Battle Management Language:
A Grammar for Specifying Reports

Dr. Ulrich Schade
FGAN-FKIE
Neuenahrer Str. 20
Wachtberg, 53343
Germany
0049-228-9435-436
schade@fgan.de

Dr. Michael R. Hieb
Center of Excellence for C4I
George Mason University
4400 University Drive
Fairfax, VA 22030
USA
001-703-993-3990
mhieb@gmu.edu

Keywords: Battle Management Language (BML), Command and Control Information Exchange Data Model (C2IEDM), Command and Control Simulation Interface Language (CCSIL), Command and Control (C2), Grammar, Linguistics, Orders, Reports, Robotics

ABSTRACT: Battle Management Language (BML) is being developed as an open standard that unambiguously specifies Command and Control (C2) transactions. BML is both a methodology and a language specification, based on doctrine and consistent with Coalition C2 standards. We argue that the communication of future C2 transactions needs to be based on a formal grammar to support Net-Centric Operations. Such a grammar provides the lexicon for a BML as well as a set of production rules defining how the lexicon elements can be concatenated to BML expressions. The grammar applies both to orders (C2 transactions communicating tasking) and reports (C2 transactions communicating situations). This grammar is intended to contribute to the Coalition Battle Management Language Specification being developed within the Simulation Interoperability Standards Organization.

In this paper, we take an existing BML grammar for Orders and develop a related, but distinct grammar for reports. There are many differences between orders and reports. First, we differentiate between status reports, event reports, and task reports. Status reports include reports giving actual positions, the current status of a unit and its material. Event reports are about incidents that are not orchestrated, e.g., sandstorms. Task reports describe operations. A second difference is that with task reports in contrast to orders - the sender of the report does not always know the unit or person executing the task being reported. Therefore grammar rules must exist to allow the description of the executor by more general terms, e.g., by unit type or person type ("two snipers"). A third difference is that reports, especially those about the enemy, must allow the expression of uncertainty ("Enemy may be preparing an ambush").

This paper describes an initial set of the grammar’s rules for reports, taking into account the differences between orders and reports. It illustrates the use of BML in general and especially the use of the report rules by giving an example of BML communication in a scenario of asymmetric warfare. In this scenario, a patrol is ambushed by snipers. The communication is between lower echelon units. In particular, it involves the patrol, its battalion headquarters, a relief force, an Unmanned Aerial Vehicle (UAV) for reconnaissance, engineers for recovery, and medical support. This example illustrates how shared word semantics provided by the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) and the grammar give meaning and illocutionary force to the BML transactions.
1. Need for a Generalized Language for Command and Control

Coalition Interoperability has typically been addressed though message exchanges between Command and Control (C2) systems. There is a transition underway to a more structured communication based on exchanging data elements of common data models. One of these models is the Joint Consultation Command and Control Information Data Exchange Model (JC3IEDM) provided by the Multinational Interoperability Programme (MIP) [20]. While the data elements exchanged are semantically well-specified by the model, the communication of information is still in ad hoc messages. The syntax and grammar of these communications is lacking. It is needed, however, to construct the meaning of messages exchanged from the semantics of these messages’ data elements.

In this paper, we present a grammar for Command and Control. We have previously presented this grammar for Orders [25]. Here we present a grammar for Reports.

These grammars are designed to be computationally tractable so that statements can be both expressive and able to be parsed. Also, these grammars will be of great benefit to both Simulations and Robotics, which currently suffer from little to no commonality in how they process and understand C2 messages.

1.1 Communication C2 Information

In a network-centric environment, information empowers the participants (both human and automated decision systems). The exchange of information has to run smoothly such that fleeting opportunities can be noticed and exploited by the participants in time. In order to optimize the information exchange and to deal in a timely fashion with the huge amount of information exchanged, forces have to be equipped with smart C2 systems which automatically pre-process the incoming information. It is not only necessary that the right information is available at the right time, it is also necessary that the information is understood correctly at the right time (cf. [2], chapter 12). To achieve these goals is difficult, and it is even more difficult if the forces in question are coalition forces. Coalition forces often rely on nation-specific C2-systems developed with nation-specific doctrines in mind. Therefore, even if information is exchanged in time on the information level, the “understanding” of the exchanged information by the C2-systems normally will be different and thus a force that receives information often reacts to it a different way than the sender had intended.

The effective empowerment of Coalition C2 is predicated on the satisfaction of two key conditions. First, as Alberts and Hayes [2, chapter 7] described, the most direct way to ensure a desired degree of interoperability is to exchange information by communicating in a common language. With respect to forces, this idea has guided the development of a specific (military) variety of English and the standardization of military messages, e.g., the standard form of an Operations Order is determined by STANAG 2014. Military doctrines have leveraged these standards, such that for example professional NATO soldiers know by heart how an Operations Order has to be structured and how such an order is to be read and interpreted. However, many military messages formulated according to military doctrine are “free text” and as such hard to process automatically. Second, as Devlin presents [10], the effectiveness of any interaction is dependent on degree and quality of a shared context among the participants.

