MSG-106 Booster Network
To connect federates at sites in the MSG-106 group, Pitch Booster has been used. Federates connected to a LAN at each site are connected to other sites over WAN (see figure 15).
Messages to more than one federate behind a Booster are sent as one package over the WAN to the Booster at that site. The Booster is then distributing the message to the subscribers at that site.
A minimum of ports needs to be opened at the firewalls at the different sites.

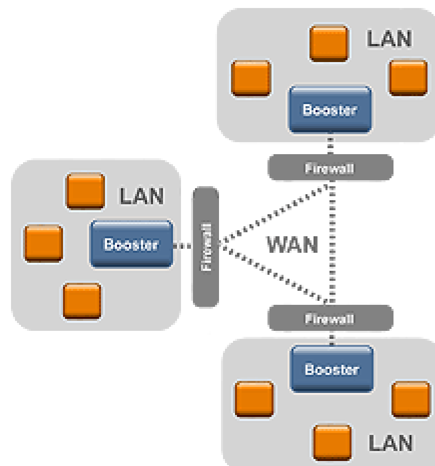

Figure 15: The architecture of a Booster network

Integration and Verification System – IVS
IVS is a self-test system used for conducting tests with prepared stimulating and responding federates. The main goal with IVS is to have the ability to test own federates with external federates before getting to the integration event. IVS has tests for Connectivity, Scalability, NETN Logistics, Behavior (Spatial, Dead Reckoning), TMR (Transfer of Modeling Responsibility) and Low Level C-BML commands and reports. IVS is adapted to RPR2 and NETN2 but is today not a FOM agile system.

Pitch Actors
Pitch Actors is a CGF mainly used in testing as a federate to stimulate other federates and for demonstration purpose. It was primary designed for the RPR FOM but is now adapted to NETN2 to manage NETN Logistics, TMR, and low level C-BML orders and reports. Entities can be given a sequence of actions to conduct in a script. Entities react to warfare interactions.

3.3 MSDL results

In order to demonstrate the work of MSG-106 group on extending MSDL, an operational scenario was created, consisting of four countries and two thousand units. It was decided to use this scenario to test the common initialization of systems.

The forces organizations (order of battle) for the complete scenario were captured using the scenario preparation features of the SWORD simulation. Using its export features, it was then possible to create an MSDL file containing all the units.
This MSDL file was then used to initialize the other simulations (see figure 16).

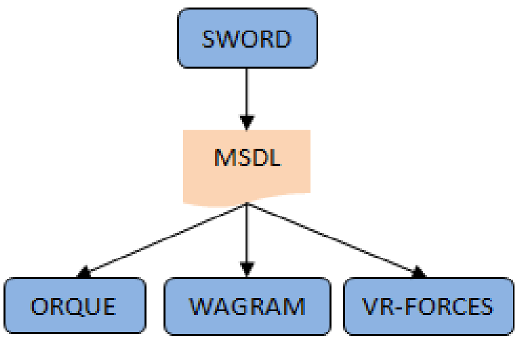

Figure 16: Share of scenario using MSDL file.

The SWORD units' database was completed with the DIS entity types of the units inside the scenario to be able to fill this information in the MSDL file. As it was used to create the scenario, SWORD's order of battle already contained the required units and doesn't need to import MSDL file for the experimentation.

The WAGRAM simulation is a land simulation (aggregated unit at battalion level). It was improved to import the new version of MSDL format. This simulation is able to insert automatically land units at battalion level.

The ORQUE simulation is the naval simulation used within the group, and is responsible for the maritime units of the scenario. The MSDL file was loaded in ORQUE to generate the order of battle.

In order to use VR-Forces, a plug-in has been assembled which is described in [9]. This simulation was also used to experiment the scenario import with the support of MSDL standard. The MSDL file could be loaded in VR-Forces to populate the order of battle of the simulation engine, but VR-Forces does not support any of the proposed MSDL extensions. Since the Deployment extension as described in section 2.1 is not supported, VR-Forces creates HLA objects for all units defined in the MSDL file. Besides this issue, VR-Forces needs probably some more adjustments to have a fully functional scenario after loading an MSDL file.

Using the extensions proposed by the group, it is now possible to automate operations which were previously done in manually process. The presence of the DIS entity type within the MSDL file allows the simulations to automatically match the units of the scenario to their model template

In this experiment, the focus was put on description of high level units in the MSDL file, in order to feed aggregated simulations such as SWORD and WAGRAM. The next steps will focus on a refined description of the platform level units to provide fully and automatically initialization for the low level CGFs.

