Workshop on Transforming
and Weaving Ontologies in Model Driven
Engineering, Denver
2009-10-04

Mapping MOF-based Requirements Representations to Ontologies for Software Reuse

Katharina Wolter*, Michal Smialek**, Lothar Hotz*,
Stephanie Knab*, Jacek Bojarski**, Wiktor Nowakowski**

(*) HITeC, University of Hamburg, Germany
(**) Warsaw University of Technology, Poland

Overall Approach

Goal:

- Support reuse of software

Approach:

- Use requirements for retrieving old software

But:

- Typical requirements are ambiguous

Thus:

- Formalize software to be a *software case*
- Use ontologies for determining meanings

Software Case

A software case consists of:

- Requirements model
- Architecture model
- Detailed design
- Code
- Transformations

Requirements Model	A	DD	C	TR

3

Query

A software case consists of:

- Requirements model
- Architecture model
- Detailed design
- Code
- Transformations

Requirements Model	A	DD	C	TR

A query consists of:

- Requirements model:
  - **Requirements specification**
  - **Domain specification**

Requirements Model

4

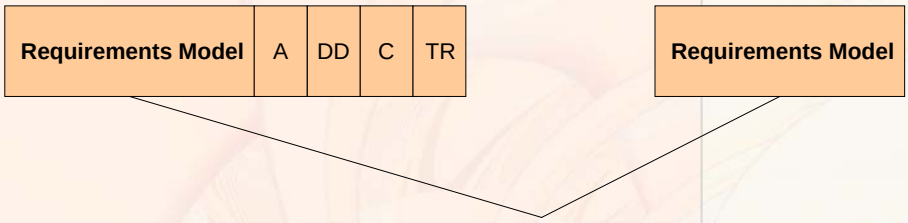
Comparing Requirements

A software case consists of:

- Requirements model
- Architecture model
- Detailed design
- Code
- Transformations

A query consists of:

- Requirements model:
  - **Requirements specification**
  - **Domain specification**



Challenges for Defining Similarity Measures

- **Similarity should reflect the intended meaning of the requirements.**
  - Take synonyms into account.
  - Take taxonomical relations into account.
- **Paraphrase and ambiguity problems should be solved.**

Ambiguity & Paraphrase Problem

Paraphrase Problem - **different words but same meaning**

- **Example:**
  - *"Client modifies purchase of merchandise"*
  - *"Customer changes order of items"*
  - *"Fireman enters data"*
  - *"Fireman types facts"*

Ambiguity Problem - **same words but different meaning**

- **Example:**
  - *"Customer changes order of items"* -> purchase is modified
  - *"Customer changes order of items"* -> sequence is modified
  - *"Fireman enters data"*
  - *"Fireman enters building"*

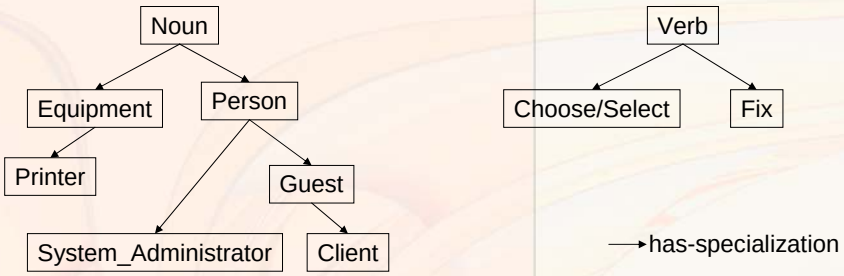
The similarity measure should recognise paraphrases.
The similarity measure should consider ambiguity of words.

7

Taxonomical Relations

Taxonomical relations - **different words but similar meaning**

- **Example:**
  - ***"Person selects equipment"***
  - ***"Guest chooses equipment"***
  - ***"Client selects printer"***
  - ***"System_Administrator fixes printer"***



The similarity measure should supply taxonomy-related similarity values.
The similarity measure should handle structure.

Overview

- Applying Description Logics
- Overview of the Ontology-based Similarity Measure
- Model-driven Software Cases
- Mapping MOF-based Models to an Ontology
- Classifying Software Cases
- Taxonomical Similarity Values
- Experiments
- Summary and Discussion

9

Primitive and Defined Concepts

Concept expressions of a DL describe classes of entities in terms of properties (unary relations) and roles (binary relations).

Main building blocks are primitive or defined concepts.

Primitive concepts: **concept => satisfied properties and relations**

satisfied properties and relations are <u>necessary</u> conditions for an object to belong to a class

