THE INTERNATIONAL

C2 JOURNAL

VOLUME 1, NUMBER 2, 2007

SPECIAL ISSUE

Decision Support for Network-Centric Command and Control

GUEST EDITOR

Sandeep Mulgund
The MITRE Corporation

Improving Planning and Replanning: Using a Formal Grammar to Automate Processing of Command and Control Information for Decision Support

Ulrich Schade
Michael R. Hieb

THE INTERNATIONAL C2 JOURNAL

The Editorial Board

Associate Editors

About the Journal

The International C2 Journal was created in 2006 at the urging of an international group of command and control professionals including individuals from academia, industry, government, and the military. The Command and Control Research Program (CCRP, of the U.S. Office of the Assistant Secretary of Defense for Networks and Information Integration, or OASD-NII) responded to this need by bringing together interested professionals to shape the purpose and guide the execution of such a journal. Today, the Journal is overseen by an Editorial Board comprising representatives from many nations.

Opinions, conclusions, and recommendations expressed or implied within are solely those of the authors. They do not necessarily represent the views of the Department of Defense, or any other U.S. Government agency.

Rights and Permissions: All articles published in the International C2 Journal remain the intellectual property of the authors and may not be distributed or sold without the express written consent of the authors.

For more information

Visit us online at: www.dodccrp.org

Contact our staff at: publications@dodccrp.org

Improving Planning and Replanning: Using a Formal Grammar to Automate Processing of Command and Control Information for Decision Support

Ulrich Schade (FGAN-FKIE, Germany)
Michael R. Hieb (George Mason University, USA)

Abstract

The face of war has changed as asymmetric warfare and operations other than war have become common in today's military environment. Troops are well connected in an information network and can therefore operate in a less hierarchical manner. The shift to network-centric operations (NCO) enables, and even calls for, rapid information exchange and automated decision support. Simulation systems suggest themselves as tools for decision support and in developing plans. However, it is currently difficult to communicate to simulated units in a natural manner, i.e., by conveying orders and the command intent and by receiving requests and reports.

In this paper we present the Command and Control Lexical Grammar (C2LG), which defines a formal, unambiguous, but also highly expressive language. This language facilitates and supports military communication including both traditional orders and reports, as well as more innovative representations supporting simulations. The language's formal properties support automatic processing. It therefore is ideal not only for communication among command and control (C2) applications and their users, but also to communicate with simulated units. The C2LG is informed by prior work in the model-

ling and simulation domain with operations orders and air tasking orders so that it can be reasoned on by simulation agents.

Introduction

In network-centric operations, command and control information needs to be communicated easily and rapidly. When planning an operation, improved collaboration can greatly assist the process and minimize the time required when humans are involved. However, while humans make decisions there is an increasing use of automated decision support applications. While these applications span a wide range of use, simulations are invariably used to predict and project the outcomes of future activities (Matthews and Davies 2003; Tolk and Kunde 2003).

Simulations used to support decisions made in planning and operations currently require (1) long lead times to prepare the simulation for specific scenarios and (2) the use of significant resources to both comprehend and make sense of situations, and to process plans for projecting them. One of the key technology gaps is the difficulty of conveying C2 information (such as orders) to simulations and getting actionable results from them (Carey et al. 2001). While force-on-force engagements are accurately portrayed, the flow of C2 information is usually not well simulated. If it is, the interfacing of simulations to the planning process is difficult at best. Plans are typically in the same format as orders.

To address this technology gap of conveying C2 information, we have developed a grammar, the Command and Control Lexical Grammar (C2LG), that defines a formal language for expressing plans and orders in a manner that reduces the ambiguity found in the military message formats commonly used: textual, XML-based, and binary (Schade and Hieb 2006a). This language is also designed to use minimal bandwidth and to support distributed and network-centric operations.