1.2 Need for Formal Grammars

A grammar, the essential part of a language, provides rules constituting how the lexical elements of the language in question can be connected and how the meaning of the catenation can be derived from the meaning of the parts if the parts are connected according to the grammar’s rules. For example, grammar determines that in the sentence “The US general ordered the German general to attack” it is the German unit that will attack whereas in the sentence “The US general promised the German general to attack” it is the US unit. In contrast, the JC3IEDM lists terms, e.g. mission terms, merely as words. E.g., “advance” is a value provided by the table “action-task-activity-code.” The semantics of the terms is given by a textual description. E.g., “advance” is “to move toward an objective in some form of tactical formation […].” This description shows the human reader what is meant by “advance”. Neither the meaning of terms like “advance”, nor appropriate rules for what can be combined with such terms, can be used by automated processes with the current design of the JC3IEDM. There is nothing but the human interpreter’s goodwill that ensures that a destination (the objective the executing unit has to move towards) should be associated with an action called “advance” in the JC3IEDM. A grammar as part of a language, however, enforces the existence of a destination and connects “advance” and its destination in the intended way. And this is exactly what is needed to convey understanding to other C2 systems, to simulation systems and to robotic forces such that the actions are carried into execution as intended.
1.3 A Grammar for Reports

In comparison to orders and requests, reports are different, especially with respect to certainty and definiteness. If a military commander orders an attack, the units that are ordered to execute it are definitely determined. To refer to these units and to give them the attack order, the commander can use their names. This is not true with respect to reports. The sender of a report may notice an attack. But, normally, he does not know by name which units execute it, especially if the attack is carried out by enemy forces. Usually, the sender only observes the behavior of troops; he sees that vehicles move and fire. From this, he may infer the type of the involved units, under the best of circumstances. In other cases, especially during operations other than war (e.g., if a convoy runs into sniper fire) the amount of objective information may even be smaller. Nevertheless, a report has to be formulated and sent, but the BML expressions that are sufficient for orders and requests are not sufficient for such reports. They, as given above, lack the power to express types of units, types of equipment, vagueness and other required concepts. We therefore have to add rules and non-terminal symbols to cope with reports.

1.4 Roadmap to Rest of Paper

The remainder of this paper is organized as follows: Section 2 gives a background on the BML and our previous grammar work for Orders. Section 3 reviews in detail the Lexical Function Grammar, the basis for our grammar work. Section 4 presents a grammar for Reports. Section 5 gives an example of using the grammar and Section 6 concludes with recommendations for future research. Throughout the text, we will also make recommendations for modification to the JC3IEDM to make it more consistent and improve it’s ability to correctly represent military reports.

2. Background – A Grammar for Orders

Our grammar work supports a current simulation initiative called Battle Management Language (BML). Within this section we describe BML, and also present the previous grammar work that has been performed for Orders.
contingent. Nevertheless, the introduction of “intelligent agent”, “command entities”, or other Command Decision Model (CDM) types of software requires unambiguous structures. Free text messages are not an option. A clear, unambiguous Battle Management Language is needed to control these agents.

C2 systems are also evolving. The future systems are incorporating automated decision aids, such as course of action development and analysis tools, and mission rehearsal simulations. While some emerging C2 systems, automatically fill certain fields when operators are entering Operations Orders, this is primarily situational awareness information (e.g. time, location, etc.) and the command information is still carried in free text form.

The Command and Control Simulation Interface Language (CCSIL) was a predecessor of BML. CCSIL was a highly structured language for communicating between and among command entities and small units of virtual platforms generated by computers for the Distributed Interactive Simulation (DIS) environment [7]. CCSIL was successful in providing an unambiguous structure, but was not consistent with the emerging C2 data standards and was not maintained as a standard.

The Simulation to Command and Control Information System Connectivity Experiments (SINCE) program is investigating interoperability issues by conducting multinational C2 experiments, supported by C2 and Simulation systems, designed to address the transformation of collaborative planning and interoperable execution in a coalition environment [19]. This is a US-German Army Bilateral Collaborative Project. SINCE uses a 5W-based Extensible Mark-up Language (XML) schema to represent the various C2 products that embody Information Exchange Requirements (IERs).

Within SISO, the Coalition BML (C-BML) Study Group was formed in September 2004 to investigate the concept of BML and, if warranted, develop a plan to develop a BML Standard. The Study Group has conducted a number of face-to-face and teleconference meetings through the year since the Fall 2004 SIW, involving a membership of over 100 persons from 11 different countries. For more details about the work of the study group see [4, 30]. As the Study Group concluded, it recommended that a Product Development Group (PDG) be formed to standardize the emerging notion of BML. The C-BML Study Group has worked closely with the Military Scenario Definition Language (MSDL) Study Group to coordinate both PDG proposals to ensure a consistent set of standards for initialization, tasking and reporting.

In parallel to the C-BML Study Group activities, the NATO Modeling and Simulation Group (NMSG) established a 3-year Technical Activity Program (MSG-048) on C-BML [17, 31]. The team, led by France, has the goal of evaluating the C-BML standard with a NATO reference implementation.

### 2.2 Order Grammar

In our work on BML grammar, we started with the development of a “tasking grammar” to formalize orders. This had been a good starting point for two reasons. First, the development of production rules for orders is easier than the development of production rules for reports as can easily be noticed by comparing the rules mentioned in this subsection to the rules provided by subsection 4.2. Second, with respect to the communication between C2 systems and simulation systems, the processing of orders is of higher priority than the processing of reports.

The format of orders is defined by the NATO standard STANAG 2014 “Format for Orders and Designation of Timings, Locations and Boundaries.” An Operational Order is divided into five sections 1) Situation, 2) Mission, 3) Execution, 4) Administration and Logistics, 5) Command and Signal, and the respective annexes. For conveying the essence of an order, Section 3 is the most applicable given the behaviors available. It is used to “summarize the overall course of action,” “assign specific tasks to each element of the task organization,” and “give details of coordination.” In the following, we will repeat the core grammar rules we proposed for STANAG 2014, section 3 in [25]. These rules form the base, on which the grammar rules for reports will be founded on.

