Battle Management Language:
Military Communication with Simulated Forces

Ms. Kellyn Kruger, Mr. Miłosław Frey and Dr. Ulrich Schade
FGAN-FKIE
Neuenahrer Straße 20
53340 Wachtberg-Werthhoven
GERMANY

email: kruger@fgan.de / m.frey@fgan.de / schade@fgan.de

ABSTRACT

In the increasingly complex worlds of battlefield, anti-terrorism, peace-keeping and disaster relief operations there is clearly a need for rapid, useful, precise and unambiguous exchange of information between military units within a given theater of activity. Orders, requests and reports need to be expressed in formal language sufficiently standardized to be unambiguous, while sufficiently expressive to convey a commander's intent or to report enemy or civilian activities. This language must also support network-centric communication requirements by facilitating automatic processing and dissemination of information. If military communication can be processed automatically, it can be exchanged not only among forces and their C2-systems, but also between commanders, C2-systems and simulation systems. Commanders then could directly command simulated forces. Simulation systems thus could be used during operations or exercises, e.g., as decision support systems or in staff exercises.

A formal, context-free language based upon the JC3IEDM, Battle Management Language (BML) is currently being developed to this end. BML's grammars for orders and requests are well-developed and have been integrated successfully into several prototype simulation systems. The grammar for reports is mostly complete and is currently under further refinement to adequately represent parameters such as source, quality and credibility of the information being reported. In this paper we will discuss the integration of BML in simulation systems and demonstrate the GUI that allows commanders to formulate BML orders to command simulated units. In addition, we will show how simulation systems may use BML to generate reports responses.

1.0 INTRODUCTION

Each day the complexity of battlefield, anti-terrorism, peace-keeping and disaster relief operations is growing. There is clearly a need for flexibility to react to continuously changing situations and to make optimal use of information obtained from a variety of sources. Retaining control of a situation and staying one step ahead requires agility [1,2], especially the capability to quickly identify and analyze situations and patterns of behavior, to evaluate these based upon knowledge available, and to recognize and exploit unexpected opportunities in order to deliver a rapid and appropriate response to a given situation.

A fundamental requirement for these capabilities is rapid, precise and unambiguous information exchange and analysis. Communications need to be networked to be available as appropriate to various actors within the theatre of operation. Human-to-human communications comprise only one part of the picture: in order to provide optimal situation analysis, this information exchange should include communications to and from both human and non-human sources such as autonomous (robotic) vehicles and sensors.

Kruger, K.; Frey, M.; Schade, U. (2007) Battle Management Language: Military Communication with Simulated Forces. In *Improving M&S Interoperability, Reuse and Efficiency in Support of Current and Future Forces* (pp. 5-1 – 5-10). Meeting Proceedings RTO-MP-MSG-056, Paper 5. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int.

Furthermore, because of the enormous volume of communications traffic in military endeavors, regardless of deployment type, standardized communication which makes automatic processing and analysis possible supports effective, efficient and appropriate decision-making.

Standardized communication which can be automatically processed has yet another advantage: it may be used for communication between C2-systems and simulation systems. Connecting C2 systems and simulation systems is desirable for many applications, e.g. for decision support and for training purposes. In decision support, simulations are used to predict and project the outcomes of future activities [13,11,23]. However, while force-on-force engagements are accurately portrayed in simulations, the flow of C2 information and communication is usually not well simulated. A key technology gap with respect to decision support systems is the difficulty of conveying C2 information, e.g., orders to be executed by simulated units, to and from simulations [5].

A similar argument holds for training. Especially, staff training can be exercised in a way that the operating troops are simulated within a simulation system. In this case, the staff's commander has to order the simulated units and the simulation has to respond by sending reports. Again, a technology gap exists with respect to communication. Often, a large number of support personnel required as workstation controllers provides the interface between the training unit and the simulation, taking orders given by the commander and converting them into a form that the simulation can understand. This can result in discrepancies. The simulation system sometimes receives orders not intended by the commander. Besides, it is not uncommon that the number of support personnel exceeds that of actual participants in the training. This is not only very resource-intensive, but also lacks the feeling of authenticity that actual combat operations present to the commander and staff.

