

FRAUNHOFER-INSTITUT FÜR KOMMUNIKATION, INFORMATIONSVERARBEITUNG UND ERGONOMIE FKIE

### **Technischer Bericht ITF/2010/02**

### **Command and Control Lexical Grammar** (C2LG) Specification

Prof. Dr. Ulrich Schade Prof. Dr. Michael R. Hieb (GMU) Dr. Miloslaw Frey Kellyn Rein

Juli 2010

© Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie 2010

Alle Rechte vorbehalten. Ohne ausdrückliche schriftliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren zu reproduzieren oder in eine für Maschinen, insbesondere Datenverarbeitungsanlagen, verwendbare Sprache zu übertragen. Dasselbe gilt für das Recht der öffentlichen Wiedergabe.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Die Veröffentlichungen in der Berichtsreihe des FKIE können bezogen werden über:

Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE Neuenahrer Strasse 20 53343 Wachtberg Germany

Telefon: +49 (0) 228/9435-0 Telefax: +49 (0) 228/9435-685 E-Mail: info@fkie.fraunhofer.de Internet: www.fkie.fraunhofer.de

### **Table of Contents**

1	The Li	nguistic Principles behind C2LG	2
	1.1 Co	onstituency and Thematic Roles	3
	1.2 Tł	ne Correct and Appropriate Type of Grammar	4
		exicality and the Principles of Completeness and Coherence	
		eech Acts	
2	-	n	
3	Orders	in C2LG	8
4	The Bu	ilding Blocks	11
		ctions – The "What"	
		enoting Organisations – The "Who"	
		pressing Spatial Aspects	
	4.3.1		
	4.3.2		
	4.3.3	1 1	
	4.4 Ex	pressing Temporal Aspects	
	4.4.1	C2LG Rules for Specific Temporal Constituents	
	4.4.2	Example of a Specific Temporal Constituent	
	4.4.3	C2LG Rules for Conditional Temporal Constituents	
	4.4.4	An Example of a Conditional Temporal Constituent	
		pressing Modifiers	
	4.5.1	1 0	
	4.5.2		
		pressing Purpose	
	4.6.1		
	4.6.2		
5		ts	
6	-		
Ũ	-	ne "(Identifaction Status-Value)" Pair	
	6.1.1		
	6.1.2		
		enoting the "Who" in Reports	
	6.2.1	C I	
		Examples for Denoting Organisations	
		pressing Spatial Aspects in Reports	
		pressing Temporal Aspects in Reports	
		pressing Certainty	
	6.5.1	C2LG Rules for Expressing Certainty	
	6.5.2	Examples of Expressing Certainty	
7		and Intent	
8		Expressed as BML Schemata	
9		nces	
-		: List of Tasking Verbs and their Rules	

This document presents a specification for a Command and Control (C2) grammar for Command and Control of Organizations. This grammar is restricted to operations of Organizations through time and space to achieve a specified intent. We call this restricted grammar the C2 Lexical Grammar (C2LG).

The specification can properly be viewed as a "core" that would be expanded in actual usage with additional domain knowledge. However, the specification as given is sufficient to communicate and model a wide variety of military and civilian organizational operations. The specification gives usage guidance and also gives directions on expanding the grammar. It is the hope of the specification's authors that the approach and core principles herein described can be used to formulate more powerful and abstract languages than the ones used currently.

The C2LG is a particular grammar that can be used to support military operations with the particular domain knowledge given. It relies upon semantics taken from an international standardization effort – the Multinational Interoperability Programme (MIP). This MIP standard is specific to the military domain and thus the C2LG is also specific to the military domain. However, as noted above, other grammars and languages can be developed using a lexical grammar approach and specified in separate specifications.

This document is organized as follows: chapter 1 provides an overview of the linguistic principles that were used to develop the C2LG. In addition, chapter 1 explains in detail the reasoning behind the grammar and the language. However, in order to use the language and to apply this specification, the knowledge of this first chapter is not assumed. Following the description of the notations and conventions used in this document (Chapter 2), the core information about C2LG, that is, the "real specification", is presented in chapters 3 to 7. Chapter 3 provides the fundamental rules used for constructing the outline of orders while Chapter 4 deals with the building blocks required to complete these orders. Chapter 5 deals with requests, and Chapter 6 with reports. Much of what is needed to generate requests and reports also is used to generate orders. Therefore, these aspects and rules will not be repeated for each type of communication. Instead, Chapters 5 and 6 focus on what is unique to requests and reports, respectively. The core specification is completed with a formulation as to how to express command intent in C2LG (Chapter 7). All core chapters include examples of C2LG expressions. These examples may guide readers interested in expressing orders, requests, and reports in C2LG themselves. The document ends with an explanation of the grammar's correlation to XML schemata (chapter 8). Last but not least, Chapter 9 provides the list of references.

### **1** The Linguistic Principles behind C2LG

In linguistics, a (formal) language is defined by a (formal) grammar (Chomsky, 1957). In short, a (formal) grammar is a quadruple (cf. Partee, ter Meulen & Wall, 1990, pp. 437ff.). It consists of a set of so-called "terminal symbols", a set of "non-terminal symbols", a starting symbol that is part of the set of non-terminals, and a set of production rules. The terminal symbols are the words of the language in question. Thus, the set of terminal symbols is nothing less than the language's lexicon. The non-terminal symbols represent meaningful expressions and thus sequences of lexical items (words), and the rules define how these lexical elements can be combined.

Although, the C2LG is a grammar, the linguistic principles we will present and discuss in this chapter are not restricted to the field of syntax. On the contrary, since the language that is based on the C2LG is designed to be used for formulating orders to be executed by simulated units, the correct meaning has to be communicated by C2LG expressions. Thus, semantic and pragmatic principles also had to be considered for C2LG development. In the following, we will begin by explaining the role of constituents (section 1.1). Constituents are defined by the grammar of a language, but they also are exactly those units of an expression that refer to real world objects. They thus build the connection between syntax and semantics. As soon as it is clear what kind of constituents are necessary for our language, we can discuss (section 1.2) what kind of formal grammar is appropriate to, on the one hand, allow automatic processing of language expression and, on the other hand, allow the construction of the constituents needed. After the correct type of formal grammar has been specified, we can go one step further and present those additional linguistic principles that support our approach best (section 1.3). These principles determine the kind of grammar we need beyond choosing the formal type of the grammar. In addition to aspects of syntax-semantic interactions, we also have to consider that the language expressions we aim at often are orders and not statements expressing encyclopaedic facts such as "Paris is the capital of France". In order to explain what is special with respect to military communication in general and orders in particular, we have to discuss pragmatics (section 1.4).

### 1.1 Constituency and Thematic Roles

In order to understand linguistic principles, it is necessary to know how a grammar works. Let us assume that there is a basic statement within a military communication, e.g., a single line of an order, like "Advance to area Alpha as soon as possible!" To analyse such a statement, multiple steps have to be undertaken. First, the statement has to be separated into constituents. Constituents are groups of words that belong together. The 5 Ws (Who, What, Where, When, Why) refer to constituents. In our example, the words "to area Alpha" form a constituent that can be categorized as a Where.

The task of a grammar is twofold. First, it has to generate all the expressions we would like to see in our language and by doing so also has to preclude those sequences of words that are not supposed to be expressions of the language. Second, it has to assign an appropriate structure to the expressions of the language (Sells, 1987, p. 9f.). Appropriate means that those words that belong together, such as "to area Alpha" in the example, are grouped together by the grammar rules. Only then can the grammar support us in finding the correct constituents in the language expression and only then can we have success with the following analyzing steps.

In natural language processing, the second step in the analysis is to assign syntactic labels to the constituents, e.g. "noun phrase" or "subject." After that, in a third step, a second label is assigned to each constituent, a label that expresses the semantic role (also called "thematic role") of the constituent in question. In our example, the constituent "*to area Alpha*" would be labelled "prepositional phrase" under the syntactic labelling and "Where" or better "destination" under the semantic labelling. One of the problems that have to be taken into account in natural language processing results from the fact that the semantic structure is not easily derived from the syntactic structure (Nirenburg & Raskin, 2004, pp. 106f.; Sadock, 2003). For example, in both statements "Lightning struck Martin" and "Martin was struck by lightning", "Martin" receives the semantic label "patient" (the one who is affected by the action) although it bears the syntactic labels "direct object" in the first statement and "subject" in the second, whereas the "subject" of the first statement is "lightning." By defining an

artificial language, one can avoid the problem of syntactic-semantic mismatch. The language can be built in a way such that it is possible to assign semantic labels directly to the constituents, that is, without an intermediate step using syntactic labels like "subject" or "object." The assignment can be based on word sequence and on keywords. We have used a combination of both methods in the development of C2LG. Thus, in the latter chapters, we will often refer to keywords (and sequence) when we explain first how to build constituents and then complete C2LG expressions.

If one defines an artificial language, one has to provide the underlying grammar. This is in our case the C2LG. In addition, in order to allow semantic interpretation of language expressions, there needs to be a good system of semantic labels (thematic roles) and an unambiguous mapping from constituents to these labels. An excellent system of thematic roles has been proposed by Sowa (2000, p. 506ff.). These labels are more fine-grained than labels like Who, What, Where, When, Why. For example, there are four labels of type Where, namely location (*stay at control point Charley*), source (*leave from control point Charley*), destination (*advance to control point Delta*), and path (*advance along route Beta*). We used Sowa's thematic roles for C2LG and ensured the unambiguity of the mapping between constituents and thematic roles by the use of keywords and the exploitation of fixed constituent sequences.

In summary, our aim has been to define an artificial and formal language for military or, more precisely, for multi-agency communication which is easily processed automatically. This processing has two steps. First, constituents are identified by using the appropriate grammar rules to calculate the constituents. Second, each constituent is labelled by a semantic label; the set of semantic labels denote the thematic role each labelled constituent plays. Technically, the expressions of the language will be transformed into XML documents in which the constituents form the contents of the elements and the labels are used as tags. In this form, the semantics of an expression can easily be interpreted automatically, e.g., by simulation systems.

### 1.2 The Correct and Appropriate Type of Grammar

As previously stated, a language is defined by a grammar. As explained in the section above, the grammar has to be chosen carefully so that its rules for grouping its lexical items (words) into constituents guarantee that the resulting constituents are appropriate ones.

Let us thus discuss briefly what kind of grammar we chose and why. Linguistic theory categorizes grammars into four types that together form the Chomsky hierarchy (Chomsky, 1957; Partee, ter Meulen & Wall, 1990, section 16.5): grammars of type 0 (unrestricted grammars), of type 1 (context-sensitive grammars), of type 2 (context-free grammars), and of type 3 (regular grammars). The types of rules used in the definition of a grammar determine its type. Only grammars of types 2 and 3 consist exclusively of rules that can easily be applied by automated systems. That is, only these kinds of grammars can be automatically processed. Therefore, our language must have a grammar of type 2 or 3. Grammars of type 3 (regular grammars) only allow two types of rules "A  $\rightarrow a$ " and "A  $\rightarrow a$ B" where "a" represents a terminal symbol (a word) and "A" and "B" represent non-terminal by any sequence of terminals and non-terminals. Thus, type 2 grammars have rules of the form "A  $\rightarrow \sigma$ " in which  $\sigma$  represent such sequences. As in type 3 grammars, the left side of every rule consists of exactly one non-terminal symbol.

Type 3 grammars (regular grammars) are not adequate to define a natural language (cf. Chomsky, 1956; Partee, ter Meulen & Wall, 1990, section 17.3.2: "Inadequacy of right-linear grammars for natural languages"). Since the arguments presented by Partee et al. are focused on embedded recursion, they are not of relevance for our purposes, as embedded recursion is not a property needed in a BML. Nevertheless, a regular grammar is still not adequate for defining a language for military communication, although the contrary has been asserted by Tolk et al. (2007). The reason is the following: regular grammars are not able to structure expressions into appropriate constituents, constituents that, like the 5 Ws, are natural and necessary building blocks of military communications. In order to illustrate this, we again take a look at the example order "advance to area Alpha as soon as possible". This expression has to be divided into three constituents, namely "advance", "to area Alpha", and "as soon as *possible*". We therefore would like to have a rule like "Order  $\rightarrow$  Task Where When" such that Task can be expanded into "advance", Where can be expanded into "to area Alpha", and When can be expanded into "as soon as possible". Such a rule is context-free and thus covered by a context-free grammar (type 2 grammar). In order to allow the generation of the example expression under a type 3 grammar, we would need regular rules of which every one has to separate exactly one word from the expression. The first of these rules, separating advance, would be "Order  $\rightarrow$  advance Non-terminal-1". In a context-free rule we would be able to use the meaningful non-terminal symbols Task, Where and When, whereas in the regular rule we have the meaningless non-terminal symbol "Non-terminal-1". Even worse, we would continue the process with the even more meaningless rule "Non-terminal-1  $\rightarrow$  to Nonterminal-2", followed by the equally meaningless rule "Non-terminal-2  $\rightarrow$  area Non-terminal-3" and so on. By their very nature, regular rules have to be expressed with meaningless nonterminals, whereas context-free rules can use non-terminals representing semantic labels like Where or When. In summary, our language has to have a grammar of type 2 so that the constituents resulting from the first step of analysis can be assigned semantic labels in the second step.

### 1.3 Lexicality and the Principles of Completeness and Coherence

In order to optimize the assignment of semantic labels (analysis step 2), we incorporated the following linguistic principles (in addition to using a grammar of type 2) into our language:

- o Lexicality,
- o the Principle of Coherence, and
- the Principle of Completeness.

Lexicality means that the grammar's rules are based on lexical elements. Standard theories on the cognitive process of language production in general and of the sub-process of grammatical encoding in particular emphasise that this process is lexically driven (Levelt, 1989, p. 235). The theory Levelt presented is based on works of Kempen and co-workers (e.g., Kempen and Hoenkamp, 1987). With respect to grammar theory, Levelt reached back to Bresnan's Lexical Functional Grammar (LFG) (cf. Kaplan & Bresnan, 1982; Bresnan, 2001). C2LG is also modelled after LFG.

The central element of a sentence is its main verb. The main verb expresses the predicate, determines the thematic roles that have to be considered in the interpretation of the sentence. As already explained, sentences in natural languages have to be analyzed syntactically before a semantic interpretation can be carried out. In LFG, this is done by determining the constituent structure (c-structure) of the sentence in question, after which, in the second step, the functional structure (f-structure) is calculated. Central to the calculation of the f-structure

is the lexical entry of the sentence's verb which provides an argument structure that is exploited in the calculation. In the following step, a semantic structure can then be assigned to the sentence on the basis of the previously calculated f-structure. The semantic structure represents the semantic interpretation of the sentence. In LFG, it is called argument structure (a-structure). As also previously mentioned, a semantic structure in an artificial language like BML can directly be assigned to the constituent structure. In this case as well, the verb plays the central role for enabling this assignment. In a BML, verbs normally denote tasks to be executed by units. Their lexical entries come with so-called frames. This term partially goes back to Minsky (1975) but is based more directly on the ideas of Fillmore (1968, 1976). Frames determine, in terms of thematic roles, what kind of constituents *must*, which kind of constituents *may*, and which kind of constituents *may not* follow the tasking verb. For many languages, prominent among them English, the frames of verbs nowadays can be looked up at FrameNet (http://framenet.icsi.berkeley.edu/). In addition, Kremer (2006) offers an analysis identifying which of the Frames listed at FrameNet were used in HUMINT reports produced by German (Bundeswehr) forces under the KFOR mandate.

