UPM

T2.2 Methods and tools for re-engineering of non-ontological resources

D2.2.4 Final version of methods for re-engineering and evaluation

Boris Villazón-Terrazas
Mari Carmen Suárez-Figueroa

{bvillazon,mcsuarez}@fi.upm.es
Ontology Engineering Group. Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid

June, 2009

ToC

- Motivation
- Formalization of the Ontology
- Formalization of the NOR
- Implementation of the Patterns
- To be included in D2.2.4

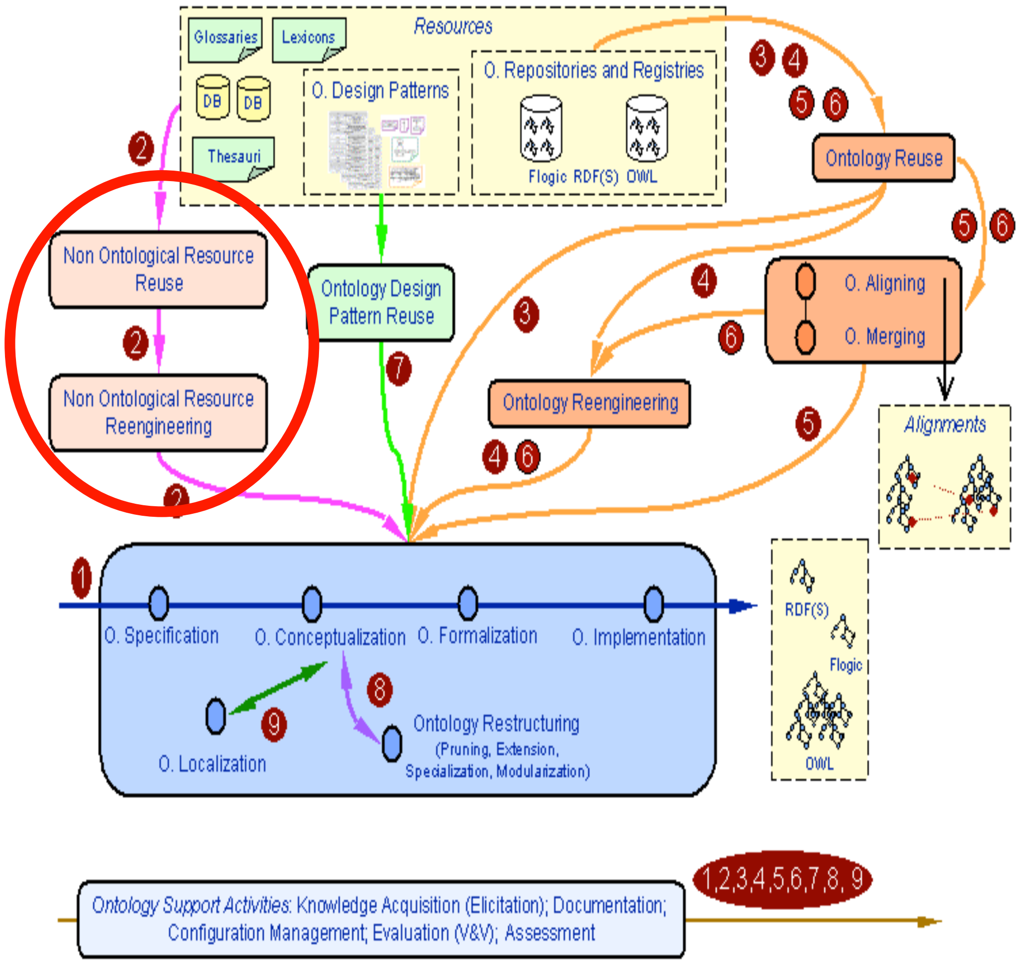
ToC

- **Motivation**
- Formalization of the Ontology
- Formalization of the NOR
- Implementation of the Patterns
- To be included in D2.2.4