In order to represent an order’s execution section, our grammar for orders starts with rule (1).

\[
S \rightarrow \text{OB}^* \text{C}_sp^* \text{C}_t^*
\]

This rule means that a BML message that is an order consists of three parts: 1) basic expressions to assign tasks to units (indicated by the non-terminal \(\text{OB}\)); 2) spatial coordinations (indicated by the non-terminal \(\text{C}_sp\)); and 3) temporal coordinations (indicated by the non-terminal \(\text{C}_t\)). The asterisk indicates that arbitrarily many of the respective expressions can be concatenated together. Basic expressions are composed of a tasking verb – a terminal symbol – and its frame. The tasking verbs are taken from JC3IEDM’s table “action-task-activity-code.” Thus, the rules to expand \(\text{OB}\) have the general form as given in (2). (3) and (4) provide two examples.
Therefore, we will discuss the principles of grammars that are modeled on Lexical Functional Grammar (LFG). The grammars for specifying reports and the tasking grammar sketched above. Both, the tasking grammar and the grammar for specifying reports are modeled on LFG. In this section, we will discuss the principles of grammars in general and of LFG in particular before presenting the details of the grammar for specifying reports.

2.3 The next step

In order to expand the order grammar to a grammar that covers military communications, reports also have to be dealt with. Thus, in the following, we will present a grammar for specifying reports that is compatible with the tasking grammar sketched above. Both, the tasking grammar as well as the grammar for specifying reports are modeled on Lexical Functional Grammar (LFG). Therefore, we will discuss the principles of grammars in general and of LFG in particular before presenting the details of the grammar for specifying reports.

3. Lexical Functional Grammar

There are two main purposes a formal grammar has to fulfill with respect to BML. First, language processing – that is, the processing of BML expressions – has to be done by systems. For example, a simulation system that receives a BML order for some of the simulated forces has to process this order such that the simulated forces react to the order appropriately. Thus, BML expressions have to be automatically processable. In [25] we argued that for this reason the optimal BML grammar is a context free grammar. A context free grammar is a grammar of type 2 in the Chomsky hierarchy of grammars [9]. It rules have the form A → γ where A is a symbol and γ is a sequence of symbols and words. The production rules of the BML tasking grammar (cf. subsection 2.2) are context free grammar rules, and the production rules of the reporting grammar that we propose by this paper (in subsection 4.2) are likewise context free. Naturally, the rules for the reporting grammar are modeled after the rules for the tasking grammar such that the grammars complement each other.

The second major purpose of a grammar is to support semantic processing. In contrast to what is suggested by the notion of the BML “triangle with five sides” [32, 33], grammar contributes to semantic processing. In order to illustrate this point let us take a look at the following sentence: “Then Thomas Johnson […] nominated Washington for the position.” [22, p. 4]. In this sentence, the term “Thomas Johnson” – and not the term “Washington” – refers to the agent of the nomination. This simple fact is determined by nothing else but the syntax of the English language: namely by the rule that in English the sequence “subject - predicate - object” is the correct one. Because the term “Thomas Johnson” – but not the term “Washington” – precedes the verb “nominated” it is the subject and thus refers to the agent of the nomination.

In order to design the BML grammar so that it has the ability to support semantic processing in a unambiguous way, we need a grammar that specifies an unambiguous mapping from the constituents of a BML expression to semantic roles (cf. [29], for a contemporary theory of semantic roles). In the example above, “Thomas Johnson” is mapped on the agent role. With respect to BML expressions we want to map their constituents to roles which roughly correspond to the 5 W. In this section, we will argue that the 5 Ws are a good starting point for defining roles for semantic processing. However, the 5 Ws have to be split up further in order to get an ample set of roles. In addition, we will argue for modeling BML grammars on the Lexical Functional Grammar (LFG) [5, 18] because LFG provides not only the linguistic...
principles necessary for a successful mapping from constituents to roles but shows a methodology to implement this mapping.

3.1 From the 5Ws to frames to LFG

Experience says that the information contained in simple orders or simple reports can roughly be broken down into the 5 Ws, namely WHAT, WHO, WHERE, WHEN, and WHY. However, it is quite obvious that often these Ws show internal structure. With respect to orders, there are two WHOs, namely the one who gives the order – the TASKER – and the one who has to execute it – the TASKEE. The same is true for the WHEN if there is information about when an ordered task begins and when it ends. In this case, there are two WHEN terms – a Start-WHEN and an End-WHEN. If a move is ordered, there might be information about the source of the movement, about the destination of the movement and about the path of the movement, which would result in at least three WHERE terms.

In our tasking grammar [25], we considered all these cases and attribute them to constituency and subcategorization. Constituency and subcategorization are syntactic concepts. They formalize the commonly understood idea of the 5 Ws in the field of linguistics and they build the bridge from language expressions to semantic roles. Constituents are sequences of words that belong together. In “phrase structure grammars”, constituents are phrases with exactly one node in the phrase structure that is a root node of all the words the phrase consists of, and no other word node is consumed under this root node. Constituents answer Wh-questions. This forms the connection between the 5 Ws and constituency. For example, in the expression “Three hostile battle tanks move from crossing X towards north on highway 25”, there are the constituents “three hostile battle tanks” – answering a who-question – and the constituents “from crossing X”, “towards north”, and “on highway 25” that all answer where-questions. However, to be more precise, we can say that “from crossing X” answers the question about the source or origin of the movement, that “towards north” answers the question about its direction, and “on highway 25” answers the question about its path. Being precise, we can map the Where-constituents of our example directly to the semantic roles “source/origin”, “direction/goal”, and “path”.