A new approach to characterizing C2 orders, known as Battle Management Language (BML) (Kleiner et al. 1998; Carey et al. 2001; Hieb et al. 2004; Tolk et al. 2004; Sudnikovich et al. 2004), has recently been developed. An extension to BML, geoBML, deals with geospatial information (Hieb et al. 2006). Within the Simulation Interoperability Standards Organization (SISO), a Coalition BML (C-BML) Product Development Group (PDG) was formed to standardize BML (Blais et al. 2005; SISO 2006). In parallel to the C-BML SISO activities, the NATO Modeling and Simulation Group (NMSG) established a 3-year Technical Activity Program (Group MSG-048) for 2006–2009. C-BML was built on standard data models such as the Multinational Interoperability Programme's (MIP's) Joint Command, Control and Consultation Information Exchange Data Model (JC3IEDM) (MIP 2007). Our linguistic approach is part of both the SISO standardization and the NATO C-BML initiative. The resulting language (Schade and Hieb 2006a)—suitable for military communication among live forces, simulations, and robotics—expands on the initial BML work. The two grammars developed as parts of the C2LG for tasking and reporting currently use the JC3IEDM (MIP 2007) as its vocabulary, but it is general and could be used with any comprehensive C2 data standard. The grammar includes a formalization of command intent. This, although traditionally being the most difficult concept to represent, is particularly relevant for network-centric operations, given the need for rapid coordination and collaboration between geographically distributed forces (Albert and Hayes 2003; 2006).

Future C2 services that would benefit from the C2LG include automated decision aids such as course of action development and analysis tools, mission rehearsal simulations, and mission planning tools. The use of a structured language to convey orders including the command intent will facilitate higher level reasoning and better behaviours. Planning would be improved and replanning could be much more rapid given that new plans would not have to be recoded into a simulation, but could be automatically sent by altering the previous plan structure in accordance with the grammar.

The next section presents the theory taken from computational linguistics that underlies our work with C2 grammars and formal languages. We need a formal grammar and thus a formal language to ensure that the C2 expressions can be processed automatically. We then review and analyze common C2 vocabularies in the next section. Following that, we describe the syntax for conveying orders and reports, which forms the basis for the formalization of command intent as given in the following section. In the last section, we conclude with how the grammar and command intent formalisms can be used in planning and operations.

Developing a Formal C2 Language

A formal language is the set of all expressions that can be generated by a formal grammar. In general, a grammar consists of a lexicon and a set of rules. The lexicon provides the words of the language and the rules determine how to construct longer expressions (e.g., sentences) using these words. In order to specify the semantics of the language, one has to give meaning to every word of the lexicon. In addition, one has to determine how to concatenate the meanings of the words to form the meaning of an expression. In principle, this entails giving meaning not only to the words but also to the rules. For example, if the terms "two," "hostile," and "sniper" are put together by a rule to form the phrase "two hostile snipers," the respective rule has to ensure that "hostile" is treated as a modifier to "sniper," which assigns a specific value to the object referred to by "sniper," and that "two" is treated as a specifier to "hostile snipers" that provides a quantity.

Linguistics uses the term "constituent" for expressions that are part of a sentence but nevertheless form an information unit. For example, in the sentence, "the 43rd Spanish Cavalry Regiment advances to phase line Star," there are two constituents besides the verb, namely: "the 43rd Spanish Cavalry Regiment" and "to phase line Star." The first constituent refers to the executer of the action and the second provides the destination. Obviously, the sequence "Regiment advances to" does not form a constituent. Constituents fill semantic roles within a sentence (cf. Sowa 2000). In the example, the roles filled are *agent* (the initiator of the action denoted by the verb: executer) and *destination* (the spatial goal assigned to this action). Semantic roles can be seen as labels assigned to the information units. A role describes the function of the constituent in question in the context of the whole sentence.

Verbs come with a frame (cf. Fillmore 1976; FrameNet 2006). The frame determines which kind of constituents are demanded (mandatory) and which are allowed (optional) by the verb. The linguistic principle of completeness demands the occurrence of the mandatory constituents and the linguistic principle of coherence prohibits the occurrence of constituents that are not at least optional. To incorporate these principles in a formal grammar is essential to assign the correct meaning to the grammar's rules. For example, the sentence "the platoon rests towards north" does not make sense, as a constituent of type *direction* is connected to the verb "rest," although "rest" neither requires nor allows such a constituent. A direction is not consistent with the doctrinal definition of "rest," which means that a unit is stationary and inactive.