Non-Ontological Resource Re-engineering

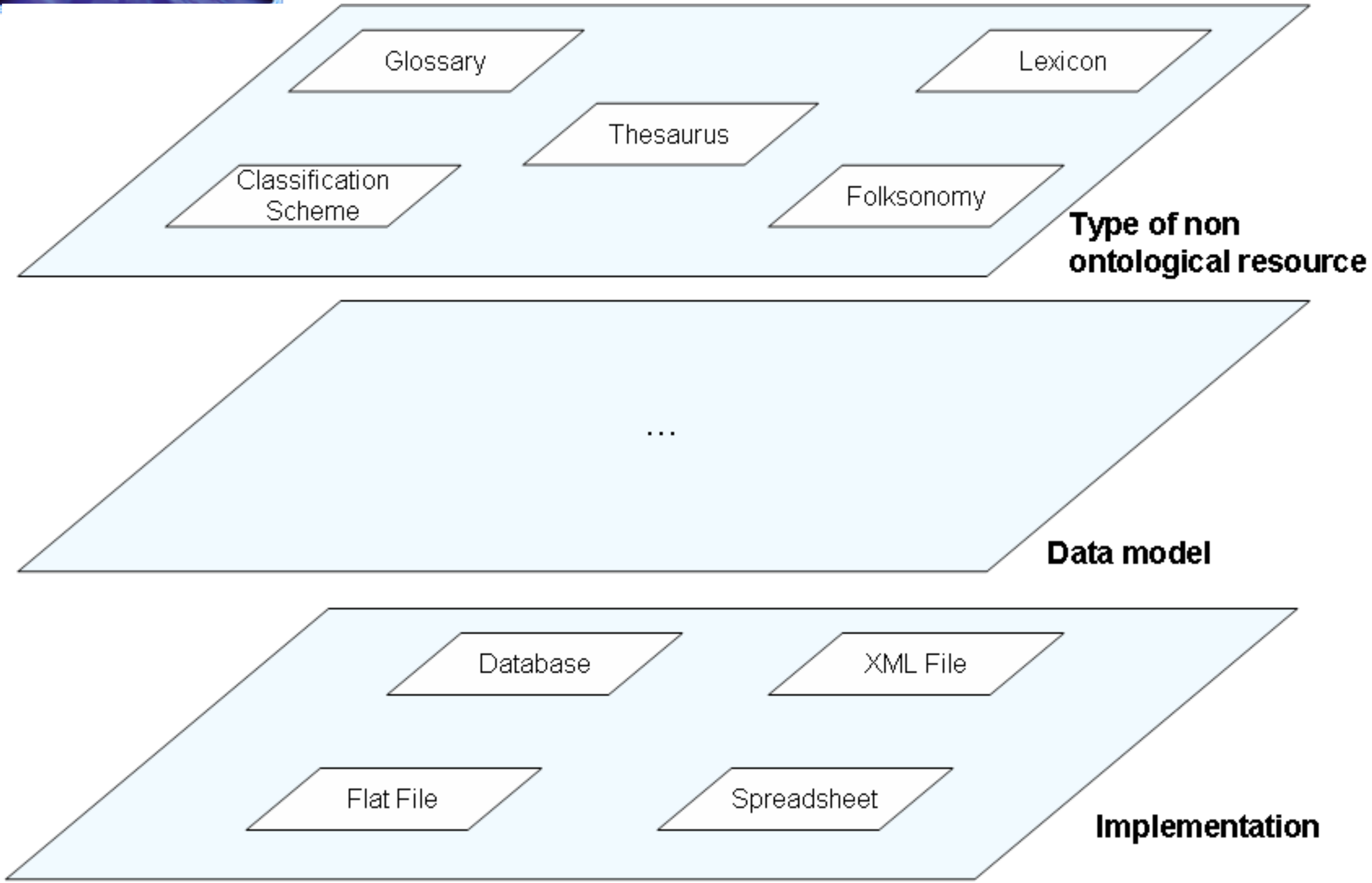
WP5

- **Non-Ontological Resource (NOR)** is an existing knowledge resource whose semantics has not been formalized yet by means of an ontology.
- **Non ontological resource reengineering** refers to the process of taking an existing non ontological resource and transforms it into an ontology.



