
Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests and Reports

*Presented to the 11th International Command and Control
Research and Technology Symposium*

Paper I-069

Dr. Ulrich Schade

FGAN-FKIE

GERMANY

schade@fgan.de

Dr. Michael Hieb

George Mason University

US

mhieb@gmu.edu

Content

1. Motivation
2. The Need for a C2 Grammar
3. A BML Tasking Grammar
4. Beyond Tasking
5. Illustration by Example
6. Conclusion

Motivation

Coalition Interoperability

requires **unambiguous communication**.

Network Centric Operations will only increase the need for more precise communications.

Military communication consists of

- Orders
- Requests
- Reports

We will argue that one needs a formal language to communicate unambiguously. To this end, we have developed Battle Management Language (BML).

Motivation

A Formal Grammar - Syntax and Semantics

- Formal Languages provide a rigorous framework for automated processing.
- The military domain provides excellent structure to terms and actions in a formal language.
- Current Message and Data-based communications do not go far enough – a grammar is needed to give additional meaning.
- The lack of a formal grammar for Orders has led to a focus on Situation Awareness rather than Decision Support

Motivation

Desired features for such a language:

- **formal**
in order to enable automatic processing
- **unambiguous**
in order to avoid misunderstandings
- **expressive**
in order to convey commander's intent

Motivation

A formal language is defined by a ***grammar***.

The grammar provides

- **a lexicon**

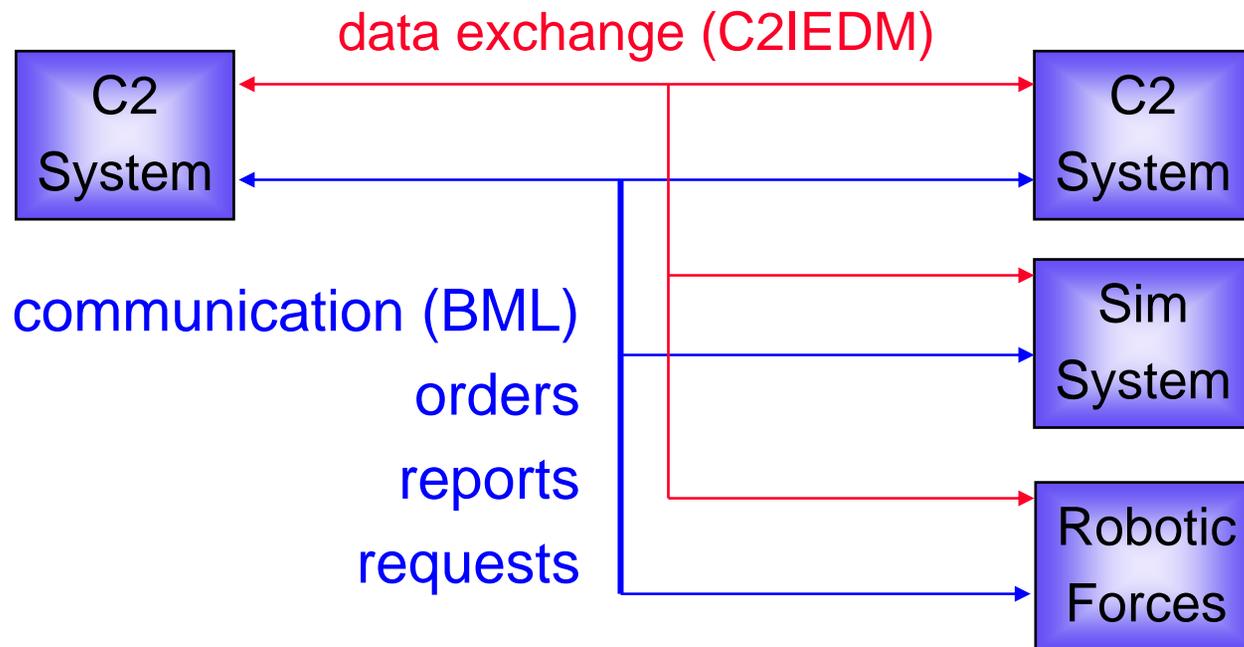
in order to determine the words which may be used as well as their semantics (their meaning);

- **a finite set of rules**

in order to determine how to concatenate the words and to give meaning to the concatenations.

