

BML-enabling national C2 systems for coupling to Simulation

Nico de Reus, Paul de Krom
TNO Defence, Security and Safety
PO Box 96864
2509JG The Hague, The Netherlands
nico.dereus@tno.nl, paul.dekrom@tno.nl

Ole Martin Mevassvik, Anders Alstad, Geir Sletten
Norwegian Defence Research Establishment, FFI
PO Box 25, NO-2027 Kjeller, Norway
ole-martin.mevassvik@ffi.no, anders.alstad@ffi.no, geir.sletten@ffi.no

PD Dr. Ulrich Schade, Dr. Miloslaw Frey
FGAN-FKIE
Neuenahrer Str. 20,
Wachtberg, 53343, Germany
schade@fgan.de, m.frey@fgan.de

Keywords:

C2 systems, C2IS, C4I systems, Battle Management Language, C-BML, JBML, SISO, MSG-048, Decision Support, Simulation

ABSTRACT: In the SISO product development group (PDG) on Coalition Battle Management Language (C-BML) a language is being defined that has the aim of enabling unambiguous interoperability between C2 systems on one hand and C2 systems, simulations or robotic forces on the other. In order to verify the C2 to simulation part of this claim, NATO MSG-048 used the JBML input to the PDG in C2-Simulation coupling experiments and demonstrated this at I/ITSEC 2007. The JBML (Joint Battle Management Language) is being developed by the C4I Centre at George Mason University (GMU). The JBML specifies the language using XML Schemas and provides web-services to store and retrieve the information in a JC3IEDM database. In the demonstration three national C2 information systems were used, namely the Netherlands ISIS system, the Norwegian NORTaC-C2IS and the US C2PC/CAPES. An editor for a formal C2 language, C2LG (C2 Lexical Grammar), provided by the German FGAN-FKIE institute was used as an add-on to two of the three national C2ISs. The C2LG has been built in order to support the BML grammar development. This paper describes the adaptation that was done to ISIS and NORTaC-C2IS using a C2LG editor. It also describes briefly the limited experiment that used the BML-enabled C2ISs to interface to two national simulators (the French SCIPIO and the Spanish SIMBAD).

1. Introduction

Advances in Coalition Interoperability are being developed along two axes – Net-Centricity and standard Command and Control (C2) Semantics. The latter is dealt with by the Multilateral Interoperability Programme (MIP) and the development of the JC3IEDM (Joint Consultation, Command and Control Information Exchange Data Model). MIP provides a shared semantic of military terms and a mechanism to exchange data. In order to allow automatic processing of military information (i.e., orders, requests, and reports) as well as capturing the intention under which they are generated, a formal mission-oriented language is needed.

Military information requires seamless transfer without delays or distortions which can only be achieved if automation is applied. The automation not only accelerates the exchange, but is also necessary because

C2 networks are composed of not only humans but systems, as well. There are C2 systems to support the C2 process and to build the common operational picture. There are sensors that provide relevant data. In the future, we expect increased use of robotic forces. And, last but not least, there are simulation systems under development to support mission preparation, mission execution, mission rehearsal, and training in a network environment. All these systems rely on automated exchange of data and information; these requirements cannot be met by the current methods of representing military information (commonly in the form of free-text). A new approach is needed.

The communication and interoperability challenges related to automated information exchange and processing require a formal common language. To be processable by systems, this language has to be clear and unambiguous. In addition it has to be sufficiently expressive to enable a

commander to convey his orders while preserving his intent and to allow for reporting all aspects relevant to an ongoing operation. In order to preserve meaning and intent, the language has to derive from standard military doctrines already defined. Such a language is a “Battle Management Language” or BML.

A great amount of effort has gone into creating a standard C2 data model by the Multinational Interoperability Program (MIP) with its data model the JC3IEDM. However, the interoperability is on a data level, passing sets of data defined by operational requirements. BML adds an additional structuring of the data using linguistic principles to develop a language for operations. This language is more general than the MIP, but maps easily into both the standard operational message formats as well as the JC3IEDM.

BML has to support communication between C2ISs and their users or related systems. To be more precise BML has to support the interoperability of C2IS to C2IS, C2IS to simulation and in the future C2IS to robotic forces. This is displayed in Figure 1.

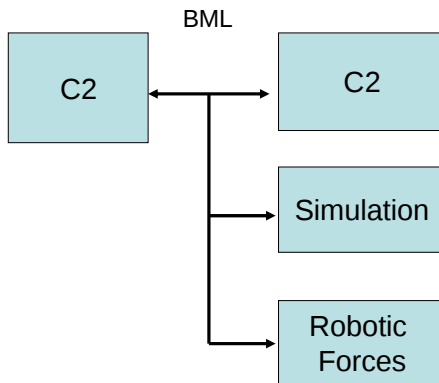


Figure 1: BML enabling interoperability

An initial demonstration of a Coalition Battle Management Language (C-BML) was given by NATO MSG-048 at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2007 in Orlando FL, where several C2ISs and simulations were coupled using BML. An overview of this demonstration is given in [1].