Resources
Glossaries
Lexicons
DB
DB
Thesauri
O. Design Patterns
O. Repositories and Registries
Flogic RDF(S) OWL
Ontology Reuse
Non Ontological Resource Reuse
Non Ontological Resource Reengineering
Ontology Design Pattern Reuse
Ontology Reengineering
O. Aligning
O. Merging
Alignments
O. Specification
O. Conceptualization
O. Formalization
O. Implementation
O. Localization
Ontology Restructuring (Pruning, Extension, Specialization, Modularization)
RDF(S)
Flogic
OWL
Ontology Support Activities: Knowledge Acquisition (Elicitation); Documentation; Configuration Management; Evaluation (V&V); Assessment
1,2,3,4,5,6,7,8, 9

Types of NOR



Glossary

Lexicon

Thesaurus

Classification Scheme

Folksonomy

Type of non ontological resource

...

Data model

Database

XML File

Flat File

Spreadsheet

Implementation

ToC

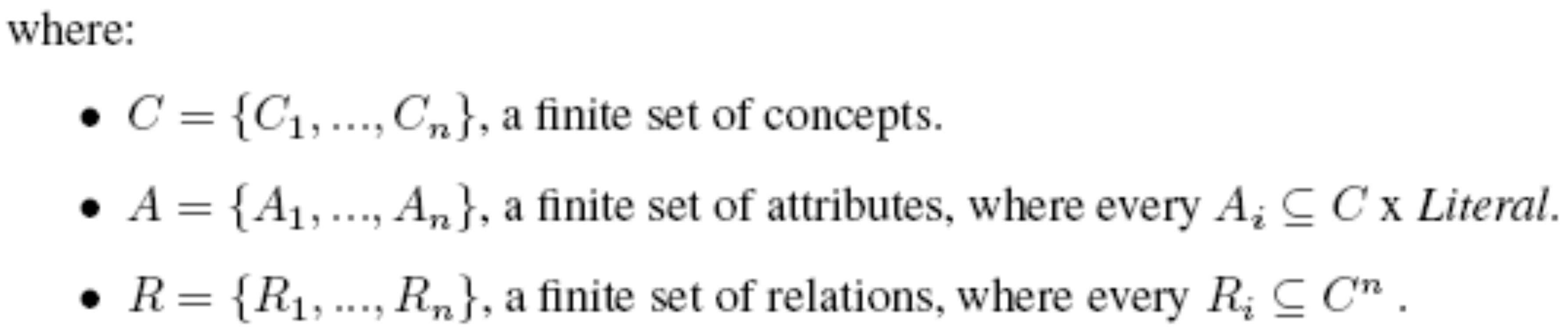
- Motivation
- **Formalization of the Ontology**
- Formalization of the NOR
- Implementation of the Patterns
- To be included in D2.2.4

Ontology

- Include formalization of the models
  - Ontology

$$O = \langle OS, KB \rangle$$

  - Ontology schema (OS – TBox)