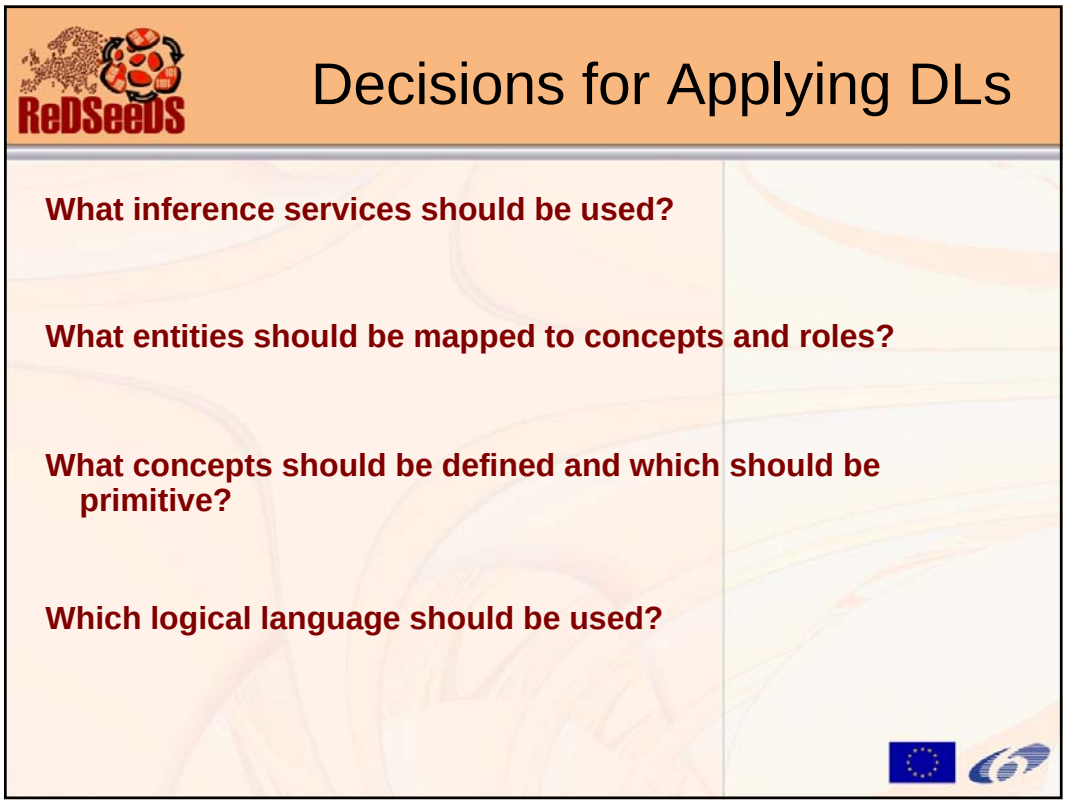
Defined concepts: **concept <=> satisfied properties and relations**

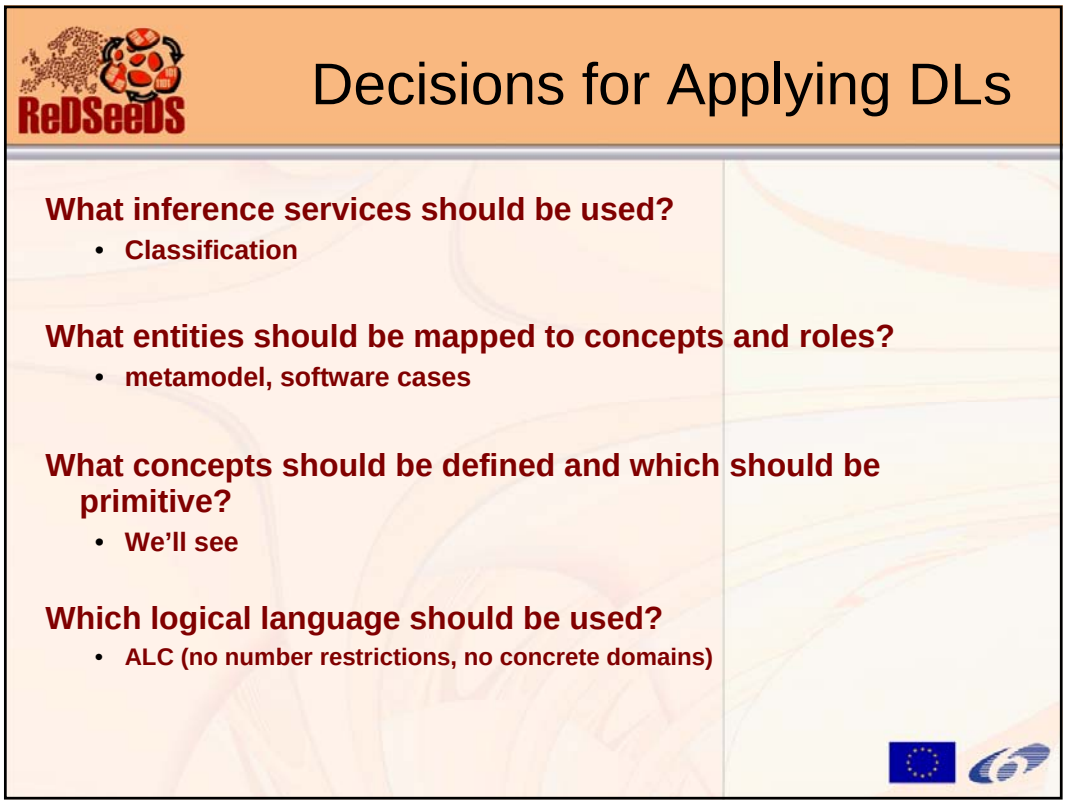
satisfied properties and relations are <u>necessary and sufficient</u> conditions for an object to belong to a class

Primitive concept "person":
(implies person (and mammal (some has-gender (or female male))))

Defined concept "parent":
(equivalent parent (and person (some has-child person)))

c/o Neumann


ReDSeeDS
Decisions for Applying DLs
What inference services should be used?
What entities should be mapped to concepts and roles?
What concepts should be defined and which should be primitive?
Which logical language should be used?