This paper elaborates on how the C2ISs were BML-enabled for this demonstration. The C2ISs were the US C2PC/CAPEs, the Norwegian NORTaC-C2IS and the Netherlands ISIS. Only C2PC/CAPEs had a native BML capability before the demonstration. This paper describes the approach chosen to BML-enable the Norwegian and Netherlands systems. The described approach and lessons

learned may be useful to other nations that want to BML-enable existing C2 systems for experimentation.

2. C-BML

The NATO technical activity MSG-048 C-BML charter is to evaluate and recommend a BML standard for NATO. BML currently under development by SISO will be called Coalition BML or C-BML. One of the inputs to the SISO Product Development Group (PDG) is JBML, the US Joint Battle Management Language (JBML), being developed by George Mason University (GMU) [2]. JBML is used in the MSG-048. The following section gives a brief description of JBML.

2.1. JBML

The US Joint Battle Management Language project started under the name Extensible Battle Management Language (XBML) in 2004. The initial thrust of this project was to demonstrate the value of Web services as a basis for interoperation of legacy software, combined with the value of the Command Control Information Exchange Data Model (C2IEDM) as the basic data model for C2-simulation interoperation [3, 4]. The project was among the first to demonstrate the rapid development potential of Service Oriented Architecture (SOA) for distributed systems. It also validated the utility of the C2IEDM in two different ways. First, it demonstrated that the C2IEDM could be used for simulation C2 data with minimal extensions (only four tables needed to be modified, out of the hundreds in the data model). Second, the C2IEDM and Web services were put to practical use very quickly when the XBML prototype was linked to the French APLET system, showing both the generic applicability of the C2IEDM and the rapid development that is possible using SOA.

The ongoing JBML project has been used as a candidate for the standard being developed by the SISO PDG. To achieve this, the development team undertook a fresh analysis of the BML problem. Where XBML had replaced “stovepipe” interfaces with Web services, JBML sought to create a truly general and extensible language, implemented using SOA. Following the work of Hieb and Schade as reported in [5] a concise and grammatically correct approach to representing BML orders was formulated and represented as an XML schema.

The grammar and XML schema in turn became the basis for a three layer architecture, following the work of Tolk as presented in [6]. In this layered approach the developer/user accesses the topmost layer while the bottom layer provides for representing the BML in the JC3IEDM. This approach enables the developers to access the JC3IEDM database avoiding the need to learn

the complex JC3IEDM data model; it also guarantees that the database will be updated in a way that cannot lead to inconsistencies due to interleaving access to tables by different transactions.

The JBML project held a successful proof of principle demonstration in May, 2007, where orders from C2 systems of ground, air, and maritime elements were passed through a common Web Service to two different, linked instances of the JSAF simulation, validating the ability to use the grammar-inspired schema and the multi-layered service for Joint operations. The JBML software that was demonstrated is available as open source on the web site of the developer, the GMU C4I Center. The JBML project will continue to expand the scope of possible orders while also developing a similarly well-structured approach to feedback of situational awareness data from simulations to C2 systems.

2.2. C2LG

BML has to be a language that can be processed automatically. For the current implementation this means, it must be possible to map BML orders automatically into the JBML schema. To be automatically processable, the BML has to be designed as a formal language. A formal language is generated by a grammar. The grammar we use is the Command and Control Lexical Grammar (C2LG) proposed by Schade&Hieb in [5] gives some background on linguistics and grammar. This section not only describes the C2LG but also contributes to the discussion initiated by Tolk et al in [14] whether a BML grammar has to be regular or context-free.

The need for a context-free grammar

A formal language is defined by a grammar. According to linguistic theory, a grammar is a quadruple. It consists of a set of so-called “terminal symbols”, a set of “non-terminal symbols”, a starting symbol that is part of the set of non-terminals, and a set of production rules. The terminal symbols are the words of the language. They form the lexicon. The non-terminal symbols represent constituents that are those sequences of words a label can be assigned to. E.g., the “5 Ws” can serve as appropriate non-terminals in a BML. The rules define how the lexical elements can be combined. Grammars are categorized into four types that form the Chomsky hierarchy [7],[9]. These types are grammars of type 0 (unrestricted grammars), grammars of type 1 (context-sensitive grammars), grammars of type 2 (context-free grammars), and grammars of type 3 (regular grammars). The type of a grammar depends on the rules used. The higher the type the more severe are the restrictions the rules have to follow. Only grammars of types 2 and 3 exclusively use

rules that can easily be applied by automated systems. Therefore, a BML has to have a grammar of type 2 or 3. Grammars of type 2 (context-free grammars) allow rule of the type “ $A \rightarrow \psi$ ”; in contrast, grammars of type 3 (regular grammars) only allow two types of rules “ $A \rightarrow a$ ” and “ $A \rightarrow aB$ ”. Here “ a ” represents a terminal symbol (a word), “ A ” and “ B ” represent non-terminal symbols, and “ ψ ” represents an arbitrary string of terminal and non-terminal symbols.