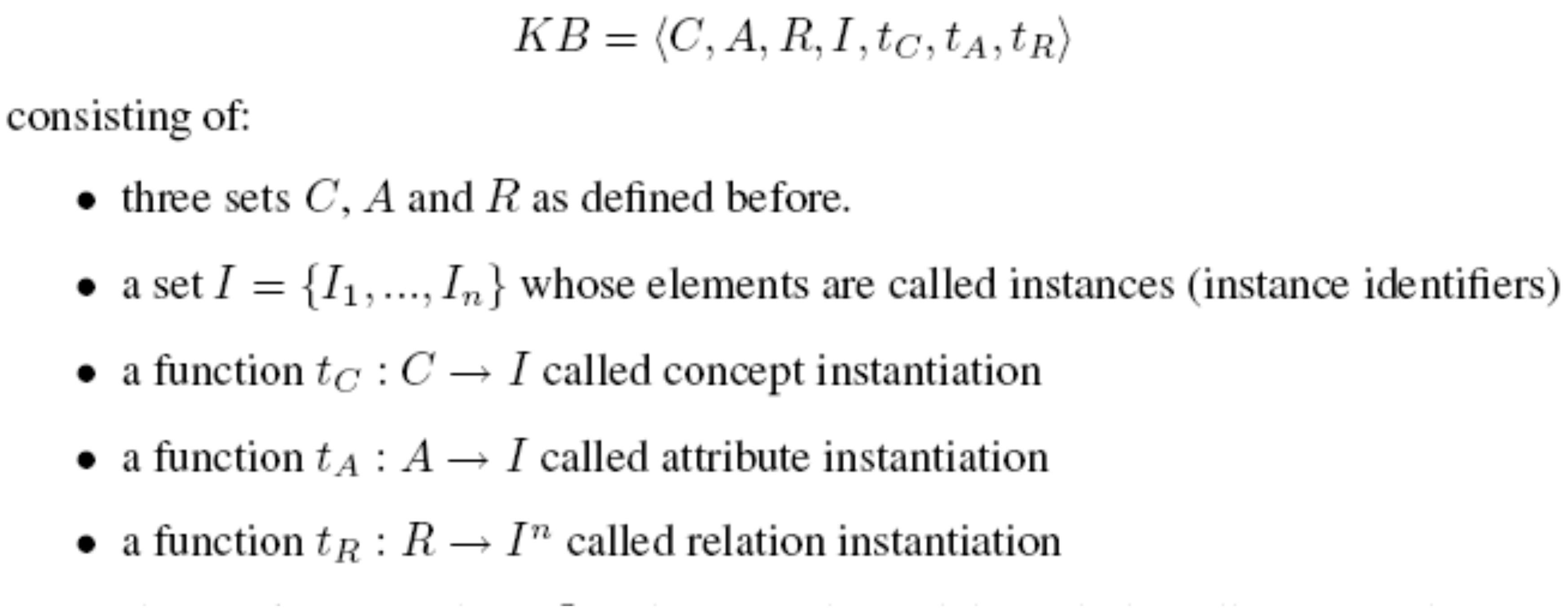


where:

- $C = \{C_1, ..., C_n\}$, a finite set of concepts.
- $A = \{A_1, ..., A_n\}$, a finite set of attributes, where every $A_i \subseteq C$ x *Literal*.
- $R = \{R_1, ..., R_n\}$, a finite set of relations, where every $R_i \subseteq C^n$.

Ontology (II)

- Knowledge Base (KB – ABox)



$$KB = \langle C, A, R, I, t_C, t_A, t_R \rangle$$

consisting of:

- three sets C, A and R as defined before.
- a set $I = \{I_1, ..., I_n\}$ whose elements are called instances (instance identifiers)
- a function $t_C : C \rightarrow I$ called concept instantiation
- a function $t_A : A \rightarrow I$ called attribute instantiation
- a function $t_R : R \rightarrow I^n$ called relation instantiation

ToC

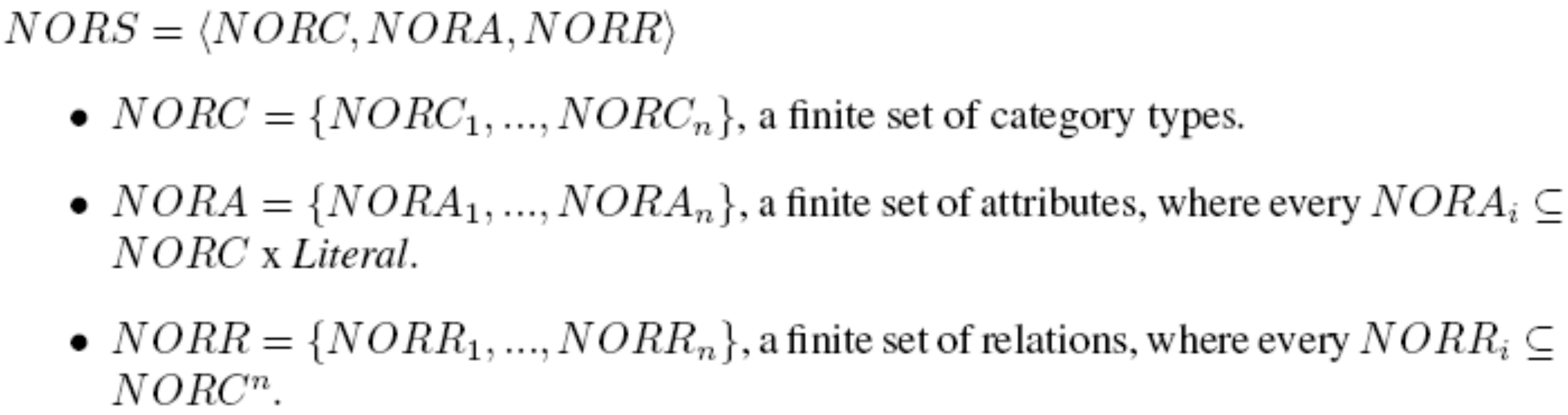
- Motivation
- Formalization of the Ontology
- **Formalization of the NOR**
- Implementation of the Patterns
- To be included in D2.2.4

