A Grammar for Battle Management Language

Dr. Ulrich Schade  
Bastian Haarmann  
Fraunhofer FKIE  
Neuenahrer Str. 20  
Wachtberg, 53343  
Germany  
0049-228-9435-376  
ulrich.schade@fkie.fraunhofer.de  
bastian.haarmann@fkie.fraunhofer.de

ABSTRACT:

In order to improve the interoperability between C2 systems and simulation systems, Battle Management Language (BML) has been developed as standard for assigning C2 tasks to units in simulation system and for sending reports from the simulation back to the C2 system.

The specification of a language always is based on a grammar. This is especially true if the language in question is used for inter-system communication: Automated systems demand the foundation of such an exchange language on a formal grammar. As a consequence, under the roof of the Simulation Interoperability Standardization Organization (SISO) a product development group extents the current BML standard which is called Version 1 standard to the BML Version 2 standard by founding BML on a grammar.

In this paper, we will discuss the demands that exist for a BML grammar and we will argue that our grammar Command and Control Lexical Grammar (C2LG) which had been the linguistic basis for “Coalition BML” – the BML version used in the successful NATO RTO MSG 048 “Coalition BML” activities meets those demands.
1 Introduction and Structure of the Paper

A language that is supposed to be processable by a system has to be a formal language. Formal languages are defined by grammars. To be more precise, according linguistics, a formal language is defined as the set of all the sentences that can be generated by its grammar. The language we want to define is a language for Command and Control known as “Battle Management Language” (BML). Although BML is defined and standardized to some extent, this standard does not include a grammar so far. This paper will discuss the requirements that a grammar for BML should fulfil. In particular, it will argue that most of those requirements are fulfilled by the Command and Control Lexical Grammar (C2LG), the BML grammar we developed under the umbrella of NATO RTO MSG-048 “Coalition BML”.

The paper is structured as follows. First, we will present a short introduction to BML so that we can refer to BML’s peculiar characteristics while arguing which properties a BML grammar should have (section 2). Then we will discuss formal grammars (section 3) as the base for grammar definition. From this, we will derive some properties our BML grammar should have (section 4). In order to broaden this set of properties, we will take a look at those grammar theories that are applied in computational linguistics to process natural language (section 5). Again, we will examine those grammars for properties our BML should have (section 6). The paper concludes summing up the results (section 7).

2 Battle Management Language (BML)

The original idea of Battle Management Language (BML) was to achieve interoperability among C2 systems and simulation systems. Thus, BML has to be a language that can express orders, requests and reports in a way that is processable by systems. More precisely, C2 systems should be able to integrate a BML service so that orders can be expressed in the C2 system by using BML. Then those BML orders have to be sent to a connected simulation system that should be able to interpret the order. In the end, the simulated units should operate within the simulation according to the order given. In addition, results from the simulation, e.g., spot reports, should be transformed into BML reports and sent to the C2 system. The C2 system should interpret the reports so that the operator is informed of the simulated unit’s actions and their results. A BML that achieves these goals can be used in many ways. First among these ways is staff training. If a staff is trained on its C2 systems, sending orders and receiving reports and request, BML provides easy and natural communication with the troops commanded even if those troops are only simulated. In addition, BML can be used whenever a C2 system is connected to a simulation system, e.g., if the simulation system serves as a decision support tool or as a vehicle for alternative operational sequences as in after-action analysis.

The first work on the BML concept was sponsored by the US Army’s Simulation-to-C4I Interoperability Overarching Integrated Product Team (SIMCI OIPT) back in 2000 (Carey et al., 2001); cf. Pullen et al. (2010) for detailed information about the different BML approaches. Among those approaches, two are of specific importance. Recognizing the value of BML, in 2005, SISO as well as the NATO RTO Modeling and Simulation Group (MSG) spent resources to develop BML versions. Within SISO, a product development group was established to define a BML standard (Blais, Galvin & Hieb, 2005). Within NATO, MSG-048 “Coalition BML” was founded to explore BML concepts and to demonstrate their feasibility.
Without timely input from the SISO standardization efforts, NATO MSG-085 proposed a preliminary standard for BML. On that basis, C2 systems and simulation systems from up to nine nations were connected to operate in concert. Communication among those systems was handled exclusively by exchanging BML orders and reports as was demonstrated at I/ITSEC 2007 (De Reus et al., 2008a; Pullen, Carey et al., 2008a, 2008b; Pullen, Hieb et al., 2008) and 2008 (De Reus et al., 2008b; Pullen et al., 2009) as well as in a substantial experiment run at George Mason University, Manassas, Virginia, November 2009 for officers from the nations participating in the MSG (Pullen et al., 2010). Because of the success of MSG-048, NATO RTO decided to renew the efforts by establishing a follow-up MSG in June 2010 which is MSG-085 “Standardization for C2-Simulation Interoperation.” In the meantime, SISO at last proposed a BML “phase 1” standard. The “phase 1” standard still lacks a grammar. A grammar, however, will be part of the upcoming “phase 2” standard. In order to achieve that standard it seems worthwhile to merge SISO’s “phase 1” standard with the results already provided by the NATO MSGs. In the following we will discuss how that can be done with respect to grammar.