In order to decide which type of grammar should be chosen for a BML, let us take a look at an example expression, namely “*advance to area Alpha as soon as possible*”. With the 5 Ws (Who, What, Where, When, Why) in mind, the automatic analysis should divide this expression into three pieces, namely “*advance*”, “*to area Alpha*”, and “*as soon as possible*”. We therefore would like to have a rule like “Order $\rightarrow$ Task Where When” such that Task (the What of the 5Ws) can be expanded into “*advance*”, Where can be expanded into “*to area Alpha*”, and When can be expanded into “*as soon as possible*” by the further application of other rules. The rule “Order $\rightarrow$ Task Where When” is a rule of a context-free grammar. This is not allowed in a regular grammar. Using a regular grammar would mean analyzing the example order by regular rules in an incremental way. A regular rule that can start the analysis of our order could be “Order $\rightarrow$ *advance* Non-terminal-1”. In contrast to the non-terminal symbols Task, Where and When we used for the context-free rule, in the regular rule we have the meaningless symbol “Non-terminal-1”. Even worse, we would need an even more meaningless rule “Non-terminal-1 $\rightarrow$ *to* Non-terminal-2” to go on with analysis followed by the equal meaningless rule “Non-terminal-2 $\rightarrow$ *area* Non-terminal-3” and so on. Obviously, regular rules have to be expressed with meaningless non-terminals whereas context-free rules use non-terminals that represent the 5 Ws.

Why is it important that the non-terminals bear meaning? The answer is easy. The analysis of a BML expression that can be done with the exploitation of its rules (the syntactical analysis) is only the first step. Next to the fact that BML orders are supposed to be human understandable, they also must be usable to command simulated units, the analysis must be processed automatically and it must be done in such a way that the final result of this process can trigger the correct simulated behaviours. In our example, the “*advance*” has to be mapped onto a move routine and “*to area Alpha*” has to become the destination of that move. Thus, the constituents (the words of the BML expressions that are grouped together by the syntactic analysis) have to be labelled in a way that gives the labels meaning, in this

case, the meaning that the labelled constituent describes the destination of a move. The 5 Ws form a fine set of such labels; an even better set is formed by the thematic roles proposed by Sowa [13]. By using a context-free grammar, the desired labels (the 5 Ws or Sowa's thematic roles) can serve as non-terminals whereas the use of a regular grammar necessarily constructs meaningless constituents (with meaningless non-terminals) and no node in the syntax tree to put the desired labels on. In summary, BML has to have a context-free grammar such that the constituents resulting from the syntactic analysis can be assigned meaningful labels in a second step. The use of a regular grammar to define a BML is not adequate, although the contrary has been asserted in [14].

Tasking Grammar

The C2LG can be used to formulate orders [5],[10] and reports [11]. In addition, there are rules for expressing the Command Intent [8]. All these parts are needed for interaction between C2 systems and simulation systems if the simulations are supposed to serve as a decision support tool [12]. In the I/ITSEC 2007 experiment, however, the task was "only" to command simulated units. Thus, it is sufficient to take a look at only those rules that are needed to task units. These rules form the subset of the C2LG rules known as "Tasking Grammar". A single task is assigned to a unit by a basic order expression. C2LG's basic order expressions have the form

(1) OB →
Verb Tasker Taskee (Affected|Action) Where
Start-When (End-When) Why Label (Mod)*

In this form Verb represents the task. Task verbs are taken from JC3IEDM's table "action-task-category-code". Tasker is the unit or individual that assigns the task, and Taskee is the unit that has to execute it. Start-When and End-When express when the task has to start and when it has to be finished, respectively. End-When is optional as indicated by the brackets. Why denotes a reason for the assignment of the task. Label is a unique identifier for the task so that it can be referred to in other expressions. Mod (modifier) is a wild card that is used for describing the formation of a movement or the manner in which the task has to be executed, e.g. as fast as possible or cautiously and without taking avoidable risks. Affected and Action are only used in some basic order rules. Affected is used if the task directly affects someone or something, e.g. the enemy in the case of **attack**. Action is used in a similar way, namely if the task in question affects another action, e.g. in the case of **assist**. As the use of these two terms in a rule depends on the verb, there is one basic order rule for each task verb in the tasking grammar. (This is one of the reasons the grammar

is a "lexical" grammar.) In addition, the type of the Where also depends on the tasking verb. Some tasks demand a Route-Where – a language expression that denotes a route -- whereas others demand an At-Where – an expression that denotes a location. In linguistics, the constituents that a verb demands and allows form the verb's so-called frame (cf. FrameNet). In order to illustrate the dependency between the task verbs and their frames, (2) lists some of C2LG's basic order rules.

- (2a) OB → **advance** Tasker Taskee
Route-Where Start-When (End-When)
Why Label (Mod)*
- (2b) OB → **ambush** Tasker Taskee Affected
At-Where Start-When (End-When)
Why Label (Mod)*
- (2c) OB → **assist** Tasker Taskee Action
At-Where Start-When (End-When)
Why Label (Mod)*
- (2d) OB → **rest** Tasker Taskee
At-Where Start-When End-When
Why Label (Mod)*

Example (3) describes the basic order to advance along a pre-defined route called "Duck" given to the 13th Netherland Mechanized Brigade (M_BDE13(NL)) by the Multinational Division West (MND-West). In the example, the Why constituent is deleted for reasons of distinctness. For details on the Why and its links to Command Intent see [8].