The linguistic principles mentioned, namely the use of constituents, the use of verb frames, and the principles of completeness and coherence, are ideally supported by the Lexical Functional Grammar (LFG) (Kaplan and Bresnan 1995; Bresnan 2001). Therefore, we modelled the C2LG on LFG. LFG analysis of an expression 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 (Sells 1985). Phrases, including those that form constituents, are organized within a syntactic tree. The syntactic tree is the result of a pure syntactic analysis of an input expression. In the second step, the syntactic tree is transformed into the f-structure ("f" for "functional"). This second step is indicated by the "F" in LFG, whereas the "L" points to the lexicon and the frames that are stored in it together with their verbs. The frames are exploited for the building of the f-structure that connects syntax to semantics. In short, the constituents are identified during the construction of the c-structure, whereas the construction of the f-structure maps them to semantic roles.

F-structures are represented as feature-value matrices. This representation simplifies the transformation into XML representations. It also allows for further processing by unification, a standard algorithm in the field of computational linguistics (Shieber 1986). For example, order expressions can be enhanced by a unification-based process based on an ontology before conveying them to simulated units in order to add parameters that the simulation system needs for the correct interpretation of the order.

Using Standard Vocabularies

With any syntax, there must also be common semantics for comprehension. The C2LG is designed so that it can be used with various vocabularies depending upon the specific domain supported. For example, the vocabulary for air operations is quite different than the vocabulary for peacekeeping. In this section, we look at some of the current and emerging lexicons appropriate for military operations.

The MIP has already produced semantics for C2 suitable for coalition operations as documented in the JC3IEDM (MIP 2007). The MIP terms form an excellent lexicon for our purpose. The terms not only possess definitions, and thus semantics, but are also agreed upon by NATO's nations and has been proposed as a backbone for the information exchange between NATO's Land Command and Control Information Services and the U.S. Maneuver Control System, version 6.4, in Afghanistan (Christman and Postal 2006). The JC3IEDM terms form a comprehensive standard vocabulary that can be used to describe the information to be exchanged in the command and control domain. Even more, the MIP approach is generic and not limited to a special level of command, force category, etc.

With the advent of the JC3IEDM defining standard semantics to C2 terms on the one hand and providing a data exchange mechanism on the other, the additional effort of developing a language may be thought to be unnecessary. However, Schade and Hieb (2006a) presented an analysis on why relying solely upon a data model is insufficient for military communication.

The Cursor-on-Target (CoT) data model defines an XML data schema for exchanging time-sensitive information on the positions of moving objects—"what," "when," and "where" information. The CoT data strategy is based on a terse XML schema and a set of sub-schema extensions and is designed for exchanging information between bandwidth-limited hardware (Konstantopoulos and Johnston 2006). In principle, our language can be seen as a generalization of Cursor-on-Target expressions. The main difference is that CoT uses its own data model optimized for targeting whereas our language is based on the NATO standard JC3IEDM. In other aspects, however, the languages have a great deal in common. CoT expressions are exchanged in XML, and so are the expressions of our language. The XML format naturally derives from LFG's f-structures because these structures are attribute-value matrices. The attributes of the matrices form the XML tags, and the values define their contents. Even more, the attributes (the tags) derive from the linguistic version of "what," "where," and "when" (Schade and Hieb 2006b; 2007) that also underlie the CoT tags. However, since linguistic principles are taken into account in the definition of C2LG's attributes, our language expressions can be semantically interpreted by systems. This allows the generation of expressions that cover a larger part of military communication, among it the formulation of command intent, without losing the property of automatic processability.

There are two other distinctive initiatives that are being developed as network-centric data models. The first is the global strike domain awareness Community of Interest (COI) data schema. This XML schema is being developed to support Joint Strike operations. It is based on a geospatial standard, the Geography Markup Language

(GML), and places the current representation of tasks and objects under GML abstract features and objects while using the location semantics of GML. As with CoT, the current implementation of the schema is narrow, but it could have broad applicability as it is based on a broader standard (GML).

The second initiative is the Universal Core Data Model (UCDM). This initiative is just starting and has not published a schema yet. It aims to apply to information sharing and interchange, as opposed to sharing static data. It also intends to leverage commercial standards outside the DoD, such as GML. It will develop a universal core of the minimally necessary concepts for supporting government agencies and military operations. The UCDM will most likely need to be substantially augmented to serve as a lexicon for the C2LG.

All of the schemas presented above are compatible with the C2LG we present below, as we are defining a generic syntax that can use various lexicons. However, the lexicon may need to be enhanced if it cannot describe the necessary concepts unambiguously in the language.