C2LG has been developed as a lexical grammar modelled on LFG. Tasking verbs form the centre of most basic C2LG expressions. The lexical entry of its task verb determines what kind of constituents (in terms of thematic roles) must, may and may not be used to complete a basic expression; these constituents are ordered in a fixed sequence and supplied with keywords such that thematic roles can be assigned to them unambiguously. The respective thematic roles are taken from Sowa (2000, p. 508). These measures ensure that the principles of coherence and completeness can be applied to C2LG with respect to the semantic interpretation of a C2LG expression and its tasking verb in the same way that these principles are applied to LFG (Bresnan, 2001, section 4.7). Completeness means that a constituent must appear in a thematic role required by the frame. Coherence means that each constituent that is part of an expression must have a thematic role assigned that is licensed by the frame.

### 1.4 Speech Acts

Not all language expressions are about facts which can be judged as true or false. In communication, a speaker has an intention that she wants to convey by her expression, and it is important that she conveys the expression in a way that the hearer can recognize that intention. The expression has to be appropriately formulated to let the hearer recognize the speaker's intention. Semantic is about truth; pragmatics concerns appropriateness and communication of intention.

In the field of pragmatics, the theory of speech acts (Austin, 1962; Searl, 1969) specifies the relationship between intention and expression (Sadock, 2006). Speech acts are expressions with an intention. In the literature, the communicative intention is called its *illocutionary force* (Austin, 1962; Dröge, 2006). Speech acts can be classified by their illocutionary force. With respect to military communication, reports belong to the class of assertives and orders and requests to the class of directives according to Searl's (1979) classification.

In an assertive, the speaker informs the addressee that something is so. There are sincerity conditions involved, by which that the speaker commits herself that she believes in the truth of what is expressed. Storing data in a database and exchanging this data with others is an assertive by its nature.

Orders are directives. "The purpose of a directive is to get the addressee(s) to do something" (Levelt, 1989, p. 60). Orders are especially strong directives. The sender assigns tasks to the addressee. Her intent is that the addressee executes the tasks following the wording of the order. The speaker can even rightfully assume that her speech act will be successful, which means that the addressee indeed acts as intended by the speaker. This assumption is in contrast to other directives such as requests, pleas, or challenges. The right to assume that the addressee will execute the order is anchored in the military hierarchy, in every military doctrine and in the self-concepts of military organisations. It is also the linguistic core of the speech act of ordering. In summary, orders initiate actions with intention, that intention being the execution of the assigned task. Assertives, in contrast, do not necessarily initiate actions. To illustrate the illocutionary force of orders, consider the two following fictional descriptions of ordering and the results achieved by the orders. The first example is from ancient times, from the Iliad, the second is from modern SF.

"Laut erscholl sein durchdringender Ruf in die Schaaren der Troer: Auf, ihr reisigen Troer, hinan! Durchbrecht der Argeier Mauer, und werft in die Schiffe die schreckliche Flamme des Feuers! Also ermahnte der Held; und Aller Ohren vernahmen's. Grad' andrang zu der Mauer die Heerschaar; Jene begierig Klommen empor die Zinnen, geschärfte Speer' in den Händen." (Homer's *Iliad*, as translated into German by Heinrich Voß, 1882, p. 188, 12<sup>th</sup> song, verse 439 to verse 444)

""Helm, roll ninety degrees port!"

No one on that wounded, half-broken bridge, and Honor least of all, perhaps, recognized the cool, sharp soprano which cut cleanly through the chaos, but the helmsman clinging to his own sanity with his fingernails recognized the incisive bite of command.

"Port broadside stand by for Fire Plan Delta Seven," that soprano rapier commanded, and confirmations raced back from *War Maiden*'s undamaged broadside [...]." (Weber, 2004, pp. 151f.)

Obviously, there is no relevant difference between the examples. In both cases, the addressees act as required, and they act without any inquiry or discussion. On the contrary, they anticipate the orders eagerly and act with raised spirit.

It has been suggested that instead of using a BML to express an order and then to send the BML order to the taskee, the order should be represented as a plan in a JC3IEDM database. After that, the "action-task-category-code" values of all included tasks could be changed from "plan" (PLAN) to "order" (ORD), and the representation and the change could be distributed to all other C2 systems or simulation systems with the assumption that every organisation or simulation system would interpret the distributed data as orders upon which it must act. This suggestion wilfully ignores the difference between an order as a directive and the information about the content of an order as it is stored in the database. As a speech act the proposed method is an assertive only.

In section 6.3, we will discuss and provide rules for communication within a complex endeavour. This will include some more remarks on speech acts since the communication situation is different if orders are substituted by other directives as taskings and requests.

### 2 Notation

In order to be usable as a specification, this document adheres to quite strict notation. The grammar of C2LG uses formal context-free rules for building constituents. We will present these rules using "Arial" typeface in the usual form for rules, in which the left side and the right side of the rule are separated by an arrow (" $\rightarrow$ ") that means "is expanded to". As the rules are context-free, the left side of each rule consists of exactly one non-terminal symbol whereas the right side is a sequence of non-terminal and terminal symbols. Non-terminal symbols start with a capital letter. Terminal symbols (words) are always given in italics. Most terminal symbols are taken from the value lists of respective attributes of the JC3IEDM. These are written in non-capitalized letters (including the first letter). However, for some of JC3IEDM's attributes, the values are represented not as "normal" words but as so-called physical values (which are much shorter). These physical values are represented completely in capital letters. For example, the physical value of "not later than" is "NLT". But, of course, when used as terminal symbols in C2LG rules, the physical values are written in italics (i.e., "NLT"). Some terminal symbols are not derived from the JC3IEDM. These are so-called keywords that are used in the C2LG in order to identify in an unambiguous manner the thematic role of the constituent they represent. These keywords are not capitalized, italicized and, to distinguish them from other terminal symbols, they are also in boldface. Elements that are part of the right side of the rule but which are optional appear in. Since the C2LG is a lexical grammar, there are many rules associated with a specific lexical item, e.g., with a tasking verb. In some cases, rule forms are given rather than individual rules. By rule forms we mean that rules of similar structure are expressed together in a short and concise manner. Instead of listing individually the rules for all possible similarly structured tasking verbs, a single rule form provides the general format which these rules follow. The rule forms are always complemented by some examples of respective rules in order to illustrate the forms.

Normally, rules and rule forms are provided separately from the explanatory text. When portions of rules are cited in the explanatory text, the rules appear enclosed by quotation marks. And, of course,, these will conform to the notation conventions described above.

Examples of C2LG expressions are sequences of terminal symbols and will, therefore, appear in italics. Those parts of such expressions which are of special interest in the respective section appear highlighted in blue.

### **3** Orders in C2LG

Since C2LG is supposed to be a grammar that defines a language for communications in an operation, C2LG expressions are messages. If facts are conveyed, the expressions are linguistically assertives (according to the theory of speech acts as discussed in section 1.4). In the context of an operation, these kinds of expressions are otherwise known as reports. Before we discuss how to express reports in C2LG, however, we will discuss how to express directives, the kind of expressions by which the sender intends to get the addressee to do something. Our examples are taken from the military domain. The standard directive in this domain is an order. In the chapters following the discussion of how to express orders, we will discuss how to express other directives (like requests) as well as assertives (reports).

As all forms of (at least military) communication should have a header which includes, at a minimum, the sender, the addressee, and the point in time when the communication is sent, we will begin our C2LG rule set with the following rules.

<u>Rules 3.0.a) to 3.0.d)</u>							
3.0.a) Order	$\rightarrow$	Header OrderBody					
3.0.b) Request	$\rightarrow$	Header RequestBody					
3.0.c) Report	$\rightarrow$	Header ReportBody					
3.0.d) Header	$\rightarrow$	Sender Addressee SendingTime (SecurityClassification)					
3.0.d) Header	$\rightarrow$	Sender Addressee Sending Time (SecurityClassification)					

"Sender" and "Addressee" are to be expanded by the name of the respective organisations, "SendingTime" is to be expanded by a datetime value (cf. section 4.4), and "SecurityClassification" can be added in military communication or whenever it makes sense. "SecurityClassification" might be expanded by a value from the list of security values that are valid for the operation in question, for example, "*NATO Unclassified*" or "*NATO Secret*" would be valid values in a NATO-led operation.

In C2LG, the representation of the content of an order is determined by the following rule:

In this rule, "CI" stands for "Command Intent", "OB" (order, basic line) is a single task assignment, "C\_Sp" represents a single spatial coordination expression, and "C\_T" a single temporal coordination expression. The star (\*) at "OB", "C\_Sp", and "C\_T" indicates that these single expressions may occur multiple times in the order's paragraph 3. The expansion of "CI" (Command Intent) will be dealt with later in this document in its own chapter (Chapter 7), as an understanding of the expression of both basic lines of orders and reports is a prerequisite to understanding this topic.

With regard to "OB", as C2LG is a lexical grammar, there is exactly one rule for every tasking verb contained in C2LG's lexicon. In its present form, the tasking verbs for C2LG were taken from the value list of JC3IEDM's attribute "action-task-activity-code". All OB rules are based upon the rule form shown in (3.1.a); specific examples of rules for five tasking verbs follow in 3.1.c) – 3.1.g); a comprehensive list is provided in Appendix A.

 $\frac{Rule\ Form\ 3.1.a)}{OB \rightarrow \ TaskingVerb\ Tasker\ Taskee\ (Affected|Action)\ Where} StartWhen\ (EndWhen)\ Mod\ Why\ Label$ 

<u>Rules 3.1.c) to 3.1.g</u>)

3.1.c)	<i>lvance</i> Tasker Taskee RouteWl artWhen (EndWhen) Mod Why	
3.1.d)	<i>sist</i> Tasker Taskee Action AtW artWhen (EndWhen) Mod Why	

- 3.1.e) OB → *block* Tasker Taskee Affected AtWhere StartWhen (EndWhen) Mod Why Label
- 3.1.f)  $OB \rightarrow defend$  Tasker Taskee Affected AtWhere StartWhen (EndWhen) Mod Why Label
- 3.1.g)  $OB \rightarrow march Tasker Taskee RouteWhere StartWhen (EndWhen) Mod Why Label$

In both the rule form and the example rules, "Tasker" is a non-terminal to be expanded by the name of the organisation 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, while "StartWhen" and "EndWhen" are non-terminals to be expanded by temporal constituents. "EndWhen" is optional as indicated by the parentheses. "Tasker", "Taskee", "StartWhen", and "EndWhen" appear in each rule for task assignment as do "Mod", "Why" and "Label". "Mod" can be expanded by modifying constituents that describe "how" the task in question must be executed. "Why" gives the purpose that the task serves, and "Label" provides a unique identifier assigned to the task so that it can be referred to by other lines of the order.

Some constituents do not appear in every rule for task assignment, specifically "Affected" and "Action", "AtWhere" and "RouteWhere". "Affected" is a non-terminal to be expanded by the name of the organisation or the object to be affected by the task; in the terms of Sowa (2000, p. 508) this is either "Patient" - in the case that an organisation is affected - or "Theme" - in the case that an inanimate object is affected. Whether "Affected" is part of a rule depends on the tasking verb. It is there when the tasking verb's frame requires it as in examples 3.1.e) and 3.1.f). Dependency on the tasking verb is also true for "Action", is to be expanded by a reference to another action, e.g., a task that is to be assisted as example rule 3.1.d). Within the parentheses "Action" is separated from "Affected" by an "exclusive or" as indicated by the symbol "|". The parentheses around "either Affected or Action" indicate that this element is optional. Therefore, rules that include neither an "Affected" nor an "Action" are allowed as shown in the example rules 3.1.c) and 3.1.g). The dependency on the tasking verb also holds for the "Where" in the rule form. In the rules themselves, the "Where" either is an "AtWhere" - in the case where the tasking verb does not involve movement as in the example rules 3.1.d), 3.1.e), and 3.1.f) - or a "RouteWhere" - in the case where the tasking verb does, e.g., example rules 3.1.c) and 3.1.g). The fact that there is no general rule for assigning tasks but rather an individual rule for each tasking verb ensures that the rules align with the frames of the tasking verbs. This is the core of lexicality. Furthermore, any potential context-sensitivity of the rules is thereby avoided.

Like single task assignments ("OB"), single spatial coordination expressions ("C\_Sp") are lexicalized. There is again a single rule for each kind of control feature (tactical spatial object), but, as with task assignments, all of these rules follow a single general rule form. This rule form as well as two examples of rules are shown below.

 $\begin{array}{c} \underline{\text{Rule Form 3.1.b}} \\ \text{C\_Sp} \rightarrow & \text{ControlFeatureType Tasker (Taskee) ControlFeatureID} \\ & \text{StartWhen (EndWhen) Label} \end{array}$ 

 $\frac{\text{Rules 3.1.h) and 3.1.i)}}{\text{3.1.h}} \quad \text{C}_\text{Sp} \rightarrow$ 

area of responsibility Tasker Taskee ControlFeatureID StartWhen (EndWhen) Label

#### 3.1.i) C\_Sp → *hazard area* Tasker ControlFeatureID StartWhen (EndWhen) Label

As in single task assignments, "Tasker" is expanded by the name of the organisation that orders (and thus owns) the control feature. "Taskee" is expanded by the name of the organisation that the control feature is assigned to. Whether "Taskee" is demanded, as it is in rule 3.1.h), or not, as in rule 3.1.i) depends on the control feature in question. Some control features simply are not assigned to any unit at all, e.g., those indicating the presence of threats. In the case where a control feature is assigned to more than one unit, as, for example, it would be for a contact point, there has to be a separate spatial coordination expression for each of these taskee units. Naturally, these expressions would only differ in the content of "Taskee". "ControlFeatureID" is the name of the control feature that being used for the purpose of spatial coordination. It is assumed that the control feature was previously defined and stored in the underlying database so that the ID can be used to look up the features coordinates if necessary. "StartWhen" and "EndWhen" define the timeperiod during which the control feature is valid. This is important: if, for example, an air corridor is referenced, an artillery unit may fire through that corridor at a time when the corridor is not valid, but is prohibited from firing through it while it is valid. And, finally, "Label" provides the unique name of the whole spatial coordination.

### **4** The Building Blocks

In chapter 3, we discussed how orders are expressed in C2LG on the sentence level. Specifically, we discussed how task assignments are constructed as sequences of constituents. In this chapter, we will now focus on these constituents. We will explain how to build the constituents as sequences of lexical items and provide examples to illustrate the discussion. We will use complete sentence-like expressions or even expressions that consist of many sentences in our examples, in which the constituent in question will appear in blue. By presenting the constituents in such a large context, we hope to show the interconnections among the constituents. In addition, in the examples, the JC3IEDM physical values normally used have been substituted by expanded values for reasons of better readability. For example, "platoon" appears instead of the "PLT" used by C2LG.

Constituents, as has already explained in section 1.1, are the linguistic specifications of the 5 Ws. They express the thematic roles more precisely than do the 5Ws and therefore are the natural building blocks of sentence-like expressions in military communication. The building blocks are

- a) verbs, predominantly task verbs specifying the "What", (discussed in section 4.1),
- b) denotations for organisations, specifying the "Who", (discussed in section 4.2),
- c) constituents describing spatial aspects, the "Where", (discussed in section 4.3),
- d) constituents describing temporal aspects, the "When", (discussed in section 4.4),
- e) modifier constituents, the "How", (discussed in section 4.5), and
- f) constituents that express purpose, the "Why", (discussed in section 4.6).

