11th International Command and Control Research and Technology Symposium

Coalition Command and Control in the Networked Era

Topics: **Coalition Interoperability, C2 Modeling and Simulation**

Paper's Title: **Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests and Reports (I-069)**

Authors:

Dr. Ulrich Schade
FGAN-FKIE
Neuenahrer Str. 20
Wachtberg, 53343
GERMANY
0049-228-9435-376
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

Abstract

Coalition Operations are characterized by the diversity of organizations involved. This diversity can cause challenges, in particular, with Command and Control (C2) due to 1) the difference in military doctrine between forces, and 2) the incompatibility of their Information Technology (IT) and their IT systems. The majority of current IT solutions do not address differences in military doctrine.

Current emerging solutions include the Command and Control Information Exchange Data Model (C2IEDM) developed by the Multinational Interoperability Programme (MIP) as a NATO standard. However, while C2 information can be stored in the C2IEDM, exchange relies upon a specific exchange mechanism. A large number of ad hoc business rules enforce consistency and semantic coherence. To incorporate military doctrine, orders, requests and reports should be conserved and exchanged as such. Therefore, development of a C2 language for military messages is clearly preferable. The keystone of such a language is a formal grammar. A formal grammar ensures that the messages can be processed automatically, a necessity under the network-centric perspective. In this paper we propose an initial C2 grammar which not only unifies military doctrine with current IT in a network-centric approach, but is also applicable to a wide range of civil operations.

1 Introduction

Advances in Coalition Interoperability are being developed along two axes – Net-Centricity and standard Command and Control (C2) Semantics. Assuming success in these endeavors, there is yet a further axis that is beyond those two. The future challenge is that of formalizing a mission-oriented "language" for C2. Our research has shown that while situational awareness is being addressed by currently developed solutions, the order generation or tasking process is not supported by these solutions.

In keeping with the lessons learned in the past 10 years, we present a methodology applicable to both military and civilian domains. A formal grammar is a firm foundation for the next generation of net-centric message exchange services, taking advantage of standard semantics in well-specified domains. A new aspect here is the leveraging of military doctrine, needed for intelligent applications such as adaptive planning and advanced decision support aids.

In this paper, we present a formal grammar for command and control of complex operations, involving diverse organizations such as military coalitions. This grammar, designed for the exchange of orders (and also the specification of military tasks), is also applicable to requests and reports with appropriate specialization.

1.1 C2 in a Network-Centric Environment

In a network-centric environment, information empowers the actors. The exchange of information has to run smoothly such that fleeting opportunities can be noticed and exploited by the actors 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. [1], 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.

As Alberts and Hayes [1, 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 North Atlantic Treaty Organization (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 incorporate "free text" and as such is hard to process automatically.

1.2 Limitations on C2 from Current IT

"Free text" that exists in current military message text formats such as Allied Data Publication-3 (ADatP-3) is for the benefit of humans. The highly trained, professional soldier has little problem dealing with this "free text" as long as it follows the standards. Current automated systems that deal with "free text" handle it as a single data field and pass the <character string> on. Understanding of the content of the <character string> does not exist within the system. However, it is required under the network-centric perspective, in order to ensure interoperability on the cognitive level.

C2 systems are evolving. The future systems must not only be interoperable on the cognitive level, they will also incorporate 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.). The command information is still carried in free text form which has all of the problems discussed above.

A recent development in simulations is the command agent or intelligent agent software. Software agents are designed to receive general "mission type" tasks, and cognitively process the tasks applying a situational awareness. Using this information and by applying knowledge of military doctrine, tactics and techniques it determines its own solution to the problem and then

issues appropriate orders and directives to the simulated forces. It subsequently monitors the task's progress against the planned progress. The intelligent agent then makes corrections as necessary. This type of simulation, layered over a more traditional simulation, can greatly help the military commander to judge the actual information and the importance of incoming information. The agents therefore will contribute to network-centric operations. Nevertheless, the introduction of "intelligent agent," "command entities," or other Command Decision Model (CDM) types of software requires unambiguous inputs. Free text messages are not an option.

All the communication and interoperability problems would collapse if there existed a clear and unambiguous language derived from doctrine already defined for military messages. Being unambiguous and thus processable by systems, this "Battle Management Language" can be used for the communication among C2 systems, it can be used to control the agents in simulations, and it can even be used for communication among C2 systems on the one hand and simulation systems on the other – with the additional possibility of a communication with future robotic forces. In addition, the incorporation of standards for military messages will contribute to understanding and interoperability.

1.3 The Multinational Interoperability Programme Approach

Development of an unambiguous C2 Language requires a vocabulary where the terms are fixed in their meaning. The Multinational Interoperability Programme has already produced semantics for C2 terms suitable for coalition operations. This is documented in the Command and Control Information Exchange Data Model (C2IEDM). It will be accepted as a NATO standard in the next version – the Joint Command, Control and Consultation Data Exchange Model (JC3IEDM).

The C2IEDM consists of both a Data Model and an Exchange Mechanism. The Data Model is intended to represent the core data types identified for exchange across multiple functional areas. It lays down a common approach to describing the information to be exchanged in the command and control domain. Thus, the approach is generic and not limited to a special level of command, force category, etc. In general, C2IEDM describes all objects of interest on the battlefield, e.g., organizations, persons, equipment, facilities, geographic features, weather phenomena, and military control measures such as boundaries using a common and extensible data modeling approach.