The C2 Lexical Grammar: Orders and Reports

As has already been mentioned, we have developed C2LG grammars for tasking (Schade and Hieb 2006a; 2006b) and reporting (Schade and Hieb 2006a; 2007) following the linguistic principles given in the preceding section. The tasking grammar and the reporting grammar together with the formalization of the command intent (Hieb and Schade 2007) form the C2LG. The set of the expressions that can be generated by applying its lexicon and its rules builds a language for military communication. In this section, we provide an overview of the most important aspects of the grammar. We begin with ordering, present how to express reports, and finish with discussing how to formally express a command intent.

We use the attributes and the values provided by the standard data model JC3IEDM as lexicon elements. Since the lexicon is grounded in the JC3IEDM, the description of the C2LG is mainly a description of the rules. C2LG uses c-structure rules to describe the concatenation of words to form constituents and basic expressions (sentences) for orders and reports. These c-structures rules—examples are given below in (1) to (3) and in (5) to (10)—by themselves constitute a context-free phrase structure grammar. In the LFG, c-structures are transformed into f-structures that represent syntactical information in feature-value matrices. The syntactical features used in f-structures like "subject" or "object" abstract from more language specific designators like "noun phrase." The abstraction serves as a bridge-building function of the f-structure, a bridge to connect expressions to semantics. BML, however, is not a natural language but a constructed one. There is no need to build in grammatical features like "grammatical gender" or "case." Even more, syntax-semantics mismatches marking grammars for natural languages (Sadock 2003) can be avoided. To do this, C2LG's c-structures (see below) use designators for thematic roles (Sowa 2000) as non-terminals. As a consequence, C2LG's f-structure uses these designators instead of abstract syntactic categories for parts of sentences like "subject" or "object" and thus can be seen as a semantic representation (an argument structure in the terms of LFG). The feature-value matrices assigned to the thematic roles in C2LG's f-structure only consist of one feature-value pair in which the feature is "predicator" (PRED). By use of Web services developed under the umbrella of the Joint Battle Management Language (JBML) project (Pullen et al. 2007) the matrices can be fleshed out. The Web services take the PRED value, most often the name of an object, and look it up in the underlying JC3IEDM database. For example, in the case of a "taskee" (the executer or agent of a task), the PRED value is the name of a unit, an object-item of the database. The database has information about the size, type, affiliation etc. of this unit. This information is inserted. The resulting structure specifies how the semantics will be interpreted by the simulation.

As can be seen from the discussion above, the most important part of the grammar is the set of its c-structure rules. In the following section, we provide an overview over these rules. We start with the rules that allow the generation of orders.

The format of orders is defined by the NATO standard STANAG 2014 (NATO 2000) "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. 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." The tasking grammar (Schade and Hieb 2006a) scope covers Section 3, execution, which consists of the command intent, the assignment of single specific tasks to specific units, as well as the giving of details of coordination. Therefore, the basic rule of the tasking grammar is:

(1) S → CI OB* C_Sp* C_T*

This rule means that a tasking expression consists of the command intent (indicated by CI), basic order expressions to assign tasks to units (OB), spatial coordination (C_Sp), and temporal coordination (C_T). The asterisk indicates that arbitrarily many of the respective expressions can be concatenated together.

According to the linguistic principles as given above, we define basic order expressions as composed of a verb and its frame. The verb denotes a task. For the tasking grammar, tasking verbs are taken from JC3IEDM's table "action-task-activity-code" (MIP 2007). Thus, the rules to expand OB have the general form as given in (2a). (2b) and (2c) give examples for the tasks "advance" and "defend," respectively.

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

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

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

Tasker is to be expanded by the name of the one who gives the order. **Taskee** is to be expanded by the name of the unit that is ordered to execute the task. **Start-When** and **End-When** are to be expanded by temporal phrases expressing when the execution of the task has to start and when it has to be finished. **End-When** is optional as indicated by the parentheses. **Tasker**, **Taskee**, **Start-When**, and **End-When** appear in each basic order rule.