3 Formal grammars

If a BML is to be processed by systems, its grammar needs to be a formal grammar. Formal grammars have been introduced into the field of linguistics by Noam Chomsky. In “Syntactic Structures,” published in 1957, Chomsky answered the question “What do we know when we know a language?” by postulating that what we know is a set of words (the lexicon of that language) and a set of production rules that can be used to generate sequences of those words, especially the language’s sentences. In short, a sequence of words is called well-formed or “grammatical” if the sequence can be generated by the production rules operating on the lexicon.

Following Chomsky’s approach, a formal grammar G is a quadruple, G = (S, N, Σ, P), where S is the starting symbol, N is a finite set of non-terminal symbols, Σ is a finite set of terminal symbols (the lexicon), and P is a finite set of production rules. A production rule expands a sequence of symbols taken from the union of N and Σ to another sequence of symbols taken from the union of N and Σ. The only restriction is that the left-hand side of a rule must contain at least one non-terminal symbol. The language generated by G, L(G), is the set of all sequences of symbols from Σ which can be produced by applying the rules of P, starting from S. Although N, Σ, and P are finite sets, L(G) need not be finite because recursion is allowed.

The definition of a formal grammar as given above is quite abstract. It allows the construction of different types of formal grammars by applying restrictions to production rules. Linguistic theory categorizes the set of formal grammars into four types that form the Chomsky hierarchy (Chomsky 1957; Partee, ter Meulen & Wall, 1990, section 16.5): 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 types of production rules which are used in the definition of a specific formal grammar define the type of that grammar. Only grammars of types 2 and 3 exclusively use rules that can easily be applied by automated systems. Thus, only these kinds of grammars can be automatically processed easily. Therefore, BML should have a grammar of type 2 or 3. In linguistics, there has been a detailed discussion as to whether grammars of type 2 (context-free grammars) are sufficient to describe natural languages like English, cf. Partee, ter Meulen & Wall (1990, section 18.6). The result of the discussion is that it is quite probable that context-free grammars are not sufficient and a grammar of type 1 (a context-sensitive grammar) might be
needed. However, it was clear that nearly all constructions in English as well as in all other natural languages can be “elegantly and efficiently” processed by context-free techniques (Gazdar & Mellish, 1989, p. 133).

As a BML will not be a natural language, it need not include such exceptional constructions. On the contrary, BML has to be a language whose expressions can be interpreted by systems. Thus, all those ambiguities and exceptions that characterize natural languages have to be avoided. Indeed, one of the central demands for a BML is its unambiguity, cf. Tolk et al. (2004). In short, it is sufficient to restrict the search for a BML grammar to context-free and regular grammars.

4 Further Demands on BML Grammar and How to Meet Them

So far, we deduced from the fact that BML expressions have to be processed by systems that BML’s grammar should be a formal grammar or, more precisely, a context-free or a regular grammar. In addition, BML expressions are supposed to be unambiguous. Of course, there are further demands on BML which impact the choice of BML’s grammar. The most important of those demands, as, for example, stated in Tolk et al. (2004) and Blais, Galvin & Hieb (2005), are a) that BML has to respect military doctrine and b) that it is compatible with the standard data model JC3IEDM; cf. Multilateral Interoperability Programme (MIP) (2008) for information on the JC3IEDM.