ReDSeeDS
Decisions for Applying DLs
What inference services should be used?
• Classification
What entities should be mapped to concepts and roles?
• metamodel, software cases
What concepts should be defined and which should be primitive?
• We'll see
Which logical language should be used?
• ALC (no number restrictions, no concrete domains)

Overview

- Applying Description Logics
- **Overview of the Ontology-based Similarity Measure**
- Model-driven Software Cases
- Mapping MOF-based Models to an Ontology
- Classifying Software Cases
- Taxonomical Similarity Values
- Experiments
- Summary and Discussion

13

Overview of the Ontology-based Similarity Measure

- **Formal MOF-based software cases**
- **Mapping the metamodel and software cases to an ontology (T-Box)**
  - Enables reasoning about metamodel and software case!
- **Classify the ontology**
  - Use WordNet's synset taxonomy for classifying structure
- **Compute the taxonomical similarity**

- **Gain:**
  - Automatic detection of taxonomical relations
  - Similarity measure which considers taxonomical relations and structure of requirements specifications

Overview

- Applying Description Logics
- Overview of the Ontology-based Similarity Measure
- **Model-driven Software Cases**
- Mapping MOF-based Models to an Ontology
- Classifying Software Cases
- Taxonomical Similarity Values
- Experiments
- Summary and Discussion

15

Modelling Requirements

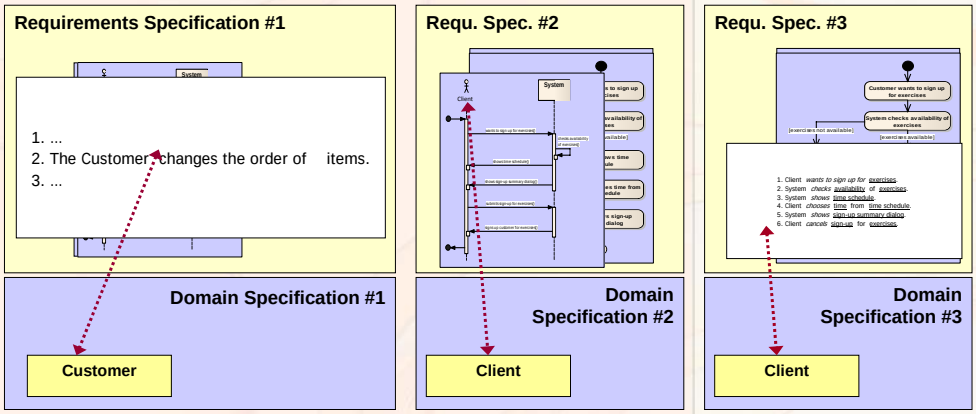
Two main parts:

- **Requirements specification**
  - Functional requirement on system
  - Functional requirement on composite
  - Use case
  - Constraint on system
  - Constraint on process
- **Domain specification**
  - Notion
  - Actors
  - UI elements
  - System elements

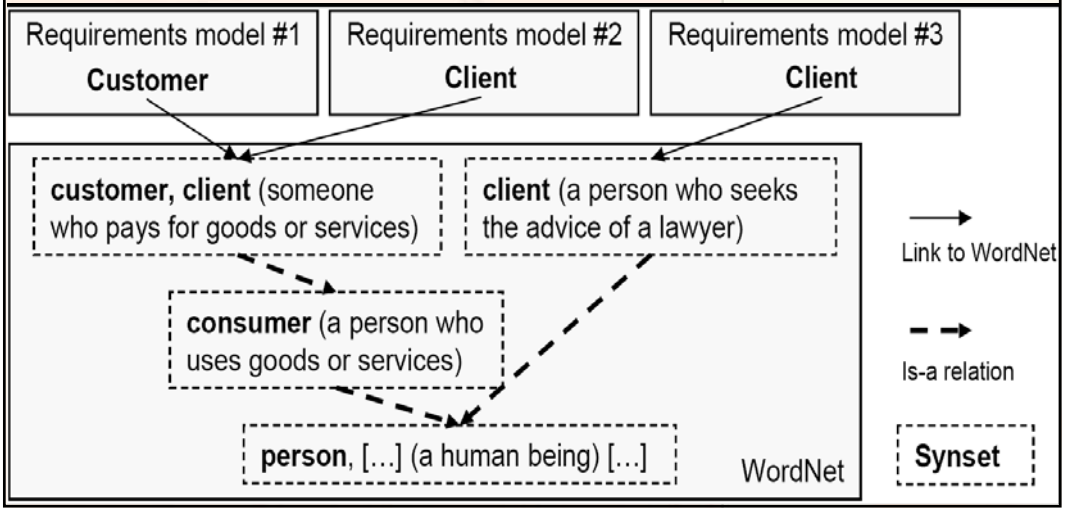
Requirements representations

- **Sentence list**
  - with natural language hypertext sentences
  - with constraint language sentences
- **Constraint language scenario**
- **Activity scenario**