Affected in (2a) has to be a term in the expression if someone, e.g., the enemy, will be directly affected by the task; in linguistic terms this is called the *patient* (Sowa 2000). Whether **Affected** is part of a rule depends on the tasking verb. For example, it is there in the case of ***attack*** or ***defend*** because the executing unit is tasked to attack the enemy or to defend against the enemy. It is not there in the case of ***advance***. The tasking verbs come with frames that express which kind of constituents are required, e.g., a constituent of type **Affected**. This enforces the principles of completeness and coherence. **Action** is similar to **Affected**. It only appears if the task affects an action, as a task of type ***assist*** does; the unit is tasked to assist the execution of another task by another unit. In addition, the type of the **Where** is also determined by the verb. It is currently an **At-Where** or a **Route-Where**. An **At-Where** denotes a location, and a **Route-Where** a path to a location. A **Route-Where** can be expanded to more complex concatenations of constituents as in "from **LocationA** to **LocationD** via **LocationB** and **LocationC**."

A basic rule ends with **Why**, **Label**, and the optional **Mod**. **Label** is a unique identifier for its task. By this identifier the task can be referred to in other expressions, especially in temporal coordinations. The optional **Mod** (for modifier) is a wild card that represents additional information that can be used to describe a particular task, e.g., *formation* to specify a particular formation for an advance, or *manner* to express, for example, whether the task in question has to be completed as fast as possible or more slowly, without taking any risks. Modifiers are particularly important for decision support.

Why represents a reason why the task specified by the rule is ordered: the mission's purpose. FM 3-90 (USA 2001) offers a list of purpose verbs (Pverbs) that can be used to express the **Why**. (Examples are "divert, enable, deceive, deny, prevent, open, envelope, surprise, cause, protect, allow, create, influence, and support"). From a linguistic perspective, the Pverbs can be used with an argument that is an object, like "in order to deceive the enemy." Other purposes delineated are (1) those that cause a state and (2) those that need another task as argument, like "in order to enable task DELTA." In accordance to this differentiation, we expand the **Why** as follows:

(3a) Why → *in-order-to* PVerb (ObjectLabel)

(3b) Why → *in-order-to* cause EndStateLabel

(3c) Why → *in-order-to* enable TaskLabel

In these cases, the purpose of a task is to influence another task, and the **Why** in (3c) can be used to make explicit the dependencies between the tasks of a course of action. For example, if a course of action is divided into three phases, and a unit has to execute Task 1 in Phase 1, Task 2 in Phase 2, and Task 3 in Phase 3, normally the completion of Task 1 is a precondition to Task 2, and the completion of Task 2 is a precondition of Task 3. Therefore, the **Why** of Task 1 is "in order to enable Task 2," and the **Why** of Task 2 is "in order to enable Task 3."

An example order expression is given in (4b) that has been generated using, among others, rules (2b) and (3a) to express (4a).

(4a) Multi-National Division (West) commands 13th Dutch Mechanized Brigade to perform a Tactical March to PL TULIP by or behind ROUTE DUCK.

(4b) advance MND-West M_BDE13(NL)
along DUCK start at Phase1A
in-order-to surprise label_3_11;

With respect to reports, the C2LG says that reports consist of arbitrarily many basic reporting expressions (RB):

(5) S → RB*

The general form of a basic reporting expression depends on whether the report is about military operations (task report), events (event report), or status (status report). The respective rule forms are given in (6a) to (6c).

(6a) RB → Task-Report Verb Executer (Affected|Action) Where When (Why) Certainty Label (Mod)*

(6b) RB → Event-Report EVerb (Affected|Action) Where When Certainty Label (Mod)*

(6c) RB → Status-Report Hostility Regarding (Identification Status-Value) Where When Certainty Label (Mod)*

Rule forms (6a) and (6b) are quite similar to the rule form for basic order expression as given in (2a). The differences are as follows: Neither (6a) nor (6b) has a Tasker. For (6a), this is because the reporter may not know the unit that has ordered the task he is reporting on, as with an action performed by the enemy. For (6b), this is because events happen and it does not make sense to say they are "commanded" by an organization. Rule form (6a) has a generalized Taskee named Executer in order to allow constituents like "four hostile snipers." (6b) has no Executer; it uses verbs (EVerb) from another JC3IEDM table "action-event-category-code" that contains expressions like "flood" or "peace conference" as verbs.

Rule form (6c) uses Regarding instead of a verb to determine what kind of status is reported. Status reports can be given about the operational status of a unit (by using the key word status-general in

Regarding), about the status of a unit's personnel (**status-person**) and about the status of a unit's materiel (**status-materiel**). In addition, the position of a unit can be reported (**position**). All report forms include **Certainty** to specify the certainty of the report.

