

Towards an Ontology for Army Battle C2 Systems

Dr. Ulrich Schade

FGAN

FKIE – ITF

Structure of the Talk

- Definition: Ontology
- About Knowledge
- Building Up the Ontology
 - The Project “*Sokrates*”
 - The Body Structure of the Ontology
 - Examples
- Future Prospects

Definition: Ontology

Gruber (1993): “An ontology is an **explicit** specification of a **shared conceptualization**.”

explicit:

Everything the system has to know must be represented in the system.

shared conceptualization:

In particular, even the information human beings do **not** communicate (since it is self-evident for us), has to be represented.

About Knowledge

➤ Declarative Knowledge

→ *facts (encyclopedic knowledge)*

→ *situational knowledge / context*

→ *rules*

➤ Procedural Knowledge

→ *how to act*

→ *how to apply rules*

About Knowledge

Specific facts are stored in the **data base**.

- *The 3./PzBtl84 is attached to the PzGrenBtl332.*

General facts are stored in the **data base** as well as in the **ontology**.

- *Each unit has a commander.*

The **Relations among Facts** are stored in the **ontology**.

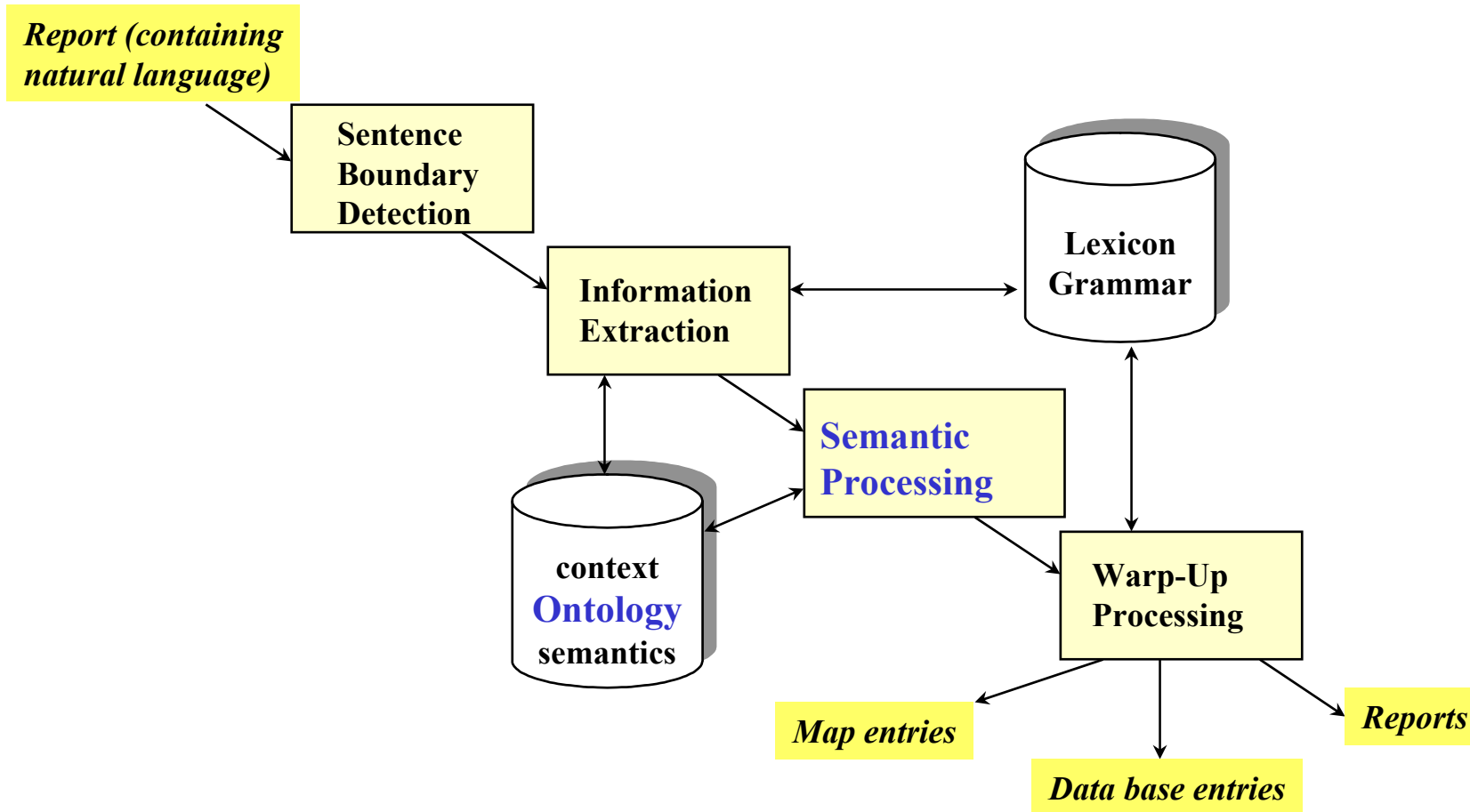
- *If unit A is under operational control of unit B then unit B exercises operational control on unit A.*