A standard representation for orders and reports plus tools that can be used to automate the simulation interface to C2 services would fill the gap. Plans and orders may be expressed in this representation format, best via a GUI, allowing commanders to input directly plans and orders to simulated forces.

In this paper, we will describe the standardized language BML ("Battle Management Language") which can be used in both real and simulated operations. We will discuss how BML looks like and how formulating BML expressions is supported by a GUI. We will also provide an overview about how BML simulation systems the BML-GUI has already be connected to. Last, but not least, we will give future directions we envision for the use of BML in simulations for increasing training effectiveness in the military.

2.0 BATTLE MANAGEMENT LANGUAGE

The broad definition of a battle management language ("BML") has defined [5] as:

***The unambiguous language used to command and control forces and equipment conducting military operations and to provide for situational awareness and a shared, common operational picture*.**

In other words, BML is a standardized language for military communication which means, it can be used to formulate orders, requests and reports.

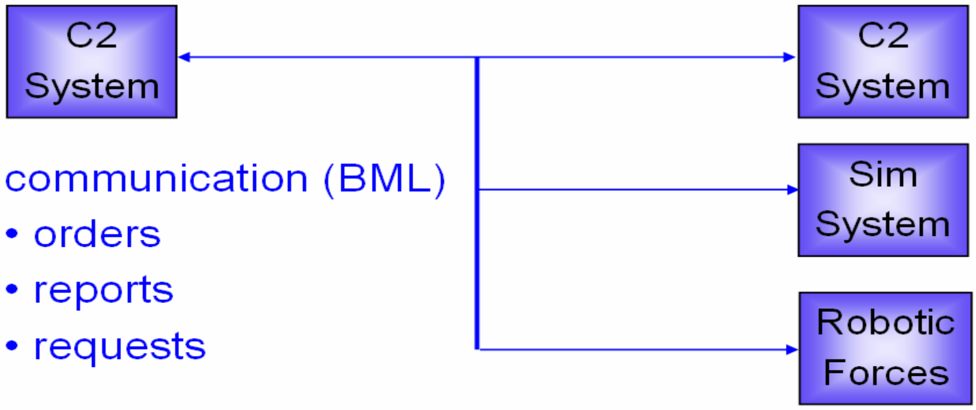

Figure 1: BML is a language for communication among C2 systems, simulation systems, and future robotic forces.

BML had its start in the SISO (simulation interoperability standardization organization). SISO installed a project group called "Coalition Battle Management Language" in order to define a BML and to suggest its standardization. First result of the work of these group are given in [4]. In addition, in 2006, the NATO RTO MSG-048, also called "Coalition Battle Management Language", had been installed to support and to power the SISO endeavors within the NATO RTO context. The NATO group is chaired by France and the US. In addition, there are members from Canada, Denmark, Germany, the Netherlands, Norway, Spain, and the United Kingdom. In order to ensure a close cooperation of the groups, some of the members work in both groups.

2.1 BML Standards

BML is designed as a formal language, i.e. the set of all expressions that can be generated by a formal grammar. A formal grammar generally consists of a lexicon which provides the words of a language and a set of production rules indicating how the elements (words, phrases) of the language may be combined to construct longer expressions (sentences) using these words. The formal structure and unambiguous usage of the language elements is important, as, among other things, it increases clarity of communication between multinational forces and makes possible automatic processing of the expressions produced by the language.