To summarize, the Data Model of the C2IEDM defines the semantics of coalition C2 terms, for a well-established and standardized vocabulary of a C2 Language.

Besides the Data Model there is an Exchange Mechanism that uses a replication protocol that allows the exchange of data between two systems that employ the C2IEDM.

1.4 The Need for a Language

With the advent of the C2IEDM defining standard semantics to C2 terms on the one hand and providing a data exchange mechanism on the other, it becomes necessary to justify additional effort, such as developing a language. To this end, we present a brief analysis on why just relying upon a data model is insufficient for military communication.

First, the C2IEDM is designed from the standpoint of a headquarters agency. It is primarily designed for planning and status, but not for actually performing tasks. This can be seen by examining the table that relates tasks to organizations – "organisation-action-association." It is arranged for representing the units that approve, control, initiate, coordinate, observe, plan, or request a task, but NOT for representing the unit(s) that (should) execute it.

Second, the C2IEDM as well as its future successor, the JC3IEDM, are (as of the publication of this paper) not able to represent orders, requests, or reports as entities which concatenate single expressions about actions and states in a cohesive way. The C2IEDM can "only" represent single expressions without providing a sensible frame to connect and to structure them. For example, "reporting-data" only can connect facts within one single table, and "context" gives the ability to connect entities but does not do so according to any set structure. The military doctrines which inspired the standardization of military messages are not incorporated in the data model.

Third, the C2IEDM is for exchanging data, and aims to create shared situational understanding. However information should be exchanged. And information exchange should be driven by military doctrine. Only then can meanings and intentions be communicated. This, however, is what a language is for.

Fourth, a grammar – an essential part of a language (cf. section 2) – 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 C2IEDM 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 C2IEDM. 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 C2IEDM. 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.

1.5 Organization of the Paper

In Section 1 the basic concepts and motivations have been introduced, setting the framework for the development of a language. Taking the language's vocabulary from the C2IEDM, a grammar is still needed to define the language. Thus, in Section 2 the development of formal grammars is described. We need a *formal* grammar to ensure that the language's expressions can be processed, automatically. Linguistic theory is examined in Section 2 and the types of grammar are reviewed to inform a discussion of how a grammar should be designed to support C2 processes. In Section 3, previous work in this area, known under the topic "Battle Management Language" (BML), is presented. Section 4 presents a formal grammar for Orders, Requests and Reports. First, the grammar for Orders is given, then it is related to Requests. A further specialization of this grammar is made for Reports. Section 5 concludes with the benefits of a grammar and two important further research areas – that of scope (how general is the grammar) and domain applicability (how can the grammar be specialized for specific communities such as different nations or different types of operations). Section 5 also looks forward to future work.

2 Grammar

In his famous book "Syntactic Structures" [6], published in 1957, Noam Chomsky answered the question "What do we know when we know a language?" by postulating that what we know is a collection of words and rules. The words form the vocabulary or lexicon of this language and the rules are used to generate sequences of those words, the sentences of this language. As a consequence, a sequence of words is defined as grammatical if the sequence can be generated by the rules operating on the lexicon.

A formal grammar is defined as an abstract description of a lexicon and production rules (cf., subsection 2.2.1). In a less formal way, however, grammar is used to refer to the set of production rules alone. Thus we can say, using the less formal definition of grammar, that a language consists of its lexicon and its grammar. Therefore, in this section, we concentrate on the development of a grammar for a C2 language.

2.1 Formal Methods

Following Chomsky's approach, a formal grammar G is defined as a quadruple, $G = \{S, N, \Sigma, P\}$, 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 here 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 to be finite because recursion is allowed.

2.2 Types of Grammars

Chomsky postulated four types of grammar. They are ordered within what is designated as the Chomsky hierarchy. Grammars of type 0 are unrestricted. Grammars of type 1 have rules of the form $\alpha A \beta \rightarrow \alpha \gamma \beta$ where A is a non-terminal symbol, α, β, and γ are sequences of terminals and non-terminals, and γ consists of at least one symbol. Such a rule can be understood as "A is expanded to γ in the context of α and β." Thus, these kinds of grammars are called *context sensitive* grammars. Grammars of type 2 have rules of the form $A \rightarrow \gamma$ where again A is a non-terminal symbol and γ is a sequence of terminals and non-terminals. Such a rule can be understood as "A is expanded to γ." In contrast to type 1 grammars, no context is taken into account. Therefore, these grammars are called *context free* grammars. Grammars of type 3 are even more restricted with respect to their rules. A type 3 rule maps a non-terminal either on a single terminal (a single word) or on a sequence consisting of exactly one terminal and one non-terminal (Grammars of type 3 are also called *regular* grammars).

2.3 Constituency and BML's type of grammar

In order to state a formal grammar for our language (which we will call a BML in the following, as it derives from previous work on BML – cf. section 3), we will specify the lexicon (the set of terminal symbols Σ), the set of non-terminal symbols N and the set of production rules P. Per the discussion above, the C2IEDM fulfils the requirements for a standard lexicon in the C2 Operational Domain. Thus we will specify the C2IEDM as the lexicon for the C2 grammar described here.