NOR (I)

- NOR formal definition

$$NOR = \langle NORS, NORD \rangle$$

  - NOR Schema - NORS

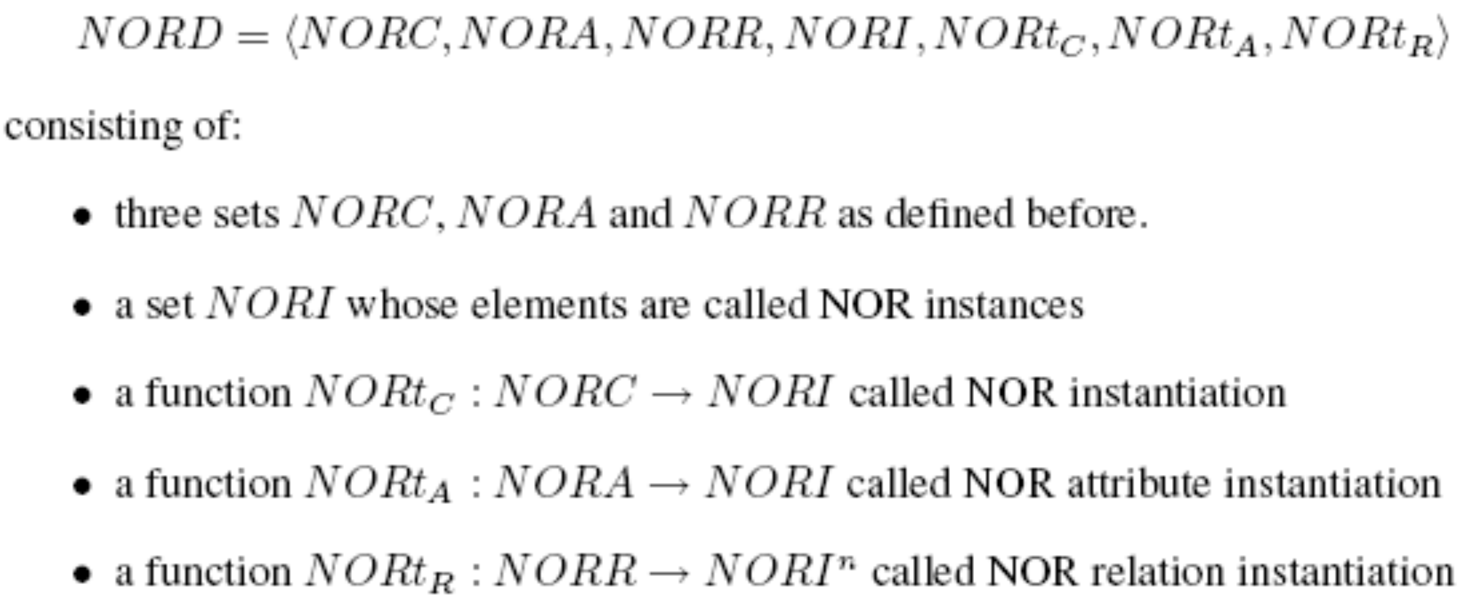


$$NORS = \langle NORC, NORA, NORR \rangle$$

- $NORC = \{NORC_1, ..., NORC_n\}$, a finite set of category types.
- $NORA = \{NORA_1, ..., NORA_n\}$, a finite set of attributes, where every $NORA_i \subseteq NORC$ x *Literal*.
- $NORR = \{NORR_1, ..., NORR_n\}$, a finite set of relations, where every $NORR_i \subseteq NORC^n$.