The SISO project group and the NATO RTO MSG-048 have decided that the attributes and values used in the JC3IEDM should comprise BML's lexicon. The same decision has been taken for other standardization approaches, e.g., for the development for AVLP (Autonomous Vehicle Command Language) [6]. JC3IEDM is the data model developed as NATO standard by the Multilateral Interoperability Programme [14]. The data model defines terms for all the elements that may need communication during military operations, both conflict and non-conflict (humanitarian) operations. These terms range from tasks for orders or requests, descriptions of units, personnel, weather, time, location, etc. Using JC3IEDM terms as lexical items in BML facilitates the mapping of information expressed in BML into the data model. In addition, the JC3IEDM provides a sort of "dictionary" of the terms. This decreases misunderstandings about their meaning and thus eliminates ambiguity.

Under the aegis of NATO RTO MSG-048 and in cooperation with the Center of Excellence for C4 at George Mason University, FGAN-FKIE proposed a formal grammar for BML, the so called C2LG (Command and Control Lexical Grammar). As required, JC3IEDM terms constitute the lexicon. In addition, rules have been defined that combine the lexical elements unambiguously into BML expressions.

As BML is for military communication, BML must convey information about military operations, the proposed production rules include a rule set for conveying orders [18], reports [19,20], and command intent [9]. In the following, we will take a look at these rules in order to illustrate the power of BML in general and the C2LG specifically.

2.2 Grammar Rules for Orders and Requests

BML has been created to cover a broad range of military communications. Orders and requests in BML are generated in a standard way to facilitate automatic processing. Orders and requests have identical syntax: in both cases there is a tasker, who produces the order or request, and a taskee, the recipient of the order, who may or may not execute the request/order. Orders and requests can be differentiated only by the relative ranking of the tasker and taskee: in an order, the tasker is in a position to mandate (e.g., of higher rank or authority), while in a request, the tasker is a peer, subordinate to or otherwise not in a position to mandate (e.g., from another nation in a multinational taskforce).

Of primary importance in the communications between a C2 system and a simulation system are orders. In the interaction between a C2 system and its users on the one side and a simulation system on the other side, communication of and reaction to orders is of the highest importance. During a staff exercise, the simulation system may fulfill the role of the forces providing the commander is able to issue orders in a way that the simulated forces can react to these orders as intended and anticipated. Further, as orders are the basis of communication between many of the actors, both real and virtual, in the simulation, it is worthwhile to examine more closely how BML is used to produce orders.

The doctrine and format of orders are defined by the NATO standard STANAG 2014 “Format for Orders and Designation of Timings, Locations and Boundaries” [15]. An Operational Order is divided into five paragraphs – Situation; Mission; Execution; Administration and Logistics; Command and Signal – plus necessary annexes. The third paragraph (“Execution”) is used to “summarize the overall course of action,” “assign specific tasks to each element of the task organization,” and “give details of coordination.” The information contained in this paragraph must be correctly interpreted by the simulation system in order to ensure that the simulated forces react properly to a given order. The standardized syntax of BML significantly reduces the chance of error, ambiguity or misinterpretation. In BML, Paragraph 3 is generated by

(1) OrderParagraph3 → CI OB* C_Sp* C_T*

In this production rule the task expression production consists of the command intent (CI), the basic order expression(s) outlining specific tasks (OB), spatial coordination expression(s) (C_Sp), and temporal coordination expression(s) (C_T). The asterisk indicates that arbitrarily many of the respective expressions can be concatenated.

The general production rule for specific tasks (OB) is given in (2a). Examples of task orders appear in (2b) and (2c).

(2a) OB → Verb Tasker Taskee (Affected|Action) Where Start-When (End-When) Why Label (Mod)*

(2b) OB → ***advance*** Tasker Taskee Route-Where Start-When (End-When) Why Label (Mod)*

(2c) OB → ***defend*** Tasker Taskee Affected At-Where Start-When (End-When) Why Label (Mod)*

Verb in (2a) indicates that a tasking verb will appear in this position in an OB rule. This verb is “advance” in (2b) and “defend” in (2c). Verbs are taken from the JC3IEDM (“action-task-category-code”) in which the tasks they describe are defined.