(3) **advance** MND-West M_BDE13(NL)
along DUCK **start at** Phase1A label_3_11;

In order to use the orders in the experiment's simulation systems, they are processed in two steps. First, for each basic order expression, its constituent structure is calculated. Second, this structure is mapped into JBML. Because JBML is based on C2LG, the XML tags of the JBML schema correspond to C2LG's non-terminals. For example, in (3) M_BDE13(NL) is the Taskee constituent. It is tagged <TaskeeWho> in JBML.

```

<bml:GroundTask>
  <bml:TaskeeWho>
    <bml:OrgName>M_BDE13(NLD)</bml:OrgName>
  </bml:TaskeeWho>
  <bml:What>
    <bml:Action>ADVANCE</bml:Action>
  </bml:What>
  <bml:Where>
    <bml:WhereLabel>DUCK</bml:WhereLabel>
    <bml:RouteWhere>
      ...
    </bml:RouteWhere>
  </bml:Where>
  <bml:StartWhen modifier="AT">
    <bml:RelativeToTask>Phase1A</bml:RelativeToTask>
  </bml:StartWhen>
  ...
  <bml:Label>label_3_11</bml:Label>
</bml:GroundTask>

```

Figure 2: JBML XML example

3. C2 systems used

This section describes the BML capable US C2 system briefly and the non-BML capable Norwegian and Dutch C2 systems in more detail.

3.1. The US C2 system C2PC/CAPES

The US C2PC is a non-classified C2IS used in the MSG-048 demonstration. It is developed by the US Marine Corps and is also used by the US Army. It features an architecture supporting injector modules that can manipulate and display external data. The US Army CAPES (Combined-Arms Planning and Execution System) and a BML module were used as injector modules to enable C2PC to generate BML.

3.2. The Netherlands C2 system ISIS

The Royal Netherlands Army C2 Support Centre (C2SC) is developing a generic, configurable and distributed Command and Control information system. This system, known as C2 Framework (C2FW), is the baseline for a suite of C2 applications that will provide staff sections, vehicles and individual combatants with a common operational picture. The C2FW is a configurable *application platform* and information system that provides generic functionality to support the C2 process.

The C2FW supports the users in building and maintaining a Common Operational Picture (COP) that provides Situational Awareness. The C2FW is the foundation for a family of C2 Information Systems. The Integrated Staff Information System (ISIS) is aimed at the static domain (compound, command post). It is developed and used within the Royal Netherlands Army as a main C2 application for issuing orders and delivering a COP throughout the mission. Other systems, based on the

framework are OSIRIS and XANTHOS which are used in the mobile (command vehicles, tanks ...) and dismounted domain (dismounted commanders, soldiers).

Figure 3 shows how ISIS enables the commander to view tactical data in the form of a COP and assemble plans to be sent out to the users. The plans are in the form: Operation plan (OPLAN), Order Of Battle and Overlay displaying the commander's plan graphically. The OPLAN is in accordance with the five paragraph NATO standard [15].

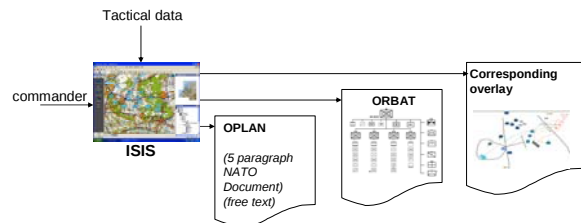


Figure 3: ISIS Input/Output

ISIS is a MIP enabled system, meaning that the data on this system can be aligned with other MIP enabled C2 systems using a MIP gateway.

The free text used in the OPLAN and the ISIS overlay do not contain enough information to unambiguously and automatically generate BML statements by non-humans. In other words, the information is there, but cannot be extracted automatically. This was the reason why the ISIS system had to be BML-enabled.

3.3. The Norwegian NORTaC-C2IS

NORTaC-C2IS is a part of the ComBatt product suite for tactical army operations and was fielded in 2002. ComBatt is developed by Kongsberg Defence & Aerospace for the Norwegian Army and includes a C2IS, fire support system and battle management system.

The main functionality of NORTaC-C2IS is planning, establishing, sharing and maintaining the recognized land picture and order of battle management. The exchange of data with external systems and to other components in the ComBatt suite makes use of MIP data models and information exchange mechanisms. NORTaC-C2IS currently supports MIP block II – the C2IEDM.

In the MSG-048 initial C-BML demonstration NORTaC-C2IS was used to present the plan to the military user. As yet, the planning capabilities as defined by MIP are limited to the presentation of graphical objects and storing

elements of a plan as free-text. However, mapping a plan as defined by the JBML into the C2IEDM data model will allow for MIP compliant systems to present additional elements of the plan.

This approach allowed for NORTaC-C2IS to give a graphical presentation of the plan showing unit positions, control measures (e.g. unit boundaries) and actions (e.g. defend). As yet there is no support for presenting the relations between units, tasks and the temporal information associated with tasks. These would be the next steps in order to introduce a limited BML planning capability.