The second syntactic concept that helps us to build up the semantics of an expression is subcategorization. Verbs, in our case the tasking verbs, are not only of category “verb” but also evoke a “frame” (frame semantics has been introduced by Fillmore [12, 13]). A verb’s frame tells us which kinds of constituents must and which kinds may go together with the verb. For example, according to FrameNet [14] the frame “motion” that is evoked by “advance” is defined as follows: “Some entity (Theme) starts out in one place (Source) and ends up in some other place (Goal), having covered some space between the two (Path). Alternatively, the Area or Direction in which the Theme moves or the Distance of the movement may be mentioned.” In short, verbs are subcategorized. Subcategories are defined by the frames. A verb belongs to a certain subcategory if it evokes the frame which defines the subcategory in question. The frame determines which kinds of constituencies (and therefore which semantic roles) are governed by the verb, and it determines which kinds of constituencies must and which may occur in an expression that is ruled by the verb. In sum, if we split an expression in constituents, starting from the verb and its frame, we can determine for each constituent the semantic role of the object denoted by the constituent.

The interplay between syntax and semantics as sketched above is ideally supported by LFG. Therefore, we modeled BML grammar on LFG. The LFG analysis of a sentence consists of at least two steps. In the first step, the so-called c-structure is derived – “c” for “categorical”. The c-structure is a classical phrase structure [27] in which phrases – the representatives of the constituents – are organized within a tree. This structure is transformed into the f-structure – “f” for “functional”. The f-structure is represented as a feature-value matrix such that unification can operate on it. As a third step, the f-structure can be transformed even further into a semantic representation, e.g., into the a-structure – “a” for “argument” [5 p. 19]. However, the main representation format of LFG is the f-structure. LFG is lexical-driven as indicated by the L. This also means that the lexical entry of a sentence’s verb determines the general look of the sentence’s f-structure.

With respect to BML, we have defined basic rules for orders that are lexical entries for the tasks and combined them with the task’s frames. These rules are task-driven. Thus, we can parse BML expressions in a LFG-manner to come up with attribute-value matrices. These matrices correspond to LFG’s f-structures. They are constructed on the tasks’ frames like an f-structure of an English sentence is constructed on its verb. Like in standard LFG, the principles of completeness and coherence apply. The principle of completeness grants that a well-formed BML expression includes all those constituents that are demanded by the task’s frame, and the principle of coherence grants that all constituents that appear in a BML expression will have a slot in the task’s frame where they fit in.
3.2 Towards Semantics

The result of a parsed BML expression is a feature-value matrix. The look of the matrix is determined by the frame of the expression’s task. In this subsection, we will list the benefits of such a representation, especially with respect to semantics. First, the matrix representation can be written in XML, if the attributes are used as tags and the values as content. Thus, an XML schema can be defined that can be used to check whether a certain matrix is valid.

Second, the principles of completeness and coherence apply during the parsing of the BML expression. Thus, BML expressions are checked for these aspects. This is not the case if data is inserted into a J3IEDM database. For example, according to the J3IEDM it is possible to associate a hostile unit as action-objective to a “rest” task, although J3IEDM’s definition of the task “rest” is given as “to impose a specific period of inactivity on an organization that is out of contact with the enemy.” For BML, the principle of coherence permits this association. According to the J3IEDM, it is also possible to not associate a destination to an advance task although the definition of “advance” – “to move forward towards an objective […]” – demands it. For BML, this kind of problem is prevented by the principle of completeness. In the J3IEDM, the definitions given for attributes and values more often than not do not have an impact on the implementation of the model itself, and it is this aspect of the J3IEDM we had in mind when we said that the tasks listed in J3IEDM’s table “action-task-activity-code” “are merely words with a vague textual description” [25].

Third, the representation of BML expressions in the form of feature-value matrices allows for further processing by unification [28], a standard method in the field of computational linguistics. For example, BML reports represented in this way can be exploited in information fusion processes [23, 24]. They also can be enhanced by information extraction or processes based on an ontology about military operations as suggested by Tolk et al. [32]. If, for example, the BML expression is a “march” task for a computer generated force, the semantic enhancement can be influenced by knowledge about the simulation system. Often, an order is not specific enough for the simulation system, and the enhancement can be tuned to provide the required degree of specificity. For example, a formation parameter can be added depending on the type of force if the simulation system a specific formation in the case of a “march” task. A feature-value matrix determining which constituent of the expression fills which semantic role is the proper starting point for semantic processes, be it information fusion, information extraction, or semantic enhancement.

4. A BML Grammar for Reports

In this section, we will present a grammar for describing reports in the context of an. The grammar is designed to specify reports so that their description can be used in automated systems.

4.1 Scope

The grammar presented in this section is restricted with respect to its scope. The idea behind this is the following. BML has to be developed step by step. Then, in each step, lessons learned during the preceding steps can be applied.

The format of reports is defined by the NATO standard ADL+3 NATO Message Text Formatting System [1] and the US FM 101-5-2, U.S. Army Report and Message Formats [11]. These publications provide the doctrinal format for reports and also groups reports into various classes.