Definition: BML

BML is an unambiguous formal **language** used for the command and control of forces and equipment conducting military operations. It also is for operating simulation systems and future robotic forces.



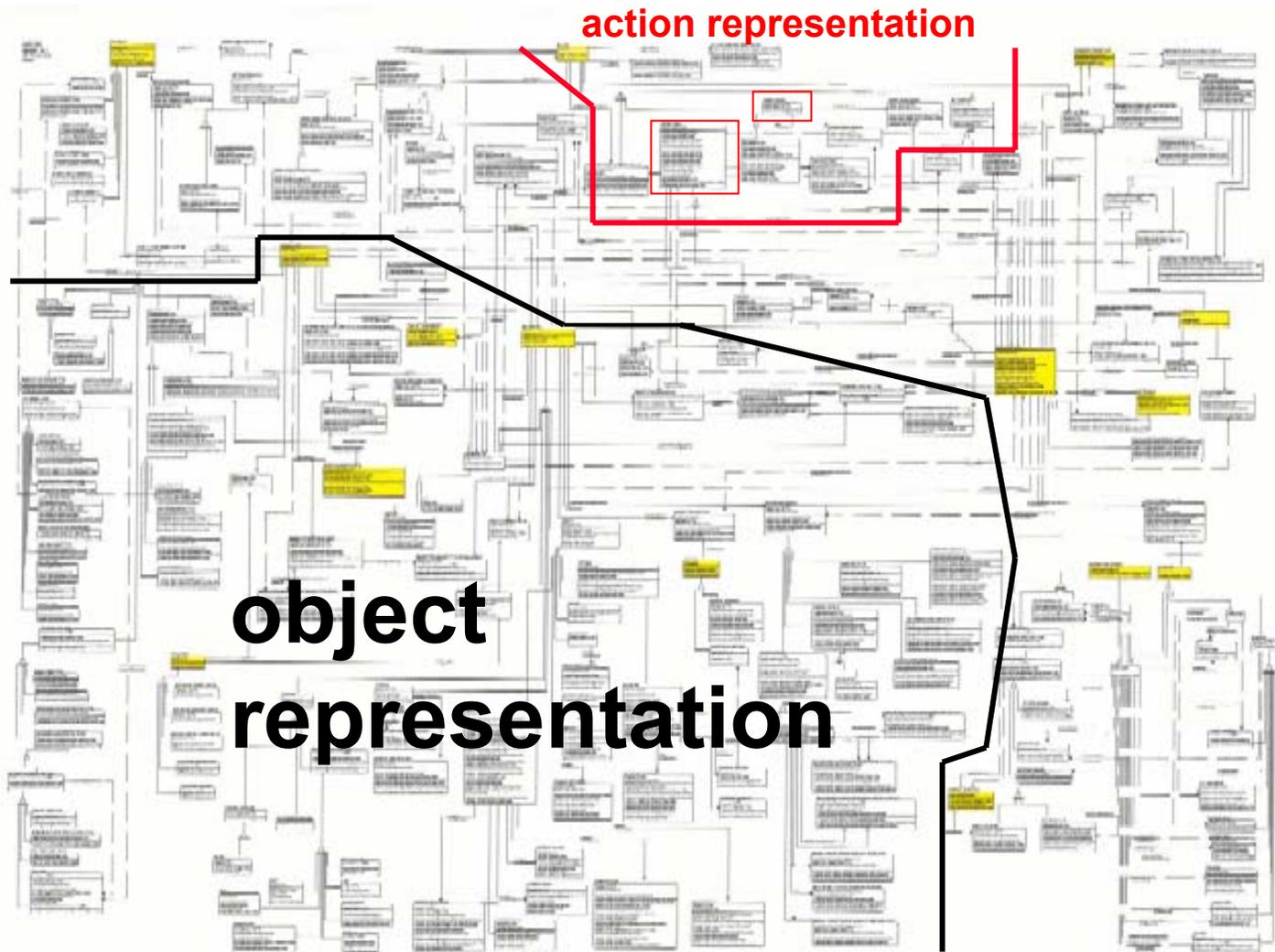
The need for a C2 Grammar

The Command and Control Information Exchange Data Model (**C2IEDM**) provides a standard Command and Control **Lexicon**.

However, the C2IEDM cannot serve as language.