Building Up the Ontology

The Ontology, this talk is about, is meant to be part of an army battle C2 system. Its **domain** therefore is the battlefield.

To be more precise, we use the ontology as part of the so-called **Sokrates-system**, a system under development.

The Sokrates-system



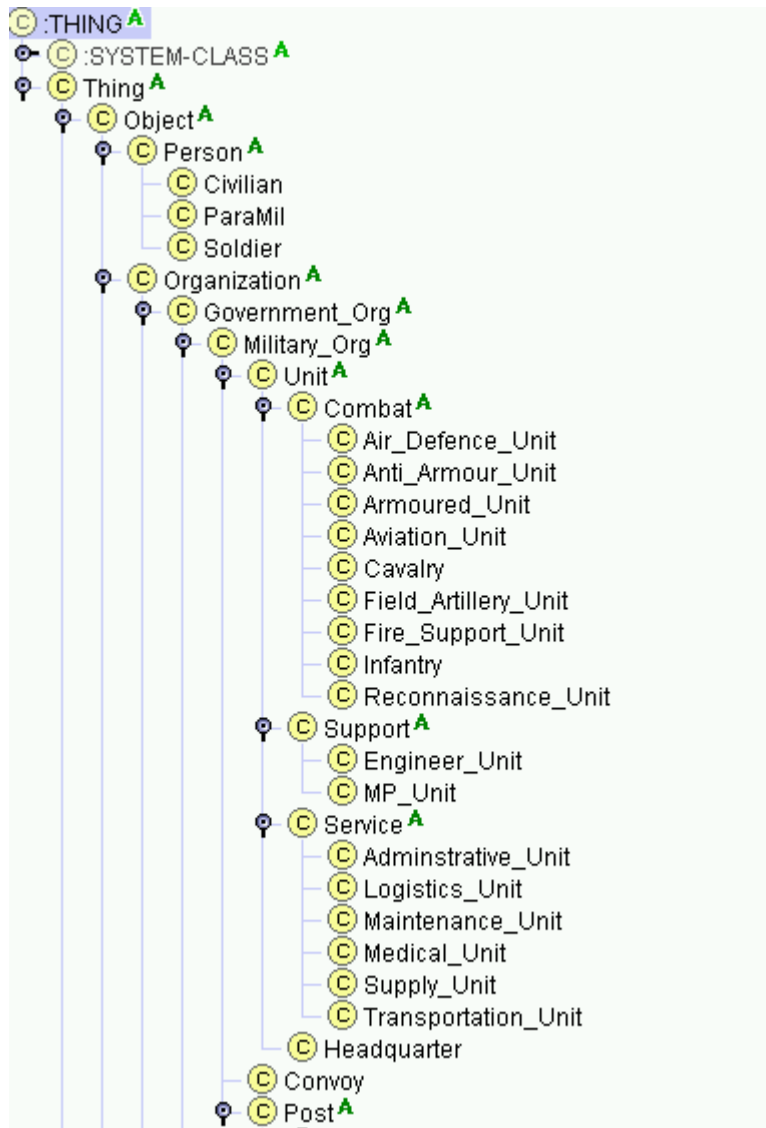
The Sokrates-system: components

<u>Components</u>	<u>Task</u>
<i>Sentence Boundary Detection</i>	The report is resolved into sentences.
<i>Information Extraction</i>	The content is represented by formal means .
<i>Semantic Processing</i>	The formal representation is enriched .
<i>Warp-Up Processing</i>	Map entries as well as data base entries are generated.