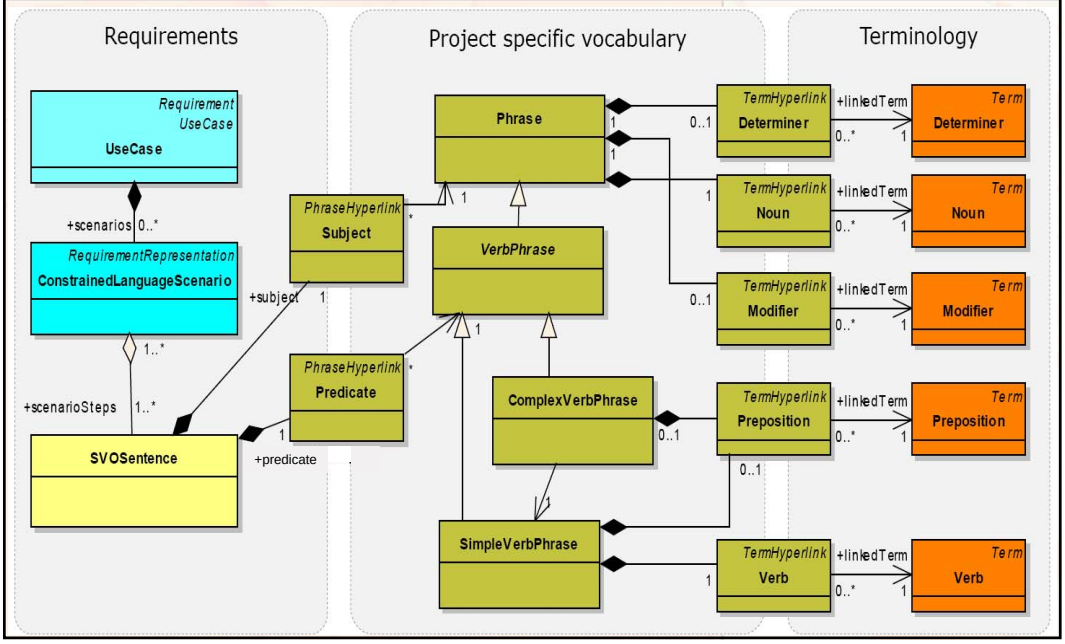
Software Case Requirements



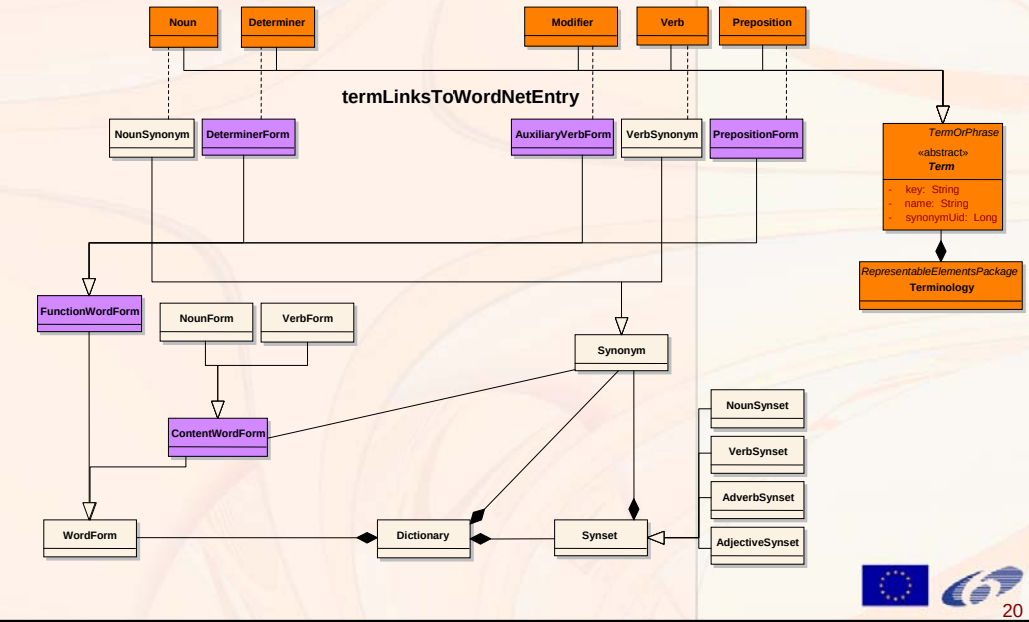
Link to WordNet

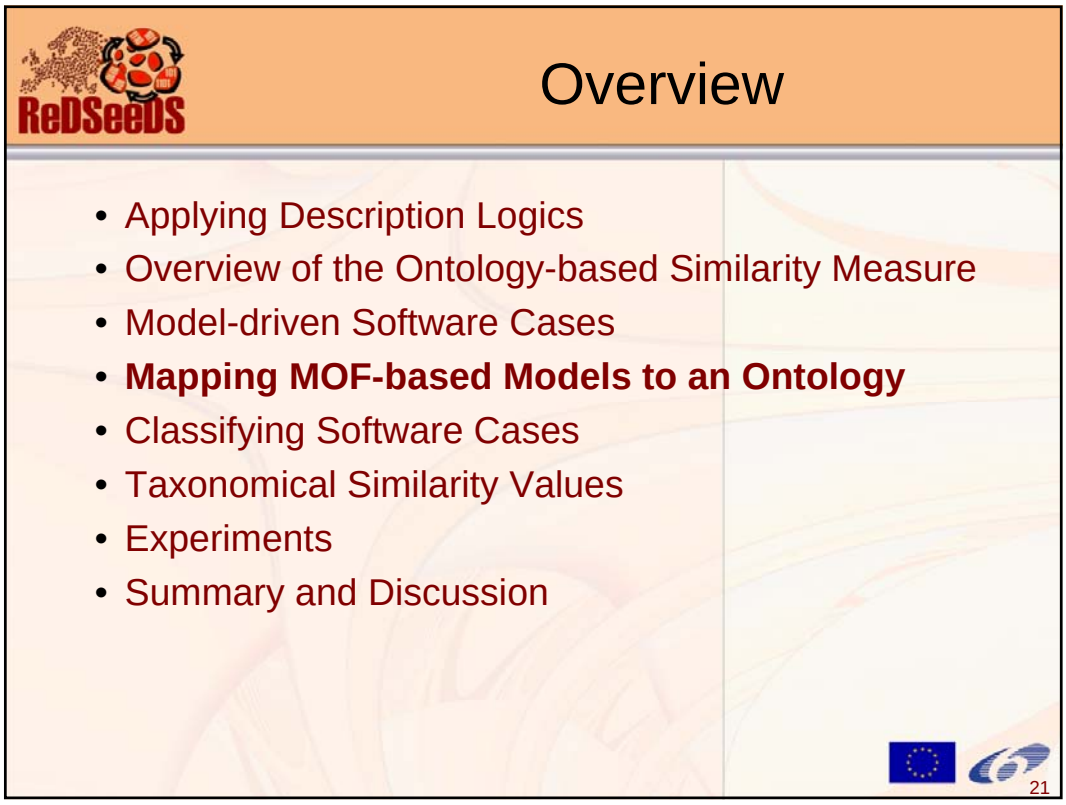


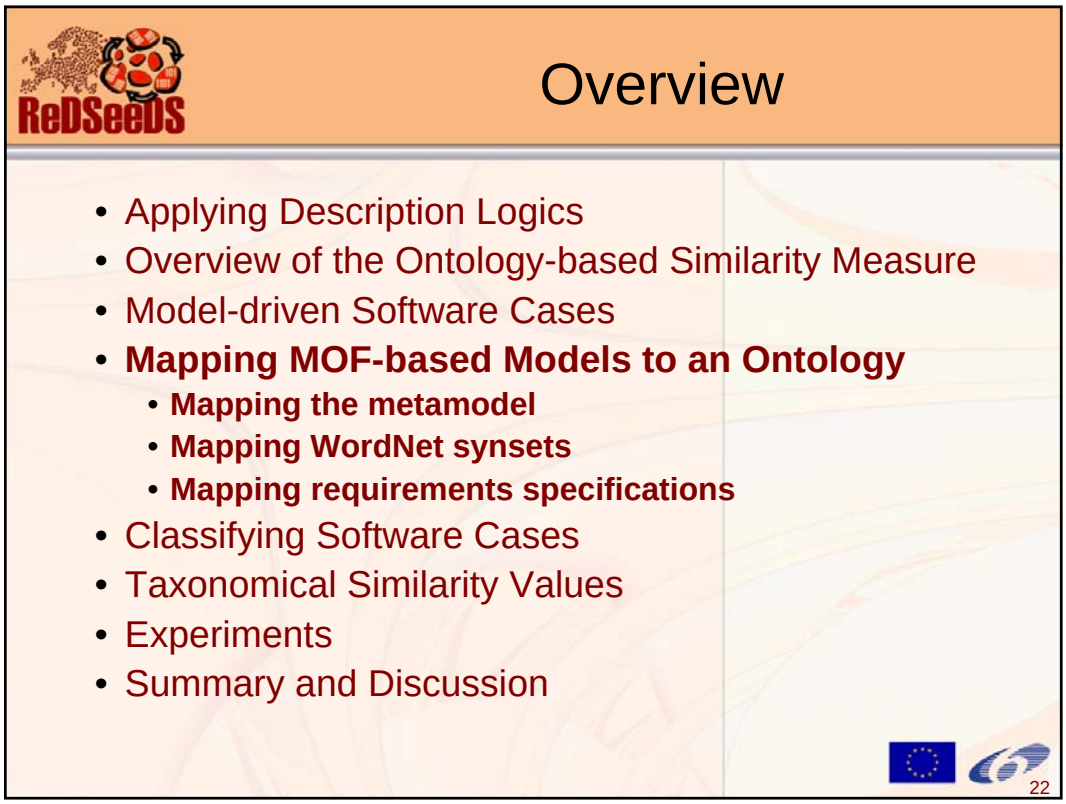
RSL Metamodel (cut-out)



RSL and WordNet Metamodel



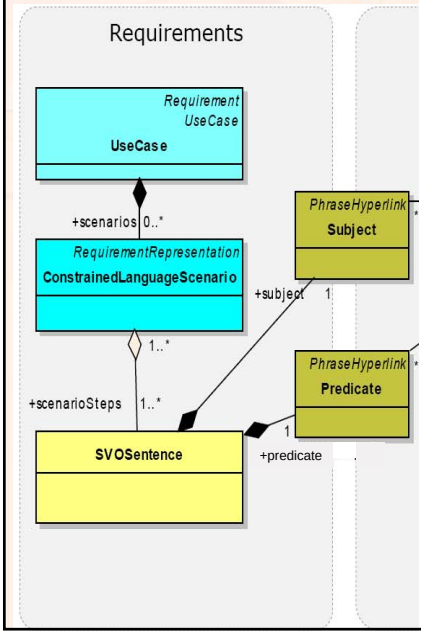

ReDSeeDS
Overview
• Applying Description Logics
• Overview of the Ontology-based Similarity Measure
• Model-driven Software Cases
• Mapping MOF-based Models to an Ontology
• Classifying Software Cases
• Taxonomical Similarity Values
• Experiments
• Summary and Discussion
21


ReDSeeDS
Overview
• Applying Description Logics
• Overview of the Ontology-based Similarity Measure
• Model-driven Software Cases
• Mapping MOF-based Models to an Ontology