Each verb spans a frame, the general form of which is shown in (2a). Tasker will be completed by the identifier of the ordering unit; Taskee is the unit being ordered to execute the task. Start-When and End-When (optional as indicated by the parentheses) are to be replaced by temporal phrases indicating when the execution of the task should start and end, respectively.

Affected in (2a) is required as part of the frame if someone, e.g., the enemy, will be directly affected by a task. For example, it is there in the case of ***attack*** to indicate exactly whom the executing unit is ordered to attack (cf. 2c). It is not there in the case of advance (cf. 2b).

Action is similar to Affected. It only appears should the ordered task modify another action (task); an example of this is the task verb ***assist***. There are two variations of Where which are also determined by the verb: Where may be either At-Where (denoting a location) or Route-Where (movement to a location). Route-Where can be expanded to sequences of spatial expressions such as "**from** LocationA **to** LocationD **via** LocationB **and** LocationC."

An OB rule ends with Why, Label and the optional Mod. Why gives the reason for or supplies a context for the ordered task, for example "divert," "enable," "surprise," "protect," etc. Label is the unique identifier by which the order may be referred to in other expressions. Mod (for modifier) is an optional wildcard which can be used, where deemed necessary, to convey additional information necessary to describe the assigned task, e.g., to specify a particular formation for an ***advance***, or the speed at which the ***advance*** should be executed.

As mentioned previously, the only difference between a request and an order lies in the hierarchical differences between the tasker and the taskee; syntactically they are identical.

2.1.2 Grammar Rules for Reports

In addition to orders and requests, BML allows for the creation of several types of reports: task reports, event reports, status reports and position reports.

Task reports are solely concerned with military activities. Within this designation fall progress reports, in which information is conveyed concerning orders received (in-progress, completion, etc) by own or coalition forces, and reports concerning observed enemy activities, i.e., troop or equipment movements or activities. When the reporter can identify the observed unit, it is named in the report, otherwise the report refers to the acting unit by its type or the report contains the number and/or rank of individuals, or the type and number of equipment observed.

Non-military activities and/or events in the theatre of operation are covered in event reports. These events may provide background information on occurrences which, while not directly military, may be useful or necessary to effective operations. Particularly in the case of peacekeeping or humanitarian deployments, these events also may trigger operations or offer essential information for appropriate actions. An event report may describe a natural occurrence such as a flood, a volcanic eruption or an earthquake; non-military human actions such as traffic accidents, political demonstrations, or strikes; or various other occurrences such as epidemics, or fires.

Finally, status reports provide current snapshots on positions, personnel, materiel, facilities or general operational status for own, allied or hostile forces. This may be for own units, coalition units or information about enemy units. As with task reports, known units would be identified by name; otherwise with count and/or type information. Locations or facilities would be identified by name or by coordinates. Important operational information also comes via non-human sources such as sensors, robots, drones, etc. Each of these sources provides a different type of information which can be transformed and assigned to the same types of BML reports discussed above. Much of what comes from non-human sources is environmental and/or intelligence information that is important within the context of the exercise. These reports may then be used within the simulation system as described later in this paper.

A report consists of arbitrarily many basic report expressions (RB) as given in (3).

(3) Report $\rightarrow$ RB*

The general form of a basic reporting expression depends whether the report is about military operations (4a: basic task report), events (4b: basic event report) or status (4c: basic status report). The latter form is also used for basic position reports.