The Body Structure of the Ontology

- Hierarchy of **Classes** (Inheritance !)
- **Instances**
- **Attributes: Feature-Value-Pairs,**

- **Constraints**
- **Reasoning-Mechanisms**

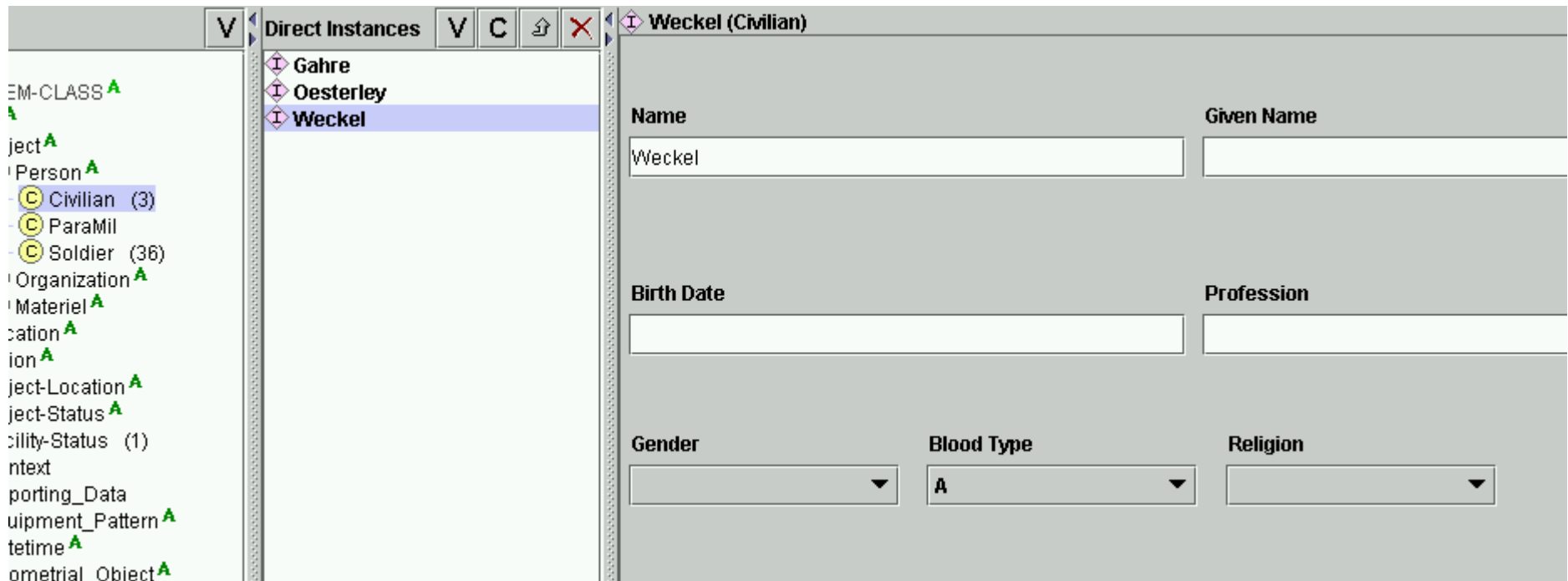


The hierarchy of classes

represents the
“X is a Y”-relation.