• Mapping the metamodel
• Mapping WordNet synsets
• Mapping requirements specifications
• Classifying Software Cases
• Taxonomical Similarity Values
• Experiments
• Summary and Discussion
22

Mapping the Metamodel

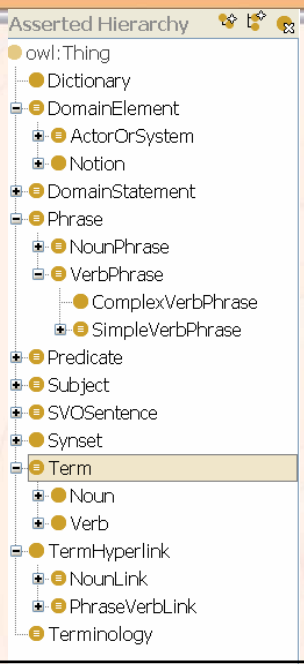
- **Map a class of the metamodel to a concept.**
  - "Upper-model concept"
- **Map generalization relations to specialization.**
  - Necessary conditions
- **Map a relation/assoziation/link to a role.**
  - Relations are necessary and sufficient conditions for a class:
  - If an object with such relations exists, then it belongs to the class.
- **The classes are disjoint.**
- **No inverse relations needed for the classification task.**
  - Only the path to WordNet terms is needed.