The next thing we must do is specify the rules and by doing so also the set of non-terminals. Specifying the rules also determines the type of grammar. As we want to have a language that can be processed automatically, we prefer more restricted grammars over less restricted ones. To be more precise, it is much easier to process (parse or generate) expressions from a type 3 or a type 2 grammar than from a type 1 or a type 0 grammar. But to be automatically processable is not the only constraint for BML. In order to convey understanding and a sender's intent, BML also should not constrain (at least not too much) the full expression of a natural language. This led us to the template of "5Ws" (Who, What, Where, When, Why) – a good starting point for discovering what has to be expressed by a BML, as well as an initial specification of the rules.

Consider the order "*Advance to phase line TULIP at 151200ZSept03.*" In this order there is a WHAT, a WHERE, and a WHEN. These Ws correspond to "*advance,*" "*to phase line TULIP,*" and to "at *151200Zsept03,*" respectively. From a language point of view, these word strings are *constituents*. Constituents are constructed out of smaller word strings in a way that also an appropriate structure is assigned to the whole. In the example, this structure is given by the Ws. Constituents are even more: a constituent of a correct or valid expression can be exchanged by another of similar type with the result that another expression arises that is also a valid expression. For example, if one exchanges the WHERE-constituent by another WHERE-constituent another sequence arises, e.g., "*Advance to (the town of) Barnstedt at 151200ZSept03.*" In principle, the idea of the 5Ws is a mirror image of the linguistic concept of constituency. Constituency is something we clearly want to have in BML. However, there is a cost. A rule like

"OB → WHAT WHERE WHEN" (OB is the abbreviation for the basic expression of an order), obviously, is not a type 3 rule. Thus, BML cannot be of type 3, and must of necessity be of type 2 or less restrictive.

To determine the type of a grammar means to find the most restrictive grammar (the higher the type the better) that generates the language. Type 3 is excluded, as stated above. As we would like BML to be nearly expressive as a natural language, say English, we can check what is the grammar type of natural languages. According to Chomsky [6], natural languages are context-sensitive, that is of type 1. However, "the overwhelming majority of the structures of any natural language" can be parsed and generated by a type 2 grammar, cf. [10]. Peter Reich [17] even proposed that Chomsky was wrong and said that natural languages are of type 2. With this in mind, we choose BML to be of type 2. Thus, its rule will have the form $A \rightarrow \gamma$, and its automatic processability is ensured.

2.4 Subcategorization

Besides constituency, there is another basic concept (cf. [21], chapter 1, for a general discussion of basic concepts of syntax) we want to use within BML, namely subcategorization. As has been discussed above, part of the function of a grammar is to determine how the meaning of a sequence of words can be calculated from the meanings of the subsequences (and finally from the meanings of the words). The meaning of the basic expressions (the sentences) will be calculated out of the meanings of its constituents. In this regard, the application of subcategorization will ensure that exactly the correct constituents are used. For example, if the task is "*advance,*" any WHERE-constituent cannot be used with this task. "*Advance at Verkeersbrug Zaltbommel*" sounds wrong (even besides the Dutch name for the bridge) as one cannot advance to the point one is already at, but "*Advance to Verkeersbrug Zaltbommel*" is fine. The point is that on the one hand the WHERE-constituents can be subcategorized in locations, destinations, directions and so on and on the other hand tasks normally must be used with specific WHERE-constituents (some of them are obligatory, as destination is for the "advance" task), and cannot be used with others. This, of course, is true not only for WHERE-constituents but for the other Ws as well. In general, subcategorization builds the base to ensure that obligatory constituents of the requested subcategory are there (principle of completeness) and that constituents of improper subcategory are not (principle of coherence). Using subcategorization and its incorporated principles we go beyond the 5Ws.

Subcategorization taps into semantics, especially into the theory of semantic roles [7, 8, 11, 13, 22], but also bears syntactic aspects. With respect to BML, we applied it to the tasks/actions (the "verbs" of BML). This also means that BML becomes lexical-driven as are the natural languages humans speak [15]: The rules for basic expressions will start with an action verb (normally, a task verb), and the task spans a *frame* as provided by [8]. The frames have slots to be filled by the respective constituents, obligatory ones and facultative one (cf. [20], appendix A, for examples of these rules). In a language like English, noun phrases like "*the 12th Spanish Cavalry Regiment*" as well as prepositional phrases like "*to Verkeersbrug Zaltbommel*" form most of the constituents, and in linguistics, there are four predominate phrase structure grammars, i.e., GB, GPSG, HPSG, and LFG. Lexical Functional Grammar (LFG) has all the properties we asked for including the

principles of completeness and coherence. Therefore, our BML grammar is designed as LFG variant. LFG was introduced by Ronald Kaplan and Joan Bresnan [14] and is further described in [3].

3 Previous Work

In the development of the grammar presented in this paper, we have been guided by our experience with a standard from the Modeling and Simulation community –known as "BML." While the original BML had specific goals related to interfacing M&S systems to C2 systems it became apparent to the M&S community that a BML is of greatest value if it is also used for sharing information in a network-centric fashion among C2 systems/services. To date, the BML initiatives have concentrated on the 5Ws. A persistent issue with the original BML work is the lack of formal syntax and semantics – in other words, the lack of a grammar. In the following we will describe the previous work done with BML. From this description, it can be seen how our work is integrated into former and current initiatives, how the former development efforts with BML inspired the development of a formal grammar and how the specific rules for the formal grammar to be presented in section 4 are motivated.

