



# A Method for Re-engineering Non-Ontological Resources for Building Ontologies

Boris Villazón-Terrazas, Mari Carmen Suárez Figueroa

{bvillazon,mcsuarez}@fi.upm.es

Ontology Engineering Group. Laboratorio de Inteligencia Artificial

Departamento de Inteligencia Artificial

Facultad de Informática

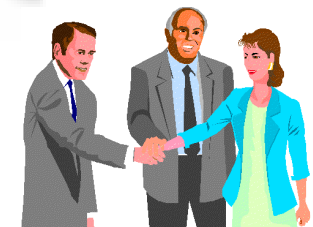
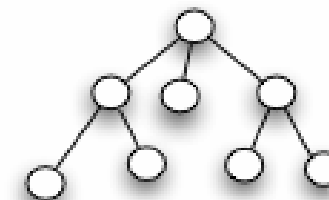
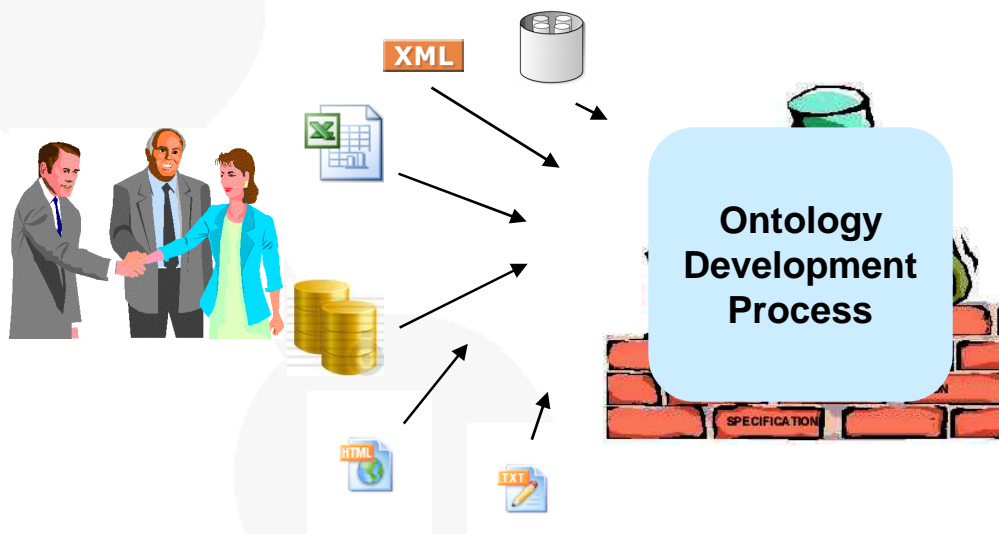
Universidad Politécnica de Madrid

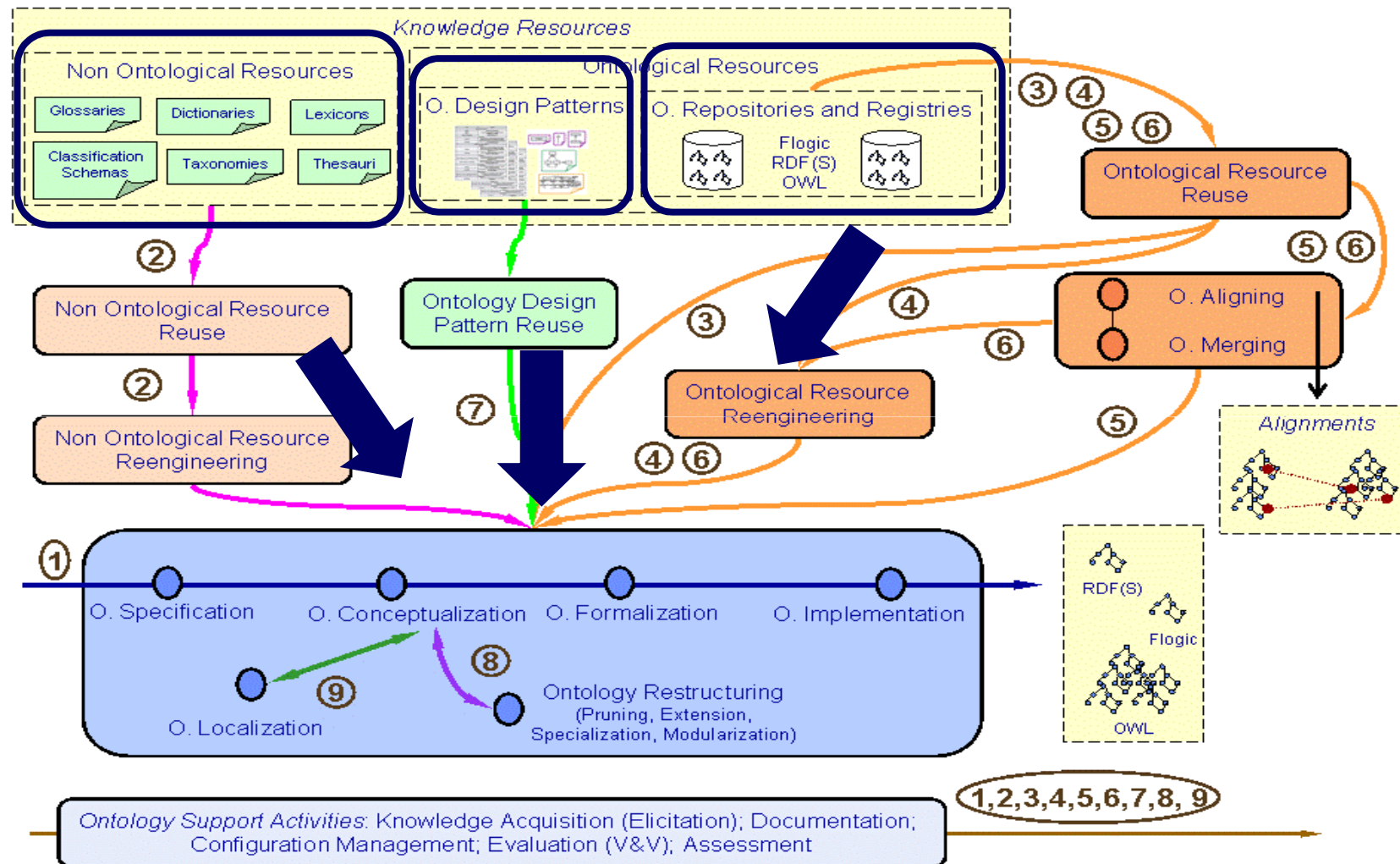
- Introduction
- A proposal for classifying/categorizing Non-Ontological Resources
  - Classification scheme
  - Thesaurus
- Method for Re-engineering Non-Ontological Resources
- PR-NORs