Example of a Concept for a class of the Metamodel




```
(equivalent SVOSentence
   (and ConstrainedLanguageSentence
        (some subject Subject)
        (some predicate Predicate)))
```

Upper-Model Concepts



Mapping WordNet

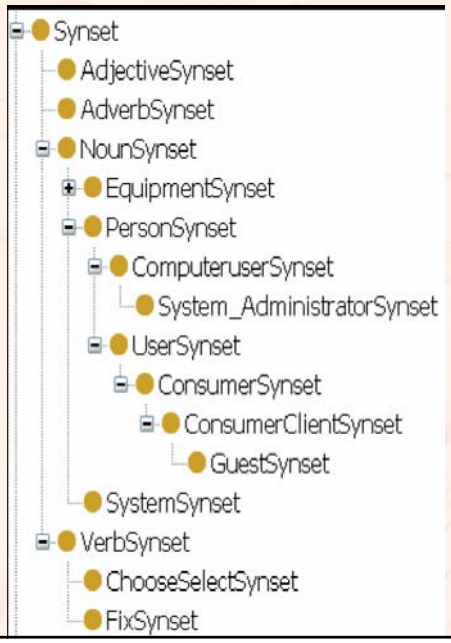
- **Map only synsets, not synonyms, nor word form etc.**
- **Map hyponym relations to specialization.**
- **Use only necessary conditions because no further roles are given.**
- **Synsets that are siblings are disjoint to each other.**
- **Synset concepts are linked to nouns via the role `termLinksToWordNetEntry`.**
- **Map only needed synsets not the complete WordNet.**

Mapping WordNet Synsets Example

```
(implies ChooseSelectSynset VerbSynset)
(implies FixSynset VerbSynset)
(implies PersonSynset NounSynset)
(implies ComputerUserSynset PersonSynset)
(implies System_AdministratorSynset
         ComputerUserSynset)
(implies UserSynset PersonSynset)
(implies ConsumerSynset UserSynset)
(implies CustomerClientSynset ConsumerSynset)
(implies GuestSynset CustomerClientSynset)

(disjoint ComputerUserSynset UserSynset)
(disjoint ChooseSelectSynset FixSynset)
```

Synset Taxonomy



Mapping Requirements Specifications

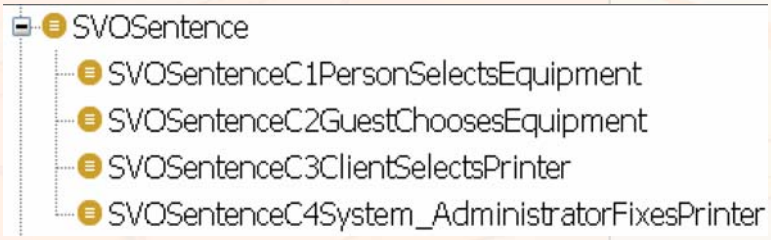
- **Metamodel approach:**
  - Requirements Specifications are represented on M1 as instances of classes of M2.
- **Those instances are mapped to concepts (*case concepts*) in the T-Box!**
- **Those concepts are subconcepts of the upper-model concepts.**
- **Roles *define* (again) the case concepts.**

- **No A-Box individuals are used.**
  - Because taxonomical relations between elements of the requirements specification can only be computed for concepts.
  - No relations between individuals can be automatically computed.

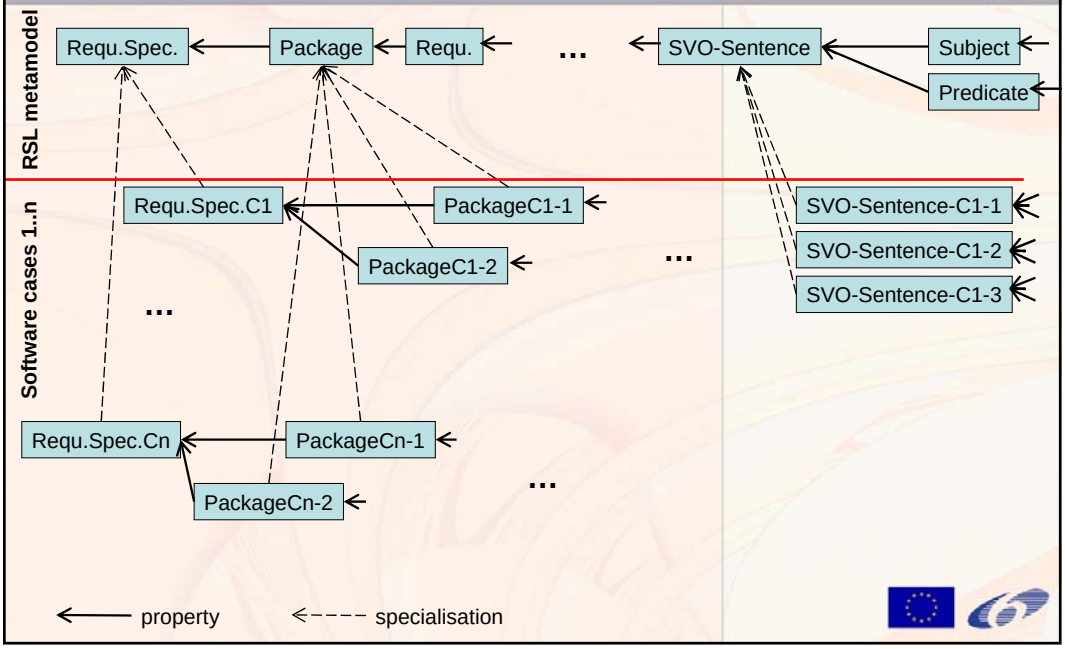
Mapping Requirements Specification - Example