Name	Type	Cardinality	Other Facet
S minefield_persistence	Symbol	single	allowed-values={permanent,re
S minefield_depth	Symbol	single	allowed-values={surface,subs
S minefield_stopping_power	Float	single	minimum=0.0, maximum=0.1
S minefield_purpose	Symbol	single	allowed-values={heavy_tactica
S minefield_pattern	Symbol	single	allowed-values={regular,thicke
S minefield_type	Symbol	single	allowed-values={anti_person,ε
S forefront	Symbol	single	allowed-values={N,NE,E,SE,S,
S monitor	Instance	single	classes={Unit}
S status	Float	single	minimum=-1.0, maximum=1.0
S state	Symbol	single	allowed-values={planned,on,o
S center_longitude	Float	required single	
S code_name	String	single	
S center_latitude	Float	required single	
S name	String	single	
S activation_value	Float	single	
S buffer_input	Float	single	
S buffer_activation	Float	single	
S center_line	Instance	required multiple	classes={Point}
S width	Float	required single	minimum=0.0
S dimensionality	Float	required single	minimum=0.0, maximum=3.0

Each class has its specific attributes.
 The attribute values are restricted.
 Value restrictions are constraints.



Most of the instances are provided by the underlying data base. The core data base matches the **LC2IEDM** which also provides class names, attribute names, and allowed attribute values.

Example 1: Attachment

The screenshot displays a software interface for managing military units. On the left, a hierarchical tree shows the structure of units, starting from 'LASS' and branching down to 'Unit' and 'Combat'. The main area shows the 'Direct Instances' of a selected unit, '2./PzBtl84 (Armoured_Unit)'. The right panel provides detailed information for this unit, including its name, nationality, size, commander, and a list of units it consists of or is subordinate to. A blue arrow points from the 'Abgegebene Us' field to the '5./PzGrenBtl332-ZugD' entry in the 'Unterstellte Us' list.

The ontology enforces that changes of operational control result in records for **all** the units (instances) affected by the change

Example 1: Attachment

The screenshot displays a software interface for unit management. On the left, a tree view shows a hierarchy of units under 'Direct Instances'. The selected unit is '5./PzGrenBtl332'. The right pane shows the details for this unit, including its name, nationality (GE), size (COY), commander (Weniger), and a list of units it consists of (5./PzGrenBtl332-ZugD, 5./PzGrenBtl332-ZugC, 5./PzGrenBtl332-ZugB, 5./PzGrenBtl332-ZugA). The 'Abgegebene Us' field lists '5./PzGrenBtl332-ZugD' and '5./PzGrenBtl332-ZugC'. A blue arrow points from the text below to this field.

This automatically includes the unit which loses the control.

Example 2: Agent of an Action

(Military) tasks are resolved by units.

An analysis of the order

“C to all, position 3.1 advance, over!”

by syntactic means, however, suggests that it is the position 3.1 which has to advance.

Example 2: Agent of an Action

The screenshot shows a software interface for defining an ontology class. On the left is a tree view of the ontology hierarchy, with 'Action' selected. The main window displays the details for the 'Action' class, which is a 'NETWORK-CLASS'. The 'Name' field contains 'Action'. The 'Role' is set to 'Abstract'. The 'Template Slots' table lists the following slots:

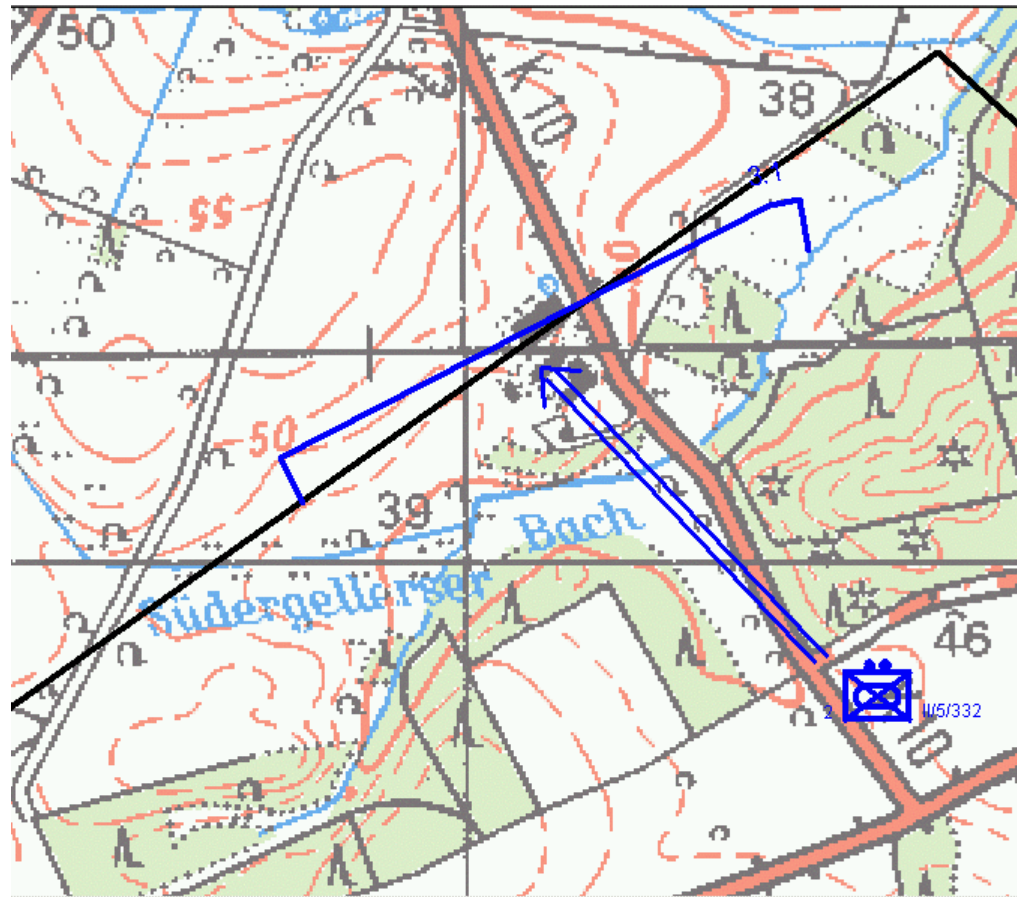
Name	Type	Cardinality	Other Facets
start_time	Instance	single	classes={Moment}
end_time	Instance	single	classes={Moment}
code_name	String	single	
duration	Instance	single	classes={Interval}
agent	Instance	single	classes={Person, Organization}
activation_value	Float	single	

The 'agent' slot's constraint 'classes={Person, Organization}' is circled in blue, with a blue arrow pointing from the text below to it.

The ontology determines that actions are carried into execution by persons or organizations, only. This is restricted even further in the case of military tasks.

Example 2: Agent of an Action

Thus,
misinterpretations
are avoided
from the outset.



Example 3: Intrinsic Reference

Local references often take the **speaker** or the **hearer** as origin of the coordinate system (**deictic reference**):

*“The sniper is behind the tree [from **my/your** point of view].”*

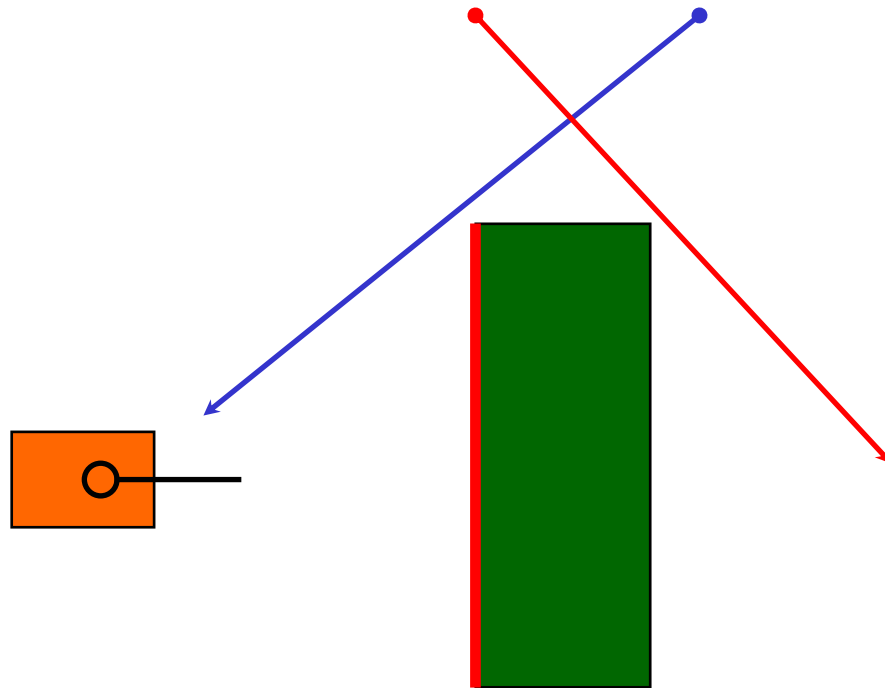
In some cases, however, the origin of the coordinate system is the **relatum** (**intrinsic reference**):