### 4.1 Actions – The "What"

As already stated, verbs are the central elements – linguists would say the "head" – of a sentence. Similarly, tasking verbs are the central elements of single task assignments. All single lines, the sentence-like expressions of C2LG, are shaped by their verbs' frames. In particular, tasking verbs determine whether there is an "Affected" in a given sentence. In the 5W-terminology, a verb is the "What". More often than not, what we call "Affected" – the verb's direct object in a natural language expression – is also considered part of this "What" in 5W-terminology. This is a misconception in the 5W-terminology because "Affected" is a thematic role. With respect to wh-words, "Affected" is a "Whom" if an organisation or a person is affected; only in the case that the action affects an (inanimate) object, e.g., a facility, we would use "What", not in the sense of "What has been done?" but in the understanding of "What has been affected?". The sequence of lexical items that fills the "Affected" role forms a constituent by itself and has to be regarded as a unique block. Since this constituent often denotes an organisation, e.g., a unit affected by tasks like ambush, attack, block etc., an "Affected" often is structured like a "Who".

In C2LG, the lexical items that are allowed as tasking verbs ("TaskingVerb") are the values of JC3IEDM's attribute "action-task-activity-code". (Remark: This attribute clearly is misnamed. According to JC3IEDM's own naming conventions it should have been named "action-task-category-code". However, that name has been given to an attribute that denotes whether a task has been requested, planned or ordered, another naming error.) It should be noted here that the change in the MIP data model from C2IEDM to JC3IEDM – "J" meaning "Joint" – had the result that the data model no longer is "land-centric". The result of this change is that the attribute which now has the most predefined values by far is "aircraft-type-model-code". The possibilities for denoting an aircraft exceed the possibilities for denoting vessels or vehicles by a factor of 100 to 1, which seems to be somewhat imbalanced. On the positive side, however, the change also means that actions like "air interdiction, battlefield" can now be tasked. The C2LG has already incorporated some of these air-specific values as tasking verbs. These verbs come with very specific "Where"s (as will be discussed in section 4.3.1), which however does not pose a problem since the C2LG is a lexical grammar and thus specific form of specific constituents can be determined by the verbs.

### 4.2 Denoting Organisations – The "Who"

In task assignments, organisations are denoted as "Tasker", as "Taskee", or as "Affected". The superior organisation whose commander issues the order is "Tasker", the organisation ordered to execute the task is "Taskee", and the organisation that is affected by the task is "Affected", (providing there is an organisation that is affected by the task). In all cases, the organisation is referred to by its unique name under which it can be unambiguously found in the underlying database.

In the 5W-terminology, organisations are of type "Who" although within individual task assignments, only the Taskee is the "*real* Who" for the task. An organisation affected by a task is a "Whom", while the "Tasker" is the "Who" of the ordering action, but not the "Who" of the ordered action. This example again shows clearly that the 5W-terminology, although often helpful, is sometimes misleading and not specific enough to guarantee correct semantic analysis.

### 4.3 Expressing Spatial Aspects

In this section, we will specify how the "Where" for individual task assignments (orders) and single report lines is expressed in CBML. More precisely, we will specify how C2LG expresses spatial information. This includes representation for the following thematic roles: origin, path, destination, direction, and location. In principle, in C2LG, there are two kinds of constituents that represent spatial information connected to assigned and reported tasks. The non-terminals associated with these constituents are called "AtWhere" and "RouteWhere". As C2LG is a "lexical grammar", the non-terminal used depends on the task verb selected. In general, task verbs that denote (among other things) movement will contain "RouteWhere" whereas task verbs that do not include movement require an "AtWhere". "AtWhere" is also used in single reports that do not cover tasks (event reports and status reports including position reports). In the case of an air task, in contrast to ground-based tasks, further spatial aspects must be taken into account. As a result, there is an additional specific rule for the "Where" in this case; the non-terminal for this constituent is called "AirRouteWhere".

### 4.3.1 C2LG Rules for Spatial Constituents

We begin with the rules for "AtWhere", which denotes the essence of spatial information, namely, a location.

<u>Rules 4.3.a) to 4.3.d)</u>							
4.3.a)	AtWhere	→ SpatialModifier Location					
4.3.b)	SpatialModifier	$\rightarrow$ {at, in-front-of, behind}					
4.3.c)	Location	→ LocationName   Coord					
4.3.d)	Coord	$\rightarrow$ Latitude Longitude (Label)   UTMREF (Label)					

As can be seen in the rules, "AtWhere" is to be expanded to a spatial modifier and "Location". Currently, we allow "*at*", "*in-front-of*", and "*behind*" as spatial modifiers. "Location" can be the name of a location ("LocationName"), as defined by a database entry or a coordinate ("Coord"). A coordinate consists of two real numbers, the first of which denotes the latitude ("Latitude") and the second the longitude ("Longitude"). As an option, a label ("Label") can be added to latitude and longitude such that this point in space can be referred to elsewhere in the respective C2LG order or report by this label. Similarly, a coordinate of type UMTREF also can be labelled.

The rules for "RouteWhere" are somewhat more complex.

<u>Rules 4.3.e) to 4.3.h)</u>							
4.3.e)	RouteWhere	→ along RouteName					
4.3.f)	RouteWhere	→ towards Location   towards Bearing					
4.3.g)	RouteWhere	$\rightarrow$ ( <i>from</i> Location) <i>to</i> Location ( <i>via</i> Location*)					
4.3.h)	Bearing	$\rightarrow$ {north, northeast, east,, northwest}   Degree					

"RouteWhere" denotes the route for an assigned or reported task that includes movement. "RouteWhere" can be expanded in numerous ways. First, when the route has already been stored and named in the database, "RouteWhere" can be expanded by the sequence of the keyword "*along*" and the unique name ("RouteName") for the route in the database. Second, when only the direction of the task's movement is known, "RouteWhere" is expanded by the sequence of the keyword "*towards*" and something that indicates the direction. This can be either a location ("Location"), following the rules previously mentioned, or a bearing ("Bearing"). A bearing can be given as a cardinal point, e.g. "*north*", or as in degrees (integer between 0 and 360). Third, "RouteWhere" can be expanded by a sequence of three spatial constituents, namely an optional starting point (also called origin) that is preceded by the keyword "*from*", a mandatory destination preceded by the keyword "*to*", and an optional path identified by the keyword "*via*". All these constituents consists of the relevant keyword and one location, in the case of the path constituent it is possible to list more than one location following the keyword "*via*", i.e., the path between origin and destination need not be a straight line.

The spatial constituents which expand "RouteWhere" can be recognized by the keyword they begin with: "*along*", "*towards*", "*from*", "*to*", and "*via*". These keywords also determine the thematic role of the constituent. The location constituent that forms "AtWhere" can be recognized as it begins with one of the spatial modifiers "*al*", "*behind*", or "*in front of*": additionally "*al*" may also serve as a temporal modifier (cf. below). When "*al*" is used as temporal modifier, it immediately follows another keyword, i.e., "*start*", "*end*", or "*ongoing*". In all other cases, "*al*" signals the beginning of a location constituent ("AtWhere").

In comparison to ground tasks, air tasks need a specific spatial constituent which is called "AirRouteWhere". Rule 4.3.i) provides an example of a rule for assigning an air task. This rule follows rule form 3.1.a) and is comparable to rules 3.1.c) through 3.1.g). It should be remarked that rule 4.3.i) not only differs from rules 3.1.c) through 3.1.g) with respect to the "Where" but also with respect to the non-terminal for the modifier ("AirMod") in 4.3.i).

<u>Rule 4.3.i)</u>

### $OB \rightarrow air interdiction$ Tasker Taskee AirRouteWhere StartWhen (EndWhen) AirMod Why Label

There are two main differences between ground tasks and air tasks. First, air tasks obviously require a height dimension, and, second, an aircraft always should go back at a secure airport. It should not only be tasked to move to a certain destination where it is supposed to affect something by acting. The air task should incorporate the route home. These additional demands are mirrored in the rules for "AirRouteWhere" and "AirMod". The expansion of "AirRouteWhere" is determined by rules 4.3.j) to 4.3.q).

<u>Rules 4.3.j) to 4.3.q</u>)

4.3.j) AirRouteWhere	$\rightarrow$	AirRouteIn OnStation AirRouteHome
<ul> <li>4.3.k) AirRouteIn</li> <li>4.3.1) AirRouteIn</li> <li>4.3.m) OnStation</li> <li>4.3.n) AirRouteHome</li> <li>4.3.o) AirRouteHome</li> </ul>	$\begin{array}{c} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{array}$	<ul> <li><i>in</i> (<i>from</i> Location) <i>to</i> Location (<i>via</i> Location*)</li> <li><i>in along</i> Route</li> <li><i>fly</i> Orbit   <i>fly</i> RaceTrack   <i>target</i> Location</li> <li><i>home</i> (<i>from</i> Location) <i>to</i> Location (<i>via</i> Location*)</li> <li><i>home along</i> Route</li> </ul>
4.3.p) Orbit 4.3.q) RaceTrack	$\rightarrow$ $\rightarrow$	<b>orbit of radius</b> Radius <b>race track of radius</b> Radius <b>and width</b> Width

"AirRouteWhere" consists of three parts, namely "AirRouteIn", "OnStation", and "AirRouteHome" (rule 4.3.j)). Both "AirRouteIn" and "AirRouteHome" expand to specify a

route in a similar way as "RouteWhere" does. Rules 4.3.k) corresponds to rule 4.3.g), and rule 4.3.l) corresponds to rule 4.3.e). The only difference is that "AirRouteln" has the additional keyword "*in*" to ensure that the incoming route cannot be mistaken for the outgoing route. Similarly, the expression denoting the route out has the keyword "*home*". Otherwise the rules for "AirRouteHome" – rule 4.3.n) and 4.3.o) – are the same as the rules for "AirRouteln". "OnStation" specifies how the aircraft is supposed to move while at the target area, i.e., whether it should fly an orbit or a racetrack or simply operate against target. The specific size of the orbit or the racetrack is determined by rules 4.3.p) and 4.3.q), respectively.

### 4.3.2 Examples of Spatial Constituents

The following examples are taken from our earlier papers on C2LG. These examples show complete single task orders or complete single reports in order to demonstrate the spatial constituents in their contexts. The spatial constituents themselves have been highlighted in blue.

Example 4.3.a) ("RouteWhere" expanded by using a predefined route):

The leader of the Multinational Division "West" (led by a Spanish element) orders the 13<sup>th</sup> Dutch Mechanized Brigade (M\_BDE13(NL)) to march along a predefined route labelled "DUCK".

[task assignment] march MND-West(SP) M\_BDE13(NL) along DUCK start at Phase1A end before Phase1E label\_3\_11;

Example 4.3.b) ("RouteWhere" expanded by providing source and destination): The leader of Brigade 66 orders her first battalion to advance from its line of departure called "Denver-North" to phase line "Jade" (in order to execute the task given in example 4.3.c).

### [task assignment] advance BDE-66 BN-661 from Denver-North to Jade start at TP0 in-order-to enable label-o12 label-o11;

Example 4.3.c) ("AtWhere" expanded by using a predefined location in a task assignment): The leader of brigade 66 ("*BDE-66*") orders her first battalion to fix an enemy unit (previously labelled "*Red-1-1-182*") at a location called "*Boston*" so that it can be destroyed (by another unit; the destroy action given to this unit is referenced to by "*label-o33*").

[task assignment] fix BDE-66 BN-661 Red-1-182 at Boston end nlt TP1 in-order-to enable label-o33 label-o12;

### 4.3.3 Additional Remarks on Spatial Aspects

In military orders, spatial information is normally at least partially represented in overlays that may be attached to the order.

If an explicit textual representation of the overlay is available (e.g., as specific XML file identified by a unique label), this overlay can be referred to. In this case, the "Where" ("AtWhere", "RouteWhere", or "AirRouteWhere", depending on the tasking verb) is expanded by the keyword "under use of" ("*under-use-of*") and the label ("OverlayName") of the overlay:

<u>Rules 4.3.r) to 4.3.t)</u>							
4.3.r)	AtWhere	$\rightarrow$	under-use-of OverlayName				
4.3.s)	RouteWhere	$\rightarrow$	under-use-of OverlayName				
4.3.t)	AirRouteWhere	$\rightarrow$	under-use-of OverlayName				

### 4.4 Expressing Temporal Aspects

In this section we will specify how the "When" for single task orders and single reports is expressed in CBML. We need to differentiate between temporal constituents that refer to points in time expressed by denoting the date time, and temporal constituents that link the temporal aspects of one action (the target action or subject action) to specific temporal specifications of another action (the relatum action or object action), for example, stating that the target action should start as soon as the relatum action ends. The first type of temporal constituents are called "specific temporal constituents", while the second type are "conditional temporal constituents".

### 4.4.1 C2LG Rules for Specific Temporal Constituents

In single task orders, a task is assigned to a unit. In accordance with the JC3IEDM, C2LG expressions can specify when this task has to start and when it has to end. To specify when the assigned task should start, "StartWhen" is used. To specify when the task should end, "EndWhen" is used.

 $\begin{array}{ll} \underline{Rules \ 4.4.a) \ and \ 4.4.b)} \\ 4.4.a) \ StartWhen \ \rightarrow \textit{start} & \mbox{TemporalModifier DateTimeValue} \\ 4.4.b) \ EndWhen \ \rightarrow \textit{end} & \ \mbox{TemporalModifier DateTimeValue} \end{array}$ 

Both expressions consist of three parts. The first part is a keyword which is "*start*" in the case of "StartWhen" and "*end*" in the case of "EndWhen". The second is a temporal modifier ("TemporalModifier") and the third is a date time expression ("DateTimeValue"). Temporal modifiers are, for example, "*at*", "*not later than*", and "*as soon as possible but not later than*". The list of temporal modifiers consists of values of JC3IEDM's attribute "action-task-start-qualifier-code". (JC3IEDM's attribute "action-task-end-qualifier-code" provides the same values.) The current list for C2LG is provided by rule 4.4.c).

 $\frac{\text{Rule 4.4.c)}}{\text{TemporalModifier}} \rightarrow \{AFT, ASAP, ASAPAF, ASAPNL, AT, BEF, NLT, NOB\}$ 

The meanings (values) of these physical values are as follows: AFT = after, ASAP = as soon as possible, ASAPAF = as soon as possible after, ASAPNL = as soon as possible but not later than, AT = at, BEF = before, NLT = not later than, NOB = not before.

There are in fact more values for these two JC3IEDM attributes, but these either denote lack of knowledge (UNK = unknown, TBD = to be determined) or are not tied to a specific date time denotation ( $ONCALL = on \ call$ ,  $ONCDWD = on \ codeword$ , INDEF = indefinite,  $UNTFRN = until \ further \ notice$ : the latter two are values for "action-task-end-qualifier-code" only). None of these values are currently part of the C2LG specification.

"DateTimeValue" either is specified as a specific date time or it is "now". When the default value "now" is used, the date time value is automatically set to the sending time of the communication (order, request, or report). At present, C2LG does not further specify the format used for denoting date. However, if C2LG is going be used for C2 communications it is highly recommended that this format is specified beforehand. In order to prevent misunderstandings, C2LG also allows expanding "DateTimeValue" with unique identifiers for specific points in time if those previously had been stored in the underlying common database. Normally, such names refer to temporal start or end points of phases defined for the operation in question.