The rules for tasking and reporting provide the basis for formalizing command intent (Hieb and Schade 2007) as discussed below.

The C2 Lexical Grammar: Formalizing Command Intent

Network-centric operations are particularly suited to a command style based on "mission command" ("Auftragstaktik") (Storr 2003; Alberts and Hayes 2006). Its key element is the command intent. Forces must operate agilely and in the spirit of the command intent even if this contradicts explicit tasking orders. NCO has increased the importance of command intent.

Obviously, a military professional views the process of communicating command intent as a special prerogative. Many view the decisionmaking process of a military commander as more art than science. There are many nuances in the expression of command intent that are difficult or impossible to convey in a formal language. These include qualifiers modifying tasks or purposes as well as emotional cues.

As an example, there are many stories where a commander will not give written orders to a subordinate for a particularly difficult mission. Instead, the commander may feel that his command intent may only be conveyed personally, as when a commander will travel to a subordinate to convey a task such as "defend to the last man." In these cases, some of the command intent is conveyed "between the lines" and not explicitly stated. In a coalition force, those subordinates who do not speak the commander's native language will probably not catch the nuances mentioned above. The illocutionary force (Austin 1962) of the command thus is not conveyed to these coalition officers. This is even more true for simulated forces.

In network-centric operations, command intent must be communicated to a potentially wide range of recipients: coalition officers, automated situational awareness systems, robotic reconnaissance units, and simulated forces within a decision support tool. The command intent in these cases must be as clear as possible, without ambiguity, and understandable. "Clear" means that the expression is concise and conforms to the doctrinal guidance given for the C2 process. "Without ambiguity" means that there is an explicit structure that the command intent can be put into and then parsed out of. It also means that only one clear and definite outcome results from the parsing. "Understandable" means that the semantics used in the command intent are available and common to all of the recipients.

Our approach has been an initial structuring of the command intent to meet the requirements of automated systems. Hieb and Schade (2007) showed how the command intent can be broken into doctrinally derived elements that can then be represented by elements already defined for tasking and reporting.

In order to create the grammar rules for command intent, its doctrine as stated in the U.S. Field Manual (FM-5) is used. Accordingly, command intent is composed of three terms: End State, Key Tasks, and Expanded Purpose. Therefore the basic rule for command intent is (7).

(7) CI → [Expanded Purpose] [Key Tasks] [End State]

The End State describes the resulting situation that is achieved when the mission is accomplished. Therefore, we modeled the End State as it would be reported at the successful conclusion of the mission. It can be represented by a combination of basic report expressions (10). The Key Tasks are tasks and conditions that are essential to accomplishing the mission; thus Key Tasks can be formulated as a sequence of basic orders and basic reports (9). The Expanded Purpose is similar to the End State, but expresses more general aspects of the resulting situation.

(8) [Expanded Purpose] → RB*

(9) [Key Tasks] → (OB|RB)*

(10) [End State] → RB*

As given by (8) to (10), the components of command intent are formalized by the use of basic order expressions and basic report expressions. There are however some specific aspects that have to be taken into account as has been discussed by Hieb and Schade (2007). Instead of broadening the discussion, we would like to provide a short example in order to illustrate the value of a formal command intent as input for a decision support system.

Let us assume that a relief convoy is carrying food to a refugee camp in a foreign country. The troops are stationed in Camp "Landing" in the city of "Beeden." The commander of Camp Landing, "CLC," wants to order a unit "Bravo" to escort the relief convoy to "Beeden Refugee Camp." The respective order is run in a decision support system to evaluate the order in general and the size of Bravo in particular.

The BML expression of the order's command intent would look like the following:

(11) [Expanded Purpose]
Task-Report *resupply* Relief-Convoy Red-Cross at Beeden Refugee Camp start at TP3 RPTFCT label-ep1;

(12) [Key Tasks]
escort CLC Bravo Relief-Convoy from Camp Landing to Beeden Refugee Camp start at TP1 in-order-to protect food-supplies label-kt-1;

(13) [End State]
Status-Report *friend position* Relief-Convoy at Beeden Refugee Camp at TP2 RPTFCT label-es-4;

The inclusion of the formalized command intent in the order sent to the decision support system ensures that the key aspects of the order are taken into account. In our example, the key aspect is the secure arrival of the relief convoy at the refugee camp. If the (simulated) escort runs into fire, it should prioritize a disengage over a fight even if the fight might result in the defeat of the opponent—an outcome that the system might favour without the additional information given by the formalized command intent.