3.1 Former Battle Management Language Initiatives

A major drawback of using computer-simulated training is the need for large contingents of support personnel to act as workstation controllers and provide the interface between the training unit and the simulation. The group of workstation controllers is often as large as, or larger than, the training audience. While this enables training opportunities at the corps and division echelon, it is still resource-intensive and lacks the degree of fidelity that actual combat operations present to the commander and staff. What is needed is a standard representation and tools that can be used to automate the simulation interface. A BML would address this need.

Each major simulation used today to represent military operational forces has a language to task simulated units. Unfortunately, each of these is specific to its own simulation and often is driven by technical constraints of the simulation system and not by operational necessities of the warfighter, e.g., [5].

Taking the widest possible interpretation, BML has been defined [2] 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.***

The objective of the BML work is to define an unambiguous language to describe the commander's intent in a way that soldiers and systems can understand and make use of it. The resulting language should be applicable not only to simulation systems, but also to operational command and control systems, and robotics.

In 2002-2003 SIMCI sponsored an Army BML Proof of Principle (PoP) demonstration, bounded by requirements to: 1) eliminate "free text"; 2) employ a realistic scenario; 3) use BML to link a C2 application to a simulation in a doctrinally consistent manner; 4) employ doctrinally correct tasks and units; and, 5) employ a scenario useable by both the C2 application and the simulation, thereby requiring a common terrain database [12].

To meet these requirements, the demonstration employed a realistic scenario in which a heavy brigade with its supporting units conducted operations at the National Training Center, Fort Irwin, California. The demonstration contained four components: 1) a C2 application to develop Courses of Action (COAs) and generate orders; 2) a simulation, used to simulate the effects of orders generated by the C2 application; 3) a BML repository (using a "5W" representation [4]); and, 4) a graphical user interface to allow staff members to view BML-based "5W" orders and create orders for subordinate units whose missions were subsequently carried out in the simulation.

The result of this effort was a series of successful BML PoP demonstrations to senior Army leaders in 2003-2004, generating enthusiasm and sanction for additional development of BML concepts. From a combatant commander's perspective, the importance of the BML PoP is in meeting the functional requirements, thus paving the way for a viable COA analysis and mission rehearsal tool for use in training and real-world operations. This work was the foundation for several other BML initiatives with Air Operations and Coalition Forces [2].

3.2 Current Coalition Initiatives

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 [16].

Within the Simulation Interoperability Standards Organization (SISO), the Coalition BML (C-BML) Study Group was formed in September 2004 to investigate the concept of BML and develop a plan for a BML Standard. The Study Group conducted a number of technical meetings involving a membership of over 100 persons from 11 different countries. In September 2005, the Study Group recommended that a Product Development Group (PDG) be formed in SISO to standardize BML [2].

In parallel to the C-BML Study Group activities, the NATO Modeling and Simulation Group (NMSG) established a 12 month Exploratory Team (ET-016) on C-BML [2, 23, 24]. The team, led by France, endorsed the requirement for a C-BML and has proposed that a 3-year Technical Activity Program should be established. Their recommendation was submitted to a meeting of the NMSG in October 2005 in Poland and a NATO Technical Activity (MSG-048) has been approved for 2006-2009.

Our linguistic approach is part of both the SISO standardization and the NATO C-BML initiative.

4 A Grammar for Command and Control

The BML grammar presented in this section applies the principles we presented and discussed above. In the first part of this section, we will concentrate on orders and the work we presented in [20]. In the second part, we will widen the scope to requests and reports.

4.1 Orders

In our work on BML grammar, we start with the development of a "tasking grammar." Its scope is to formalize orders. This is 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. Reports include a greater richness of linguistic phenomena, e.g., modality terms like "most probably," "apparently," "possibly" and so on. To cover these phenomena requires additional rules (cf. section 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 currently the most applicable given the behaviors available. Section 3 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 outline our solution to these aspects by presenting and discussing the rules needed to generate and to parse.

In order to represent the major parts of an order's execution section, our grammar starts with a single rule:

(1) S → OB* C_Sp* C_T*

This rule means that a BML message which is an order consists of three parts, basic expressions to assign tasks to units (indicated by the non-terminal OB), spatial coordinations (indicated by the non-terminal C_Sp), and temporal coordinations (indicated by the non-terminal C_T). The asterisk indicates that arbitrarily many of the respective expressions can be concatenated together.

According to the principles we want to hold for the grammar, expressions are composed of a terminal symbol and its frame. An order's basic expression's terminal symbol is a tasking verb, taken from C2IEDM's table "action-task-activity-code." Thus, the rules to expand OB have the general form as given in (2a). (2b) to (2f) give examples.

(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 → ***assist*** Tasker Taskee Action At-Where
Start-When (End-When) Why Label (Mod)*

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

(2e) OB → ***defend*** Tasker Taskee Affected At-Where
Start-When (End-When) Why Label (Mod)*
(2f) OB → ***march*** Tasker Taskee Route-Where
Start-When (End-When) Why Label (Mod)*

Tasker is a non-terminal to be expanded by the name of the one who gives the order, Taskee is a non-terminal to be expanded by the name of the unit that is ordered to execute the task, and Start-When and End-When are non-terminals to be expanded by temporal phrases. The temporal phrases for Start-When are given in (3a) and (3b). End-When expands analogously, but is optional as indicated by the parentheses. Tasker, Taskee, Start-When, and End-When appear in each basic order rule.

