A Grammar for Battle Management Language

Bastian Haarmann\textsuperscript{1}
Dr. Ulrich Schade\textsuperscript{1}
Dr. Michael R. Hieb\textsuperscript{2}

\textsuperscript{1}Fraunhofer Institute for Communication, Information Processing and Ergonomics
\textsuperscript{2}George Mason University

bastian.haarmann@fkie.fraunhofer.de
ulrich.schade@fkie.fraunhofer.de
mhieb@gmu.edu
A Grammar for Battle Management Language

Overview

- Battle Management Language (BML)
- Formal Grammars
- What BML Requires from its Grammar
- Implications from the Requirements
A Grammar for Battle Management Language

Battle Management Language (BML or Coalition-BML / C-BML)
unambiguous language for C2-communication

It is mainly used for commanding simulated units, to train staffs or as decision support application.

Exchange of
- Orders
- Requests
- Reports

Figure by Ole Martin Mevassvik (FFI)
NATO RTO MSG-048 "C-BML"
A Grammar for Battle Management Language

Battle Management Language (BML)
A Grammar for Battle Management Language

Battle Management Language (BML)

Battle Management Language is under development by NATO RTO, Modeling and Simulation Group 048 “Coalition BML” (2006-2010) and currently by its successor MSG-085 “Standardization for C2-Simulation Interoperation“.

It is under standardization by the Simulation Interoperability Standardization Organization (SISO).
Responsible: The Product Development Group “C-BML”
BML expressions must be interpretable by a system.

A formal language is the set of all sentences that can be generated out of a formal grammar.

Thus, we need a formal grammar as basis for BML.
Noam Chomsky asked:

„What do we know when we know a language?”
(Syntactic Structures, 1957)

Chomsky’s answer was:

- a lexicon (= a set of words) and
- a set of rules (= how to combine the words)
Grammar Research that supports SISO C-BML Phase 2

Formal Grammar: Definition

A formal grammar is a quadruple, $G = \{S, \Sigma, N, P\}$.

- $S$ is the start symbol
- $\Sigma$ is a finite set of terminal symbols (the lexicon)
- $N$ is a finite set of non-terminal symbols (Who, Where, …)
- $P$ is a finite set of production rules
A Grammar for Battle Management Language

Formal Grammars

Production rules
terminal- and non-terminal symbols → terminal- and non-terminal symbols

E.g., S → What Where When

Restriction: The left-hand side of a rule must contain at least one non-terminal symbol.
A Grammar for Battle Management Language

Requirements

There are two types of requirements:

- doctrinal requirements
- requirements of interoperability
A Grammar for Battle Management Language

Doctrine for military communication:

- field manuals
- NATO standard agreements (STANAGS)

Example: NATO’s STANAG 2014 “Formats for Orders and Designation of Timings, Locations and Boundaries” and US Army’s Field Manual 6-0 “Mission Command: Command and Control of Army Forces”
A Grammar for Battle Management Language

Doctrinal Requirements

Paragraph 3 “Execution” in the format for operation orders:
Tasks are assigned to units

**Doctrinal** refers to the “5 Ws”:

- **What** (what kind of task is to be executed)
- **Who** (the unit that is ordered to execute the task)
- **Where** (the spatial conditions of the task)
- **When** (the temporal conditions of the task)
- **Why** (the task’s purpose)
A Grammar for Battle Management Language

Doctrinal Requirements

Constituency

→ some words in an expression belong together

To respect the doctrinal demand of using the 5 Ws, BML expressions should consist of constituents.
Formal grammars can be classified hierarchically in four levels:

- unrestricted
- context-sensitive
- context-free
- regular

Regular expressions are best for automatic processing, but to allow constituency, it is necessary to use a context-free grammar.
A Grammar for Battle Management Language

Doctrinal Requirements
A Grammar for Battle Management Language

Interoperability Requirements

**JC3IEDM**: Joint C3 Information Exchange Data Model primary standard for exchanging C2 data between C2 systems (www.mipsite.org).