Reports are classified according to what triggers them and according to what they are about. With respect to triggers, we differentiate between: 1) scheduled reports; 2) spontaneous reports; and 3) reports on order. Scheduled reports have a defined schedule which determines when reports are submitted. The schedule can be time or event driven. For example, a report to be submitted after crossing a phase line is event driven. It is related to an action as well as to a control features – the phase line of the example. In German doctrines, control features that are referred to in scheduled reports are even renamed as “Meldelinie” (report line) or “Meldepunkt” (reporting point). Time driven reporting, (e.g., a troop is ordered to report its position every 15 minutes) is of particular relevance to blue force tracking as well as robots and simulations because in these cases position reports are required in short intervals.

Spontaneous reports are given if something – e.g., an action or an event – is observed which is newsworthy. Spontaneous reports push information. In contrast, scheduled reports combine information push with information pull. Reports on order, which are reports demanded by a superior, constitute information pull. In order to implement information pull reports, there has to be a BML “verb” that tells one unit to make a report to another. Since the value “report” is missing in J3IEDM’s table “action-task-activity-code” we have to add a special value “report” to that table. Therefore, we put “report” into the BML lexicon along with the rules.
(5) \( OB \rightarrow \text{report Tasker Taskee Hostility Regarding} \)  
\((\text{At-Where}) \text{Start-When} (\text{End-When}) \text{Why Label} \)

(6) \( \text{Regarding} \rightarrow (\text{position} | \text{status-general} | \text{status-material} | \text{status-person}) \)

(5) is the basic rule for ordering a report. It follows the general rule form as given by (2): Tasker orders Taskee to give a report. Regarding, together with its companion Hostility, denotes what Tasker wants Taskee to report about. It is expanded according to rule (6). We will come back to Regarding and rule (6) in subsection 4.2.

According to [1] and [11], reports differ with respect to what a report is about. We therefore also differentiate between reports about position or status, reports about events, and reports about military tasks. In the current state, only reports about position or status can be ordered. However, the grammar already allows giving reports about events or military actions (as spontaneous reports) nevertheless.

4.2 Syntax

In this section, we will discuss the grammar rules for reports. In principle, a report consists of arbitrarily many basic report expressions (RB) as given in (7).

(7) \( S \rightarrow \text{RB}^* \)

The look of a basic report expression depends on what the report expression is about. (8a) shows the general rule form of a basic report expression about a military task, (8b) the general rule form of a basic report expression about an event, and (8c) the general rule form of a basic report expression about position or status.

(8a) \( \text{RB} \rightarrow \text{Task-Report Verb Executer (Affected|Action) Where When (Why) Certainty Label (Mod)*} \)
(8b) \( \text{RB} \rightarrow \text{Event-Report EVerb (Affected|Action) Where When Certainty Label (Mod)*} \)
(8c) \( \text{RB} \rightarrow \text{Status-Report Hostility Regarding (Identification Status-Value) Where When Certainty Label (Mod)*} \)

The rule form (8a) is similar to the basic rule form for order expressions. However, there are three differences. First, there is no Tasker in the rule form. Second, instead of Taskee, there is the term Executer. Third, the non-terminal Certainty is inserted.

Certainty expands to a modality operator (a terminal symbol) to denote the certainty, the sender attributes to his report. These symbols are taken from JC3IEDM’s table “reporting-data-credibility-code.” The corresponding rules are given in (9a)-(9d).

(9a) \( \text{Certainty} \rightarrow \text{fact} \)
(9b) \( \text{Certainty} \rightarrow \text{plausible} \)
(9c) \( \text{Certainty} \rightarrow \text{uncertain} \)
(9d) \( \text{Certainty} \rightarrow \text{indeterminate} \)

The JC3IEDM provides many more attributes and corresponding values for specifying “reporting-data.” However, “reporting-data” in the JC3IEDM does not mean “data from a report.” To be more precise, “reporting-data” is associated with each table which contains information that may change. Thus, not all of JC3IEDM’s attributes for “reporting-data” can be used for our purpose. In [26], we discussed what JC3IEDM’s other attributes represent and which of them may be useful for future extensions of the BML grammar.

The other two differences between basic report form (8a) and the basic order form also mirror differences between reports and orders: There is no Tasker in the report form for reporting military tasks because the Tasker is not normally known to the reporter, in particular if the report is about a task that is under execution by an enemy force. A similar consideration is behind the substitution of Taskee, by the term Executer. The reporter normally does not know the name of the unit executing a task. Therefore, Executer may be expanded by Taskee, Agent, or Theme (10a) – (10c), and in the latter cases a Label may be added for referring to the executing unit in other orders, requests, or reports.

(10a) \( \text{Executer} \rightarrow \text{Taskee} \)
(10b) \( \text{Executer} \rightarrow \text{Agent (Label)} \)
(10c) \( \text{Executer} \rightarrow \text{Theme (Label)} \)

Executer can be expanded to Taskee if the name of the unit is known to the reporter, and Taskee itself then can be expanded to the name of the executing unit. Executer is expanded to Agent if not the name of the executing units but its type is know to the reporter. Agent then can be expanded by rule (11a) to a constituent that includes the size of the unit, its hostility, and its type, e.g., battalion (of) hostile infantry. Last, but not least, Executer can be expanded to Theme if only the type of main equipment used by the executing unit is known to the reporter. Theme expands by rule (11b) to a constituent that includes the number of this equipment, the hostility, and the type of the equipment – e.g., four hostile battle tank(s).

(11a) \( \text{Agent} \rightarrow \text{Size Hostility Unit_type} \)
(11b) \( \text{Theme} \rightarrow \text{Count Hostility Equipment_type} \)

Besides reporting about a military task, a reporter can also report about an event. In the formal grammar this is done by rule (8b). In the case of events, the verbs are not taken
from JC3IEDM’s table “action-task-activity-code”, but from the table “action-event-category-code.” This is encoded by EVerb in (8b) in contrast to Verb in (8a). An example is “flood” (cf. 12a) by which the natural disaster of a flood can be reported. Another example is “peace conference” (cf. 12b).