4.1 Doctrinal Demands on BML Grammar

Doctrine for military communication is set in respective field manuals of the nations and in NATO standard agreements (STANAGS) for NATO as a whole. For example, the standard for orders is documented in NATO’s STANG 2014 “Formats for Orders and Designation of Timings, Locations and Boundaries” and in the US Army’s Field Manual 6-0 “Mission Command: Command and Control of Army Forces”. In particular, these documents describe the operation order, or “five paragraph field order”. The core of the operation order is paragraph 3 “Execution,” in which tasks are assigned to units. The assignment of the tasks to units is also the core information that has to be delivered to a simulation system in order to allow that system to simulate the execution of the order received. Thus, statements that express those assignments constitute a central part of BML.

With respect to task assignment, doctrine refers to the so-called “5 Ws”: the What (what kind of task is to be executed), the Who (the unit that is ordered to execute the task), the Where (the spatial conditions of the task), the When (the temporal conditions of the task), and the Why (the task’s purpose which links it to the concept of operations). Linguistic theory uses the concept of “constituency” to refer to the idea that some words in an expression belong together because as a unit they denote a real world object, an idea, a spatial object, a temporal object or something similar. Constituents are not only the building blocks of natural-language sentences, but also represent the linguist version of the 5Ws; cf. Sells (1985, chapter 1, section 2) or Jurafsky & Martin (2009, section 12.1) for more details on constituency. In order to give an example, the sentence “Occupy Prins Willem-Alexander Brug at Parnass as soon as possible” includes the verb “occupy” and the constituents “Prins Willem-Alexander Brug” (the direct object of the verb), “at Parnass” (a spatial adverbial), and “as soon as possible” (a temporal adverbial). With respect to the 5 Ws, the verb and the direct object constitute the
What whereas the spatial adverbial constitutes the Where and the temporal adverbial the When. In summary, to respect the doctrinal demand of using the 5 Ws, BML expressions should consist of constituents that mirror the 5 Ws. As we already have shown in Schade & Hieb (2010) – see also Carnie (2010, p. 81f.) – context-free and higher grammars allow constituents but regular grammars do not. Thus, our BML grammar needs to be a context-free grammar.

In the context of the MSG-048, we already defined such a context-free grammar for BML (Hieb & Schade, 2007; Schade & Hieb, 2006, 2007; Schade, Hieb, Frey & Rein, 2010). This grammar is called “Command and Control Lexical Grammar” (C2LG). It had been the basis for the BML used in the MSG-048 demonstrations and experiments. In C2LG, a task assignment is expressed according to context-free production rules that follow the rule format as given in (1). A task assignment expression corresponding to the rule in (1) is given in (2).

(1) TaskAssignment →
       TaskVerb TaskerWho TaskeeWho (Affected) Where When Modifier Why Label

(2) occupy BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start at TP1 in-manner fast in-order-to enable label-o24  label-o23;

Example (2) represents the English example discussed. The verb in the example, which expands the non-terminal TaskVerb of the rule format (1), is “occupy”. All task assignments of C2LG’s BML version begin with the verb denoting the task to be assigned. The verb is followed by the constituents. In contrast to natural languages, the sequence of the constituents is fixed. Although English enforces a quite fixed constituent order in its sentences – in contrast to nearly all other languages of the world – it is at least allowed to shift adverbials to the beginning of a sentence as in “On June 15 the first troops of the Army of Northern Virginia [...] slip across the Potomac at Williamsport and begin the invasion of the North” (Shaara, 1974, p. ix). A flexible constituent order induces syntactic ambiguity. For example, the German sentence “Die Mutter liebt die Tochter” is ambiguous. “Die Mutter” might be the subject – then the sentence translates to “The mother loves the daughter” – or it might be the object – then the sentence translates to “The daughter loves the mother.” To exclude this kind of ambiguity and such respect the demand that BML has to be unambiguous, C2LG enforces a totally strict constituent sequence as expressed in (1). Following the tasking verb, the next constituents are “Tasker” and “Taskee,” the Whos. In (2) the Tasker, the organization that assigns the task, is the battalion 661 (BN-661), and the Taskee, the organization that has to execute the task, is its 2nd company (Coy2). The “Affected” of the task is whatever is affected. In (2), this is the Prince Willem Alexander Bridge (Prins Willem-Alexander Brug). “At Parnass” is the denotation of a location, the Where; “start at TP1” is the When, “in-manner fast” is a Modifier, and “in-order-to enable label-o24” is a Why. The “label-o24” part of the Why refers to another task assigned to the same company, namely to secure the bridge. All C2LG expressions end with such a label so that they can be referred to by other expressions. The label in (2) is “label-o23”.