### 4.4.2 Example of a Specific Temporal Constituent

The following example is taken from our previous papers on C2LG. It illustrates the usage of a specific temporal constituent.

### Example 4.4.a)

The leader of the Multinational Division "West" (led by a Spanish element) orders the  $13^{\text{th}}$  Dutch Mechanized Brigade (M\_BDE13(NL)) to march along a predefined route labelled "DUCK" during the first phase (lasting from the point in time "*Phase1A*" to the point in time "*Phase1E*") of an operation.

[task assignment] march MND-West(SP) M\_BDE13(NL) along DUCK start at Phase1A end before Phase1E label\_3\_11;

### 4.4.3 C2LG Rules for Conditional Temporal Constituents

In single task orders the temporal constituents can also refer to another task or to an event. With respect to these cases, the JC3IEDM denotes the assigned task as "subject task", and the action (task or event) referred to as "object action". From a linguistic point of view, the names "target task" and "relatum action" denote them more precisely.

When referencing other tasks or events, "StartWhen" and "EndWhen" are extended by rules 4.4.h) and 4.4.i) below rather than by the rules given above in section 4.4.1:

 $\begin{array}{ll} \underline{Rules \ 4.4.h) \ and \ 4.4.i)} \\ 4.4.h) \ StartWhen \rightarrow \textit{start} \\ 4.4.i) \ EndWhen \rightarrow \textit{end} \end{array} \begin{array}{l} \text{TemporalAssociationModifier1} \ \text{RelatumAction} \\ \text{TemporalAssociationModifier2} \ \text{RelatumAction} \end{array}$ 

The keywords in these rules are the same as previously described in Section 4.4.1 above. The temporal modifier in rules 4.4.a) and 4.4.b) is substituted by a temporal association modifier that specifies the association by which the target task is related to the relatum action. Rather than referring to a point in time, this rule refers to the relatum action ("RelatumAction"). The reference uses a unique label that must have been assigned to the relatum action in a preceding C2LG expression.

The following is a list of the values C2LG allows as temporal association modifiers. They are taken from or inspired by the list of physical values designed for JC3IEDM's attribute "action-temporal-association-category-code". However, for our purposes the list of these

physical values from the JC3IEDM was incomplete, so it had be expanded to what is proposed by rules 4.4.j) and 4.4.k).

Rules 4.4.h) and 4.4.i)

4.4.j) TemporalAssociationModifier1  $\rightarrow$  {*STRBEF, STRSTR, STRDUR, STREND, STRAFT*} 4.4.k) TemporalAssociationModifier2 $\rightarrow$ {*ENDBEF, ENDSTR, ENDDUR, ENDEND, ENDAFT*}

The meanings of these values are as follows:

STRBEF	=	target action starts before relatum action starts
STRSTR	=	target action starts when relatum action starts
STRDUR	=	target action starts during relatum action
STREND	=	target action starts when relatum action ends
STRAFT	=	target action starts after relatum action ends
ENDBEF	=	target action ends before relatum action starts
ENDBEF ENDSTR	= =	target action ends before relatum action starts target action ends when relatum action starts
ENDSTR	=	target action ends when relatum action starts
ENDSTR ENDDUR	=	target action ends when relatum action starts target action ends during relatum action

### 4.4.4 An Example of a Conditional Temporal Constituent

The following shows an example in which a conditional temporal constituent is used. Naturally, in the example, there two task assignments such that the temporal constituents of the one task can serve as relatum information in the conditional temporal constituent.

### Example 4.4.c)

The leader of the battalion 661 orders his 2<sup>nd</sup> company to occupy the "Prins Willem-Alexander" bridge at Parnass. This task is labelled by "*label-023*". The temporal constituents for this task are expressed as specific temporal constituents.

Occupying the bridge is the pre-condition for securing it. Consequently, the next task, namely securing the bridge, should start when the occupation has been accomplished: this chaining is seen in the second task assignment in this example.

[*task assignment*] occupy BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start at TP1 end before TP2 in-order-to enable label-o24 label-o23;

[task assignment] secure BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start STREND label-o23 label-o24;

### 4.5 Expressing Modifiers

In this section the specification for expressing the "How" for single task orders in C2LG will be presented.

### 4.5.1 C2LG Rules for Modifyer Constituents

Currently, there are three kinds of sub-constituents for the modifier constituent, namely an instrument constituent, a formation constituent and a manner constituent. All are optional as indicated by the parentheses in the rules. These rules are as follows:

<u>Rules 4.5.a) to 4.5.d)</u>							
4.5.a)	$Mod \to$	( <b>by</b> Instrument) ( <i>in-formation</i> Formation) ( <i>in-manner</i> Manner)					
4.5.b)	Instrument	$\rightarrow$ Vehicle   Vessel   Aircraft   foot					
4.5.c)	Formation	$\rightarrow$ column   patrol   wedge   line   coil   echelon   herringbone					
4.5.d)	Manner	ightarrow fast   slow   cautious   aggressive					

The instrument constituent starts with the keyword "**by**". Currently, this constituent is used only for expressing the means of transport for units. These means are vehicles, vessels or aircraft. In addition, the lexical element "*foot*" can be used resulting in the constituent "**by** *foot*". This expresses explicitly that, for example, an infantry unit has to move without using vehicles. Values for "Vehicle" are taken from JC3IEDM's table "vehicle-type-category-code", values for "Vessel" from table "vessel-type-category-code" and values for "Aircraft" from tables "aircraft-type-category-code" and "aircraft-type-model-code". If in the future, the MIP adds the attributes "vehicle-type-model-code" and "vessel-type-model-code", the respective values could be used as well.

The formation constituent uses keyword "*in-formation*" together with one of the words listed in the respective rule.

The manner constituent uses keyword "*in-manner*" together with one of the words listed in the respective rule.

As had already been mentioned in section 4.3, air tasks not only have their own specific spatial constituents, they also have their own specific modifier constituents. In air task assignments and in reports about air tasks, "AirMod" is used instead of "Mod". "AirMod" has already been shown in rule 4.3.i). It expands according to rule 4.5.e) rather than rule 4.5.a).

# $\frac{\text{Rule 4.5.e)}}{\text{AirMod} \rightarrow (by \text{Instrument}) (in-formation \text{Formation})} \\ (by-speed \text{Speed}) (at-flight-level \text{FlightLevel})}$

In rule 4.5.e), the manner constituent is substituted by constituents that directly denote the ordered speed and the ordered flight level. Both constituents consist of a keyword ("**by**-**speed**" and "**at-flight-level**") and a numerical value (expanding "Speed" and "FlightLevel", respectively). It should be mentioned, however, that the numerical values are associated with a unit of measure. Units of measure should be agreed upon among communicators before exchanging orders and reports. Otherwise, interoperability problems occur.

### 4.5.2 Examples of Modifier Constituents

### Example 4.5.a)

Taking example 4.4.c), we might additionally require that the 2<sup>nd</sup> company of the battalion occupy "Prins Willem-Alexander Brug" in very near future. In this case, the applicable manner constituent is added.

#### [task assignment] occupy BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start at TP1 in-manner fast in-order-to enable label-o24 label-o23;

### Example 4.5.b)

It might also be the case that in order to facilitate the occupation of the bridge, the battalion commander orders the  $2^{nd}$  company to go there by helicopter (*AIRRW* = *rotary wing*). In this case the applicable instrument constituent is also added.

[task assignment] occupy BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start at TP1 by AIRRW in-manner fast in-order-to enable label-o24 label-o23;

### 4.6 Expressing Purpose

In this section the C2LG specification for the "Why" for single task orders, for single request lines, and for single lines of task reports is discussed.

### 4.6.1 C2LG Rules for the Purpose Constituent

The purpose constituent starts with the keyword "*in-order-to*". This keyword is followed by a task-influencing verb and the label of a single task order. If the order as a whole includes a command intent expression, the label of one line of this command intent might be used instead of the label of a single task order.

<u>Rules 4.6.a) to 4.6.c)</u>

4.6.a) Why  $\rightarrow$  *in-order-to* TaskInfluencingVerb TaskLabel

4.6.b) Why  $\rightarrow$  *in-order-to* TaskInfluencingVerb IntentExpressionLabel

4.6.c) TaskInfluencingVerb  $\rightarrow$  {*allow, assist, cause, enable, prevent*}

Task-influencing verbs are verbs that have an action or a state as "theme". So, for example, in C2LG one cannot say "attack ... enemy ... in order to destroy enemy". Instead, the destruction of the enemy has to be expressed as a task on its own. Say the task to destroy the enemy has been given the label "task-d". Then the original attack order that should result in the destruction of the enemy is expressed as "attack ... enemy ... in order to enable task-d". In this manner, tasks can be chained together so that the logic behind the operation is expressed. Consequently, one can refer not only to other tasks but also to parts of the command intent so that the chain of tasks ultimately results in the desired state expressed in the intent. In general, the task-influencing verbs "allow", "assist", "enable", and "prevent" are followed by a label referring to a task. Normally this is a task assigned to an own unit in paragraph 3b or paragraph 3c. It also might be a task listed as a key task in paragraph 3a "Execution – Intent", or a task described in paragraph 1a "Situation – Enemy Forces" or in paragraph 1b "Situation – Friendly Forces". Especially the influencing verb "prevent" is – of course - often followed by a task whose execution by the enemy has been reported or is anticipated, and is thus mentioned in paragraph 1a. In contrast to the influencing verbs discussed above, "cause" normally is followed by a state which is a desired end state from paragraph 3a.

### 4.6.2 Examples of Purpose Constituents

This section includes two examples: a short one (example 4.6.a) and an elaborated one (example 4.6.b).

### Example 4.6.a)

Example 4.4.c), as presented above, already shows the chaining made feasible by the using the Why constituents. In that example (repeated here as example 4.6.a) the execution of the first assigned task is a precondition for the second: Occupying the bridge enables securing it. Thus, the basic order expression for the occupation task identifies the securing task (labelled by "*label-o24*") as its purpose.

### [task assignment] occupy BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start at TP1 end before TP2 in-order-to enable label-o24 label-o23;

### [task assignment] secure BN-661 Coy2 Prins Willem-Alexander Brug at Parnass start STREND label-o23 label-o24;

### Example 4.6.b)

This example is a longer one. It has been taken from the preparation for NATO RTO MSG-048's 2009 experiment, but has been simplified in some details for clarity. The Combined Arms Battalion 66 is ordered to attack southeast, taking and securing two objectives (objective DOG and objective LION), thereby defeating or bypassing enemy forces. The enemy forces are assumed to have already been reduced to 40% combat effectiveness. Battalion 661 consists of two armoured squadrons (Coy1 and Coy3), each having 14 main battle tanks, two mechanized companies (Coy2 and Coy4) and a recce platoon. The example shows the orders for one of the armoured squadrons (Coy1) and one of the mechanized infantry companies (Coy3) who are ordered to cooperate in coordination. The sketch for an overlay of this operation is given in Figure 1 below for better understanding.

In the example, categorization terms are not included and the labels of the sentence-like expressions are coloured in red while the Why-constituents are coloured in blue. The detailed explanation following the example uses the red coloured labels to refer to the respective sentence-like expressions.

Command intent: secure BN-661 Lion at Lion **ongoing** at TP4 RPTFCT label-ci1;

Tasks assigned to Coy1: deploy BN-661 Coy1 at Denver end before TP0 in-order-to enable label-o11 label-o10; advance BN-661 Coy1 from Denver to Boston start at TP0 in-order-to enable label-o12 label-o11; fix BN-661 Coy1 Red-1-182 at Boston end nlt TP1 in-order-to enable label-o33 label-o12; advance BN-661 Coy1 to Austin start at TP1 in-order-to enable label-o14 label-o13; fix BN-661 Coy1 Red-2-194 at Dog end nlt TP2 in-order-to enable label-o35 label-o14; advance BN-661 Coy1 to Atlanta start at TP2 in-order-to enable label-o16 label-o15; fix BN-661 Coy1 Red-2-196 at Atlanta end nlt TP3 in-order-to enable label-o37 label-o16; advance BN-661 Coy1 to Ruby start at TP3 in-order-to enable label-o18 label-o17; fix BN-661 Coy1 Red-2-191 at Lion end nlt TP4 in-order-to enable label-o39 label-o18; seize BN-661 Coy1 Lion at Lion end nlt TP4 in-order-to cause label-ci1 label-o19; Tasks assigned to Coy3: deploy BN-661 Coy3 at Denver end before TP0 in-order-to enable label-o32 label-o30; support BN-661 Coy3 Coy1 at Troy start at TP0 end at TP4 label-031; attspt BN-661 Coy3 Red-1-182 from Denver to Boston start at TP0 end nlt TP1 in-order-to enable label-o12 label-o32; destroy BN-661 Coy3 Red-1-182 at Boston end nlt TP1 in-order-to enable label-o13 label-o33; attspt BN-661 Coy3 Red-2-194 from Boston to Dog start at TP1 end nlt TP2 in-order-to enable label-o14 label-o34; destroy BN-661 Coy3 Red-2-194 at Dog end nlt TP2 in-order-to enable label-o15 label-o35; attspt BN-661 Coy3 Red-2-196 from Dog to Atlanta start at TP2 end nlt TP3 in-order-to enable label-o16 label-o36; destroy BN-661 Coy3 Red-2-196 at Atlanta end nlt TP3 in-order-to enable label-o17 label-o37; attspt BN-661 Coy3 Red-2-191 from Atlanta to Lion start at TP3 end nlt TP4 in-order-to enable label-o18 label-o38; destroy BN-661 Coy3 Red-2-191 at Lion end nlt TP3 in-order-to enable label-o19 label-o39;

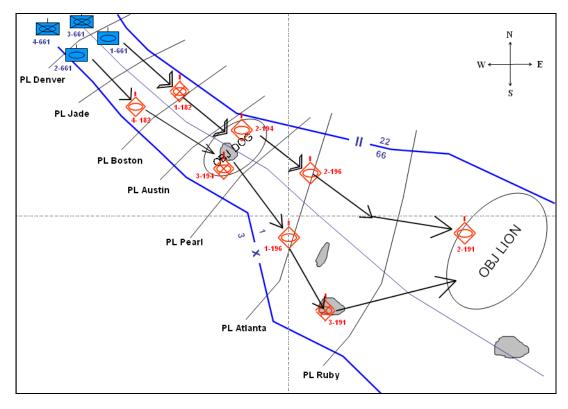


Figure 1: The figure shows the sketch of an overlay for an operation. Part of the tasks assignments of the order by which that operation is ordered is in given in example 4.6.b)

The operation is divided into four phases: from TP0 to TP1, from TP1 to TP2, from TP2 to TP3, and from TP3 to TP4. The points in time, labelled TP0 to TP4, respectively, are specified by exact date times in the underlying database.

The first line of the example, labelled by "*label-ci1*", is part of the command intent. It says that at the end of the ordered operation, at TP4, the Combined Arms Battalion should be in a position such that it is securing objective Lion. The last basic order given to Coy1, labelled by "*label-o19*", has as purpose to cause this state to become true.

The first basic orders to Coy1 and Coy3 (*label-o10, label-o30*) concern the deployment of the units at line of departure "*Denver*". Departure is the precondition for starting the operation. Thus, departure enables the first moving actions towards "*Lion*", the advance (in strength) of Coy1 to phase line "*Boston*" (*label-o11*) and Coy3's supporting attack (*label-o32*) in coordination with Coy1's fixing the enemy unit "*Red-1-182*" (*label-o12*). The advance of Coy1 and the supporting attack of Coy3 both enable the fixing of the enemy ("*Red-1-1-182*").