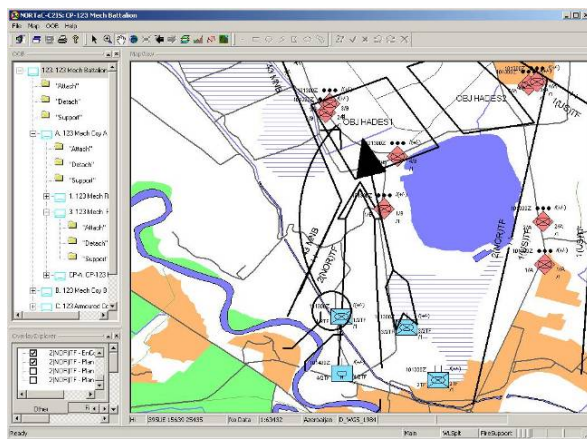


Figure 4: Phase 1 of the Norwegian OPLAN

Phase 1 of the Norwegian OPLAN and perceived enemy situation displayed in NORTaC-C2IS are shown in Figure 4.

3.4. The German C2LG GUI

FGAN-FKIE has developed an editor (Figure 5) that can be used to formulate orders (and reports) according to C2LG's rules. This editor is referred to as C2LG GUI. The GUI includes plug-ins that allows it to be connected to other systems. The plug-in mechanism has been used to interface with the Norwegian and Netherlands C2 systems.

As can be seen in Figure 5, the GUI uses drop-down menus and also a map. Units, facilities, features and locations can be selected in the map to speed up formulation of the order, especially formulation of the "Where". When a BML order is completed it is processed in two steps. First, the constituent structure is calculated; second, this structure is transformed into a functional structure and thematic labels that serve as tags in the JBML are assigned to the constituents. The output format

is XML, respecting the JBML schema. As such, the output can be delivered to the simulation systems by the JBML web services.

The initialization input plug-in is used to pre-fill the GUI with Order Of Battle (ORBAT) information and location of units which is drawn from the JBML database.

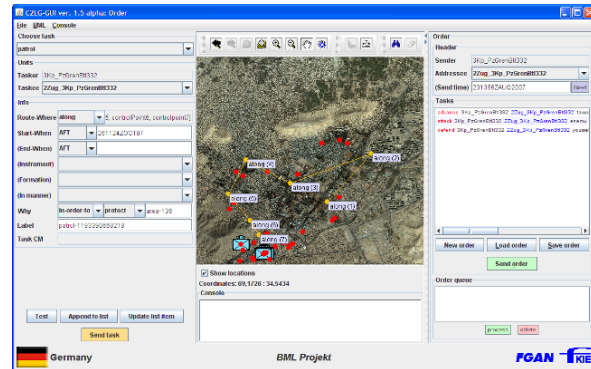


Figure 5: Snapshot of the C2LG GUI showing the formulation of a patrol order

4. BML-enabling the Norwegian and Netherlands C2ISs

The Norwegian and Netherlands C2ISs used in the MSG-048, ISIS and NORTaC-C2IS, are not BML-capable as described in the previous sections. A BML interface was developed by using the C2LG GUI and by utilizing their existing interfaces, which for NORTaC is the MIP support and for ISIS is the C2-Framework (C2FW). The architecture used for connecting the systems is shown in Figure 6.

JBML plug-ins were developed for the C2LG GUI in order to output JBML to a database through web services and to receive incomplete JBML from ISIS and NORTaC-C2IS. For assembling an order, the data flow is as follows: the order available in the C2 systems (ISIS or NORTaC, respectively) is transformed into JBML and loaded into the C2LG GUI via the input plug-in. The C2LG GUI checks this order and enforces the addition of missing information.

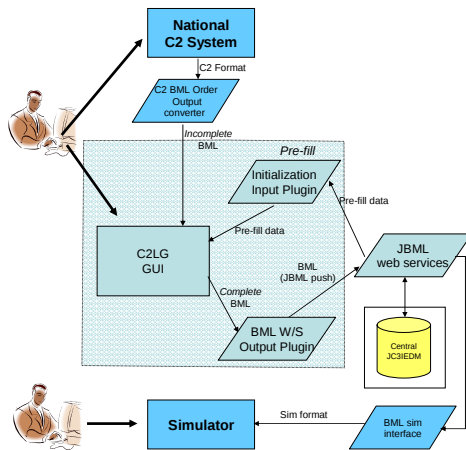


Figure 6: Architecture for the C2 systems

4.1. ISIS / C2LG GUI coupling

The ISIS(C2FW)-HLA Gateway developed at TNO [16] has been used as a basis to develop the ISIS / C2LG GUI coupling. This gateway was developed for the purpose of incorporating ISIS in a Federation of the TNO KIBOWI constructive simulation. The code to get information from the C2FW was reused.

As explained in section 3.2. the information in ISIS contains the free text 5 paragraph NATO order, the ORBAT and the tactical overlay. The concept that is used for the transformation of the C2 information available in ISIS to JBML is to generate an *incomplete BML order* from the information in the tactical overlay on ISIS (not using the free text for this) and have the commander completing this information in the C2LG GUI.

The elements on the overlay consist of control measures, tasks and units. These elements are pulled from the C2 Framework into the modified C2FW-HLA gateway, assembled into an incomplete order and sent to the C2LG GUI. This incomplete order contains all the relevant information available in ISIS. For the current scenarios in the demonstration this means that all the tasks in the order overlay on ISIS are present in the incomplete order. The only choice the commander still has to make in the C2LG GUI is which units are to execute which tasks.