(3a) Start-When → **start** Qualifier1 Point_in_Time
(3b) Start-When → **start** Qualifier2 Action

In (3a) and (3b), respectively, Point_in_Time expands to a point in time (a datetime), Action expands to a label which refers to an action, e.g. another task, Qualifier1 expands to a value from C2IEDM's table "action-task-start-qualifier-code," e.g. to **nlt** (not later than), and Qualifier2 expands to a value from table "action-temporal-association-category-code." (3b) refers to a relative point in time, e.g. at the start of a particular action (whenever this may occur).

Affected in (2a), is a non-terminal to be expanded by the name of the one to be affected by the task; in linguistic terms this is the patient. Whether Affected is part of a rule depends on the tasking verb. It is there if the tasking verb's frame requires it as in (2d) and (2e). The same is true for Action in (2a) – separated from Affected by the exclusive or "|". The same is also true for the Where in (2a). It is either an At-Where or a Route-Where as determined by the verb. A Where has to be expanded by location phrases. These expansions are complex expansions, especially in the case of Route-Where. E.g., Route-Where can be expanded to "**from** Location **to** Location **via** Location **and** Location." Some of the respective phrase rules are given in (4).

(4a) At-Where → **at** Location
(4b) Route-Where → Source Destination Path | Source Path | Destination Path | …|
along Route
(4c) Source → **from** Location
(4d) Destination → **to** Location

A basic rule ends with the non-terminals Why, Label and the optional Mod. Why represents a reason why the task specified by the rule is ordered. At the moment, it could be expanded by a single tasking verb (a value of "action-task-activity-code"). It is to be determined whether a more complex expansion is necessary, e.g., an expansion by a reduced basic expression. Label is expanded by a unique identifier. By this identifier the single order represented by the respective basic expression can referred to in other expressions, especially in temporal coordinations. The optional Mod (for modifier) is a wild-card that represents additional information necessary to describe a particular task, e.g., formation – to specify a particular formation for an advance, or speed – to specify the speed of a road march.

The abstract rule for spatial coordination is (5a); (5b) and (5c) give examples.

(5a) C_Sp → Control_Feature Tasker (Taskee) Start-When (End-When) Label
(5b) C_Sp → ***area of responsibility*** Tasker Taskee
Start-When (End-When) Label
(5c) C_Sp → ***hazard area*** Tasker Start-When (End-When) Label

The spatial coordination rules have key words that denote control features, e.g., lines or areas. These are taken from C2IEDM's table "control-feature-type-category-code." In this case the "area of responsibility" is assigned by a commander to be used by a subordinate and is an area defined by natural features or control measures for the exclusive operation of the subordinate unit's forces. However, a "hazard area" is identified by a unit, but not assigned to a subordinate unit, hence there is no Taskee argument in (5c).

The abstract rule for temporal coordination is (6a); (6b) is not a rule, but an example expression, denoting that the action referred to by "*label_3_12*" is ordered to start exactly when the action referred to by "*label_3_11*" ends.

(6a) C_T → Temporal-Term Qualifier2 Action Action
(6b) **start at-the-end-of** *label_3_12 label_3_11*

In temporal coordinations, the non-terminals Action have to be expanded by different unique identifiers that serve as labels for basic expressions. Temporal-Term is either **start** or **end** signifying whether the start or the end of the first Action is determined by the expression. Qualifier2 is expanded by a relational expression that determines how the start (or the end) of the first Action is related to the temporal interval the second Action defines. As has already been said with respect to (3b), Qualifier2 is taken from C2IEDM's table "action-temporal-association-category-code."

Additional examples of BML rules used for formalizing orders are given in [20], Appendix A.

4.2 Beyond Orders: Requests and Reports

Besides orders, military communication also includes reports and requests. Even though requests are not mentioned as often as reports and orders, they form a relevant subclass of military messages. A military commander whose units do not possess the capability for specific tasks to be done will send a request to his superior such that the required capability will be provided. For example, military commanders may request fire support from artillery or close air support to pin down or destroy enemy forces, they may request medical support, or they may request the recovery of incapacitated equipment.

In principle, request expressions look like orders. The main difference is the relation between sender and addressee. In the case of an order, the sender is the superior of the addressee; in the case of a request, it is vice versa. Therefore, we will use the same BML rules for requests as for orders. Whether the expression is an order or a request can easily be deduced from the relationship between Tasker and Taskee. There is an additional reason to treat orders and requests alike. Under NCW conditions, the operating units may be edge organizations, and command is established by conditions (cf. [1], p. 218). Accordingly, under NCW conditions, the difference between orders and requests blur which is mirrored by their similarity in BML.

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 behaviour 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.

As this is a work in progress, the rules for reports are preliminary and will be developed further. In principle, a report consists of arbitrarily many basic report expressions (RB) as given in (7).

(7) S → RB*

In its general form, a basic report expression looks like this:

(8) RB → Verb (Executer) (Affected|Action) Where When (Why) Certainty Label (Mod)*

It still has to be decided whether the non-terminal Tasker should be included between Verb and Executer. In addition, there are three main differences to the basic order expressions (and to the basic request expression as well). First, the non-terminal Certainty is inserted. Second, the verb can denote not only a task but also an event. Third, Executer has to be handled differently from Taskee in Orders and Requests.