(4a) RB $\rightarrow$ ***task-report*** Verb Executer (Affected|Action) Where When (Why) Certainty Label (Mod)*

(4b) RB $\rightarrow$ ***event-report*** EVerb (Affected|Action) Where When Certainty Label (Mod)*

(4c) RB $\rightarrow$ ***status-report*** Hostility Regarding (Identification Status-Value) Where When Certainty Label (Mod)*

Rule forms (4a) and (4b) are comparable to the rule form for basic order expression (2a). However, there are some differences: 1) they do not have a Tasker, 2) Taskee is substituted by Executer in (4a) and left out in (4b), and 3) all basic report expressions contain Certainty. With respect to (4a), Tasker is dropped because normally a reporter does not know who has ordered the task being observed and reported on. Taskee here is generalized to Executer in order to allow use of expressions like “four hostile snipers” to describe the agent. Additionally, (4b) uses actions from another JC3IEDM table, namely “action-event-category-code” that are coded EVerb. EVerb covers events like “flood” or “peace conference”.

Rule form (4c) uses Regarding instead of Verb or EVerb. Regarding determines what kind of status the report is about. Status reports can be given about the operational status of a unit (by using the key word status-general in Regarding), about the status of a unit’s personnel (status-person) and about the status of a unit’s materiel (status-material). In addition, the position of a unit can be reported (position).

All report forms include Certainty to specify the certainty of the report, a label and an optional modifier. More about the grammar rules for reports can be found in [20].

3.0 BML GRAPHICAL USER INTERFACE

In order to formulate BML messages, the user is provided with a GUI. The GUI ensures that formulated messages conform to the grammar and therefore can be processed automatically. The user is guided through the GUI starting with choosing whether he wants to give an order, make a request, or make a report. Then the header is filled out: the sender is automatically set (it is the user); the addressee is chosen and the send time is automatically inserted. In figure 2, the user has chosen to send an order, and the header is filled out as can be seen in the upper right corner of the GUI.

After the header has been filled out, the user can proceed. In the case of an order, tasks can be chosen and assigned to units. In accordance with the grammar, after choosing a task additional fields appear which the user has to (in case of mandatory fields) or may (in case of optional fields) fill out. These fields can be seen under “Info” in the GUI screenshot (figure 2). Which kind of fields appear depends on the task chosen. In the example of figure 2, the task is “patrol”. So, the fields represent the route, the starting time, the end time, instrument (which kinds of vehicles should be used), formation, a modifier to enter the desired speed, why and label. The end time, instrument, formation and speed modifier are optional. Most of the fields are drop-down menus. However, all geo-information – in this example, the route – can be determined by clicking on the map, and the label – an ID that can be used to refer to this task in other communications (orders, reports) – is provided by the GUI itself.

In the GUI, the buttons below the Info fields are used to enter the completed task in the “Tasks” list, the lists of all tasks contained in the order. This tasks list is displayed on the right side of the GUI. The buttons

In the GUI, the buttons below the Info fields are used to enter the completed task in the "Tasks" list, the lists of all tasks contained in the order. This tasks list is displayed on the right side of the GUI. The buttons below this list are for editing the list and for sending a completed order to the simulation.