4.2. Coupling NORTaC-C2IS to C2LG GUI

As described in section 3.3. the chosen approach was to store the complete OPLAN defined in SQL into the NORTaC-C2IS C2IEDM database. This allowed NORTaC-C2IS to present the order graphically. The complete JBML formatted OPLAN may also be sent from

NORTaC-C2IS to the C2LG GUI. A NORTaC JBML translator was developed for this purpose.

NORTaC-C2IS has two databases, one internal database which interacts with the GUI, and a C2IEDM database for data exchange with other systems. Mapping rules can be set up to automatically synchronise the two databases. The operational plan is submitted to the C2IEDM database by the use of SQL scripts. Although the use of SQL scripts is not a practical way of inputting plans into a C2IS, this approach made a proof of concept on BML-enabling MIP compliant systems. In order to demonstrate how data created by the C2IS can be used as a starting point for creating orders, control measures (areas and lines) created by NORTaC-C2IS were translated into JBML and transferred to the C2LG GUI.

The JBML translator's main function is to map between a C2IEDM relational database and a JBML XML document. The translator application needs to read a fairly large number of tables in the C2IEDM database and map them to XML elements defined by the JBML XML Schemas. This implementation task was greatly simplified by the use of open source projects and code generation tools.

Because of its wide support and the many available open source projects, Java was selected to create the translator application. The open source Java project Hibernate was used to map data from the relational database to Java objects. XML mapping data enabled Hibernate to perform SQL queries to move data between Java objects and database tables.

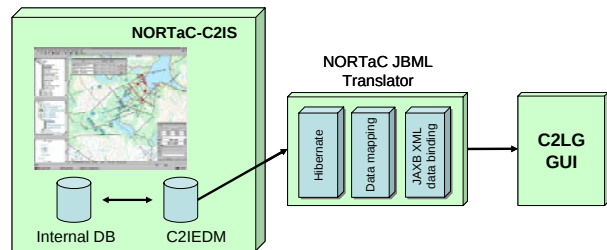


Figure 7: BML-enabling the NORTaC-C2IS

A plug-in to the Eclipse Integrated Development Environment (IDE) named "Hibernate Tools" was used for reverse engineering the C2IEDM database, generating Java code and mapping XML to be used by Hibernate. By using this tool basically all the Java classes necessary to interact with the C2IEDM database were automatically generated. While this made the development easier and faster, it also results in code with fewer bugs.

The Java API for XML Binding (JAXB) was used to provide an XML data binding. JAXB generates Java classes corresponding to an XML Schema, in our case the JBML XML Schemas. Thus the majority of the work to develop the NORTaC JBML Translator was to perform the mapping from C2IEDM Java objects to JBML Java objects. Figure 7 shows the architecture for the translator application.

5. MSG-048 initial demonstration

This section briefly describes the demonstration held by NATO MSG-048 at I/ITSEC 2007. A more comprehensive description can be found in [1].

5.1. Scenario; Main orders and FRAGO

The overall scenario developed for the experiment is described in [1]. The Commander Joint Forces Land Component Commander (CJFLCC) has decided to commit the 43rd Multi-National Brigade (43 MNB - Figure 3) composed of three Battalion (Bn) Task Forces (TF) to secure the Kazi Magomed Airport. For this mission, the Commander 43 MNB requires the use of two of his three BN TFs. He has selected 1(USA)TF, a US Combined Arms Battalion, and has to decide between 2(NLD)TF (Netherlands) and 2(NOR)TF (Norway) for his second TF. Figure 8 shows the order of battle.

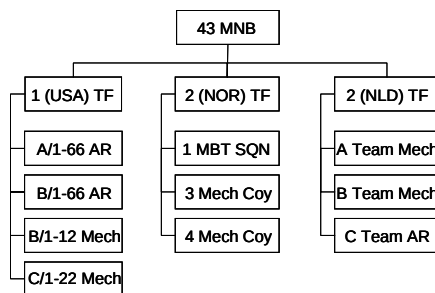


Figure 8 Order of battle

The 43MNB has tasked the commanders of NLD (Maneuver Battalion) and NOR (Telemark Battalion) both to plan for the same assignment. Therefore 2(NOR)TF and 2(NLD)TF each developed a Course of Action (COA). The outcome of the simulation-based COA-analysis will support the commander's decision.

In the case of the 2(NOR)TF an OPLAN was developed by the Norwegian Army Combat Manoeuvre Training Centre (NACMTC) in accordance with STANAG 2014 [15] and national army planning framework. The plan consists of three phases, each basically securing one of three defined objective areas (Hades 1, 2 and 3). 2(NOR)TF consisted of one main battle tank squadron

and two mechanized infantry companies, of which one was reinforced by a main battle tank platoon. The OPLAN was formulated at the company level, that is, the tasks are assigned to the companies and the squadron only. The plan is shown in Figure 9.



Figure 9 Plan for the 2(NOR)TF

In the case of the 2(NLD)TF an OPLAN was developed by the Military Subject Matter Experts of TNO in accordance with the NLD army doctrine. The plan consists of two phases. In phase 1, objectives Hades 1 and 2 will be secured and during phase 2 Hades 3 will be taken and secured by the task force's reserve team. 2(NLD)TF consisted of 2 mechanized infantry teams and one armoured reconnaissance team. The plan is shown in Figure 10.