4.2 Demands with respect to JC3IEDM compatibility

The JC3IEDM is the primary standard for exchanging C2 data developed by the MIP [http://www.mip-site.org]. BML is supposed to be JC3IEDM compatible (Blais, Galvin & Hieb, 2005). For the BML version developed by MSG-048 this means that the language terms (according to formal grammar theory these are terminals of set Σ, the words of the language)
have been taken from JC3IEDM’s set of attribute values. Thus, these terms are well-defined in agreement with military doctrine. As a consequence, MSG-048 was able to develop a mapping service by which the BML orders and reports exchanged during the demonstrations and experiments were mapped into and stored in JC3IEDM database.

As a data model, the JC3IEDM proposes a structure. In principle, a grammar can be calculated from this structure to serve as a BML grammar. Such an approach would even enhance the compatibility between BML and the JC3IEDM beyond what has been achieved by MSG-048. JC3IEDM’s structure is, of course, optimized with respect to database demands. The structure is a database’s structure. In contrast, the structure that is assigned to BML expressions by the production rules of a BML grammar is a syntactic structure, optimized according to linguistic principles. Often these two kinds of structures go hand in hand but sometimes differences are recognizable. In particular, JC3IEDM’s structure does not comply with constituency. Temporal information is subordinated under task information which would make the When to a part of the What. In addition, the temporal information is split into many disjointed pieces so that a When is not constituted in JC3IEDM’s database structure. Similar problems exist for the Where; for more details about these problems see Schade & Hieb (2010). In summary, using JC3IEDM’s structure as basis for the production rules of a BML grammar would contradict the efforts to represent military doctrine by BML. Thus, the solution taken by the MSG-048 is preferable.

5 Contemporary Grammar Theories

One of Chomsky’s key insights was that grammaticality is independent from meaning. Chomsky gave the example (3) of a grammatical but not meaningful sequence in order to illustrate this point. In addition, word sequence can be meaningful even when not grammatical as is shown in example (4).

(3) Colorless green ideas sleep furiously.
(4) The unit to the phase advances.

Additionally, there is no trivial mapping from syntax to semantics (Sadock, 2003). For example, in “He loves Virginia above all, the mystic dirt of home. He is the most beloved man in either army.” (Shaara, 1974, p. xi, [about Robert Edward Lee]), both instances of “he” constitute a constituent. Both bear the syntactic role “subject.” However, in the first sentence, “he” bears the semantic role “agent” (“an active animated entity that voluntary initiates an action” – Sowa, 2000) whereas the second sentence does not describe an action at all: the second “he” refers to a person an attribute is assigned to. Obviously, the meaning of the second sentence is similar to “He is respected by the soldiers of both armies.” In this sentence, “he” again bears the syntactical role “subject.” However, the semantic role “agent” has to be assigned to “the soldiers of both armies.”

Contemporary syntactic theories like Head-Driven Phrase Structure Grammar (HPSG) (Pollard & Sag, 1994; Sag, Wasow & Bender, 2003) or Lexical Functional Grammar (LFG) (Bresnan, 2001; Kaplan & Bresnan, 1982; Sells, 1985, chapter 4) focus on syntax. They calculate the syntactic structure of natural language expressions. In addition, they provide starting points for Semantic Role Labeling (Jurofsky & Martin, 2009, section 20.9). Semantic Role Labeling is the process that assigns semantic roles to constituents on the basis of a syntactic analysis. As such, Semantic Role Labeling is the link between syntactic and
semantic interpretation. In order to calculate the syntactic structure of an expression, in order to assign syntactic roles and in order to prepare Semantic Role Labeling, grammars like HPSG and LFG enhance the lexical entries with matrices of attribute-value pairs. The information stored in these matrices is then used in analytic processes. For example, a context-free production rule like $S \rightarrow NP \ VP$ (meaning a sentence $S$ is composed of a noun phrase $NP$ – which will receive the role “subject” – and a verbal phrase $VP$ can only be applied some specific parts of the respective attribute-value matrices correspond to each other (under the operation of “unification,” cf. Shieber, 1986). Sag, Wasow and Bender (2003, p. 69) discuss the example of subject-verb agreement in English to illustrate with respect to the “$S \rightarrow NP \ VP$” rule: the subject noun phrase and the verb have to correspond with respect to the grammatical features “person” and “number.” So, in the matrices, there are attribute-value pairs using “person” and “number” as attribute so that the unification process can check whether the respective values are identical in the subject noun phrase matrix and the verb matrix.