NOR (II)

– NOR Data - NORD

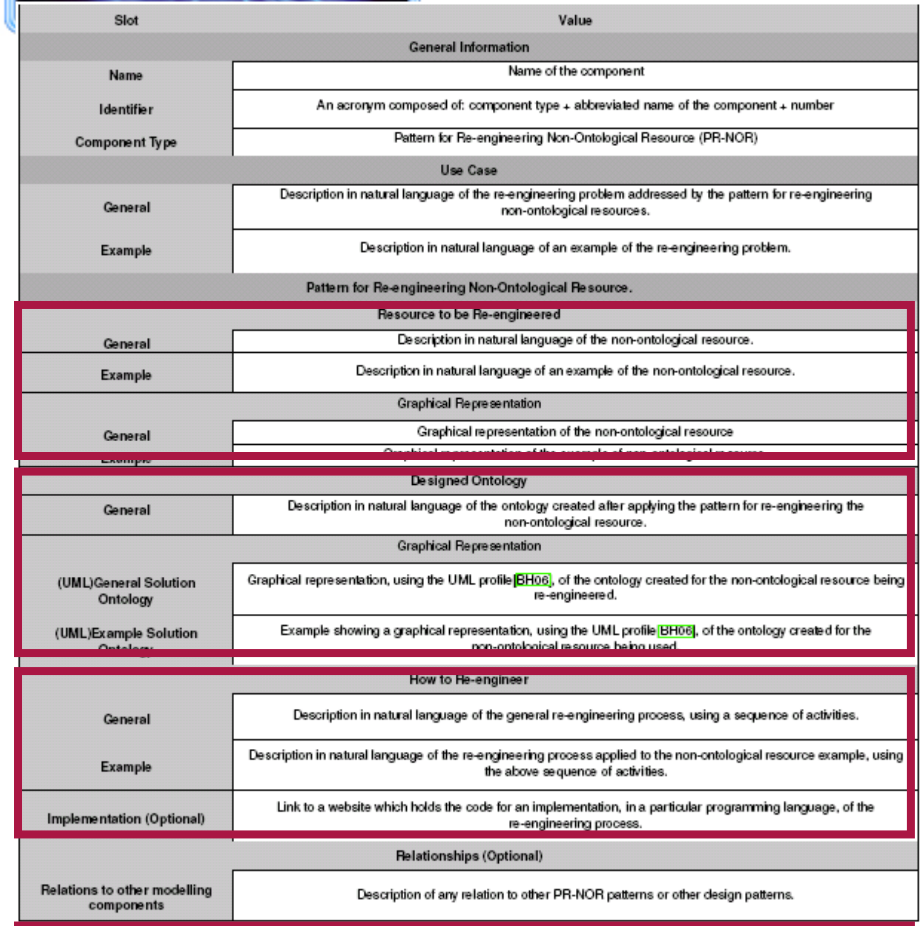


$$NORD = \langle NORC, NORA, NORR, NORI, NORt_C, NORt_A, NORt_R \rangle$$

consisting of:

- three sets $NORC$, $NORA$ and $NORR$ as defined before.
- a set $NORI$ whose elements are called NOR instances
- a function $NORt_C : NORC \rightarrow NORI$ called NOR instantiation
- a function $NORt_A : NORA \rightarrow NORI$ called NOR attribute instantiation
- a function $NORt_R : NORR \rightarrow NORI^n$ called NOR relation instantiation