BML is supposed to be JC3IEDM compatible.
A Grammar for Battle Management Language

Interoperability Requirements

BML’s vocabulary (the \( \Sigma \) of the BML grammar) consists of attributes and values from the JC3IEDM.

The structure cannot be mirrored in BML’s production rules in total.
So far, a BML grammar should look like the following:

- It should have context-free production rules.
- Its lexical terms should be taken from the JC3IEDM.
A Grammar for Battle Management Language

Unambiguity Requirements

BML expressions should not be ambiguous.

The interpretation of a language expression is done by assigning semantic roles to constituents.

Example: “The unit advanced to phase line Alpha.”
The unit = Agent
to phase line Alpha = Destination
A Grammar for Battle Management Language

Unambiguity Requirements

**Word order** and **keywords** support the correct assignment of semantic roles to constituents

“to phase line Alpha” = Destination
“from phase line Alpha” = Origin

Passive voice or pronouns blur the assignment. The 5 W’s constituents should be fix in BML expressions.

If sub-roles of *When* and *Where* are used, the respective constituents should start with a specific unique keyword to indicate the sub-role meant.
A Grammar for Battle Management Language

Implications

So far, a BML grammar should look like the following:

- It should have context-free production rules
- Its lexical terms should be taken from the JC3IEDM
- Its non-terminals should denote semantic roles
- The sequence of the constituents should be fixed
- If the sub-roles of When and Where are used, the respective constituents should start with a specific unique keyword to indicate to which sub-role a specific constituent belongs
A Grammar for Battle Management Language

Intermediary Result

BML phase 1 standard “light”
- grammar in the form of an XML schema
- suggested by George Mason University
- complies with the requirements
Lexicalization:

The type of action determines which kind of constituents may appear in a respective expression.
C2LG Lexicalized Rules

OB  →  block Tasker Taskee Affected AtWhere
          StartWhen  EndWhen  Mod  Why  Label

OB  →  escort Tasker Taskee Affected RouteWhere
          StartWhen  EndWhen  Mod  Why  Label

RouteWhere  →  along RouteName
RouteWhere  →  towards Location | towards Bearing
RouteWhere  →  (from Location) to Location (via Location*)
Location  →  LocationName | Coordinate

Example:  A orders B to escort 2./LogBtl168(DEU):

escort A  B  2./LogBtl168(DEU)  What / Who / Affected
from Tango to Tulip  Where
start at now  When
in-order-to enable label-o169  Why
label-o168;  Label
A Grammar for Battle Management Language

„Command and Control Lexical Grammar“ (C2LG)

C2LG is a context-free lexicalized grammar.

It allows the expression of orders, requests and reports.

C2LG is built on

- military doctrines,
- the concept of the 5 Ws,
- the JC3IEDM (providing the lexical items), and
- principles of linguistics.
A Grammar for Battle Management Language

„Command and Control Lexical Grammar“ (C2LG)

C2LG is modeled on Lexical Functional Grammar (LFG). It incorporates

- **lexicality** (task driven)
- **constituency** (5Ws)
- the principle of coherence (semantic check)
- the principle of completeness (semantic check)

We recommend a lexicalized grammar such as the C2LG as a basis for the SISO C-BML standard, phase 2 (grammar phase).
Thank you for your attention!

bastian.haarmann@fkie.fraunhofer.de
A Grammar for Battle Management Language

C2LG example (backup slide)

Example

A orders B to occupy a certain building:

occupy A  B  Building2109  
at Melkar Square  
start at now  
in-order-to enable label-o169  
label-o168;  
What / Who / Affected  
Where  
When  
Why  
Label
Interpretation ambiguities may arise if they are not excluded by the grammar.

For example, there are semantic overlaps between “AffectedWho” and “CandidateTargetList” as well as between “Resource” and “TaskeeWho” in BML, version 1 standard “full”.