*“The tank stopped in front of **the city hall**.”*

Example 3: Intrinsic Reference

Ambiguities have to be resolved.

(Is the reference **deictic** or **intrinsic** ?)



- reporting unit:
*“Battle tank stopped
in front of barrier.”*

Example 3: Intrinsic Reference

Sperre_315 (Minefield)

Name	Code Name
Sperre_315	Sperre_315

Status: State: on Center Line:
sp315w
sp315e

Forefront: N Dimensionality: 2.0

Monitor: [V] [C] [+] [-] Center Latitud:

Minefield Type: Minefield Depth:

Minimal requirement:

Objects allowing for **intrinsic reference**

must be represented together with their orientation.

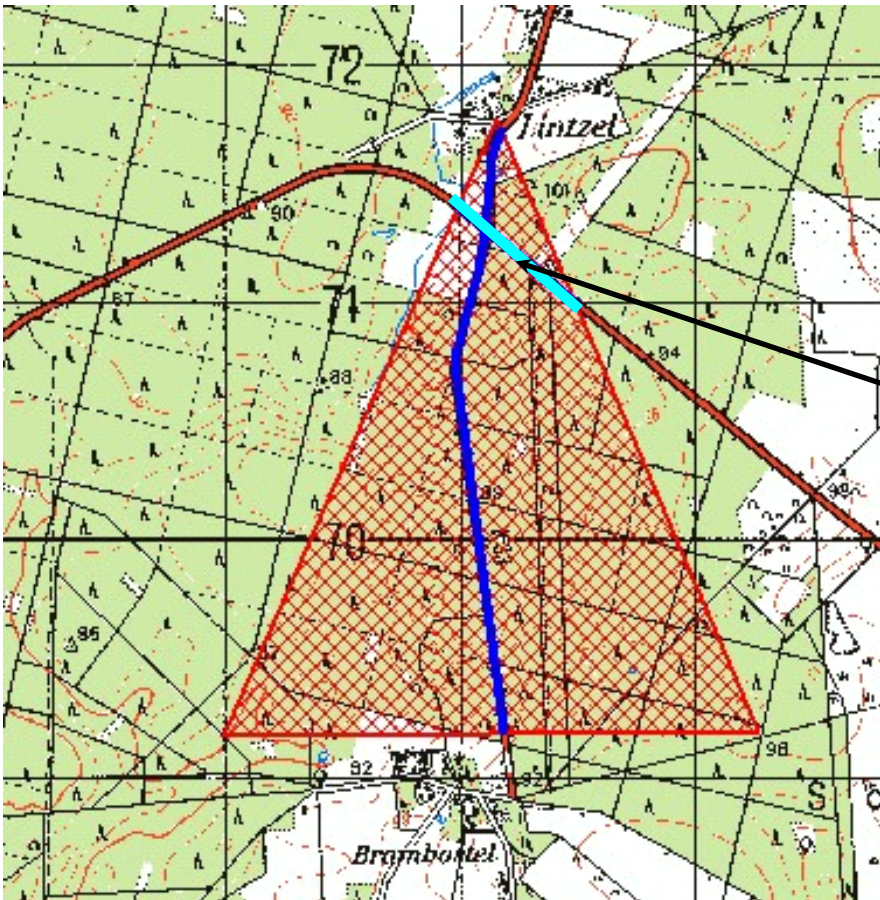
Example 4: Vague Reference

Sometimes, references are vague:

“South of Lintzel, 18 trucks, moving westwards.”