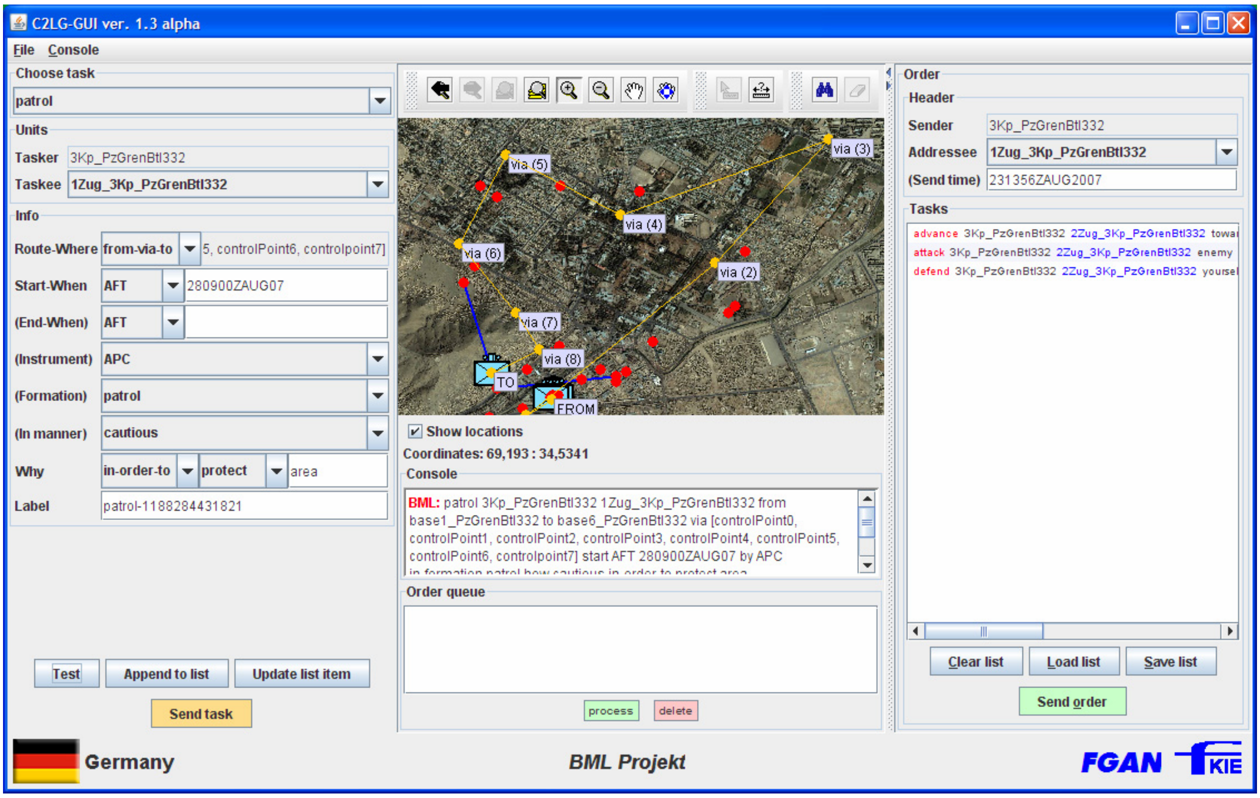

Figure 2: Example of an order displayed in the BML graphical user interface.

4.0 BML IN A SIMULATION CONTEXT

In order to illustrate our approach, we have started to link the BML GUI to simulation systems. As a first example, the GUI has been integrated into the so-called C2SimProxy (developed by the German companies IABG and ESG). The C2SimProxy had been used in the SINCE Project [12] to couple a C2 system to the PABST simulation system (IABG). So, by integrating the GUI into the C2SimProxy, we have been able to command simulated units in the PABST system. This solution has already been demonstrated at the AFCEA exhibition in Bad Godesberg, Germany, May, 8th-9th, 2007.

A similar approach was undertaken by the Center of Excellence in C4I at George Mason University. The GUI has been connected to SitaWare, a framework for CCIS, developed by the Danish company Systematic [21]. Again, simulated units have been commanded by using the GUI. This solution was demonstrated during the 2007 C4I Center Review in Fairfax, Virginia, on May 18th, 2007.

An even more ambitious approach has been started in cooperation with the Fraunhofer Institut IAIS. The IAIS is developing ITSimBw on behalf of the Federal Office of the Bundeswehr for Information Management and Information Technology. ITSimBw is an agent-based simulation system, and the agents within the system are communicating using a BML based on C2LG. In order to enable this approach, we have added rules to the C2LG that express effects. The solution will be presented at the 2007 Winter Simulation Conference in Washington, DC, in December 2007 [10].

Last but not least, under the aegis of NATO RTO MSG-048, a demonstration is under development that will combine the Dutch C2 system ISIS, the Norwegian C2 system NorTAC, the French simulation system SCIPIO, and the US simulation system JSAF to run a scenario in which simulated troops are commanded using ISIS and NorTAC. The C2LG-GUI is connected to the C2 systems and allows the formulation of orders in BML. The GUI is coupled to the simulation systems as well as to a shared JC3IEDM data base by JBML web services [16] provided by the Center of Excellence in C4I, George Mason University. This demonstration will be shown at I/ITSEC 07 in Orlando, Florida, November 2007.