New PR-NOR Template

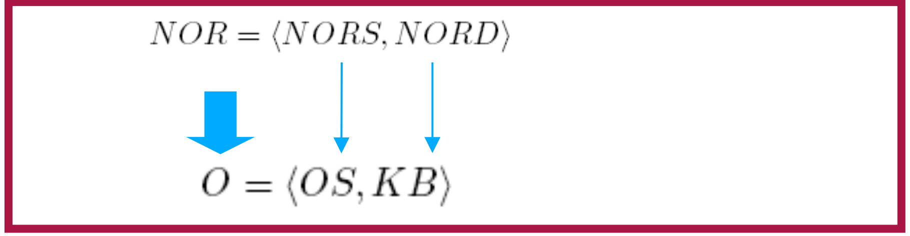


Slot	Value
General Information	
Name	Name of the component
Identifier	An acronym composed of: component type + abbreviated name of the component + number
Component Type	Pattern for Re-engineering Non-Ontological Resource (PR-NOR)
Use Case	
General	Description in natural language of the re-engineering problem addressed by the pattern for re-engineering non-ontological resources.
Example	Description in natural language of an example of the re-engineering problem.
Pattern for Re-engineering Non-Ontological Resource.	
Resource to be Re-engineered	
General	Description in natural language of the non-ontological resource.
Example	Description in natural language of an example of the non-ontological resource.
Graphical Representation	
General	Graphical representation of the non-ontological resource
Example	Graphical representation of the example of non-ontological resource
Designed Ontology	
General	Description in natural language of the ontology created after applying the pattern for re-engineering the non-ontological resource.
Graphical Representation	
(UML)General Solution Ontology	Graphical representation, using the UML profile [BH06], of the ontology created for the non-ontological resource being re-engineered.
(UML)Example Solution Ontology	Example showing a graphical representation, using the UML profile [BH06], of the ontology created for the non-ontological resource being used.
How to Re-engineer	
General	Description in natural language of the general re-engineering process, using a sequence of activities.
Example	Description in natural language of the re-engineering process applied to the non-ontological resource example, using the above sequence of activities.
Implementation (Optional)	Link to a website which holds the code for an implementation, in a particular programming language, of the re-engineering process.
Relationships (Optional)	
Relations to other modelling components	Description of any relation to other PR-NOR patterns or other design patterns.

INPUT:Non-Ontological Resource

OUTPUT: Ontology

PROCESS: How



$NOR = \langle NORS, NORD \rangle$

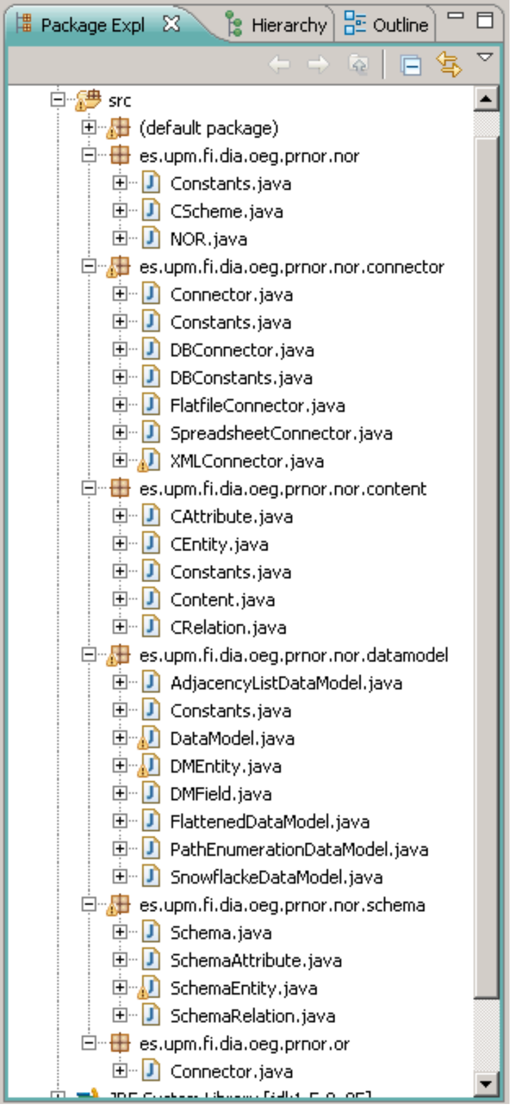
$O = \langle OS, KB \rangle$

Formal Transformation

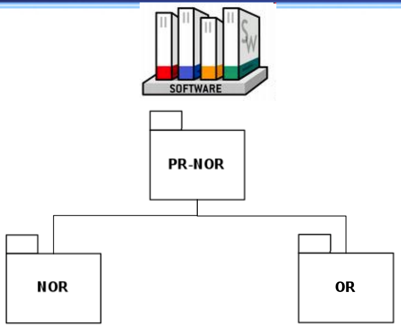
ToC

- Motivation
- Formalization of the Ontology
- Formalization of the NOR
- **Implementation of the Patterns**
- To be included in D2.2.4

Implementation of the software libraries that follow the guidelines provided by the patterns.