Enhancing Planning and Replanning

In this paper, we describe a formalism, the C2LG, that can be used to structure C2 information. This formalism covers a grammar for orders, a grammar for reports, and a representation of command intent. The C2LG grammar for orders has already been used in the JBML (Pullen et al. 2007). In this work, JBML Web services were developed in order to send orders from existing C2 systems (for land, maritime, and air operations) to simulations. The C2LG was used to structure the various order formats to support the development of the underlying Web services. This initial use of the C2LG shows that it can be implemented in a Web-services architecture.

The C2LG has been designed to support the development of future network-centric applications. It addresses the key technology gap of structuring C2 information to preserve the deep meaning of orders and plans, while still rendering this information amenable to automated processes. The alternative is to have humans interpret each plan and order developed in order to put them into the formats and representations specific to each C2 application that is used. This alternative means a loss of time and—without the formal determination of common semantics—a susceptibility for specific interpretation errors made by the humans in the loop, especially in coalition operations in which people have different training backgrounds and different native languages. Meaning will be lost in general and, in particular, command intent will not be preserved.

Simulations are commonly used to project plans into the future to determine their effects. There are many efforts to use BML for this purpose. An assessment of using the C2LG is given by Borgers et al. (2007). The benefits of this approach go beyond a reduction in the number of people that control the simulations and a reduction in the time needed to run the simulation analysis. Running the simulation on C2LG orders, the simulations have metarules embedded to send C2LG reports back to the users whenever a task or subtask is accomplished or, conversely, whenever a simulated unit is no longer able to accomplish its tasks in time. The communication with the simulation thus is much closer to the real communication. The human decisionmakers can both better understand what is going on in the simulation and see the results earlier than they would have. Therefore, they can more easily interrupt the simulation execution of plans that are showing undesired results and concentrate on those plans that are promising in order to optimize them.

When planning complex operations, various processes have been developed to structure the workflow. Many of these processes are sequential due to the time and effort involved. With the advent of network-centric operations, it is possible that these processes can be restructured, but in the area of decision support tools, and particularly simulations, there need to be new technologies developed that can rapidly predict many types of outcomes, not only kinetic, to determine the most effective plan. The C2LG is a fundamental enabler for both simulations and other automated decision support tools that require detailed C2 information.

Acknowledgments

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 of Excellence in C4I at George Mason University.

References

Alberts, D.S., and R.E. Hayes. 2003. *Power to the edge*. Washington: CCRP.

Alberts, D.S., and R.E. Hayes. 2006. *Understanding command and control*. Washington: CCRP.

Austin, J.L. 1962. *How to do things with words*. Oxford: Clarendon Press.

Blais, C., M.R. Hieb, and K. Galvin. 2005. Coalition battle management language (C-BML) study group report. 05F-SIW-041. Paper presented at the 2005 Fall Simulation Interoperability Workshop, September 2005, in Orlando, FL.

Borgers, E., W. Huiskamp, N. de Reus, and J. Voogd. 2007. Research and development towards application of the military scenario definition language and coalition battle management language in The Netherlands. 07S-SIW-025. Paper presented at the 2007 Spring Simulation Interoperability Workshop, March 2007, in Norfolk, VA.

Bresnan, J. 2001. *Lexical-functional syntax*. Malden: Blackwell.

Carey, S., M. Kleiner, M.R. Hieb, and R. Brown. 2001. Standardizing battle management language: A vital move towards the army transformation. 01F-SIW-067. Paper presented at the 2001 Fall Simulation Interoperability Workshop, September 2001, in Orlando, FL.

Christman, G.J., and M. Postal. 2006. Coalition interoperability: A modeled approach. Paper presented at the 11th ICCRTS, September 2006, in Cambridge, UK.

Fillmore, C.J. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Science* 280: 20–32.

FrameNet. 2006. http://framenet.icsi.berkeley.edu/

Hieb, M.R., A. Tolk, W.P. Sudnikovich, and J.M. Pullen. 2004. Developing extensible battle management language to enable coalition interoperability. 04E-SIW-064. Paper presented at the 2004 European Simulation Interoperability Workshop, June 2004, in Edinburgh, UK.