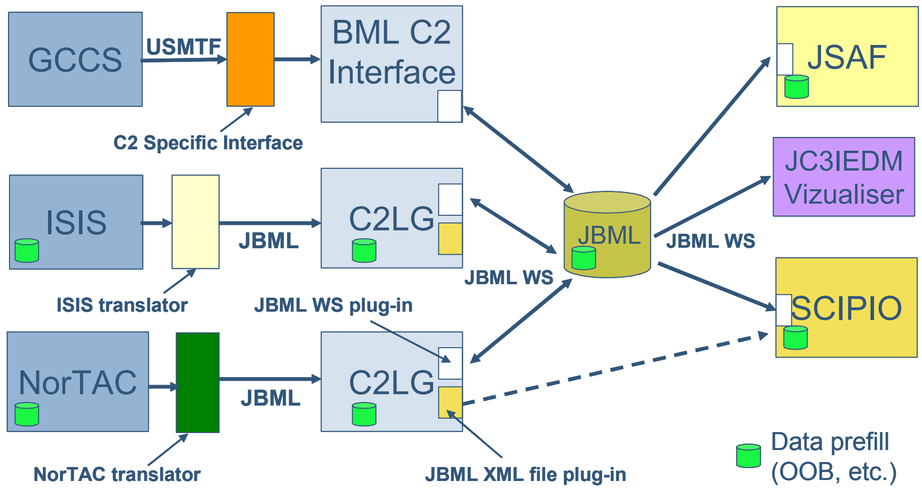

Figure 3: Architecture for the I/ITSEC demonstration

5.0 CONCLUSIONS

In near future, we would like to expand the solution presented in the following way. German troops are traditionally commanded by “Auftragstaktik” (mission command) introduced by Moltke the Elder to the Prussian army in the 19th century, cf. [7,17]. This command style means that the commander’s intent is conveyed to the troops. They know the goal their commander wants them to reach as well as the end state that should be achieved. They are not told “how” to do it. We would like to command simulated units by mission command. We not only want this because it is the German command style, but also because network-enabled operations demand this style [1,2,22].

To command simulated units by mission command in the frame we sketched and discussed above needs the following. First, command intent has to be expressed in BML. In [9], it is described how this can be done. The needed grammar rules have been developed. The next step is to integrate these rules – and thus the command intent – into the GUI. After this will be done, a commander will be able to express his intent in BML by using the GUI. This formalized intent, however, cannot directly be thrown into the simulation system like orders expressed in BML so far. The command intent instead describes the goal of a – still unknown – plan. So, there has to be a component that takes the intent (as given in BML) and calculates a plan, cf. [8]for planning systems. This plan than has to be inserted in the simulation system in a way that the system can run the plan and thus reacts as intended to mission command.

6.0 REFERENCES

[1] Alberts, D.S. and Hayes, R.E., *Power to the Edge*. Washington: CCRP, 2003.

[2] Alberts, D.S. and Hayes, R.E., *Understanding Command and Control*. Washington: CCRP, 2006.

[3] Biermann, J., "Understanding Military Information Processing – An Approach To Supporting the Production of Intelligence in Defence and Security." In: Shahbazian E. & Rogova, G. (Eds.), NATO Science Series: Computer & Systems Sciences: *Data Fusion Technologies on Harbour Protection*. Amsterdam, NL: IOS Press, 2006.

[4] Blais, C., Hieb, M.R. and Galvin, K., "Coalition Battle Management Language (C-BML) Study Group Report." Paper 05F-SIW-041, *Fall Simulation Interoperability Workshop*, Orlando, Florida, September 2005.