Certainty expands to a modality operator (a terminal symbol) to denote the certainty, the sender attributes to his report. These symbols are taken from C2IEDM's table "reporting-data-credibility-code." The corresponding rules are given in (9a)-(9d).

(9a) Certainty → *fact*
(9b) Certainty → *plausible*
(9c) Certainty → *uncertain*
(9d) Certainty → *indeterminate*

The C2IEDM provides many more attributes and corresponding values for specifying "reporting-data." However, "reporting-data" in the C2IEDM 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 C2IEDM's attributes for "reporting-data" can be used for our purpose.

The attribute "reporting-data-accuracy-code" denotes how often a fact has been reported by different sources. A value is calculated for this attribute through aggregation of multiple reports. The attribute "reporting-data-category-code" denotes the nature of a data element, e.g., whether it

is assumed, inferred, result of planning, or reported. In the case of a report, the data always is "reported".

The attributes "reporting-data-counting-indicator-code" (denoting whether the data is based on a count) and "reporting-data-timing-category-code" (denoting whether the When of the data refers to absolute or relative timing) are explicitly dealt with in the content of the report, by Executer (see below) and by When, respectively. The handling of the attributes "reporting-data-reliability-code" and "reporting-data-source-type-code" is not completely developed in our grammar at the moment. These attributes are important if the sender reports information learned from a "third party," e.g., from a prisoner of war, a refugee, a captured document, or a radar the sender unit operates. These attributes are also of importance if the sender is a device, e.g., a sensor, a UAV, or a satellite, which automatically supplies the information network with its measurement results. Then, by "source-type," the sender can denote the type of the source, and in a device's report this attribute is set to the respective value (e.g., "unattended ground sensor"). By "reliability," the sender can judge the reliability of his source, and the device's reliability can be associated to its report. At the moment, in our approach, these denotations have to be coded under Mod, but in future versions, these attributes will be included in (8) represented by their own non-terminal symbols.

The second difference in reports compared to orders and requests is the ability to report on events in addition to tasks. In this case, the verbs are from C2IEDM's table "action-event-category-code," for example, denoting natural disasters as with a flood (cf. 10). Executer and Why are omitted in this type of rule.

(10) RB → ***flood*** (Affected|Action) Where When Certainty Label (Mod)*

With respect to task verbs, like those used for basic order rules, Executer is obligatory, but may be expanded in various ways – (11a)-(11e).

(11a) Executer → Taskee
(11b) Executer → Agent Agent_Label
(11c) Executer → Agent_Label
(11d) Executer → Theme Theme_Label
(11e) Executer → Theme_Label

Taskee is for those cases in which the executing unit can be referred to by name like in orders and requests. It can be used for reports about own forces. Agent is for those cases in which the sender wants to refer to a unit by its type; Theme is for those cases in which he wants to refer by principle equipment. In order to allow anaphoric references, specific labels are added. During the processing of the report, the labels are stored in a list of discourse referents such that in anaphoric references – Executer, then, is expanded by a respective label – the object referred to can be looked up. Example (12) shows one of the rules for expanding Theme. That rule can be used for expressions like *four hostile battle tanks*.

(12) Theme → Count Hostility Equipment_type

As has already been mentioned defining rules for report expressions is a work in progress. A corpus of reports has been analysed in order to recognize what kind of linguistic phenomena can appear in reports. Although some of these phenomena are already considered in the discussion above, like modality, some others still are not, like disjunction or negation.

4.3 Example

In order to illustrate BML expressions, we will give an example in this subsection. This example is from the execution part of an order. The original order was used in the "Integrated Operational Test and Evaluation" exercise of the "Multilateral Interoperability Programme (MIP)," September 8th to 26th, in the city of Ede in the Netherlands.

This exercise order in question was released from the Multi-National Division (West) led by Spain and directed – among others — to the 13th Dutch Mechanized Brigade and the 12th Spain Cavalry Regiment. The following shows some of its content:

3. EXECUTION.

[…]

Tasks to Manoeuvre Units.

13 NL MECH BDE:
Phase 1A: Fast Tactical March to PL TULIP by or behind ROUTE DUCK.
Phase 1B: Defense in depth sector EAST, blocking penetration ALFA.
Phase 1C: Assist the rearward passage of the 12 (SP) Cavalry Regiment

[...]
12 SP CAVALRY RGT:
[...]
Phase 1C: be assisted in the rearward passage of lines through the three sectors.

[...]

In BML this would be translated into:

march MND-West(SP) M_BDE13(NL) **along** DUCK **start at** Phase1A label_3_11
defend MND-West(SP) M_BDE13(NL) **at** EAST **start nlt** Phase1B label_3_12;
block MND-West(SP) M_BDE13(NL) MIR320(BL) **at** TULIP **start nlt** Phase1B label_3_13;
assist MND-West(SP) M_BDE13(NL) label_3_57 **at** EAST **start nlt** Phase1C label_3_14;
[...]
withdraw MND-West(SP) CAV_REG12(SP) **to** EAST **start nlt** Phase1C label_3_57;
[…]

In the BML version of the order, the TASKER is the Multi-National Division West (abbreviated MND-West(SP)), and the TASKEEs are the 13th Dutch Mechanized Brigade (abbreviated M_BDE13(NL)) and the 12th Spain Cavalry Regiment (abbreviated CAV_REG12(SP)). Within the WHERE-phrases, the control features are denoted by their names DUCK, EAST, and TULIP. The Start-WHEN-phrases use the key word **start**, qualifiers from C2IEDM's table "action-task-start-qualifier-code," namely **at** and **nlt** ("not later than"), and names which denotes points in time (Phase1A, Phase1B, Phase1C). The last BML order sentence for the 13th Dutch Mechanized Brigade (***assist***) illustrates the use of a label defined elsewhere. The ***assist*** task has as its object the rearward passage (***withdraw***) of the 12th Spanish Cavalry Regiment. The ***withdraw*** received the label label_3_57, which is used in the assist.