Hieb, M.R., and U. Schade. 2007. Formalizing command intent through development of a command and control grammar. Paper presented at the 12th ICCRTS, June 2007, in Newport, RI.

Hieb, M.R., M. Powers, M. Kleiner, and J.M. Pullen. 2006. A geospatial battle management language (geoBML) for terrain reasoning. Paper presented at the 11th ICCRTS, September 2006, in Cambridge, UK.

Kaplan, R.M., and J. Bresnan. 1995. Lexical-functional grammar: A formal system for grammatical representation. In *Formal issues in lexical-functional grammar*, ed. M. Dalrymple, R.M. Kaplan, and J.T. Maxwell III. Stanford: CSLI. Originally published in *The mental representation of grammatical relations*, ed. J. Bresnan. (Cambridge: MIT Press, 1982).

Kleiner, M.S., S.A. Carey, and J. Beach. 1998. Communicating mission-type orders to virtual commanders. Paper presented at the 1998 Winter Simulation Conference, December 1998, in Washington, DC.

Konstantopoulos, D., and J. Johnston. 2006. Data schemas for net-centric situational awareness. Paper presented at the 2006 CCRTS, June 2006, in San Diego, CA.

Matthews, K., and M. Davies. 2003. Simulations for joint military operations planning. Paper presented at the Simulation Technology and Training (SimTecT) Conference, 2003, in Adelaide, Australia.

MIP. 2007. http://www.mip-site.org

NATO Military Agency for Standardization, STANAG 2014. 2000. *Formats for orders and designation of timings, locations, and boundaries*. 9th ed. Brussels.

Pullen, J.M., M.R. Hieb, S. Levine, A. Tolk, and C. Blais. 2007. Joint battle management language (JBML): US contribution to the coalition battle management language product development group and the NATO MSG-048 technical activity. 07E-SIW-029. Paper presented at the 2007 European Simulation Interoperability Workshop, June 2007, in Genoa, Italy.

Sadock, J.M. 2003. Mismatches in automomous modular versus derivational grammars. In *Mismatch: Form-function incongruity and the architecture of grammar*, ed. E.J. Francis and L.A. Michaelis. 333–354. Stanford: CSLI.

Schade, U., and M.R. Hieb. 2006a. Development of formal grammars to support coalition command and control: A battle management language for orders, requests, and reports. Paper presented at the 11th ICCRTS, September 2006, in Cambridge, UK.

Schade, U., and M.R. Hieb. 2006b. Formalizing battle management language: A grammar for specifying orders. 06S-SIW-068. Paper presented at the 2006 Spring Simulation Interoperability Workshop, April 2006, in Huntsville, AL.

Schade, U., and M.R. Hieb. 2007. Battle management language: A grammar for specifying reports. 07S-SIW-036. Paper presented at the 2007 Spring Simulation Interoperability Workshop, March 2007, in Norfolk, VA.

Sells, P. 1985. *Lectures on contemporary syntactic theories (CSLI lecture notes 3)*. Stanford: CSLI.

Shieber, S.M. 1986. *An introduction to unification-based approaches to grammar (CSLI lecture notes 4)*. Stanford: CSLI.

SISO. 2006. *Coalition battle management language (C-BML) study group report*. SISO-REF-016-2006-V1.0.

Sowa, J.F. 2000. *Knowledge representation: Logical, philosophical, and computational foundations*. Pacific Grove: Brooks and Cole.

Storr, J. 2003. A command philosophy for the information age: The continuing relevance of mission command. In *The big issue: Command and combat in the information age*, ed. D. Potts. Washington: CCRP.

Sudnikovich, W.P., J.M. Pullen, M.S. Kleiner, and S.A. Carey. 2004. Extensible battle management language as a transformation enabler. *Simulation* 80: 669–680.

Tolk, A., and D. Kunde. 2003. Decision support systems in defence. In *Innovations in decision support systems*, ed. G. Tonfoni and L. Jain. 175–205. Adelaide: AKI.

Tolk, A., M.R. Hieb, K. Galvin, and L. Khimeche. 2004. Merging national battle management language initiatives for NATO projects. Paper presented at the RTA/MSG conference on M&S to address NATO's new and existing Military Requirements, RTO-MP-123, October 2004, in Koblenz, Germany.

USA. 2001. *Field manual 3-90 (FM 3-90), tactics.*