The second order for Coy3 (*label-o31*) is a more general one, saying that Coy3 should support Coy1 during the whole operation.

Fixing the enemy by Coy1 allows its destruction by Coy3 (*label-o33*). This destruction then enables the advance of Coy1 to phase line "*Austin*" (*label-o13*). From there, enemy unit "*Red-2-194*" can be fixed at objective Dog (*label-o14*). Again, this fixing is supported by Coy3 (*label-o34*). The fixing allows the destruction of the enemy at Dog by Coy3 (*label-o35*). And, again, this enables the further advancing of Coy1 to phase line "*Atlanta*" (*label-o15*). There, another enemy unit ("*Red-2-196*") is fixed by Coy1 (*label-o16*) under support of Coy3 (*label-o36*) allowing the destruction of "*Red-2-1-196*" by Coy3 (*label-o37*). This enables Coy1 to advance to phase line "*Ruby*" (*label-o17*) and start the attack of objective Lion by fixing enemy unit "*Red-2-191*" there (*label-o18*). Coy3 supports the attack (*label-o38*). The fixing of the enemy in the end allows its destruction by Coy3 (*label-039*) which enables Coy1 to seize the objective (*label-019*). As previously stated, this allows securing the objective such that the desired state as formulated in the intent is reached.

### **5** Requests

From the perspective of speech act theory (Searl, 1979), orders and requests are both directives. The purpose of a directive is to get the addressee to execute an action as intended by the speaker (or the sender). The difference between orders and requests lies in the right the sender has to demand the execution of the action in question. With respect to an order, the sender is superior and can expect rightfully that the addressee will execute the tasks that are assigned to him by the sender. In contrast, in a request, the sender is not superior to the addressee and thus cannot expect that requested tasks are executed.

In C2LG, single task assignments ("OB") are not only used for orders but also requests. Of course, the command and control relationship between the tasker organisation and the taskee organisation is different in a request. It should be mentioned, however, that the use of C2LG expressions in the context of complex endeavours (or in other cases in which the communication is among organisations that are not integrated in a command and control

hierarchy), requests and not order are the standard form of directives. In these cases, it is necessary to provide language expressions for confirmations and commissives, as we already recommended in Schade & Hieb (2008). Using a confirmation the addressee of a request can confirm that she received the request whereas a commissive is a speech acts by which one commits oneself to a future course of action. Thus, the addressee of the request can both confirm the request's reception and, by sending a corresponding commissive, she can commit herself to making sure that the request will be fulfilled. Confirmations and commissives thus ensure coordinated action.

When requests, confirmations and commissives are used regularly in inter-organisation communication, all sentence-like C2LG expressions should to be supplemented by categorization terms which are keywords for types of sentence-like expressions. More precisely, they should start with a categorization term ("CT"). The categorization term of a task assignment ("OB") is [*task assignment*], the one for a request is [*request*] and so on. As a result, the rule for a basic request confirmation ("CONB") and the rule form for a basic commissive ("COMB") are as follows:

### $\frac{\text{Rule 3.2.a)}}{\text{CONB}} \quad [request confirmation] \ regarding \text{RequestLabel}$

### Rule Form 3.2.a)

COMB → [*commissive*] TaskingVerb Executer (Affected|Action) Where Start-When (End-When) Mod Why Label

The basic request confirmation consists only of a reference to the request using the request's label ("RequestLabel") preceded by the keyword "regarding", whereas a commissive looks similar to a task assignment. This allows the sender of the commissive to express specifically how she will see the request fulfilled, e.g., which of her subordinated units she would send or to what point in time the execution of the requested tasks is committed. There are a few differences to the basic task assignment, however. First, there is no "Tasker" as the execution of the requested task is not yet assigned. Second, "Taskee" is substituted by "Executer" a term that gives more leeway for establishing the executing unit. "Executer" is also used in Reports and will therefore be discussed in greater detail in the next section, and in even more detail in section 4.2. Third, commissives allow an additional sub-constituent for the modifier constituent – the standard ones are presented and discussed in section 4.5 – which is optional. It corresponds to the extension of "CONB" - without a categorization term similarly consisting of the keyword "regarding" followed by "RequestLabel". By this modifying subconstituent it is possible within the commission itself to refer to the request that initiated the commissive. This is a necessary feature if the execution of more than one action was originally requested. More details about confirmations and commissives are presented in Schade & Hieb (2008).

### **6** Reports

Reports are different from orders and requests. From the perspective of speech act theory, they are assertives. The speaker's (sender's) intention of communicating a report is to inform the addressee that something is so.. Formally, assertives in general and reports in particular are not about getting the addressee to execute an action, although conveying information may result in an action. There are also differences 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 the units executing 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 types of the units involved, 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 C2LG expressions that are sufficient for orders and requests may not be sufficient for such reports. Thus, the C2LG rules and lexical items for reports differ from the rules and lexical items for orders and requests. In the following, we give an updated version of our earlier texts about how to express reports in C2LG, the first among these Schade & Hieb (2007a).

The format of reports is defined by the NATO standard ADatP-3 NATO Message Text Formatting System and the US Field Manual 101-5-2 "U.S. Army Report and Message Formats". These publications provide the doctrinal format for reports and also group reports into various classes.

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

In C2LG, a report consists of its header and its body (see above). The body consists of arbitrarily many basic report lines ("RB"). These correspond to the basic lines for task assignment ("OB") in orders. A basic report line contains one reported fact. According to doctrine, reports differ with respect to these facts. In C2LG, basic report lines differentiate between reports about position or status, reports about events, and reports about military tasks. Thus, the following rules and rule forms hold:

 $\frac{\text{Rule 3.3.a})}{\text{ReportBody}} \rightarrow \text{RB}^*$ 

<u>Rule Forms 3.3.a) to 3.3.c)</u> 3.3.a) (the Rule Form for Task Reports)  $RB \rightarrow TaskingVerb Executer (Affected|Action) Where When Mod (Why) Certainty Label$  3.3.b) (the Rule Form for Event Reports)

 $RB \rightarrow EventVerb$  (Affected|Action) AtWhere When Certainty Label

3.3.c) (the Rule Form for Status Reports)

 $RB \rightarrow$  Hostility Regarding (Identification Status-Value) AtWhere When Certainty Label

The rule form for task reports is similar to the form for task assignment. Once again, a "TaskingVerb" denotes the task. As before, the "TaskingVerb" comes with its own rule that determines the constituents belonging to this verb. The following aspects differentiate the rule form for task reports from the rule form for task assignment: First, there is no Tasker constituent because, for any given reported task, the tasker normally is not known. Second, instead of "Taskee", there is "Executer". The rules for expanding "Executer" allow referring to the organisation that executes the task either a) by name, b) by a description that includes the type of the organisation (e.g., company of mechanized infantry) or c) by referring to the vehicles or weapons systems the organisation operates (cf. section 4.2.1 for details). Third, instead of "StartWhen" and "EndWhen" there is a "When" because reports reflect a specific point in time. Consequently, "When" expresses that something has started, ended, or is ongoing at that specific point in time (cf. section 4.4.1 for details). Fourth, there is "Certainty", a constituent for expressing how certain the reporter is about the truth of the reported information (cf. section 4.7.1).

The rule form for event reports – reports concerning non-military occurrences such as floods, earthquakes, traffic accidents, political demonstration and similar occurrences – resembles the form for task reports. Naturally, "TaskingVerb" is substituted by "EventVerb", while both "Executer" and "Why" are dropped as events are not executed by someone for a reason but simply happen on their own. "Where" is substituted by "AtWhere" because the alternative expansion to "RouteWhere" is, clearly, not appropriate for events. In addition, "Mod" (for modifier) was dropped because currently it can only be expanded to add information about instruments (e.g., vehicles or weapon system) beneficial for executing tasks, about formation (information restricted to the execution of tasks), or about manner (again, specific for task execution).

The rule form for status reports looks different. Instead of starting with a verb ("TaskingVerb" or "EventVerb"), the rule begins with "Regarding". "Regarding" is expanded by one of the following six lexical items: "*position*", indicating that a position is reported in this basic report line a, and five status reports: "*status-general*", indicating that the report concerns the general status of a unit, "*status-person*", concerning one or more persons; "*status-materiel*", concerning inanimate objects (materiel), e.g., a vehicle or a weapon system; "*status-facility*" dealing with facilities; and "*status-task*", used to report about the progression of a task.

In event reports, there are event verbs in place of tasking verbs. The lexical items that are allowed as event verbs ("EventVerb") are those values of JC3IEDM's attribute "action-event-category-code" which are not also values of the attribute "action-task-activity-code". (It may be noted that, in this case, the JC3IEDM follows its own naming code creating an obvious discrepancy between the two domain names "action-task-activity-code" and "action-event-category-code".)

In addition to the verbs that serve as tasking verbs and event verbs, the lexical items that can be used to expand "Regarding" in single status report lines (i.e., "*position*", "*status-general*", etc., cf. section 3.3) play the verb role (and thus the "What") in these lines. "Regarding" is

always preceded by the determiner "Hostility", and generally also followed by the pair "(Identification Status-Value)" (cf. section 3.3). The hostility determiner and the pair "(Identification Status-Value)" are specific for status reports. They cannot be assigned to any of the other building blocks (the constituents that represent the "Who", the "Where", the "When", the "How", the "Why", and reports' "Certainty"). Thus, we will discuss them in this section under the "What" topic.

"Hostility" as the determiner preceding "Regarding" indicates whether the status report line is about own forces or others. It expands to a lexical item which is an element of the value list of JC3IEDM's attribute "object-item-hostility-status-code". Examples of these values are "FR" (*friend*), "*HO*" (*hostile*), "*NEUTRL*" (*neutral*), and "*SUSPCT*" (*suspect*) but there are many more. In addition to the values of "object-item-hostility-status-code" "*OWN*" (*own*) is allowed. The difference between "*OWN*" and "*FR*" is the following: the content of a report marked by "OWN" is about the unit of the sender of the report itself. It therefore is very trustworthy. In contrast, a report marked by "FR" is about some other but friendly unit or organisation and assigned the usual estimation of uncertainty that is common for reports.

In some sense, the hostility determiner is redundant in this position because a hostility statement is also included in denotations of objects that are not referred to by name (cf. section 4.2). Thus, if the status of an organisation, a person, materiel, or a facility is reported and the object in question is not referred to by name, the respective hostility value will be repeated in the object's description. However, having a hostility determiner at the beginning of all status reports helps to identify the branch in the addressee's unit that is chiefly responsible for processing the respective line of the incoming report. For example, if the report is designated "own status-person", the information goes to the G1/S1 branch. If the report is designated "hostile status-general", it goes to the G2/S2 branch and so on.

### 6.1 The "(Identifaction Status-Value)" Pair

In the following, we will discuss the "(Identification Status-Value)" pair in detail.

### 6.1.1 C2LG Rules for the "(Identification Status-Value)" Pair

The "(Identification Status-Value)" pair in status reports denotes the object whose status is reported by "Identification" as well as the status value (by "Status-Value") that is assigned to that object by the report.

If the report is a <u>general status report</u> the "Regarding" of rule form 3.3.c) is set to "statusgeneral", the "Identification" of the form is set to "OrganisationIdentification" and the ol"Status-Value" is set to "OperationalStatus". Thus, the rule for a general status report – matching rule form 3.3.c) – is rule 4.1.a).

### Rule 4.1.a)

# $RB \rightarrow$ Hostility *status-general* OrganisationIdentification OperationalStatus AtWhere When Certainty Label

"OrganisationIdentification" is expanded by an organisation denotation. The rules for such an expansion will be discussed in section 4.2. "OperationalStatus" is a value from JC3IEDM's attribute "organisation-status-operational-status-code". The values are "OPR" (operational),

"SOPR" (substantially operational), "MOPS" (marginally operational), "NOP" (not operational), and "TNOPS" (temporarily not operational).

If the report is a *status report about persons* "Regarding" is set to "*status-person*", "Identification" to "PersonIdentification" and "Status-Value" to "PersonStatus". The corresponding rule is rule 4.1.b).

### <u>Rule 4.1.b)</u>

### RB → Hostility *status-person* PersonIdentification PersonStatus AtWhere When Certainty Label

"PersonIdentification" is expanded by a denotation of a person or a group of persons. This denotation can be the name of the person, if the report is about exactly one person and this person is known, or a sequence of "Count", "Hostility", and "PersonType" as given in rule 4.1.c).

 $\frac{\text{Rule 4.1.c)}}{\text{PersonIdentification}} \rightarrow \qquad \text{Count Hostility PersonType}$ 

"Count" denotes the number of the persons and is expanded to an integer. "Hostility" has previously been discussed. "PersonType" refers to the type of the persons. It is expanded to a lexical item which is an element of the value sets of JC3IEDM's attributes "person-type-category-code" and "person-type-subcategory-code" and "person-type-rank-code". Examples for these values are "CIV" (civilian), "OF5" (colonel), "JRNLST" (journalist), or "VIP" (very important person). "PersonStatus" is a value from the value sets of JC3IEDM's attributes "person-status-duty-status-code", "person-status-physical-status-code", and "person-status-physical-status-code", and "person-status-physical-status-code", "INJRD" (injured), and "WNDD" (wounded). We are using JC3IEDM values as the source for the CBML lexicon. However, we recommend incorporating, at a minimum, the values for triage (Mackway-Jones, Marsden & Windle, 2006) here.

If the report is a *status report about materiel* (or other material objects) "Regarding" is set to "*status-materiel*", "Identification" to "Materieldentification" and "Status-Value" to "MaterielStatus". The corresponding rule is rule 4.1.d).

Rule 4.1.d)

## $RB \rightarrow$ Hostility *status-materiel* MaterielIdentification MaterielStatus AtWhere When Certainty Label

"Materielldentification" is, in the current version of the C2LG, expanded by a denotation of vehicles or weapon systems. This follows the rule for "Theme" as given in section 4.2.1, rule 4.2.h). "MaterielStatus" is a value from JC3IEDM's attribute "materiel-status-operational-status-code". The values are the same as the values from "organisation-status-operational-status-code" (see above). In addition, the values from JC3IEDM's attribute "materiel-status-operational-status-qualifier-code" are also allowed. Examples of these values are "DSTRYD" (destroyed), "IMMBLS" (immobilised), "LST" (lost), and "SCRPPD" (scrapped).

If the report is a *status report about facilities* "Regarding" is set to "*status-facility*", "Identification" to "FacilityIdentification" and "Status-Value" to "FacilityStatus". The corresponding rule is rule 4.1.e).

# $\frac{\text{Rule 4.1.e)}}{\text{RB} \rightarrow \text{Hostility status-facility FacilityIdentification FacilityStatus}}$ AtWhere When Certainty Label

"FacilityIdentification" is expanded by a denotation of a facility. This denotation can be the name of the facility, if the report is about exactly one facility and the name is known, or a sequence of "Count", "Hostility", and "FacilityType" as given in rule 4.1.f).