Currently, there are no WHY terms in this example as they remain to be developed in the grammar.

The second example is for Reports:

In this instance, the 12th Spanish Cavalry Regiment 1) reports back its own position near the Dutch town Luyksgestel, and 2) later reports that some enemy force movement (30 large armoured trucks that were identified as most likely of a generic type TAM150 and nationality "Bradyland") was spotted:

1) 091200ZSEP03 by CAV_REG12(SP):
Own position at crossing Kr39 (Luyksgestel).

2) 091225ZSEP03 by CAV_REG12(SP):
Convoy (30 TAM150) passed own position at 1215 to Eindhoven.

Again in our BML this is translated into the following:

reconnaissance CAV_REG12(SP) **at** Kr39 (Luyksgestel) **start at** 091200ZSEP03 fact label_12_31;

march 30 ZB TAM150 **via** Kr39 (Luyksgestel) **to** Eindhoven
start bef 091215ZSEP03 plausible label_12_32;

ZB is the C2IEDM's code for Bradyland. In BML the sender has to put in an "adjective" between the count (30) and the equipment type (TAM150) in a theme. The "adjective" encodes the hostility (friend – neutral – hostile – unknown) or the geopolitical affiliation. The adjective is present in order to make the BML expression unambiguous. In the "real" report, the sender "forgot" to insert the affiliation information. The addressee had to infer it from the situation, something that is no problem for humans but impossible for computers.

5 Outlook

In this section, we address some methodological aspects of BML and the grammar proposed. First, we will sum up its consequences for the interaction among C2 systems. Second, we will discuss how the interoperability can further be enhanced by ontological means. Third, we will discuss possible generalizations of our approach. We will suggest the scope of such a generalization and the necessary steps to build up languages for other similar domains as well.

5.1 Consequences of using a Language for the interaction between C2 systems

With respect to the interaction among C2 systems, BML enhances the interoperability strongly, at least up into the cognitive level (cf. [1], chapter 7). In order to use BML, a system needs to be aligned to the MIP data model, the C2IEDM or in future its successor, the JC3IEDM. This adjustment is necessary in order to ensure that the information exchanged is interpreted correctly with respect to its semantics. Orders, requests, and reports exchanged via BML among the C2 systems are kept and understood as orders, request, or reports, respectively. In contrast, in the actual versions of the standard data model, military messages are not represented as a package. Instead, the content of a military message is distributed over many tables. Thus, if the exchange is handled by the MIP data replication mechanism it is very hard if not impossible to reconstruct the message itself. In our opinion it is reasonable to exchange data about the situation by the MIP data replication and use BML to exchange military messages and thus preserving the structure and principles of the applied military doctrine.

Besides, in order to use BML, it is not necessary for a system to have the MIP data replication mechanism (the MIP data exchange interface) implemented; only the semantics of the data model are needed. This is especially helpful with respect to the lowest echelons due to the restricted computational power in their hardware (PDAs etc.).

The grammar proposed is very advantageous with respect to semantic analysis and attributing meaning to expressions. It especially prohibits the generation of expressions which contradict the principles of completeness and coherence. However, this might not be enough for simulation systems and robotic forces to operate according to the intention of a commander sending a respective BML message. Thus, we presently not only specify a set of grammar rules, but also are developing an assistant system. This assistant system on the one hand has to bridge gaps in BML messages (especially orders) and on the other hand has to check for semantic consistency. The former means semantic enrichment. For example a “move” order has to be inserted between a “deploy” and a “defend” order if the respective Wheres are in different locations. The latter could trigger checks like “Does the Tasker have command and control authority over the Taskee?,” “Does the Taskee have the capability and the necessary equipment to execute the ordered task?,” and “Is the route selected in the order clear?” This enrichment, as well as the checks, will be based on an ontology for military operations [18, 19].

5.2 Considerations on the Scope of a BML Grammar

To be clear about our intent, we view a BML grammar as a key component of a more generic

task representation language. We see this more generic language as more of an operational tasking language and its grammar as more of a C2 grammar. While the semantics of a military task have unique aspects, we hypothesize that its syntax is general for a certain class of "operations" that we define as "a planned activity involving many people performing various actions" [25]. This is similar to the notion of "action" in the C2IEDM, where an action is "an activity, or the occurrence of an activity, that may utilize resources and may be focused against an objective." In operations like peacekeeping, police and fire operations, and other disaster relief operations, the efforts of the acting organizations and persons have to be coordinated like in the military context. Thus, information has to be exchanged in a network-centric fashion in these operations as well. Once again, the problems of interoperability abound and cannot be allowed to hinder the exploitation of fleeting opportunities and self-organization. An operational tasking language is again needed as well as a respective grammar. Therefore, our approach is to base such a grammar on formal Linguistic theory and design it to be applicable to the types of operation mentioned.