In order to place a symbol on the map,
vague references
have to be transformed into coordinates.

Example 4: Vague Reference



On the basis of the vague reference “*South of Lintzel*” a sector is calculated. Within this sector, road sections are marked, and the most probable one is determined.

The center point of the determined road section is calculated and the respective coordinates are used for mapping.

Future Prospects: Example 5

Part of the functionality provided by the ontology
will be inspired by

psycholinguistic models of language processing:

Kintsch, W. (1998): *Comprehension*. Cambridge University Press.

Step 1

The tank commander reports to the platoon leader:

“2 in position, no enemy contact, over.”

Future Prospects: Example 5

Step 2

Information extraction transforms the report into a formal representation.

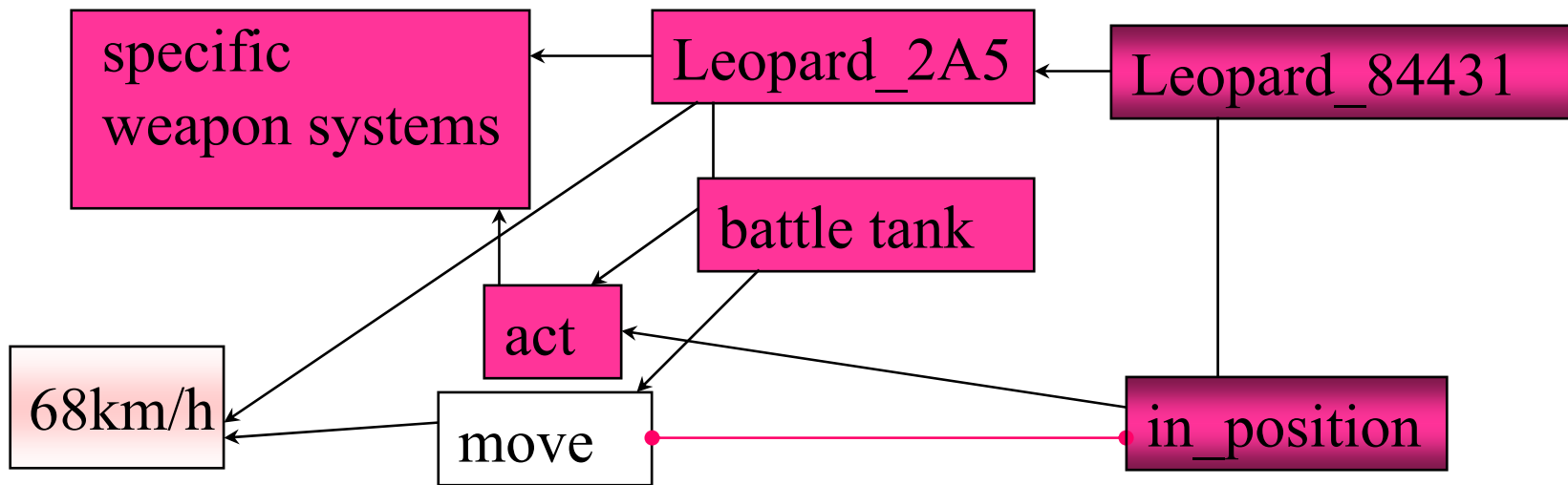
The representation is enriched by the ontology.

This includes the explication of elliptical terms.

Future Prospects: Example 5

Step 3

The ontology operates as (neural) network. Entities mentioned in the report become active. Active items spread their activation. Thus, associated items also become active.



Future Prospects: Example 5

Step 4

Within the ontology network, a tank representation stabilizes. The representation focuses the tank's capability to act and ignores its capability to move.

This representation is used to analyze following reports.

If no contradicting facts are reported, the tank is assumed to stay in the reported position.