Rule 4.1.f)

FacilityIdentification  $\rightarrow$  Count Hostility FacilityType

"Count" and "Hostility" have already been discussed. "FacilityType" refers to the type of the facility. It is expanded to a lexical item which is a value of JC3IEDM's attribute "facility-type-category-code". Examples of such values are "*CTT*" (*control tower*), "*FOBSPS*" (*forward observer position*), "*GVTBLD*" (*government building*), or "*POLSTA*" (*police station*). "FacilityStatus" is a value from JC3IEDM's attribute "facility-status-operational-status-code". These values, again, are the same as the values from "organisation-status-operational-status-code" (see above). In addition, the values from JC3IEDM's attribute "facility-status-operational-status-operational-status-qualifier-code" are also allowed. Examples of these values are "*BRNOUT*" (*burned out*), "*CVRFIR*" (*covered by fire*), "*PASABL*" (*passable*), and "*UNCNST*" (*under construction*). Here, we recommend incorporating the HAZUS values (National Institute of Building Sciences, 1997) as well.

If the report is a *status report about the progression of a task* "Regarding" is set to "*status-task*", "Identification" to TaskIdentification" and "Status-Value" to "TaskStatus". The corresponding rule is rule 4.1.g).

 $\frac{\text{Rule 4.1.g})}{\text{RB} \rightarrow \text{Hostility status-task}} \text{TaskIdentification TaskStatus}$  AtWhere When Certainty Label

"TaskIdentification" is expanded by a label that refers to a task. "TaskStatus" is a value from JC3IEDM's attribute "action-task-status-progress-code". Examples of these values are "ABO" (aborted), "COM" (complete), "IPR" (in progress), and "PAU" (paused).

If the report is a *position report*, "Regarding" is set to "*position*" and "Identification" is set to "OrganisationIdentification". "Status-Value" is not necessary because the value in question is the position of the organisation which is already provided by the "AtWhere" in the report. The corresponding rule is rule 4.1.h).

### <u>Rule 4.1.h)</u>

 $RB \rightarrow$  Hostility *position* OrganisationIdentification AtWhere When Certainty Label

"OrganisationIdentification", previously mentioned in connection with rule 4.1.a), is expanded by a denotation of the organisation in question as expressed by rules 4.2.a) to 4.2.e) and 4.2.g), from section 4.2.1.

### 6.1.2 Examples of "(Identification Status-Value)" Pairs

The following shows several report examples which illustrate what "(Identification Status-Value)" pairs in status report lines look like. The first example is an example of a position report. In it, the second company ("Coy2") reports that it is at control point 3.

#### <u>Example 4.1.a</u> [**position report**] own position Coy2 at ControlPoint3 **ongoing** at now fact label-pr-29;

The second example is a general status report. It says that a hostile reconnaissance company is reduced to the status of marginally operational soon before the report is sent.

#### Example 4.1.b)

### [status general report] hostile status-general COY hostile RECCE MOPR at LYNX start at now fact EYOBSN label sr-35;

The third example is a status report about persons. It says that five soldiers below officer rank who belong to a friendly unit but not to the reporting unit are at duty at checkpoint Tango.

#### Example 4.1.c)

[*status person report*] friend status-person 5 friend other ranks ADU at Tango ongoing at now fact EYOBSN label-sr-47;

The fourth example is a status report about materiel. It comes from a patrol that reports that two of its armoured personal carriers have been immobilized under sniper fire.

### Example 4.1.d)

[*status materiel report*] friend status-material 2 friend APC immobilized at ControlPoint3 ongoing at now fact EYOBSN label sr-15;

The fifth example is a status report about a facility, the "Prins Willem-Alexander" bridge at Parnass. The report says that the bridge is operational.

#### Example 4.1.e)

[status facility report] neutral status-facility Prins Willem-Alexander Brug OPR at Parnass ongoing at now fact EYOBSN label-sr-52;

The sixth example is a status report about the progress of a task. The task in question has been labelled "*label-o128*". The task has been assigned to the reporting unit which reports the task just has been completed.

### Example 4.1.f)

### [*status task report*] own status-task label-o128 COM at Damis **end** at now fact label-sr-232;

NOTE: Within the reports above there are some expressions that will only be discussed in detail later in this chapter. This could not be avoided because the examples for building blocks, in this case, the examples of "(Identification Status-Value)" pairs, are always presented in the context of full sentence-like expressions. This is so that the examples are of greater benefit for readers who want to apply C2LG after having read the complete specification and therefore will be later using this document as a reference manual. Let us

repeat, the forms for the rules to build sentence-like expressions – along with example rules – have already been presented in chapter 3. The chapter 3 rules refer to constituents, the building blocks, being presented in detail in this chapter. Next are the rules for denoting organisations.

### 6.2 Denoting the "Who" in Reports

In reports, organisations are denoted as "Executer" or as "Affected". As already mentioned, the reporter does not always know the name of the organisation contained in the report. As a result, reports contain rules for expressing references to organisations, in addition to the rule in which the organisation is referred to by its name.

### 6.2.1 C2LG Rules for Denoting Organisations in Reports

"Executer" refers to the organisation that executes the task contained in a task report. There are three possible expansions of "Executer": when the name of this organisation is known, "Executer" can be expanded by "Taskee" (which is then itself expanded by the unique name of the organisation). Otherwise, it is expanded by either "Agent" or "Theme", both of which are discussed in more detailed below. In these latter two cases, there is the option to add a label which can then serve as a "name" for the organisation.

 $\begin{array}{ll} \underline{\text{Rules 4.2.d) to 4.2.f}} \\ \hline 4.2.d) & \underline{\text{Executer}} & \rightarrow & \underline{\text{Taskee}} \\ \hline 4.2.e) & \underline{\text{Executer}} & \rightarrow & \underline{\text{Agent (Label)}} \\ \hline 4.2.f) & \underline{\text{Executer}} & \rightarrow & \underline{\text{Theme (Label)}} \end{array}$ 

"Executer" is expanded to "Agent" when the name of the executing organisation is not known to the reporter but its unit type is. "Agent" is further expanded by the sequence "Size", "Hostility", and "UnitType".

 $\frac{Rule \ 4.2.g)}{\text{Agent} \ \rightarrow \ \text{Size Hostility UnitType}}$ 

"Size" refers to the size of military units. It expands to a lexical item which is a value of JC3IEDM's attribute "echelon-size-code" (in earlier version of the JC3IEDM this attribute more reasonably had been "unit-type-size-code"). Examples of these values are "BN" (*battalion*), "COY" (*company*), and "PLT" (*platoon*). "Hostility" refers to the perceived hostility status of the organisation. It is expanded as already discussed in section 4.1. "UnitType" refers to the military type of the organisation. It is expanded to a lexical item which is a value of JC3IEDM's attribute "unit-type-arm-category-code". Examples of these values are "ARMOUR" (*armour*), "ARTLYR" (*artillery*), "INF" (*infantry*), and "RECCE" (*reconnaissance*). In the current state of the C2LG, "Agent" can only be used for military organisations. In order to use it for organisations of other types, the necessary lexical items have to be added to the grammar's lexicon.

"Executer" is expanded to "Theme" when the acting organisation can only be identified by the type of main equipment it uses. "Theme" is expanded to a sequence that is analogous to that of "Agent", namely the sequence "Count", "Hostility", and "EquipmentType".

Rule 4.2.h)

Theme  $\rightarrow$  Count Hostility EquipmentType

"Count" refers to the number of the equipment that has been observed. It is expanded to an integer. "Hostility" is as described above for "Agent". "EquipmentType" refers to the type of the equipment under use by the unknown organisation. It is expanded to a lexical item which is, for land-based operations and tasks, a value of JC3IEDM's attributes "vehicle-type-category-code" and "weapon-type-subcategory-code". Examples of these values are "APC" (armoured personnel carrier), "TRUCK" (truck), "ARV" (armoured reconnaissance vehicle), or "HOWIT" (howitzer).

### 6.2.2 Examples for Denoting Organisations

This subsection shows two examples that illustrate denoting an organisation.

Examples 4.2.a) and 4.2.b) (Organisations denoted by "Executer"  $\rightarrow$  "Agent", "Executer"  $\rightarrow$  "Taskee"  $\rightarrow$  "OrganisationName", and "Affected  $\rightarrow$  "OrganisationName", respectively): In the example line it is reported that a team of hostile snipers is ambushing the platoon "PTL-318-2-2" at Control Point 3. In this report, the hostile sniper unit also receives the name "label-en-1" for further reference.

## <u>4.2.a)</u> [*task report*] ambush team hostile sniper label-en-1 PTL-318-3-B at ControlPoint3 ongoing at now fact label-tr-1;

Later, platoon "*PTL-318-3-B*" is relieved and the team of hostile snipers is attacked by the relief force ("COY-318-2"). Beforelong, the snipers cease fire. This is reported by the relief forces making use of the previously assigned name for the sniper team ("*label-en-1*").

### <u>4.2.b)</u> [task report] disengage label-en-1 COY-318-2 at ControlPoint3 ongoing at now fact label-tr-38;

In both reports, the affected organisation is an own organisation (in fact, the reporting unit itself). Thus, in both cases, "Affected" has been expanded by the name of the reporting unit.

### 6.3 Expressing Spatial Aspects in Reports

Spatial constituents are the same whether used in orders, requests, or reports. The rules for building them have been discussed in 4.3. Here is an example illustrating their usage in reports:

Example 4.3.d) ("AtWhere" expanded by naming the location in a report):

The reporting unit forwards the statement of a prisoner of war (POW) saying that the hostile squad labelled "groupX6" that is located behind the town hall is (still, in spite of losses) substantially operational.

[status report] hostile status-gen groupX6 SOPR behind Townhall ongoing at 161330ZMAY08 plausible POW mostly reliable label-r1210924659465;

### 6.4 Expressing Temporal Aspects in Reports

Temporal constituents remain much the same whether they are used in orders, requests, or reports.

The only difference is that, in reports, rather than "StartWhen" and "EndWhen", there is only a "When", which expands as follows.

Rules 4.4.d) to 4.4.f)

4.4.d)When  $\rightarrow$  startatDateTimeValue4.4.e)When  $\rightarrow$  ongoingatDateTimeValue4.4.f)When  $\rightarrow$  endatDateTimeValue

The structure of "When" corresponds to that of "StartWhen" and "EndWhen" described earlier. Once again, there is a keyword, a temporal modifier and a date time value. There are however, two minor differences: First, in the current C2LG version, for single reports the only modifier allowed for "When" is "*at*". Second, in reports, a third keyword "*ongoing*" is allowed in addition to "*start*" and "*end*". It is used for reporting that an action is still ongoing. Again, temporal constituents can be recognized by their keywords.

The following is an example illustrating the usage of temporal constituents in reports.

Example 4.4.b)

The leader of the  $2^{nd}$  company reports to her battalion leader:

"Arrived at Control Point 3; starting to attack snipers."

(The group of snipers was labelled "*label-en-1*" in an earlier communication. Therefore, this label is used in "Affected" in the task report. )

[**position report**] own position Coy2 at ControlPoint3 **start** at now fact label-rpos4; [**task report**] attack Coy2 label-en-1 at ControlPoint3 **start** at now fact label-rtsk4;

### 6.5 Expressing Certainty

In C2LG reports, modifiers are used as in orders whereas purpose constituents are normally not used at all. When purpose constituents are used in reports, it is in the same way as in orders. Thus, for both modifiers and purpose consituents, the rules presented in chapter 4 hold for reports, too. However, there are some constituents specific to reports, namely those that express certainty (or uncertainty).

### 6.5.1 C2LG Rules for Expressing Certainty

The "Certainty" constituent is present in all single report lines. It consists of three subconstituents, namely the mandatory "Credibility", the optional "InformationSource", and the optional "Reliability".

 $\frac{\text{Rule 4.7.a)}}{\text{Certainty}} \rightarrow \text{Credibility} (InformationSource (Reliability))}$ 

"Credibility" expresses the degree of trustworthiness of the information reported as evaluated by the reporter. It expands to one of four values, namely the values "*reported as fact*" (*RPTFCT or "fact*"), "*reported as plausible*" (*RPTPLA or "plausible*"), "*reported as uncertain*" (*RPTUNC* or "*uncertain*"), and "*indeterminate*" (*IND*). "InformationSource" denotes the type of source from which the reporter obtained the information. It expands to one of the values for JC3IEDM's attribute "reporting-data-source-type-code". Typical values are "eyeball observation" (EYOBSN) if the reporter herself witnessed the reported facts happening, "forward observer" (FO), "human intelligence" (HUMINT), or "prisoner of war" (POW).

"Reliability" expresses the degree by which the source can be trusted according to the reporter. This degree of trust somewhat depends on "InformationSource". For example, if "InformationSource" is set to "eyeball observation" it is unlikely that "Reliability" will be given as "unreliable". This dependency can be seen by the parentheses in the "Certainty" rule (above). "InformationSource" is optional, as is "Reliability", however, "Reliability" can only appear when "InformationSource" is present. "Reliability" expands to one of the six values for JC3IEDM's attribute "reporting-data-reliability-code". These values are "completely reliable" (*A*), "usually reliable" (*B*), "fairly reliable" (*C*), "not usually reliable" (*D*), "unreliable" (*E*), and "reliability cannot be judged" (*F*).

### 6.5.2 Examples of Expressing Certainty

For the example, we assume that the reporting unit wants to report that there are five wounded civilians in front of a specific building (Building 2109). Let us assume further that the unit sees these wounded persons when reporting. Then, the report would be either

Examples 4.7.a) and 4.7.b)

[person status report] neutral status-person 5 neutral civilian (label C169) wounded at Building 2109 ongoing at now fact label-r-05;

or

[person status report] neutral status-person 5 neutral civilian (label C169) wounded at Building 2109 ongoing at now fact eyobsn label-r-05;

As the observation is made by the reporting unit itself, it is not necessary to provide a reliability value in the certainty constituent for this report. If, however, the same information is provided by a prisoner of war, the reporter might assume that, while the information might be true ("*plausible*"), the reliability of this prisoner of war cannot be judged ("F"). Thus, the corresponding report would be:

Example 4.7.c)

[person status report] neutral status-person 5 neutral civilian (label C169) wounded at Building 2109 ongoing at now plausible pow F label-r-05;

### 7 Command Intent

Command and Control is an art. There are many unforeseeable aspects that can influence an operation. Explicit and detailed orders quickly become outdated and thus, following them by letter is the surest way to disaster. The answer to this problem is adaptability. Under the heading "Network Centric Warfare" Alberts and Hayes discussed "adaption" as "the ability to alter force organisation and work processes as the situation and/or environment changes"

(2003, p. 153). After the Napoleonic Wars, the Prussian Army established adaptability and agility by "Mission Command", a command style that matches the necessities of network centric operations: "information age [...] creates conditions where such a command philosophy [mission command] is the essential bedrock for success" (Storr, 2003, p. 93). It has to be mentioned that this command style had been at least occasionally used throughout all periods of history. E.g., Alberts and Hayes use Nelson's tactic at the Battle of Trafalgar (1805) as their prime example to illustrate their reasoning in favour of agility (2003, pp. 28ff.). Traditionally, however, Mission Command is attributed to Prussia's supreme commander, Helmuth Karl Bernhard Graf von Moltke (Moltke the Elder) (1800-1891) who established mission command doctrinally (Regling, 1983, p. 384; Borgert, 1983, p. 427f.).