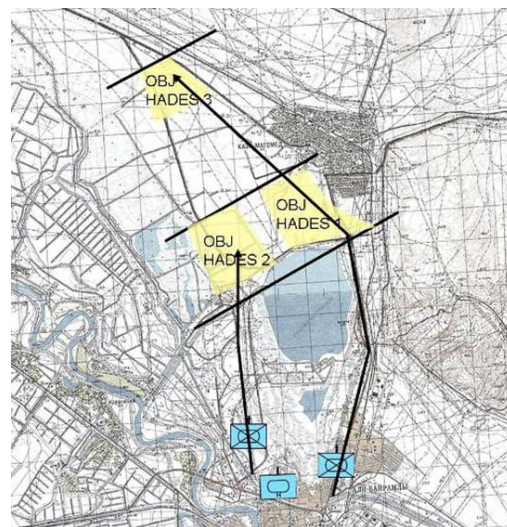


Figure 10 Plan for the 2(NLD)TF

5.2. Demonstration Architecture

In order to perform Course Of Action Analysis for both task forces with the indicated variations (1(USA)TF, 2(NOR)TF, 2(NLD)TF), the following architecture was used.

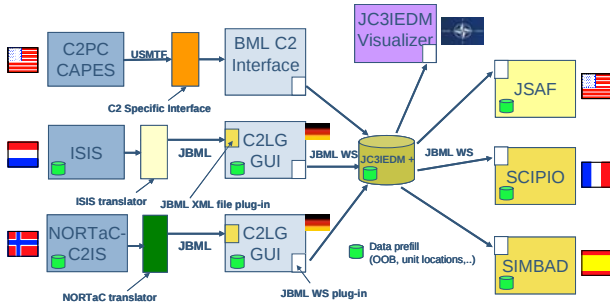


Figure 11 Demonstration Architecture

Three C2ISs and three simulations were involved. The C2ISs produced in JBML format the OPLANs generated by the TF commanders.

Joint Semi-Automated Forces (JSAF) was used to simulate the OPLAN of 1(USA)TF as assembled on C2PC/CAPEs. SCIPIO was used to simulate the OPLANs of 2(NOR)TF and 2(NLD)TF as assembled on NORTaC and ISIS. SIMBAD also simulated the OPLAN for 2(NLD)TF as assembled on ISIS. The OPLANs were stored in the JBML database using web services and pulled from the database for execution by the simulators. The JC3IEDM Visualizer presented the content of the plan database.

6. Lessons Learned

Simulations and C2 systems from different coalition partners were successfully integrated by the use of BML in just a few months. The C2 systems and simulations stored and retrieved plans through JBML web services serving as a front end to a JC3IEDM database. The use of service oriented architectures facilitates loose coupling between systems and asynchronous information exchange, making systems easy to integrate.

The lessons learned in the experiment from the simulation point of view are reported in [1]. One issue that influenced the capabilities of the C2ISs, side was that only a limited set of tasks could be implemented by the simulations during the time available. This was especially challenging due to the fact that the simulations implemented their national doctrine. This meant for the

demonstration that the commander on the C2 side was confined in the choices he could make. Solving this would call for simulators being capable of performing more tasks that are in line with the JC3IEDM task list.

The OPLANs, developed by military personnel, were mapped to JBML with only minor changes. Some challenges were:

- There was a lot of flexibility built into the plan, e.g. it included back up tasks like “on order be prepared to support unit x”.
- The order of battle could change during the operation (attachment and detachment of units).
- Control measures might be activated and deactivated during the operation, e.g. fire coordination lines.
- The demonstration required more elaborate task scheduling: The conditional starting of orders (start order B after finishing order A) was not possible. Extending BML with situation reports flowing from the simulation to C2IS is one step towards solving this problem. This will allow the implementation of external scheduling mechanisms or allowing a human to perform this task. Situation reports are planned to be part of the next phase of JBML development.

While the JBML is rooted in the JC3IEDM, the JBML XML Schema and the C2 language grammar make it easy to use the language without being a JC3IEDM expert. The JBML XML Schema allowed us to describe and validate our plans in JBML in parallel with the development of C2IS gateways and JBML web services plug-in for C2LG GUI. The use of the C2LG GUI as a generic module to generate JBML has proven to be very powerful and flexible..

In the current version of JC3IEDM there are structures to represent plans and orders according to STANAG 2014. It is questionable, though, whether this will make any significant impact (or any at all) regarding the ability to represent an order formally. The details would still be captured in the current action structure; and the additions will basically cater for more textual structuring in between these actions.

Only a small number of extensions have been made to the JC3IEDM in the JBML specification. Furthermore, the major challenges listed in the beginning of this section may be modelled fairly well. This underpins the suitability of JC3IEDM as a basis for further development of BML.

7. Way forward

The next phase of the MSG-048 experimentation will expand the C2 capabilities in the following areas:

- C2IS to C2IS coupling that enables higher level C2 to command lower level C2. This will require the C2 systems to pull orders from the JBML database.
- Situation reports flowing back from the simulations will be processed and presented by the C2ISs.
- With the introduction of reports from the simulation to the C2 systems there will be requirements for a two-way interface. Thus, further development on BML exchange mechanisms (web services) should consider requirements for synchronous information exchange or a publish/subscribe architecture.