- It is object-centric (noun-centric), whereas languages are action-centric (verb-centric).
- It violates the principles of completeness and coherence.
- It does not provide meaning to concatenations.

The need for a C2 Grammar: The object-centric view of the C2IEDM



The need for a C2 Grammar: Completeness

Tasks are listed and verbally defined in the C2IEDM table
“**action-task-category-code**”

Example: **advance**

In C2IEDM, version 6.1.5e, its meaning is given as:

*“To move forward **towards an objective** in some form of tactical formation. This is a transitional phrase between operations which may or may not result in contact with the enemy.”*

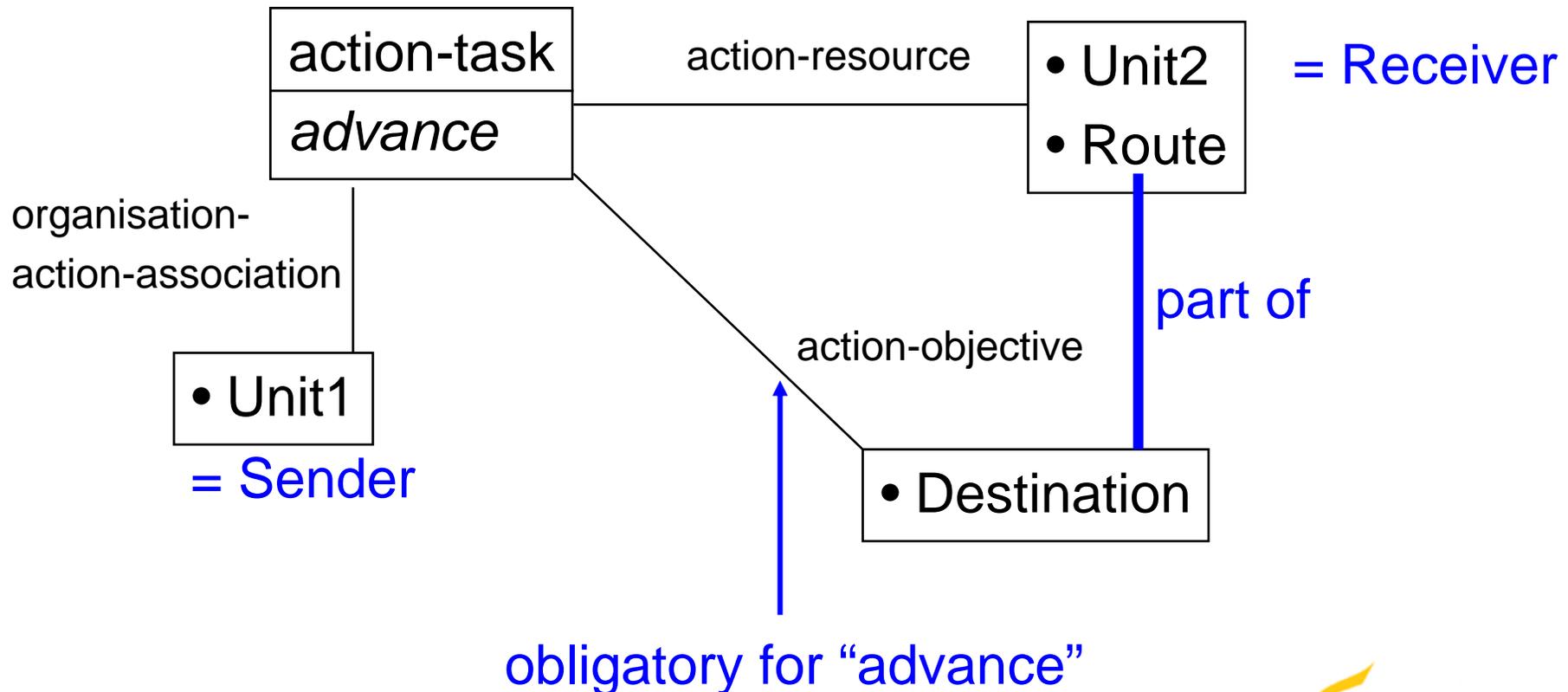
This meaning is for humans, not for machines.

The machines have to be informed that advance needs an objective.

The need for a C2 Grammar: Completeness

Example (C2IEDM: black – missing: blue)

“Advance from assembly area Alpha to phase line Tulip!”