How command intent is represented in C2LG has been discussed in detail in Hieb & Schade (2007). In this chapter, we will summarize and bring up to date the most important aspect given there. Albert and Hayes discuss intent in "Understanding Command and Control" and distinguish between intent, Command Intent, and Commander's Intent (Albert & Hayes, 2006, p. 38). Commander's Intent implies a single individual in command, while Command Intent is a newer term that implies a group or collective making decisions. The term "intent" is more general yet. With respect to C2LG, we prefer (and thus use) "Command Intent". It is, in our opinion, the best term for discussing transitioning from a well-established command process relying upon written and verbal communication to a more flexible command environment that can still support the same processes, but is built upon a formal grammar.

According to Wade (2005, p. 1-22), who uses the classical term, commander's intent consists of three components, namely, the desired *End State*, summing up the conditions under which a mission is accomplished; a list of *Key Tasks*, those tasks that the commander thinks must essentially be met to achieve the desired end state; and, in the context of the broader operational context of the mission, the *Expanded Purpose*. Following this list, we set the following rule for the expansion of command intent ("*Cl*"):

Rules 5.a)

 $CI \rightarrow$  (KeyTasks) EndState (ExpandedPurpose)

With respect to commanding simulated units, only the desired end state is of relevance. The broader picture often will not be part of the simulation. Therefore it will not help to express an expanded purpose for a simulation system. The key tasks would also be "normal" tasks assigned to some units. Thus, they are already given to the simulation system in question, and emphasising them only directs a human's attention to them but does nothing for a system. The desired end state however can be exploited for the simulation if a planning system is added to the architecture that is able to transform the desired states to plans (cf. Nazih & Schade, 2009). Because of these reasons, "ExpandedPurpose" as well as "KeyTasks" are optional whereas "EndState" is mandatory in the rule given above.

As "EndState" represents the desired situation that will be reported if the mission is successfully accomplished, "EndState" can be defined by a sequence of basic report expressions. In these report expressions the temporal constituents refer to a future point in time, when the desired situation should be achieved. Additionally, the certainty constituent in each is set to "*reported as fact*", indicating that, at that time in the future, the desired state can be reported as fact. Key Tasks are, as the name indicates, tasks. Thus "KeyTasks" can be formulated as both task and status basic order expressions. Expanded Purpose is similar to the End State, but expresses more general aspects of the expected final situation. In short, the End State is about the resulting situation from the military perspective whereas the Expanded

Purpose also considers other, e.g., political, consequences and results. Being the description of a state, "ExpandedPurpose" like "EndState" is expanded to a sequence of basic report expressions; however, here one will also find reports of event type referring to political situations, such as "*POW return*" or "*peace conference*". The rules for the parts of command intent therefore are

<u>Rules 5.b) to 5.d)</u>

<u>5.b)</u>	KeyTasks	$\rightarrow OB^{*}$
<u>5.c)</u>	End State	$\rightarrow RB^*$
<u>5.d)</u>	Expanded-Purpose	$\rightarrow \text{RB}^*$

With respect to "KeyTasks" there is an additional feature that has to be considered. In a normal task assignment as expressed in paragraphs 3b) and 3c) of the operation order, tasks are assigned directly to a subordinate unit. Thus, Tasker and Taskee are well known and can be referred to by name. In a key task as expressed in a command intent, however, the commander might order his subordinate to assign the key task in questio, to one of the subordinate's units. In this case, the commander does not specify which unit should execute the task. On the contrary, it is assumed that the subordinate commander will choose the most appropriate of his units. In other words, while expressing a key task in which the final assignment of the task is given to the subordinate, the commander cannot refer to the taskee by name. For this, C2LG uses the replacement term "*OPEN*" that allows for expansion of "Taskee" in any key task expression of a command intent. The following example illustrates the use of this replacement term.

Example 5.a)

This example is based on example 4.6.b). That example shows parts of an order by which the commander of battalion BN-661 commands an operation that ultimately ends with the seizing and securing of objective "Lion". In his own command intent, the commander of brigade BDE-66, superior to the commander of the battalion BN-661, could order the seizing of "Lion" as a key task of the brigade's broader operation by commanding the commander of BN-661 to assign that task to one of his companies, e.g. in order to enable another crucial task.

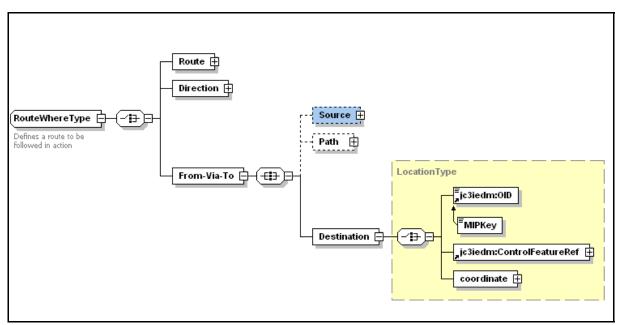
[command intent – keytasks] seize BN-661 OPEN Lion at Lion **end** nlt TP4 **in-order-to** enable label-ci13 label-ci12;

### 8 C2LG Expressed as BML Schemata

In this final chapter, we will illustrate the relationship between C2LG and XML schemata that are based on the grammar.

Once a grammar has been defined, it can easily be transformed into an XML schema. In order to provide an example let us take a look at the "RouteWhere" rules for spatial constituents as rules 4.3.e) to 4.3.g) and 4.3.c) given in section 4.3.1.

<u>Rules 4.3.e) to 4.3.g) and 4.3.c)</u>						
4.3.e)	RouteWhere	→ along RouteName				
4.3.f)	RouteWhere	→ towards Location   towards Bearing				
4.3.g)	RouteWhere	$\rightarrow$ ( <i>from</i> Location) <i>to</i> Location ( <i>via</i> Location*)				
4.3.c)	Location	$\rightarrow$ LocationName   Coord				



A possible transformation of these rules into a schema is shown in Figure 2.

Figure 2: Part of an XML schema that can be developed "directly" out of C2LG's rules for "RouteWhere"

In order to interpret the schema, the following must be noted. First, the schema uses a so called "LocationType" determining that a location is either the name of a JC3IEDM object (JC3IEDM: OID) or a coordinate. This corresponds to the "Location" rule above, although it could be restricted even more by determining that the JC3IEDM object in question has to be of type facility or feature. Second, the keywords that are used in C2LG will be used to identify the corresponding schema elements. Thus, a constituent starting with the keyword "*along*" will use the "Route" element in the schema, a constituent starting with "*towards*" the "Direction" element, a constituent with "*from*" the "Source" element, a constituent with "*to*" the "Destination" element. As determined by the third "RouteWhere" rule above, the latter three are part of the "From-Via-To" element are optional, whereas the "Destination" element is mandatory.

C2LG's rules have been transformed into a schema under the aegis of NATO RTO MSG-048 "Coalition Battle Management Language" (cf. Pullen, Hieb et al., 2008). However, the resulting schemata (the IBML schema and its predecessor the JBML schema) are designed to be used by quite a range of organizations for the exchange of orders and reports among systems. This means, other (not necessarily military) stakeholders have also influenced how the schema should look and what structure it should have so that their own needs are supported. Thus, although the C2LG's rules have by and large shaped the IBML schema there are deviations. Most importantly, in the IBML schema the task verbs do not come with their own sub-schemata. Instead, the rule form for basic order lines (cf. section 3.1) has been transformed into the schema. Thus, it is possible to represent a task assignment in the schema that includes a "RouteWhere" although its "TaskingVerb" - according to the C2LG and according to normal semantic constriction - demands an "AtWhere" (and vice versa). This is not a problem as long as orders and reports are put into the sender system using a C2LG-GUI that allows only those inputs that conform to the grammar rules, or as long as the input is correct, independent of any check. Then the input is transformed into a schema representation that is valid not only according to the schema but also according to the grammar.

### **9** References

ADatP-3 Baseline 11.0.1 (1999). Brussels, Belgium: HQ-NATO C3S-IOB.

Alberts, D.S., Garstka, J.J. & Stein, F.P. (1999). Network Centric Warfare. Washington, DC: CCRP.

Albert, D.S. & Hayes, R.E. (1995). *Command Arrangements for Peace Operations*. Washington, DC: CCRP.

Alberts, D.S. & Hayes, R.E. (2003). Power to the Edge. Washington, DC: CCRP.

Albert, D.S. & Hayes, R.E. (2006). Understanding Command and Control. Washington, DC: CCRP.

Alberts, D.S. & Hayes, R.E. (2007). Planning: Complex Endeavors. Washington, DC: CCRP.

Austin, J.L. (1962). How to do Things with Words. Oxford, UK: Clarendon Press.

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

Borgers, E., Huiskamp, W., de Reus, N. & Voogd, J. (2007). Research and Development towards Applications of MSDL and C-BML in The Netherlands. *2007 Spring Simulation Interoperability Workshop* (= Paper 07S-SIW-025), Norfolk, VA, March 2007.

Borgert, H.-L. (1983). Grundzüge der Landkriegsführung von Schlieffen bis Guderian. In: Militärgeschichtliches Forschungsamt (Ed.), *Deutsche Militärgeschichte 1648-1939*. Hersching, DE: Pawlak.

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

Carey, S., Kleiner, M., Hieb, M.R. & Brown, R. (2001). Standardizing Battle Management Language – A Vital Move Towards the Army Transformation. 2001 Fall Simulation Interoperability Workshop, Orlando, FL, 2001.

Chomsky, N. (1957). Syntactic Structures. Den Haag, NL: Mouton.

Chomsky, N. (1963). Formal properties of grammar. In: Luce, R.D., Bush, R.R. & Galanter, E. (Eds.), *Handbook of Mathematical Psychology*, Vol. II (S. 323-418). New York: John Wiley & Sons.

Chomsky, N. (1965). Aspects of the Theory of Syntax. Cambridge, MA: MIT Press.

Chomsky, N. (1981). *Lectures on Government and Binding: The Pisa Lectures*. Holland: Foris Publications. Als Nachdruck veröffentlicht: Berlin: Mouton de Gruyter, 7. Auflage,1993.

Chomsky, N. (1995). The Minimalist Program. Cambridge, MA: MIT Press.

Davis, D., Blais, C. & Brutzman, D. (2006). Autonomous Vehicle Command Language for Simulated and Real Robotic Forces. *2006 Fall Simulation Interoperability Workshop*, Orlando, FL, September 2006.

De Reus, N., De Krom, P., Mevassvik, O.M., Alstad, A., Sletten, G., Schade, U. & Frey, M. (2008a). BML-enabling of national C2 systems for coupling to Simulation. *2008 Spring Simulation Interoperability Workshop* (= Paper 08S-SIW-095), Providence, RI, April 2008.

De Reus, N., de Krom, P., Pullen, M. & Schade, U. (2008). BML – Proof of Principle and Future Development. *I/ITSEC*, Orlando, FL, December 2008.

Dörge, F.C. (2006). *Illocutionary Acts – Austin's account and what Searle made out of it.* Ph.D. Thesis. Tübingen University. (http://w210.ub.uni-tuebingen.de/dbt/volltexte/2006/2273/)

Fillmore, C.J. (1968). The case for case. In Bach, E. & Harms, R.T. (Eds.), *Universals in Linguistic Theory*. New York: Holt, Reinhart & Winston.

Fillmore, C.J. (1976). Frame Semantics and the Nature of Language. Annals of the New York Academy of Science: Conference on the Origin and Development of Language and Speech, **280**, 20-32.

Gazdar, G. & Mellish, C. (1989). Natural Language Processing in PROLOG: An Introduction to Computational Linguistics. Workingham, UK: Addison-Wesley.

Haegeman, L. (1994). *Introduction to Government and Binding Theory* (2. Auflage). Oxford, UK: Blackwell.

Hieb, M.R, Kleiner, M., Carey, S. & Schade, U. (2009). Characterizing doctrine through a formalization of C2 processes. 14<sup>th</sup> ICCRTS, June 2008, Washington, DC.

Hieb, M.R., Mackay S., Powers, M.W., Harland, Y., Kleiner, M. & Pullen, J.M. (2007). Geospatial Challenges in a Net Centric Environment: Actionable Information Technology, Design and Implementation (= Paper 6578-43). SPIE Defense and Security Symposium, Defense Transformation and Net Centric Systems Conference. Orlando, FL.

Hieb, M.R., Powers, M.W., Pullen, J.M., Kleiner, M. (2006). A Geospatial Battle Management Language (geoBML) for Terrain Reasoning. *11th ICCRTS*, Cambridge, UK, September 2006.

Hieb, M.R. & Schade, U. (2007). Formalizing Command Intent Through Development of a Command and Control Grammar. *12<sup>th</sup> ICCRTS*, Newport, RI, June 2007.

"Homers Ilias" edited by Johann Heinrich Voß, reprinted after the 1882 edition. *Die bibliophilen Taschenbücher*, 73. Dortmund, Germany 1978.

Hügelmeyer, P, Schade, U. & Zöller, T. (2007). Application of BML to Inter-Agent Communication in the ITSimBw Environment. *Winter Simulation Conference*, Washington, DC, December 2007.

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

Kempen, G. & Hoenkamp, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science*, *11*, 201-258.

Kremer, C. (2006). *Eine Untersuchung von Bewegungsverben im KFOR-Korpus im Vergleich zu FrameNet*. Bonn, DE: Universität Bonn, Institute for Communication Sciences, Master Thesis.

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

Mackway-Jones, K., Marsden, J. & Windle, J. (2006). Ersteinschätzung in der Notaufnahme. Das Manchester-Triage-System. Huber, Bern.

Mayk, I., Klose, D., Chan, A., Mai, M. & Negaran, H. (2005). Technical and Operational Design, Implementation and Execution Results for SINCE Experimentation. *10<sup>th</sup> ICCRTS*, Tyson's Corner, VA, June 2005.

Mey, H.H. & Krüger, M.K.-D. (2003). Vernetzt zum Erfolg ? Frankfurt a.M.: Report-Verlag.

Minsky, M. (1975). A framework for representing knowledge. In Winston P.H. (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.

National Institute of Building Sciences (1997). *Earthquake Loss Estimation Methodology HAZUS*, Technical Manual, Vol. I-III.

NATO Military Agency for Standardization (2000). STANAG 2014: Formats for Orders and Designation of Timings, Locations, and Boundaries, 9<sup>th</sup> Edition. Brussels, BE.

Nazih, M. & Schade, U. (2009). Einsatz der Battle Management Language zur Befehligung von Einheiten in einem Simulationssystem. *FKIE-Bericht*, **174**. Wachtberg: FGAN.

Nirenburg, S. & Raskin, V. (2004). Ontological Semantics. Cambridge, MA: MIT Press.

Partee, B.H., ter Meulen, A. & Wall, R.E. (1990). *Mathematical Methods in Linguistics*. Dordrecht: Kluwer.

Perme, D., Hieb, M.R., Pullen, J.M., Sudnikovich, W. & Tolk, A. (2005). Integrating Air and Ground Operations within a Common Battle Management Language. *2005 Fall Simulation Interoperability Workshop*, Orlando FL, September 2005.

Pullen, M., Carey, S., Cordonnier, N., Khimeche, L., Schade, U., de Reus, N., LeGrand, G., Mevassvik, O.M., Cubero, S.G., Gonzales Godoy, S., Powers, M. & Galvin, K. (2008a). NATO MSG-048 Coalition Battle Management Language Initial Demonstration. *2008 Spring Simulation Interoperability Workshop* (= Paper 08S-SIW-082), Providence, RI, April 2008.