About the allocation of the unit among the simulation (identifier of the federate responsible for each unit), preliminary tests were done on aggregated units using SWORD and WAGRAM simulations in aim to load only the interesting units from the MSDL file. Additional tests have to be conducted with a deployment of detailed units (platforms level) across low level simulations simulation instances.

During the tests, the merges of MSDL files was done manually. A tool to automatically perform this merge may be helpful for next experimentations. The so-called C2LGGUI from FKIE could be used for this functionality [10].

3.4 BML-C2 results

FKIE C-BML offers two ways to exchange MSDL and C-BML information: either by Web Service or either by JMS (Java Message Service). The first lessons learnt of this C-BML server are the following:

- The JMS interface is the fastest interface, especially for the MDSL subscription.
- It is not possible to create topics to organize the messages.

In addition, a stimulus tool was implemented to interact with the C-BML server through a graphical interface. The available services are the followings:

- Writing C-BML or MSDL information,
- Reading C-BML or WSDL information.

SICF was the only C2 system used for the experimentation. As it was not compliant with the C-

BML standard, a gateway called SPIDER was used to publish the order writing in SICF to the C-BML server and to extract the relevant messages for SICF. All the actions done in SPIDER are manual. The number of processed orders is limited.

The BML-HLA gateway is the component which enables the interoperability between the C-BML server and the HLA federation. The Gateway manages the connection on both sides through a graphical interface which provides a view on the current and past activity (status of connection, log…). In order to control exchanges between the two sides, the gateway allows setting filters to block or confirm the delivery of a C-BML message from HLA to C-BML Server or from C-BML Server to HLA.

Some simulations don't need the BML-HLA gateway to exchange data to the C-BML server. For example, SWORD is able to connect directly to the C-BML server, to receive orders, to execute the high level orders and to send back reports to the server.

3.5 High level simulation results

High level simulations (WAGRAM and ORQUE) were improved to process the following complete use case:

- To execute orders from a C2 C-BML order as an HLA object,
- To acknowledge the reception and the execution of the tasks (HLA interactions),
- To send tasks reports, situation reports and logistic reports (HLA interactions).

The order sent by the C2 system is encapsulated by the BML-HLA gateway in an HLA object. The BML content of the HLA object is used by high level simulations to run tasks defined in the order.
A C-BML order received in WAGRAM is translated into WAGRAM mission format. The WAGRAM operator can check and modify the task if necessary before sending it to the model. Tasks implemented by WAGRAM are move, attack, reconnaissance, observe, withdraw and Special Forces operations.
ORQUE can execute the task automatically if it is well formatted, or execute it after modification. Tasks implemented by ORQUE are attack, patrol and observe.

The Acknowledges and Reports (task and situation) sent by simulations are stored by the BML-HLA gateway in the FKIE C-BML server.

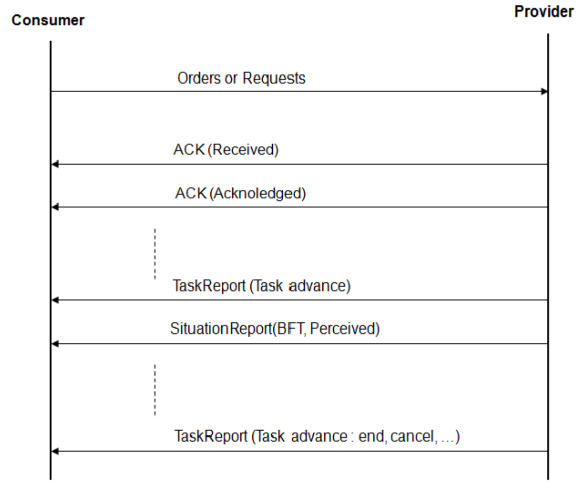

Figure 17: Sequence of C-BML messages.

3.6 Low level simulation results

The Low-level BML FOM module being developed in MSG-106 has many similarities with TNO's earlier implementation of Low-level BML. The new module has been made compatible with typical NETN data types and models a complete set of tasks and reports, but lacks the scenario management functions listed in 0. If these functions are needed in the future, then they will be added to another, more appropriate NETN FOM module.

The C2 Agent federate used in the MSG-106 experiments is based on the Belief-Desire-Intention (BDI) paradigm 0 and implemented on the agent software framework JADEX 0. The HLA interface of both the C2 Agent federate and the VR-Forces plug-in were adapted to be compatible with the NETN FOM modules.