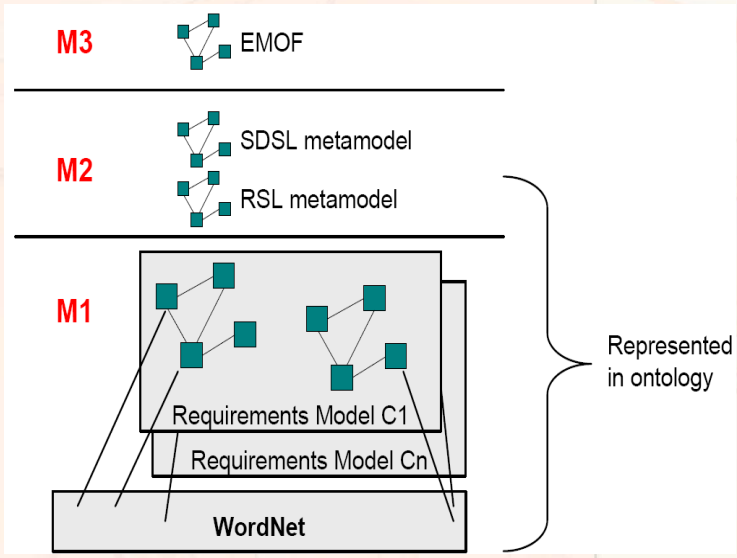
```
(equivalent SVOSentenceC2GuestChoosesEquipment
   (and SVOSentence
        (some subject SubjectC2Guest)
        (some predicate PredicateC2ChoosesEquipment)))
(equivalent PredicateC2ChoosesEquipment
   (and Predicate
       (some verbPhrase SimpleVerbPhraseC2Choose)))
(equivalent SimpleVerbPhraseC2Choose
   (and SimpleVerbPhrase
        (some verb PhraseVerbLinkC2Choose)
        (some object NounPhraseC2Equipment)))
(equivalent PhraseVerbLinkC2Choose
   (and PhraseVerbLink
        (some linkedVerb VerbC2Choose)))
(equivalent VerbC2Choose
   (and Verb
        (some termLinksToWordnetEntry ChooseSelectSynset)))
```

Case Concepts



TBox





Overview

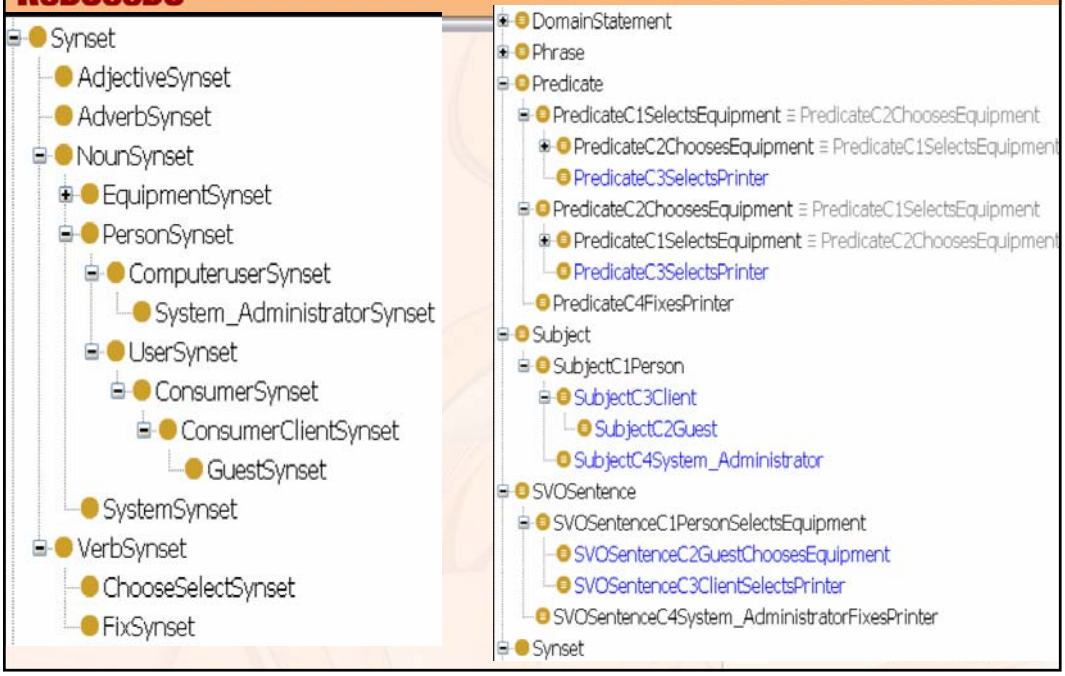
- Applying Description Logics
- Overview of the Ontology-based Similarity Measure
- Model-driven Software Cases
- Mapping MOF-based Models to an Ontology
- **Classifying Software Cases**
- Taxonomical Similarity Values
- Experiments
- Summary and Discussion