(12a) \( \text{RB} \rightarrow \text{flood} (\text{Affected|Action}) \text{ Where When Certainty Label (Mod)*} \)

(12b) \( \text{RB} \rightarrow \text{peace conference When Certainty Label (Mod)*} \)

As can be seen, Executer and Why are omitted in event reports.

The third form of reports are status reports. This includes reports about position. Currently, status reports can be made about the position, the general operational status of a unit, the status of its personnel, and the status of its equipment. The kind of status report is indicated by the term Regarding that accordingly expands as has already been given in (6).

(6) Regarding \( \rightarrow (\text{position | status-general | status-material | status-person}) \)

Regarding denotes the topic of the report. It is accompanied by Hostility. Hostility is either “own” or a value from JC3IEDM’s table “object-item-status-hostility-code” (e.g., “friend”, “hostile”, “suspect” etc.). The term “own” is deictic. It refers the reporting unit itself. This wide range of values provided by the JC3IEDM is in contrast to the “unit situation report [UNITSITREP]” [11] that only provides the values “own”, “enemy”, and “non hostile” to code the affiliation of units. The term “own” as defined by [11] corresponds to JC3IEDM’s term “friend”. We opt for JC3IEDM’s categorization of affiliation term because it is more rich in detail. But we also opt for the additional differentiation between “own” and “friend” because a unit bears a much higher responsibility for accurateness of those reports that it gives about itself than about those reports it gives about other friendly troops.

Regarding says whether the report will be about a position (e.g., a report with “own position” is about the position of the reporting unit) or about status. As has already been mentioned, currently, three different kinds of status reports can be expressed: reports about the general operational status of a unit, reports about the status of a unit’s material, and reports about the status of a unit’s personal. Reports about the general operational status of a unit have the name or the type of this unit as Identification – cf. rule (8c) – and as Status-Value a value from JC3IEDM’s table “organization-status-operational-status-code”.

Reports about the status of a unit’s personnel use a Count together with a value from JC3IEDM’s table “person-type-rank-code” as identification. The respective Status-Value denotes the physical status of persons. These values are taken from JC3IEDM’s tables “person-status-duty-status-code”, “person-status-physical-status-code”, and “person-status-physical-status-qualifier-code”. The table “person-status-duty-status-code” has to be included here because its values “deceased” and “assumed killed in action” (in contrast to “at duty”, “arrested”, and so on) are the only values that denote “dead”. [Both values are questionable, however, because “deceased” violates the Gricean Maxim of Manner [15, 16] as it is an obscure expression for "dead" in the context of warfare, and because “assumed killed in action” adds a term of uncertainty to “killed in action" which – if necessary – has to be undertaken by the attached reporting data.]

Reports about a unit’s material is expressed in the same way as reports about a unit’s personnel. An example of such reports is given in section 5.2, (13b).

4.3 Semantics

As has already been mentioned, the semantics of the terminals are names denoting units and other objects of the real world or are taken from JC3IEDM tables. In the latter case, the JC3IEDM provides semantic definitions for the terms. The meaning of the report expressions is combined from the semantics of the terminals according to the grammar rules. It is represented as a corresponding f-structure. If the reports origin from simulated units, the meaning of the reports represent the respective states of the simulation system in a very concrete sense.

5. Using BML

In order to illustrate what BML communication looks like, we will present a scenario and give its C2 communication in military natural language as well as in BML. In the explanations of the BML expressions we will concentrate on the reports.

5.1 Patrol Scenario

The scenario in which the C2 communication takes place is a patrol scenario. Patrolling is a standard task for military units, especially for coalition forces in current coalition operations, and it is a task which becomes more and more important as war changes its face back to asymmetric warfare [21]. In our scenario, the patrol is run by a platoon in an urban environment. The patrol is
commanded and controlled by a mechanized infantry battalion that is part of a Multi-National task force. Other forces of the task group are placed in support to the battalion. The patrol consists of four armored wheeled vehicles (for the German forces involved these are called “Dingos”). In the scenario, it is ambushed by snipers who immobilize two of the vehicles and wound patrol members during the fire fight that follows. The battalion leader who commands and controls the patrol sends an armored company as relief and a UAV (Unmanned Aerial Vehicle) as reconnaissance to the back of the building that the sniper fire is coming from. The armored company joins the fire fight and tips the balance. The snipers cease fire. A few moments later the UAV ground base reports that some individuals with weapons board a car and move away fast. The battalion commander concludes that the some individuals with weapons board a car and move away fast. The battalion commander concludes that the snipers, and the label label-en-1 represents the report about the ambush. The second and the third line are status reports. The second line is about the status of the patrol’s material and the third is about the status of its members. The latter one is not a direct transformation of the respective natural language report in (13a) This is because it is not possible to use negation in BML in its current state. Therefore, “nobody is wounded” is expressed as “all the 16 soldiers are still fit.” The value “fit” is from JC3IEDM’S table “person-status-physical-status-qualifier-code”, and “wounded” is provided by table “person-status-physical-status-qualifier-code”.