During the experiments the C2 Agent federate was able to transform C-BML orders, e.g. an attack order for a battalion, into sequences of lower level tasks. Platoons simulated by two different CGF tools, i.e. VR-Forces and Pitch Actors, could execute the tasks as expected and could send reports back to the C2 Agent federate.

4. Lesson learnt and way ahead

The results of the NATO MSG-106 experimentation demonstrate the feasibility of the combination of the MSDL, C-BML, HLA standards for system interoperability (C2-Simulations). New C2-Simulation experimentations are planned during the second half of 2013 and all the year 2014. United

Kingdom will be a new partner in these tests by providing the BCIP C2 system and the JSAF simulation.

A research direction could be the introduction of TMR (Transfer of Models Responsibility, i.e. enhancement of HLA transfer of ownership) in aim to link the “high level” and “low level” simulations.

References

[1] Simulation Interoperability Standards Organization, “Standard for: Military Scenario Definition Language”, SISO-STD-007-2008, 2008.

[2] Simulation Interoperability Standards Organization, “Standard for: Coalition Battle Management Language (C-BML)”, SISO-STD-011-2012-DRAFT, 4 April 2012.

[3] A. Alstad et al., “Low-level Battle Management Language”, in Proceedings of Spring Simulation Interoperability Workshop 2013, 2013.

[4] VT MÄK, “Computer Generated Forces – VR-Forces”, http://www.mak.com/products/simulate/computer-generated-forces.html.

[5] M. E. Bratman, “Intention, Plans and Practical Reason”, 1997, Harvard University Press, Cambridge, MA.

[6] “Distributed Systems and Information Systems Group”, Jadex - http://jadex-agents.informatik.uni-hamburg.de (09.04.2011)

[7] Enumerations for Simulation Interoperability - SISO-REF-010-2011.1-RC1 - 24 August 2011 – SISO DIS PSG

[8] “The improvement of simulation interoperability by introducing the C-BML (Coalition Battle Management Language) benefits into the HLA world” – 12F-SIW-009 – J. Ruiz (DGA)

[9] MASA SWORD : Automated, aggregated constructive simulation for efficient training and analysis http://www.masagroup.net/products/masa-sword/

[10] B. Gautreau et al., “Lessons learned from NMSG-085 CIG Land Operation demonstration”, in Proceedings of Spring Simulation Interoperability Workshop 2013, 13S-SIW031, 2013.

Author Biographies

JOSE RUIZ is a technical and operations research manager at the CATOD Center, French Ministry of Defense, Arcueil, France. Since 14 years, he is in charge of simulation interoperability projects dedicated to joint exercises and experiments, in national and international context (ELLIPSE project). He is the manager of French technical activities in the NATO MSG-106.

DAVID DESERT is a Project Manager and Engineer at EADS CASSIDIAN in SDC department (System and Design Center). Actually in charge of training and simulation interoperability throw French ELLIPSE project. He is part of the French technical team for the MSG-106.

PATRICE GUILLOU is a system engineer at Thales Raytheon Systems. He is in charge of system engineering management for Land WAGRAM simulation and French ALLIANCE simulation federation. He is part of the French technical team for the MSG-106.

ANTONY HUBERVIC is a consultant in the professional services team at MASA Group where he is in charge of interoperability issues. As a former member of French Ministry of Defense and of large companies, he has several years of experience on defense simulation projects. He is part of the French technical team for the MSG-106.

ROGER JANSEN is a member of the scientific staff in the Modeling, Simulation and Gaming department at TNO, The Netherlands. He works in the field of distributed simulation and his research interests include distributed computing and simulation interoperability.

NICO DE REUS is a member of the scientific staff in the technology area Modeling & Simulation and Gaming, working in the Defense, Safety and Security theme in the Netherlands’ TNO. His work focuses on the application of Modeling & Simulation and on C2-Simulation interoperability.

HENK HENDERSON is a member of the scientific staff in the technology area Modeling & Simulation and Gaming, working in the Defense, Safety and Security theme in the Netherlands’ TNO. His work focuses on the software engineering aspects of Modeling & Simulation.

KJELL MAGNE FAUSKE is a Senior Scientist at FFI (Norwegian Defense Research Establishment). His research interest is within the area of modeling and simulation, with emphasis on simulation systems interoperability.

LENNART OLSSON is a member of the Service Group at Pitch Technologies. He holds a M.Sc. in Computer Science from Linköpings University. At Pitch he has worked with System Design & Development as a Senior Integration Engineer and Senior Federation Architecture Engineer. He has a long experience of federate development.