Classification Goals

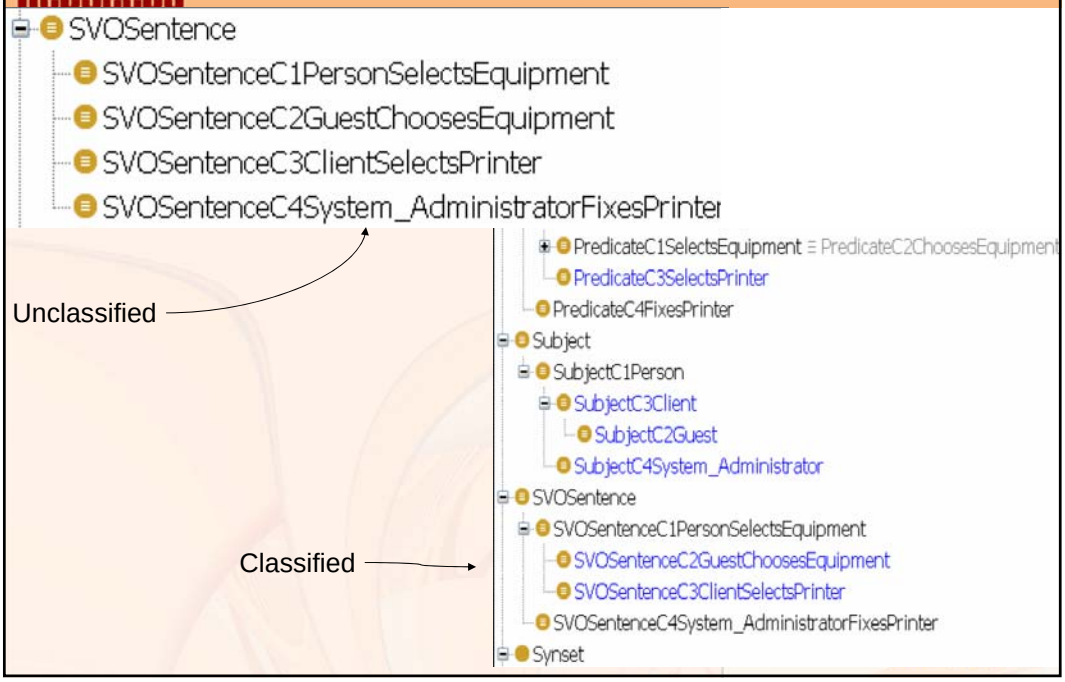
- **The requirements specifications do not contain information that directly relate concepts of *different* models.**
- **All concepts are direct subconcepts of upper-model concepts, thus, the taxonomical similarity is equal to each other.**
- **Only the synset taxonomy provides a hierarchy.**
- **This hierarchy is used for inferring other taxonomical relations through classification.**

- ***Thus, press the button "Classify Taxonomy".***

Classification Result - Example



Classification Result - Example

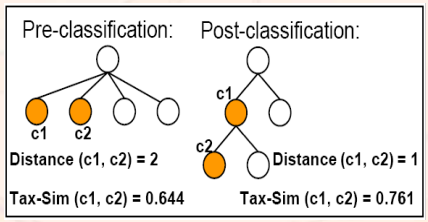


Overview

- Applying Description Logics
- Overview of the Ontology-based Similarity Measure
- Model-driven Software Cases
- Mapping MOF-based Models to an Ontology
- Classifying Software Cases
- **Taxonomical Similarity Values**
- Experiments
- Summary and Discussion

38

Taxonomical Similarity Values



lcs = leastCommonSubsumer(c1, c2)

distance1 = pathLength(c1, lcs)

distance2 = pathLength(c2, lcs)

distance = distance1 + distance2

return $\frac{1}{\ln(distance + e)}$

(There is more on similarity but not discussed here.)

Overview

- Applying Description Logics
- Overview of the Ontology-based Similarity Measure
- Model-driven Software Cases
- Mapping MOF-based Models to an Ontology
- Classifying Software Cases
- Taxonomical Similarity Values
- **Experiments**
- Summary and Discussion

40

Currently Running Experiments

- **Reduce the number of concepts**
  - Only include upper-model concepts and case concepts that are needed for similarity computation.
- **Experiments with particular requirement elements**
  - With the ontology-based similarity measure the intended ranking could be achieved.
- **Industrial experiments**
  - 261 requirements leading to 30.000 concepts and 8.000 after reduction.
  - Up to 1.000 concepts can be classified with Pellet or Racer.
  - Large implicit disjunction leads to non-tractable reasoning.

Overview

- Applying Description Logics
- Overview of the Ontology-based Similarity Measure
- Model-driven Software Cases
- Mapping MOF-based Models to an Ontology
- Classifying Software Cases
- Taxonomical Similarity Values
- Experiments
- **Summary and Discussion**

42

Summary and Discussion

- **Metamodel mapped to upper-model concepts.**
- **Requirements specification mapped to concepts.**
- **Defined concepts are needed for enabling classification.**
- **Classification as the main reasoning facility of DLs used.**
- **Similarity values that take meaning into account could be achieved.**
- **Size of handled models is limited due to DL-reasoner generality.**

Thank You for Your attention!