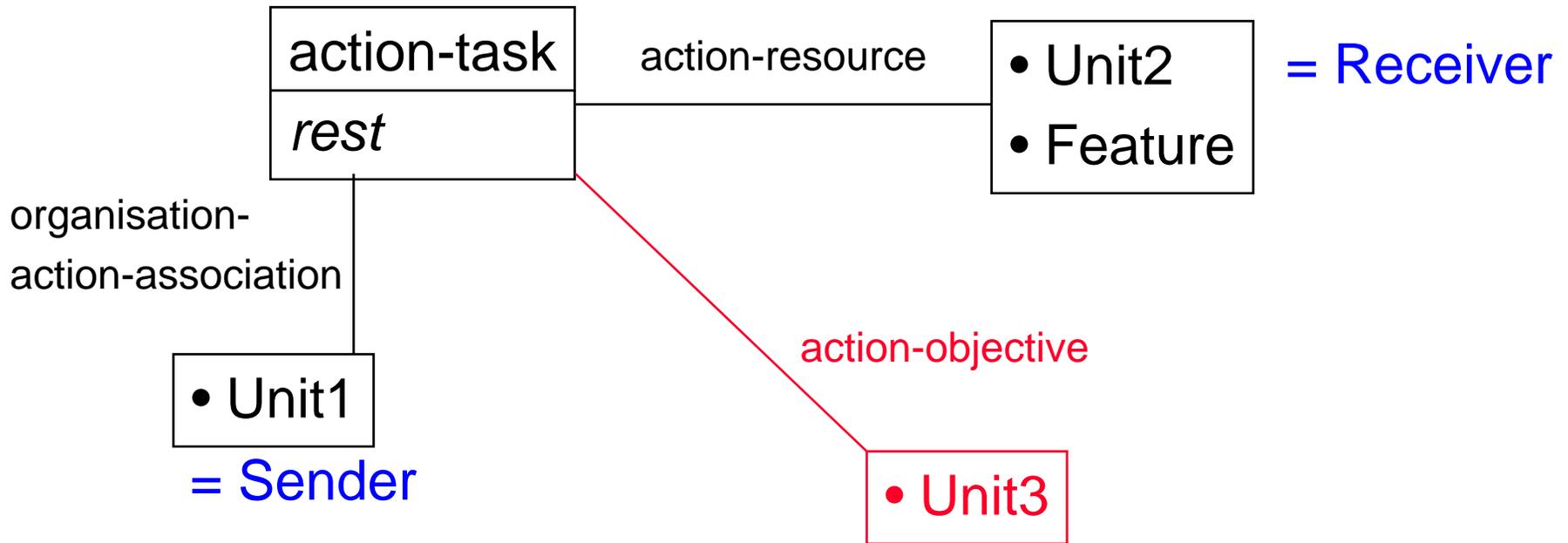


The need for a C2 Grammar: Coherence

(C2IEDM: black – missing: blue – red: superfluous)

“Rest at Fox!”