[5] Carey, S., Kleiner, M., Hieb, M.R. and Brown, R., "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation." Paper 01F-SIW-067, *Fall Simulation Interoperability Workshop*, Orlando, Florida, September 2001.

[6] Davies, D., Blais, C. and Brutzman, D., "Autonomous Vehicle Command Language for Simulated and Real Robotic Forces." Paper 06F-SIW-004, *Fall Simulation Interoperability Workshop*, Orlando, Florida, September 2006.

[7] Dupuy, T.N., *A Genius for War: The German Army and General Staff, 1807-1945*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[8] Ghallab M., Nau D., Traverso P., *Automated Planning, Theory & Practice*, Amsterdam, NL: Elsevier/ Morgan Kauffman, 2004.

[9] Hieb, M.R. and Schade, U., "Formalizing Command Intent Through Development of a Command and Control Grammar." *12th ICCRTS*. Newport, Rhode Island, June 2007.

[10] Hügelmeyer, P., Schade, U. and Zöller, T., "Application of BML to inter-agent communication in the ITSimBw simulation environment." In: Henderson, S.G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J.D. & Barton, R.R. (Eds.), *Proceedings of the 2007 Winter Simulation Conference*, Washington, DC, December 2007.

[11] Huis in 't Veld, M., van de Ven, J., Essens, P., le Grand, N., Huiskamp, W. and Voogd, J., "Improving the Commanders' Speed and Agility in Decision-making." *12th ICCRTS*, Newport, Rhode Island, June 2007.

[12] Mayk, I., Klose, D., Chan, A., Mai, M., and Negaran, H., "Technical and Operational Design, Implementation and Execution Results for SINCE Experimentation 1." *10th ICCRTS*, Tysons Corner, Virginia, June 2005.

[13] Matthews, K. and Davies, M., "Simulations for Joint Military Operations Planning." *Simulation Technology and Training (SimTecT) Conference 2003*, Adelaide, Australia, 2003.

[14] MIP website: < http://www.mip-site.org >, 2007.

[15] NATO Military Agency for Standardization, *STANAG 2014: Formats for Orders and Designation of Timings, Locations, and Boundaries*. Brussels, BE, Edition 9, 2000.

[16] Pullen, J.M., Hieb, M.R., Levine, S., Tolk, A., and Blais, C., "Joint Battle Management Language (JBML) – US Contribution to the Coalition Battle Management Language Product Development Group and the NATO MSG-048 Technical Activity." Paper 07E-SIW-029, *European Simulation Interoperability Workshop*, Genoa, Italy, June 2007.

[17] Regling, V., "Grundzüge der Landkriegsführung zur Zeit des Absolutismus und im 19. Jahrhundert." In: Militärgeschichtliches Forschungsamt (Ed.), *Deutsche Militärgeschichte 1648-1939*. Hersching, Germany: Pawlak, 1983.

[18] Schade, U. and Hieb, M.R., "Formalizing Battle Management Language: A Grammar for Specifying Orders," Paper 06S-SIW-068, *Spring Simulation Interoperability Workshop*, Huntsville, Alabama, April 2006.

[19] Schade, U. and 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.

[20] Schade, U. and Hieb, M.R., "Battle Management Language: A Grammar for Specifying Reports." Paper 07S-SIW-036, *Spring Simulation Interoperability Workshop*, Norfolk, Virginia, March 2007.

[21] SitaWare website: < http://www.systematic.dk/UK/Products/SitaWare/ >, 2007.

[22] Storr, J., "A command philosophy for the information age: The continuing relevance of mission command." In: Potts, D. (Ed.), *The Big Issue: Command and Combat in the Information Age*. Washington: CCRP, 2003.

[23] Tolk, A. and Kunde, D., "Decision Support Systems in Defence." In: Tonfoni, G. & Jain, L. (Eds.), *Innovations in Decision Support Systems* (pp. 175-205). Adelaide, AUS: AKI, 2003.