In (13c) there are three reports and one order. The first line is a report about a military task executed by the enemy (hostile). This can be read as: Task-Report <about an> ambush <executed by> team hostile sniper label-en-1 <affecting> PTL at ControlPoint3 at now <with certainty> fact <with a unique label> label-tr-1. The label label-en-1 is introduced to the discourse to allow referring to the snipers, and the label label-tr-1 represents the report about the ambush. The second and the third line are status reports. The second line is about the status of the patrol’s material and the third is about the status of its members. The latter one is not a direct transformation of the respective natural language report in (13a) This is because it is not possible to use negation in BML in its current state. Therefore, “nobody is wounded” is expressed as “all the 16 soldiers are still fit.” The value “fit” is from JC3IEDM’S table “person-status-physical-status-qualifier-code”, and “wounded” is provided by table “person-status-physical-status-qualifier-code”.

In principle, the information exchanged by the natural language reports and the respective BML reports is the same. The main difference between (13a) and (13c) is that every BML report expression includes the location (Control Point 3) and a time stamp (at now) in order to avoid ambiguity. Simulation systems as well as robotic forces would need a specified Where and a specified When. The deictic time stamp “at now” will be instantiitated by the actual date time during the processing. The actual date time is provided by the system.

The last line of (13c) is an order. The company C2 is ordered to relieve the patrol. The Why (“in order to destroy enemy”) indicates the offensive intention of this order. Please notice that the “enemy” in the Why is given by its label. It is put within brackets. The brackets mean that the nature of the Why has not yet been decided. Syntactically, the Why might be a single task verb ([in order to] destroy) or a kind of sub clause (in order to destroy the enemy). With respect to the content, the Why might be a reference to a higher order or to the mission, or it might denote the desired effect the ordered task should create.

The following examples show the reports of the relief company when it arrives at the area of action and opens

5.2 Example of BML communication

In this subsection, we will look at the communication that takes place in the patrol scenario given above. We will translate the natural language used into BML expressions. These expressions, then, can be discussed in order to illustrate strengths and shortcomings of BML.

In the first example (13), the battalion leader (BtlL) informs the company leader (C2L) of his second company (C2) about the state of the patrol (PTL) and orders the second company to relieve the patrol.

(13a) BtlL to C2L:
“Patrol under fire at Control Point 3; Two Dingos immobilized; Nobody’s wounded; Send relief patrol at once”

(13b) Report Rules
RB → ambush Executor Affected At-Where When Certainty Label
Executor → Agent Label
Agent → Count Hostility Unit-type

RB → Status-Report Hostility Regarding Identification Status-Value Where When Certainty Label

OB → relieve Tasker Taskee Affected At-Where Start-When Why Label

(13c) BtlL to C2L [BML]:
Task-Report ambush team hostile sniper label-en-1 PTL at ControlPoint3 at now fact label-tr-1;
Status-Report friend status-material 2 Dingo immobilized at ControlPoint3 at now fact label-sr-1;
Status-Report friend status-personnel 16 Enlisted fit at ControlPoint3 at now fact label-sr-2;
relieve Btl C2 PTL at ControlPoint3 begin at now in-order-to destroy [label-en-1] label-o-1;

In (13c) there are three reports and one order. The first line is a report about a military task executed by the enemy (hostile). This can be read as: Task-Report <about an> ambush <executed by> team hostile sniper label-en-1 <affecting> PTL at ControlPoint3 at now <with certainty> fact <with a unique label> label-tr-1. The label label-en-1 is introduced to the discourse to allow referring to the snipers, and the label label-tr-1 represents the report about the ambush. The second and the third line are status reports. The second line is about the status of the patrol’s material and the third is about the status of its members. The latter one is not a direct transformation of the respective natural language report in (13a) This is because it is not possible to use negation in BML in its current state. Therefore, “nobody is wounded” is expressed as “all the 16 soldiers are still fit.” The value “fit” is from JC3IEDM’S table “person-status-physical-status-qualifier-code”, and “wounded” is provided by table “person-status-physical-status-qualifier-code”.

In principle, the information exchanged by the natural language reports and the respective BML reports is the same. The main difference between (13a) and (13c) is that every BML report expression includes the location (Control Point 3) and a time stamp (at now) in order to avoid ambiguity. Simulation systems as well as robotic forces would need a specified Where and a specified When. The deictic time stamp “at now” will be instantiitated by the actual date time during the processing. The actual date time is provided by the system.
fire against the enemy snipers (14) and when the enemy snipers cease fire (15). Report (14) starts with a position report and follows up with the report about a military task. Report (15) is again about a task. In all the following examples, the labels for the reports have been deleted to concentrate on the content.

(14a) C2L to BtlL:
“Arrived at Control Point 3; start attack on snipers.”

(14b) C2L to BtlL [BML]:
Status-Report own position C2L at ControlPoint3 at now fact;
Task-Report attack C2L label-en-1 at ControlPoint3 at now fact;

(15a) C2L to BtlL:
“Enemy ceased fire.”

(15b) C2L to BtlL [BML]:
Task-Report disengage label-en-1 C2L at ControlPoint3 at now fact;

As already mentioned, “at now” is a deictic term. It is instantiated by the system. Therefore, all the When Terms will have different values when inserted into the data bank. The next example (16) gives the task report of the UAV ground station that some suspect individuals with guns leave the building the sniper fire had come from, immediately after the fire ceases. These individuals board a car and move away quickly.

(16a) UAV ground station to BtlL:
“Four armed subjects leaving target building:
They boarded a car and are moving towards Check Point 5 fast.”