5.3 Domain Specialization

In order to develop a language for a specific type of operation, first, its vocabulary has to be adapted to the respective domain. In contrast to the military context, we would not be able to take an already developed data model and use its semantically well-defined terms as lexical items. However, parts of the C2IEDM, e.g., its temporal relations or its descriptions for persons, can be used for other domains as well. Other parts, e.g., terms for special fire fighting equipment, have to be defined from scratch. Unfortunately, terms for many actions, tasks as well as events, have to be defined as well. The definition of the action terms (the verbs of a more generic language) will result in new rules for basic expressions. To formulate these rules would be an important second step towards a C2 grammar for another domain. Fortunately, the rules will follow the general forms which are already set – (2a) for orders and requests and (8) for reports. In addition, the rules that will describe the constituents, e.g., (3a) to (4d), can be adopted one to one from our BML grammar. In summary, the development of a specific tasking language in another domain on the basis of our BML is mainly the task to define the vocabulary, another reason why the development of a BML grammar is of great value.

5.4 Future Work

The grammar presented is an initial step toward a complete C2 Grammar that is general enough for multiple domains, but can be specialized as required. Many difficult research tasks remain as we have already described. Near term tasks will be to develop several test cases in very well-defined domains for evaluation. Also, in the area of linguistics, modality, negation and disjunction remain as research topics.

6 Acknowledgements

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. This work could not have been performed without the intellectual contributions from these individuals.

7 References

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

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

3) Bresnan, J., *Lexical-Functional Syntax*. Malden, MA: Blackwell, 2001.

4) 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, 2001.

5) CCSIL Message Content Definitions, Salisbury, M., "Command and Control Simulation Interface Language (CCSIL): Status Update," Twelfth Workshop on Standards for the Interoperability of Defense Simulations, 1995 (http://ms.ie.org/cfor/diswg9503/diswg9503.pdf)

6) Chomsky, N., *Syntactic Structure*. The Hague, NL: Mouton, 1957.

7) Fillmore, C.J., "The Case for Case," In: Bach, E. and Harms, R.T. (Eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehart and Winston, 1968.

8) FrameNet: http://framenet.icsi.berkeley.edu/, 2006.

9) Galvin, K., "Does the United Kingdom need a Battlespace Management Language?," Paper 04F-SIW-051, Fall Simulation Interoperability Workshop, September 2004.

10) Gazdar, G. and Mellish, C., *Natural Language Processing in PROLOG: An Introduction to Computational Linguistics*. Wokingham, UK: Addison-Wesley, 1989.

11) Gruber, J.S., *Lexical Structures in Syntax and Semantics*. Amsterdam, NL: North Holland, 1976.

12) Hieb, M.R., Tolk, A., Sudnikovich, W.P., and Pullen, J.M., "Developing Extensible Battle Management Language to Enable Coalition Interoperability," Paper 04E-SIW-064, European Simulation Interoperability Workshop, June 2004.

13) Jackendoff, R.S., *Semantic Structures*. Cambridge, MA: MIT Press, 1990.

14) Kaplan, R.M. and Bresnan, J., "Lexical-Functional Grammar: A formal system for grammatical representation," In: Bresnan, J. (Ed.), *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press, 1982. Reprinted in: Dalrymple, M.,

Kaplan, R.M., and Maxwell III, J.T. (Eds.), *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI, 1995.

15) Levelt, W.J.M., *Speaking: From Intention to Articulation*. Cambridge, MA: MIT Press, 1989.

16) Mayk, I., Klose, D., Chan, A., Mai, M., and Negaran, H., "Technical and Operational Design, Implementation and Execution Results for SINCE Experimentation 1," 10th International Command and Control Research and Technology Symposium, Tysons Corner, VA, June 2005.

17) Reich, P.A., "The finiteness of natural language," *Language, 45* (1969), 832-843.

18) Schade, U., "Towards an Ontology for Army Battle C2 Systems," In: *Proceedings of the 8th ICCRTS, June 17-19, 2003*. National Defense University, Washington, DC, 2003.

19) Schade, U., "Towards a higher level of interoperability: Ontology components for command and control systems," In: *Proceedings of the NATO R.T.O. IST-Panel Symposium on Coalition C4ISR Architectures and Information Exchange Capabilities*. Den Haag, 2004.

20) Schade, U. and Hieb, M.R., "Formalizing Battle Management Language: A Grammar for Specifying Orders," Paper 06S-SIW-068, Spring Simulation Interoperability Workshop, April 2006.

21) Sells, P., *Lectures on Contemporary Syntactic Theories (= CSLI Lecture Notes 3).* Stanford, CA: CSLI, 1985.

22) Sowa, J.F., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks and Cole, 2000.

23) Tolk, A., Galvin, K., Hieb, M. R., and Khimeche, L., "Coalition Battle Management Language," Paper 04F-SIW-103, Simulation Interoperability Standards Organization, Fall Simulation Interoperability Workshop, Orlando, FL, September 2004.

24) Tolk, A., Hieb, M. R., Galvin, K., and Khimeche, L., "Merging National Battle Management Language Initiatives for NATO Projects," Paper 12 in Proceedings of the RTA/MSG Conference on "M&S to address NATO's new and existing Military Requirements," RTO-MP-123, Koblenz, Germany, October 2004.

25) WordNet, English Dictionary, http://www.wordreference.com, 2006