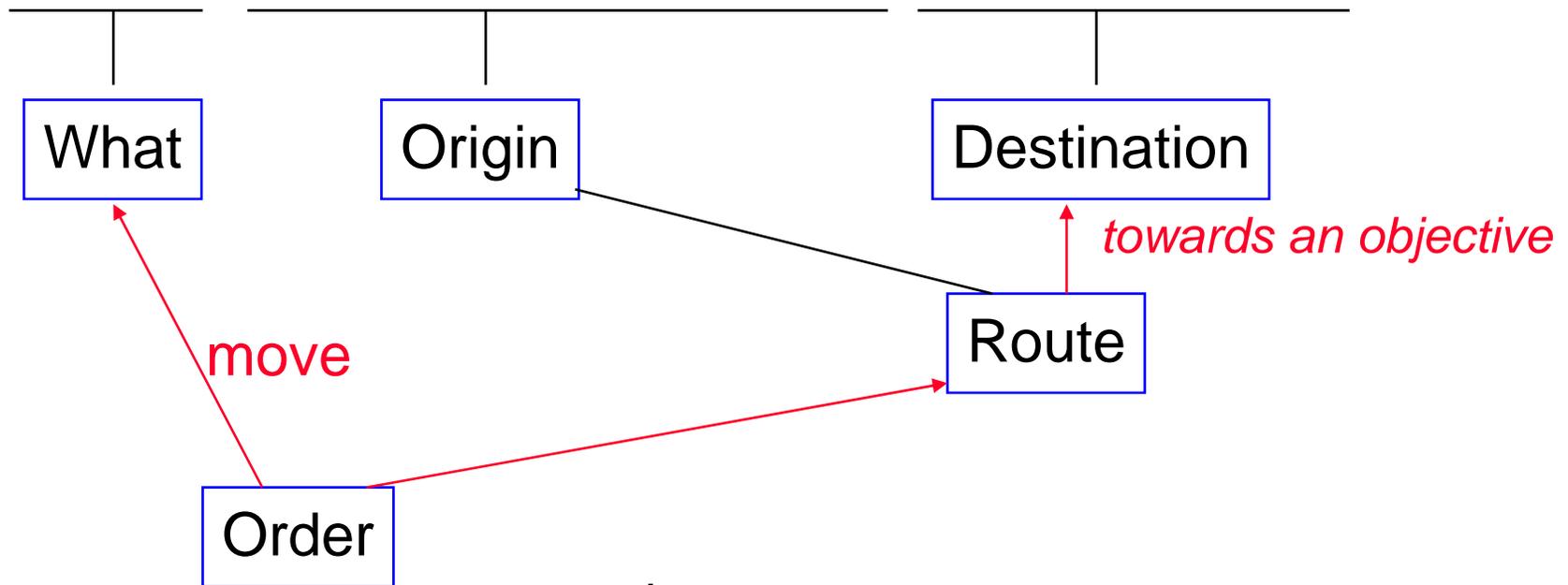
(rest: “... out of contact with enemy”)



The need for a C2 Grammar

The concatenation of meaning

“Advance from assembly area Alpha to phase line Tulip!”



advance:

“move ... towards an objective”

A BML Tasking Grammar

The production rules for the **basic expressions** have the following general form:

$B \rightarrow$ Verb Tasker Taskee (Affected | Action)
Where Start-When (End-When) Why Label (Mod)*

“Verb” is an action, normally a task;

“Tasker” is a “Who”, the unit which commands the task;

“Taskee” is a “Who”, the unit which executes the task;

“Affected” is a “Who”, the unit which is affected by the task;

“Action” is another action/task affected by the task;

A BML Tasking Grammar

The production rules for **basic expressions** have the following general form:

$B \rightarrow$ Verb Tasker Taskee (Affected | Action)
Where Start-When (End-When) Why Label (Mod)*

“Where” is a “location phrase”;

the “When”s are “time phrases”;

“Why” gives the purpose of the action;

“Label” is a label given to the task in order allow it to be referred in other basic expressions.

A BML Tasking Grammar

The production rules for **basic expressions** have the following general form:

$B \rightarrow$ Verb Tasker Taskee (Affected | Action)
Where Start-When (End-When) Why Label (Mod)*

Whether there is “Affected” or “Action” is determined by the verb. This is indicated by the round brackets. The Verb also determines the kind of Where (At-Where or Route-Where) to be used.

A BML Tasking Grammar

Rules for **basic expressions** (examples)

(“verbs” are taken from C2IEDM-table “action-task-category-code”)

B →	<i>advance</i>	Tasker	Taskee		Route-Where	Start-When	(End-When)	Why	Label
B →	<i>ambush</i>	Tasker	Taskee	Affected	At-Where	Start-When	(End-When)	Why	Label
B →	<i>assist</i>	Tasker	Taskee	Action	At-Where	Start-When	(End-When)	Why	Label
B →	<i>attack</i>	Tasker	Taskee	Affected	Route-Where	Start-When	(End-When)	Why	Label
B →	<i>block</i>	Tasker	Taskee	Affected	At-Where	Start-When	(End-When)	Why	Label
B →	<i>defend</i>	Tasker	Taskee	(Affect.)	Route-Where	Start-When	(End-When)	Why	Label

Rules for **constituents** (examples)

Start-When → *start* Qualifier1 Point_in_Time

Start-When → *start* Qualifier2 Action

Qualifier1 → { *AFT, ASAP, ASAPNL, ASAPNL, AT, BEF, NLT, NOB* }

C2IEDM-table “action-task-start-qualifier-code”