Pullen, M., Carey, S., Cordonnier, N., Khimeche, L., Schade, U., de Reus, N., LeGrand, N., Mevassvik, O.M., Cubero, S.G., Gonzales Godoy, S., Powers, M. & Galvin, K. (2008b). NATO MSG-048 Coalition Battle Management Initial Demonstration Lessons Learned and Follow-on Plans. *2008 Euro Simulation Interoperability Workshop* (= Paper 08E-SIW-064), June 2008, Edinburgh, UK.

Pullen, M., Corner, D., Singapogo, S.S., Clark, N., Cordonnier, N., Menane, M., Khimeche, L., Mevassvik, O.M., Alstad, A., Schade, U., Frey, M., de Reus, N., de Krom, P., LeGrand, N. & Brook, A. (2009). Adding Reports to Coalition Battle Management Language for NATO MSG-048. *2009 Euro Simulation Interoperability Workshop* (= Paper 09E-SIW-003), Istanbul, Turkey, July 2009.

Pullen, M., Hieb, M.R., Schade, U., Rein, K., Frey, M. & Orichel, T. (2008). Enabling the MSG-048 Multinational Demonstration 2007 with the Command and Control Lexical Grammar and JBML Web Services. *NATO MSG Conference*, Vancouver, Canada, October 2008.

Pullum, G.K. & Gazdar, G. (1982). Natural languages and context-free languages. *Linguistics and Philosophy*, *4*, 471-504. Reprinted in: Savich, W.J., Bach, E., Marsh, W. & Safran-Naveh, G. (Eds.) (1987), *The Formal Complexity of Natural Languages*. Dordrecht, NL: Reidel.

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

Sadock, J. (2003). Mismatches in Autonomous Modular versus Derivational Grammars. In Francis, E.J. & Michaelis, L.A. (Eds.), *Mismatch: Form-Function Incongruity and the Architecture of Grammar* (pp. 333-354). Stanford, CA: CSLI.

Sadock, J. (2006). Speech Acts. In: Horn, L.R. & Ward, G. (Eds.), *The Handbook of Pragmatics*. Malden, MA: Blackwell.

Schade, U. & Hieb, M.R. (2006a). Formalizing Battle Management Language: A Grammar for Specifying Orders. Paper 06S-SIW-068, *2006 Spring Simulation Interoperability Workshop* (= Paper 06S-SIW-068), Huntsville, Alabama, April 2006.

Schade, U. & Hieb, M.R. (2006b). Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports. 11<sup>th</sup> ICCRTS, Cambridge, UK, September 2006.

Schade, U. & Hieb, M.R. (2007a). Battle Management Language: A Grammar for Specifying Reports. Paper 07S-SIW-036. 2007 Spring Simulation Interoperability Workshop (= Paper 07S-SIW-036), Norfolk, VA, March 2007.

Schade, U. & Hieb, M.R. (2007b). Improving Planning and Replanning: Using a Formal Grammar to Automate Processing of Command and Control Information for Decision Support. *The International C2 Journal*, **1**(**2**), 69-90.

Schade, U. & Hieb, M.R. (2008). A linguistic basis for multi-agency coordination. *Proceedings of the 13<sup>th</sup> International Command and Control Research and Technology Symposium*, June 2008, Bellevue, WA.

Searl, J.R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, UK: Cambridge University Press.

Searl, J.R. (1979). *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge University Press.

Sells, P. (1987). *Lectures on Contemporary Syntactic Theories* (= CSLI Lecture Notes, 3). Stanford, CA: CSLI.

Shieber, S.M. (1985). Evidence against the context-freeness of natural languages. *Linguistics and Philosophy*, **8**, 333-343. Reprinted in: Savich, W.J., Bach, E., Marsh, W. & Safran-Naveh, G. (Eds.) (1987), *The Formal Complexity of Natural Languages*. Dordrecht, NL: Reidel.

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

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

Sudnikovich, W., Pullen, J.M., Kleiner, M. & Carey, S. (2004). Extensible Battle Management Language as a Transformation Enabler. *SIMULATION*, 80, 669-680.

Sudnikovich, W., Ritchie, A., de Champs, P., Hieb, M.R. & Pullen, J.M. (2005). NATO Exploratory Team – 016 Integration Lessons Learned for C2IEDM and C-BML. *2005 Spring Simulation Interoperability Workshop*, San Diego, CA, April 2005.

Tolk, A., Diallo, S. & Turnitsa, C. (2006). Merging Protocols, Grammar, Representation, and Ontological Approaches in support of C-BML. *2006 Fall Simulation Interoperability Workshop*, Orlando, FL, September 2006.

Tolk, A., Diallo, S. & Turnitsa, C. (2007). A System View of C-BML. 2007 Fall Simulation Interoperability Workshop (= Paper 07F-SIW-054), Orlando, FL, September 2007.

USA, Headquarters, Department of the Army (1999). Field Manual 101-5-2: US Army Reports and Message Formats. Washington, DC.

USA, Headquarters, Department of the Army (2001). Field Manual 3-0 (FM 3-0): Operations. Washington, DC.

USA, Headquarters, Department of the Army (2005). *Field Manual 5-0 (FM 5-0): Army Planning and Orders Production*. Washington, DC.

Wade, N.M (2005). *The Battle Staff SMARTbook: Doctrinal Guide to Military Decision Making and Tactical Operations, 2<sup>nd</sup> Revised Edition.* Lakedale, FL: The Lightning Press.

Weber, D. (2004). Ms. Midshipwoman Harrington. In Fawcett, B. (Ed.), *The Warmasters*. Riverdale, NY: Baen.

### **Appendix A: List of Tasking Verbs and their Rules**

The appendix consists of three lists of the rules matching rule form 3.1. Each rule refers to one tasking verb marked in blue. The tasking verbs in the first list refer to ground operation tasks, the ones in the second list to air operation tasks, and the ones in the third list to those tasks that might be relevant for crisis relief operations.

Rules for ground operation tasks

OB →	advance		Taskee		Route-Where		(End-When)		
$OB \rightarrow$	ambush			Affected	At-Where		(End-When)	•	
$OB \rightarrow$	arrest(legal)	Tasker	Taskee	Affected	(At-Where)	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	arrest/obstruct	Tasker	Taskee	Object	(At-Where)	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	assemble	Tasker	Taskee	Material	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	attack	Tasker	Taskee	Affected	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	attack	Tasker	Taskee	Affected	Route-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	avoid	Tasker	Taskee	Action	(At-Where)	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	block	Tasker	Taskee	Affected	At-Where		(End-When)		
OB →	breach	Tasker	Taskee	(Affected)	At-Where		End-When	Mod Why	
OB →	build-up			Material	At-Where		End-When	Mod Why	
OB →	canalize			Affected	Route-Where		(End-When)	Mod Why	
OB →	capture			Material	(At-Where)		(End-When)	Mod Why	
OB →	clear(land)		Taskee	material	At-Where		End-When	Mod Why	
$OB \rightarrow OB \rightarrow$	clear(obstacle)			Material	At-Where		End-When	Mod Why	
$OB \rightarrow OB \rightarrow$	concentrate		Taskee	Wateria	At-Where		End-When	Mod Why	
OB → OB →	conduct			Affected	Route-Where				
$OB \rightarrow OB \rightarrow$							(End-When)	Mod Why	
	confiscate			Material	(At-Where)	Start-When		Mod Why	
OB →	consolidate		Taskee	Ohiast	At-Where		(End-When)	Mod Why	
OB →	constitute		Taskee	•	At-Where		End-When	Mod Why	
OB →	contain			Affected	At-Where		(End-When)	Mod Why	
OB →	counter attack			Affected	At-Where		(End-When)	Mod Why	
OB →	cover			Affected	At-Where		(End-When)	Mod Why	
$OB \rightarrow$	defeat			Affected	At-Where		(End-When)	Mod Why	
$OB \rightarrow$	defend			Affected	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	deflect	Tasker	Taskee	Action	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	demolish	Tasker	Taskee	Affected	Route-Where	Start-When	End-When	Mod Why	Label
$OB \rightarrow$	deny	Tasker	Taskee	Affected	At-Where	Start-When	(End-When)	Mod Why	
$OB \rightarrow$	deploy	Tasker	Taskee		At-Where	Start-When	End-When	Mod Why	Label
$OB \rightarrow$	destroy	Tasker	Taskee	Affected	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	disengage	Tasker	Taskee	Affected	(At-Where)	Start-When	End-When	Mod Why	Label
$OB \rightarrow$	disrupt	Tasker	Taskee	Affected	(At-Where)	Start-When	(End-When)	Mod Why	
$OB \rightarrow$	distribute	Tasker	Taskee		At-Where	Start-When	End-When	Mod Why	
$OB \rightarrow$	envelope	Tasker	Taskee	Affected	Route-Where		(End-When)	Mod Why	
OB →	escort	Tasker	Taskee	Affected	Route-Where		(End-When)	Mod Why	
OB →	establish			Feature	At-Where		(End-When)	Mod Why	
OB →	evacuate		Taskee		At-Where		(End-When)	Mod Why	
OB →	exploit		Taskee	•	(At-Where)		(End-When)	Mod Why	
OB →	fix			Affected	At-Where		End-When	Mod Why	
$OB \rightarrow$	follow and assum				Action		(End-When)	Mod Why	
$OB \rightarrow OB \rightarrow$	follow and suppor				Action		(End-When)	Mod Why	
$OB \rightarrow$ $OB \rightarrow$	quard			Affected	At-Where		(End-When)	Mod Why	
OB → OB →	harass			Affected	At-Where		(End-When)	Mod Why	
OB → OB →	hide		Taskee	Anecleu	At-Where		· /		
$OB \rightarrow OB \rightarrow$	hold defensive			Facility	At-Where		(End-When)	Mod Why Mod Why	
OB → OB →							(End-When)		
	hold defensive			Feature	At-Where		(End-When)	Mod Why	
OB →	hold offensive			Facility	At-Where		(End-When)	Mod Why	
OB →	hold offensive			Feature	At-Where		(End-When)	Mod Why	
OB →	identify		Taskee	Object	(At-Where)		(End-When)	Mod Why	
OB →	illuminate		Taskee		At-Where		(End-When)	Mod Why	
OB →	infiltrate			Affected	At-Where		(End-When)	Mod Why	
OB →	isolate			Affected	At-Where		(End-When)	Mod Why	
OB →	mob up			Affected	At-Where		(End-When)	Mod Why	
$OB \rightarrow$	march		Taskee		Route-Where		(End-When)	Mod Why	
$OB \rightarrow$	move	Tasker	Taskee		Route-Where		(End-When)	Mod Why	
$OB \rightarrow$	observe			Facility	At-Where		(End-When)	Mod Why	
$OB \rightarrow$	оссиру	Tasker	Taskee	Facility	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	patrol	Tasker	Taskee	-	Route-Where		(End-When)	Mod Why	Label
$OB \rightarrow$	penetrate	Tasker	Taskee	Affected	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	plan	Tasker	Taskee	Action	(At-Where)		End-When	Mod Why	Label
	-				. ,			,	

$OB \rightarrow$	pursue	Tasker Taskee	Affected	(Route-Where)	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	reconnaissance	Tasker Taskee		At-Where	Start-When	End-When	Mod Why	Label
$OB \rightarrow$	recover	Tasker Taskee	Object	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	reinforce	Tasker Taskee	Affected	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	rest	Tasker Taskee		At-Where	Start-When	End-When	Mod Why	Label
$OB \rightarrow$	screen	Tasker Taskee	(Affected)	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	secure	Tasker Taskee	Facility	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	secure	Tasker Taskee	Feature	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	seize	Tasker Taskee	Feature	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	set up	Tasker Taskee	Facility	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	support	Tasker Taskee	Affected	At-Where	Start-When	(End-When)	Mod Why	Label
$OB \rightarrow$	withdraw	Tasker Taskee	(Affected)	Route-Where	Start-When	(End-When)	Mod Why	Label

Rules for air operation tasks

$\textbf{OB} \ \rightarrow$	fly_route	Tasker Taskee	RouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	aerial refuelling	Tasker Taskee Affected	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	air defence	Tasker Taskee Aff_Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	airborne assault	Tasker Taskee Aff_Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	airborne c2	Tasker Taskee Aff Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	airdrop	Tasker Taskee Aff Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	air interdiction	Tasker Taskee Affected	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	airlift	Tasker Taskee Affected	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	air superiority	Tasker Taskee Aff_Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	air-to-air sweep	Tasker Taskee Aff_Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	clear air	Tasker Taskee Aff_Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	close air support	Tasker Taskee Aff Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	combat air patrol	Tasker Taskee Aff Area	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	communication r	elay Tasker Taskee	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	defensive counte	er air Tasker Taskee Affected	d AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	offensive air supp	port Tasker Taskee Affected	d AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	offensive counter	r air Tasker Taskee Affected	AirRouteWhere	StartWhen (EndWhen)	Mod Why Label
$OB \rightarrow$	scramble	Tasker Taskee	OnStation	StartWhen (EndWhen)	Mod Why Label
OB →	suppress enemy	air defence Tasker Taskee A	ff_Area AirRouteW	/here StartWhen (EndWhen)	Mod Why Label

### Rules for crisis relief operation tasks

~ ~			• •		
OB →	abort	Tasker Taskee		(At-Where)	Start-When (End-When) (Mod) (Why) Label
OB →	alert	Tasker Taskee		(At-Where)	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	arrest(legal)	Tasker Taskee		(At-Where)	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	assemble	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	assist	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	be on standby	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	clear	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	decontaminate	Tasker Taskee	Affected	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	decontaminate	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	defuse	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	demolish	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	dismount	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	dispense	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	drive	Tasker Taskee		Route-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	escort	Tasker Taskee	Affected	Route-Where	Start-When (End-When) (Mod) (Why) Label
OB →	establish	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	evacuate	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	extinguish fire	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	follow	Tasker Taskee		(At-Where)	Start-When (End-When) (Mod) (Why) Label
OB →	qo	Tasker Taskee		Route-Where	Start-When (End-When) (Mod) (Why) Label
OB →	guide spiritually	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	hand over	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	hand over	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	illuminate	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
OB →	install	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	lift	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	maneuver	Tasker Taskee		Route-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	measure	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow OB \rightarrow$	mount	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
OB → OB →	nurse	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow OB \rightarrow$	observe	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow OB \rightarrow$					
$OB \rightarrow OB \rightarrow$	obstruct	Tasker Taskee		At-Where	Start-When (End-When) (Mod) (Why) Label
$OD \rightarrow$	operate	Tasker Taskee	Object	At-Where	Start-When (End-When) (Mod) (Why) Label

$OB \rightarrow$	park	Tasker Taskee	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	patrol	Tasker Taskee	Route-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	plan	Tasker Taskee Action	(At-Where)	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	prevent distributi	ion Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	protect	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	provide	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	pump	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	pursue	Tasker Taskee Affected	(Route-Where)	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	rally	Tasker Taskee	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	recover	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	remove	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	rescue	Tasker Taskee Affected	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	scatter	Tasker Taskee	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	scout	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	screen	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	search	Tasker Taskee Affected	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	search	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	sight	Tasker Taskee Affected	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	slaughter	Tasker Taskee Object	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	supply	Tasker Taskee Affected	At-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	task	Tasker Taskee Affected	Action (At-Where)	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	transport	Tasker Taskee Affected	Route-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	transport	Tasker Taskee Object	Route-Where	Start-When (End-When) (Mod) (Why) Label
$OB \rightarrow$	treat medically	Tasker Taskee Affected	At-Where	Start-When (End-When) (Mod) (Why) Label