Geospatial research that is currently performed by GMU will result in a data model that presents a common model for geospatial objects, referred to as geoBML. Integration of geoBML into C-BML experiments will be part of future BML research. The simulation side of the C2 – simulation coupling is discussed in [1].

8. References

- [1] Pullen, J., Carey, S., Cordonnier, N., Khimeche, L. Schade, U., de Reus, N., Le Grand, N., Mevassvik, O.M., Galan, S., Gonzales Godoy, S., Powers, M., Galvin, K., "NATO MSG-048 Coalition Battle Management Initial Demonstration – Lessons Learned and Way Forward", IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2008
- [2] Levine, S., M. Pullen, M. Hieb, C. Pandolfo, C. Blais, J. Roberts and J. Kearly, "Joint Battle Management Language (JBML) Phase 1 Development and Demonstration Results," IEEE Fall Simulation Interoperability Workshop, Orlando, FL, 2007
- [3] Sudnikovich, W., J. Pullen, M. Kleiner, and S. Carey, "Extensible Battle Management Language as a Transformation Enabler," in SIMULATION, 80:669-680, 2004
- [4] Tolk, A. and J. Pullen, "Using Web services and Data Mediation/Storage Services to Enable Command and Control to Simulation Interoperability," 9th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2005), Montreal, Canada, 2005
- [5] Schade, U. & Hieb, M.R. „Formalizing Battle Management Language: A Grammar for Specifying Orders”, Spring Simulation Interoperability Workshop, Huntsville, Alabama, April 2006.
- [6] Tolk, A, Diallo, S., Dupigny, K., Sun, B. and Turnitsa, C., “A Layered Web services Architecture to Adapt Legacy Systems to the Command & Control Information Exchange Data Model (C2IEDM)”, (IEEE European Simulation Interoperability Workshop, 2005
- [7] Chomsky, N. (1957), “Syntactic Structures”, The Hague, NL: Mouton.
- [8] Hieb, M.R. & Schade, U., „Formalizing Command Intent Through Development of a Command and Control Grammar”, 12th ICCRTS, June 2007. Newport, Rhode Island.
- [9] Partee, B.H., ter Meulen, A. & Wall, R.E. (1990). “Mathematical Methods in Linguistics”, Dordrecht, NL: Kluwer
- [10] Schade, U. & Hieb, M.R., „Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports”, 11th ICCRTS, Cambridge, UK, September 2006
- [11] Schade, U. & Hieb, M.R., „Battle Management Language: A Grammar for Specifying Reports”, Spring Simulation Interoperability Workshop, Norfolk, Virginia, March 2007
- [12] Schade, U. & Hieb, M.R., „Improving Planning and Replanning: Using a Formal Grammar to Automate Processing of Command and Control Information for Decision Support”, The International C2 Journal, 1(2), 69-90
- [13] Sowa, J.F. (2000), “Knowledge Representation”, Logical, Philosophical, and Computational Foundations. Pacific Grove, CA: Brooks/Cole.
- [14] Tolk, A., S. Diallo & C. Turnitsa, “A System View of C-BML”, Fall Simulation Interoperability Workshop, Orlando, Florida, September 2007.
- [15] NATO: STANAG 2014, Formats for orders and designation of timings, locations and boundaries, 17 October 2000, NATO/PfP Unclassified.
- [16] Huiskamp, W., Kwaijtaal, A., Fiebelkorn, S., “Interoperability between the RNLA C2 Workstation and the ‘KIBOWI’ Constructive Simulation Tool for Operational Support and Training”, NATO M&S Conference 2003, Antalya, Turkey, RTO-MSG022-SY003-Paper 9.

9. Author Biography

NICO DE REUS is a member of the scientific staff in the M&S department at TNO Defence, Security and Safety in the Netherlands. His current work focuses on modelling & Simulation in general and bringing Simulation to the Battlefield in specific. His current work focuses on C2-Simulation interoperability.

PAUL de KROM is a member of the scientific staff in the M&S department at TNO Defence, Security and Safety in the Netherlands. His current work focuses on C2-Simulation interoperability.

OLE MARTIN MEVASSIK is a Principal Scientist at the Norwegian Defence Research Establishment (FFI). His research interest is within the area of modelling and simulation, with application to training and experimentation.

ANDERS ALSTAD is a Research Scientist at the Norwegian Defence Research Establishment (FFI). His work is within the field of modelling and simulation. Anders is interested in design and development of distributed systems.

GEIR SLETTEN is a Senior Scientist at the Norwegian Defence Research Establishment (FFI). His current work is focusing on data modelling, and he has been involved in the development of JC3IEDM and earlier versions in MIP for 10 years.

ULRICH SHCADE is a Senior Scientist at the Research Institute for Communication, Information Processing and Ergonomics that is part of FGAN financed by the German MoD and is Associate Professor for Linguistics to the Institute for Communication Science, Bonn University. He serves as principal investigator of Bundeswehr IT-Office's BML projects.

MILOSLAW FREY is Research assistant at the Research Institute for Communication, Information Processing and Ergonomics that is part of FGAN financed by the German MoD. He received his master for Physics from Cracow University, Poland, and his Ph.D. for Linguistics from Bonn University, Germany. His research interests are connectionist systems and semantic networks.