A BML Tasking Grammar

Rules for **constituents** (examples, continued)

At-Where → *at* **Location**

Route-Where → **(Source) Destination (Path)**

Route-Where → *along* **Route**

Route-Where → *towards* **Direction**

Source → *from* **Location**

Destination → *to* **Location**

Path → *via* **Location***

A BML Tasking Grammar

The **main rule** (for an order, part 3 “Execution”)

S → **B*** **Coord_Space*** **Coord_Time***

B is a **basic expression**

Coord_Space is a **spatial coordination**

Coord_Time is a **temporal coordination**

Examples for **coordination rules**

Coord_Space → *area of interest* Tasker Taskee Start-When (End-When) Label

Coord_Time → *start* Qualifier2 Action Action

Beyond Tasking

A BML grammar must also cover **requests** and **reports**.

- With respect to requests, the same basic rules can be used as for orders.
- The only difference here is the hierarchical relation between the Tasker and the Taskee.

Beyond Tasking

Rules for reports are somewhat more tricky, however.

First, there is the difference between reporting a **task** and reporting an **event**:

The “event”-rule type:

RB → Verb (Affected | Action) Where When ...

Example

RB → **flood** At-Where Start-When ...

Beyond Tasking

The “task”-rule type:

RB → Verb Executer (Affected | Action) Where When ...

Example

RB → **rest** Executer At-Where Start-When ...

Beyond Tasking

Second,

“Executer” in the “task”-rules for reports differs from “Taskee”.
It can be expanded in numerous ways:

- Executer → Taskee (Executer is known by name.)
- Executer → Agent (Executer is identified by type.)
- Executer → Theme (Executer is identified by equipment used.)

Theme → Count Hostility Equipment_type
four hostile battle tanks

Beyond Tasking

Third,

report expressions include a modality term.

The modality term is called “Certainty” and its values are taken from C2IEDM’s “reporting-data-credibility-code”:

RB → Verb ... When (Why) **Certainty** Label (Mod)*

Certainty → RPTFCT (= reported as fact)

Certainty → RPTPLA (= reported as plausible)

Certainty → RPTUNC (= reported as uncertain)

Certainty → IND (= indeterminate)

Illustration by Example

Example: [MIP-Exercise, Ede, NL, Nov. 2003](#)

Extract of an (Section 3b) by MND-West (SP) to

13 NL MECH BDE:

PH 1a: Fast Tactical March to PL TULIP by ROUTE DUCK.

PH 1b: Defense in depth sector EAST, blocking penetration ALFA.

PH 1c: Assist the rearward passage of the 12 (SP) CAV. RGT.

PH 2: On order attack in direction ECHO.

PH 3: Be prepared to conduct peace support ops
along the border within boundaries.

Illustration by Example

Translation to BML

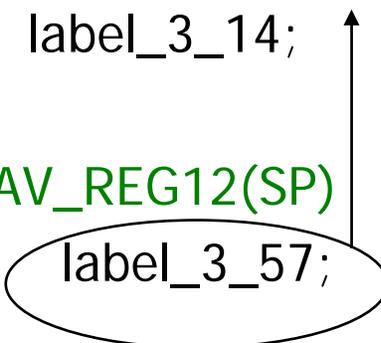
march MND-West(SP) MECH_BDE13(NL)
along DUCK *start nlt* phase1a label_3_11;

defend MND-West(SP) MECH_BDE13(NL)
at EAST *start nlt* phase1b label_3_12;

block MND-West(SP) MECH_BDE13(NL) MIR320(ZB)
at TULIP *start nlt* phase1b label_3_13;

assist MND-West(SP) MECH_BDE13(NL) label_3_57
at EAST *start nlt* phase1c label_3_14;

withdraw MND-West(SP) CAV_REG12(SP)
to EAST *start nlt* phase1c label_3_57;



Conclusion

Current Military Communications rely upon a mix of Text and Structured Messages.

While extremely expressive, this communication is not easily processed, especially in Coalition Operations.

Developing a formal language for Orders, Requests and Reports (BML) is necessary for both semantic interoperability and better decision support.

BML better supports Network Centric Operations with the ability to tailor information to military tasks.



Thanks for Your Attention !

**Questions and Comments
are appreciated.**