- src
  - (default package)
  - es.upm.fi.dia.oeg.prnor.nor
    - Constants.java
    - CScheme.java
    - NOR.java
  - es.upm.fi.dia.oeg.prnor.nor.connector
    - Connector.java
    - Constants.java
    - DBConnector.java
    - DBConstants.java
    - FlatfileConnector.java
    - SpreadsheetConnector.java
    - XMLConnector.java
  - es.upm.fi.dia.oeg.prnor.nor.content
    - CAttribute.java
    - CEntity.java
    - Constants.java
    - Content.java
    - CRelation.java
  - es.upm.fi.dia.oeg.prnor.nor.datamodel
    - AdjacencyListDataModel.java
    - Constants.java
    - DataModel.java
    - DMEntity.java
    - DMField.java
    - FlattenedDataModel.java
    - PathEnumerationDataModel.java
    - SnowflackeDataModel.java
  - es.upm.fi.dia.oeg.prnor.nor.schema
    - Schema.java
    - SchemaAttribute.java
    - SchemaEntity.java
    - SchemaRelation.java
  - es.upm.fi.dia.oeg.prnor.or
    - Connector.java



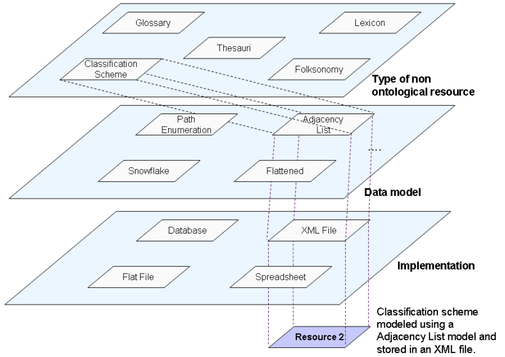
```
public static void testDBConnectorPathEnumerationDataModel() {
    //////////ISCO 88 COM
    NOR myNOR = new CScheme("ISCO-88COM");

    //Connector norConnector = new DBConnector(DBConstants.MYSQL,"digitalis","root","root","localhost","3306
    //myNOR.setConnector(norConnector);

    //Connector
    //Connector norConnector = new DBConnector(DBConstants.MSACCESS,"isco88com","","","","");
    Connector norConnector = new DBConnector(DBConstants.MYSQL,"isco88com","root","root","localhost","3306")
    myNOR.setConnector(norConnector);

    //Implementation
    DataModel norDataModel = new PathEnumerationDataModel();
    ((PathEnumerationDataModel)norDataModel).setMainEntity("isco");
    ((PathEnumerationDataModel)norDataModel).setPathField("code");
    ((PathEnumerationDataModel)norDataModel).setPathSeparator("");
    myNOR.setDataModel(norDataModel);

    myNOR.load();
```



Implementation ongoing

ToC

- Motivation
- Formalization of the Ontology
- Formalization of the NOR
- Implementation of the Patterns
- To be included in D2.2.4

To be included in D2.2.4 (UPM)

- D2.2.4 Final version of methods for re-engineering and evaluation moved to M44 – October 31st, 2009
  - QA?

- Include the formalization of the models.
- Include the ABox re-engineering patterns.
- Implementation of the patterns.
- Evaluation of the patterns.

UPM

T2.2 Methods and tools for re-engineering of non-ontological resources

D2.2.4 Final version of methods for re-engineering and evaluation

Boris Villazón-Terrazas
Mari Carmen Suárez-Figueroa

{bvillazon,mcsuarez}@fi.upm.es
Ontology Engineering Group. Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid

June, 2009