(16b) UAV ground station to BtlL [BML]:
Task-Report move four suspect paramilitary label-susp-1 from ControlPoint3 toward CheckPoint5 at now fact by car fast;

In (16b), the phrase “four suspect paramilitary label-susp-1” denotes the suspect individuals. The phrase consists of count, hostility and person-type (cf. rule 11). The value paramilitary is taken from JC3IEDM’s table “person-type-category-code.” It refers to the class of persons that are members of an irregular armed force. The phrase is labelled by “label-susp-1”. The label is provided by the system and inserted into the discourse representation list like “label-en-1” before. From (16) on, “label-susp-1” can be used as unique name for the suspect subjects. However, at the battalion an information fusion process identifies “label-susp-1” with “label-en-1”. Thus, in (17) “label-en-1” is used again.

(17a) BtlL to AirCav (Air Strike Unit):
“Enemy withdraws from Control Point 3 towards Check Point 5 by car fast;
Pursue enemy in order to destroy them.”

(17b) BtlL to Cav [BML]:
Task-Report withdraw label-en-1 from ControlPoint3 toward CheckPoint5 at now fact by car fast;
pursue Btl Cav label-en-1 towards CheckPoint5 begin at now in-order-to destroy [label-en-1];

Example (17) again shows a task report and an order. In summary, it is easy to transform natural language reports into BML reports respecting the rules given in section 4. The major problem is that some of JC3IEDM’s tables do not represent the military requirements good enough to provide all the vocabulary that is needed.

BML expressions can be generated with the help of a Graphical User Interface (GUI) developed at FGAN-FKIE. The GUI is based on the grammar as specified in this paper and is being evaluated in a number of systems.

6. Conclusions

In this paper we have presented a report grammar for BML. By utilizing the previous grammar for Orders we have created a new grammar that complements the grammar for Orders. The grammar for reports deals with additional concepts such as probability that are representative for reports.

Of particular interest is the applicability of the grammar to the C-BML prototyping and evaluation. Any BML grammar must be developed and refined through a variety of uses and applications. The grammar must be used to create specific BML content for different domains (e.g. Army, Navy, Air Force). The content for supporting specific missions will consist of elaborating the semantics and defining production rules that are sufficient and necessary for the missions of interest. And once the BML content is established, it then has the potential to increase the standardization of simulation behaviors.

The next step in the development of a BML grammar is the evaluation of a prototype grammar as used by a simulation system. For this purpose, mappings from the BML defined by the grammar into the languages of appropriate simulation systems will need to be developed. Work on these mappings is under way. In principle, military orders can be formulated with the help of a BML GUI. The orders are stored in the C2 systems in a JC3IEDM data base. They are also automatically transferred into the language of connected simulation systems. Within these simulation systems, the orders are
executed by the respective simulated forces. The execution provides an evaluation of the BML representation of the orders as well as a evaluation of the mapping. The result of the execution is reported back to the C2 system as BML expressions. These reports are stored in the data base as well as presented as part of the common operational picture. Again, this picture provides evaluation power. In this case, the BML reports and its underlying grammar can be evaluated.

The work on C-BML and its grammar, thus results in an automated information exchange between systems and its users. In the case that a C2 system exchanges information with a simulation system, the information exchange via BML expressions can be evaluated because the representations within the simulation systems provide an explicit representation of what is expressed by BML. This allows future adjustments of not only BML but also of the JC3IEDM. In the text we already pointed out some suggestions to improve the JC3IEDM.

The fact that BML expressions are based on a formal grammar as it is provided by our work ensures that the semantic and the meaning of these expressions can be represented (and exchanged) in a formal format which is the format of f-structures. F-structures allow the seamless interplay of syntax based processes, semantic based processes and ontology based processes as recommended by Tolk et al. [32].

7. Acknowledgements

Many questions arose when writing this paper how reports are handled in military doctrine. Scott Carey, Gerd Dürr, and Martin Kleiner provided authoritative answers and we are indebted to them. Dr. Schade performed this work at FGAN's Research Institute for Communication, Information Processing, and Ergonomics in cooperation with Bundeswehr IT office, section A5. Dr. Hieb performed this research under the Center for Excellence in C4I at George Mason University. We thank all of the SISO C-BML and NATO MSG-048 participants for their contributions to this area and acknowledge that this work could not have been performed without the intellectual contributions from these individuals.

8. References

8) Chairman of the Joint Chiefs of Staff Instruction “Global Status of Resources and Training System” (= CJCSI 3401.02), (http://www.dtic.mil/doctrine/).
13) Fillmore, C.J., “Frame semantics,” in: Linguistics of the Morning Calm (pp. 111-137), Seoul, South Korea: Hanshin 1982.
14) FrameNet website: http://framenet.icsi.berkeley.edu/.


Author Biographies

ULRICH SCHADE 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 a Lecturer at the Institute for Communication Research and Phonetics, Bonn University. Dr. Schade received his MA in Mathematics in 1986 and his PhD in Linguistics in 1990 at Bielefeld University (Germany), developing a connectionist model for language production processes. He has written many papers and book articles in the areas of Language Production, Ontology Development, and Cognitive Models.

MICHAEL HIEB is a Research Associate Professor with the Center of Excellence in C4I at George Mason University. Dr. Hieb was the Co-Chair of the SISO C-BML Study Group and also was on the team that developed the initial BML concept for the US Army while with Alion Science and Technology. Dr. Hieb was the Technical Director of the MRCI interface project while at SAIC. He received his PhD in Information Technology at George Mason University in 1996, developing an instructable Modular Semi-Automated Forces agent. He has published over 70 papers in the areas of BML, Simulation Interoperability with Command and Control Systems, and Multistrategy Learning.