As has already been mentioned, the attribute-value matrices are also used for preparing the Semantic Role Labeling. This is possible because the lexical elements not only include attribute-value pairs for storing pure syntactical information but also pairs for storing information that is at least in part semantic. For example, in LFG, the lexical entries of the verbs “persuade” and “promise” include information about which constituent of the main clause will serve as subject in the complement clause (Sells, 1985, p. 166, example 49). During the analysis, this information determines whether “Herb” or “Louise” receives the semantic role “agent” of the “following” action in the sentences “Herb persuaded Louise to follow” and “Herb promised Louise to follow,” respectively.

6 Implications for BML Grammars

Processing a natural language expression is complex and laborious. Often the analysis of natural language expressions consists of many processing steps. For example, a LFG analysis of a sentence is a three-step process. First, the constituent structure (c-structure) of the sentence is calculated according to pure context-free parsing. In the second step, the c-structure and the syntactic information stored in the attribute-value matrices of the lexical entries are exploited to calculate the sentence’s functional structure (f-structure). The constituents receive their own attribute-value matrices which include their assigned syntactic role. Third, the argument structure (a-structure) is calculated from the f-structure which means that the constituents receive their semantic roles.

A BML is not a natural language. It is not the task of a BML grammar to more or less copy an existing natural language. On the contrary, the BML grammar defines BML. Thus, the BML grammar can be defined so that a) the constituent can easily be calculated and b) each constituent can be assigned a semantic role directly and unambiguously. Neither syntactic roles nor syntactic features need to be part of a BML grammar.

At this point, we can sum up what a BML grammar should look like.

- It should have context-free production rules.
- Its lexical terms should be taken from the JC3IEDM.
- Its non-terminals should denote semantic roles like those mentioned in (1) which are Tasker, Taskee (= Agent), Affected (= Patient), When, or, even better, When’s sub roles like Start-When (= Start) and End-When (= Completion), Where, or, better, Where’s sub-roles like At-Where (= Location), Origin, Route (= Path), Destination,
and Direction, as well as Why, Instrument and so on. The role names in brackets are the linguistic names for the roles as given by Sowa (2000).

- The sequence of the constituents should be fixed. If the sub-roles of When and Where are used, the respective constituents should start with a specific unique keyword such as “from” in the case of “origin” or “towards” in the case of “direction” to indicate which sub-role a specific constituent belongs to.

All the points mentioned above have been realized in the BML version used by MSG-048, which surely contributed to the success of the group.

However, it might make sense to learn further from grammars like HPSG and LFG and take on the principle of lexicality as well, at least with respect to BML’s “verbs”, the terms that denote the type of task to be assigned to units. Lexicality had been proposed by C2LG (Schade & Hieb, 2006, section 2.3; Schade, Hieb, Frey & Rein, 2010, section 1.3) but, in contrast to the points listed above, was not taken into consideration in MSG-048’s implementation of a BML.

Using lexicality would allow for taking into account the semantic specifics of tasks and thus respecting their doctrinal content. This is beneficial with respect to spatial constraints that come with specific tasks. Tasks require their own specific sets of control features, e.g., a “march” needs, among other things, a route and a release point, whereas a “move to contact” needs a phase line called limit of advance. In order to achieve interoperability and to make sure that the simulation systems respond to orders for their simulated units as intended, it is necessary to follow doctrine and to provide the specific control features along with the task assignments. The information about the control features in question can be provided in the form of attribute-value matrices (or in equivalent XML structures) as a task-verb’s Where. How they should look depends on the task-verb in question. The respective constraints could be determined in the verbs’ lexical entries. Part of this has already been proposed in C2LG’s specification (Schade, Hieb, Frey & Rein, 2010). However, additional work is needed in order to cope with all the doctrinal subtleties.

7 Conclusion

A Battle Management Language has to be a formal language to respect its requirements. This means, it has to be based on a grammar. In this paper, we discussed how the requirements for a BML translate into the demands for its underlying grammar and how they can be met. By reflecting grammar theory and discussing some aspects of grammars like HPSG and LFG we deduced a list of properties a BML should have. This list is met by the C2LG which thus can be recommended to serve as a grammar for BML.

8 References


Sells, P., Lectures on Contemporary Syntactic Theories (= CSLI Lecture Notes 3). Stanford, CA: CSLI.