## From scratch

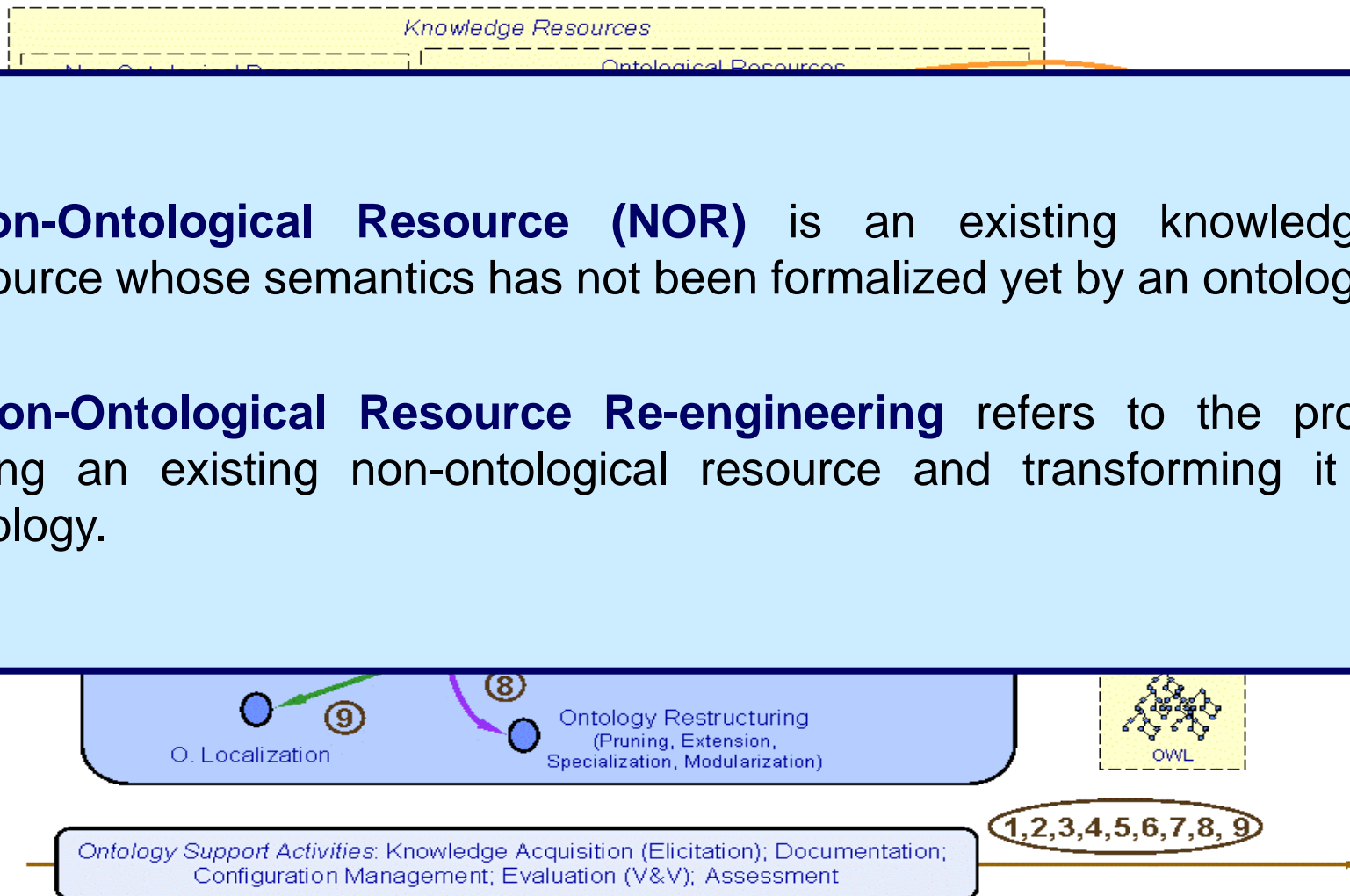


## Reusing and re-engineering non-ontological resources



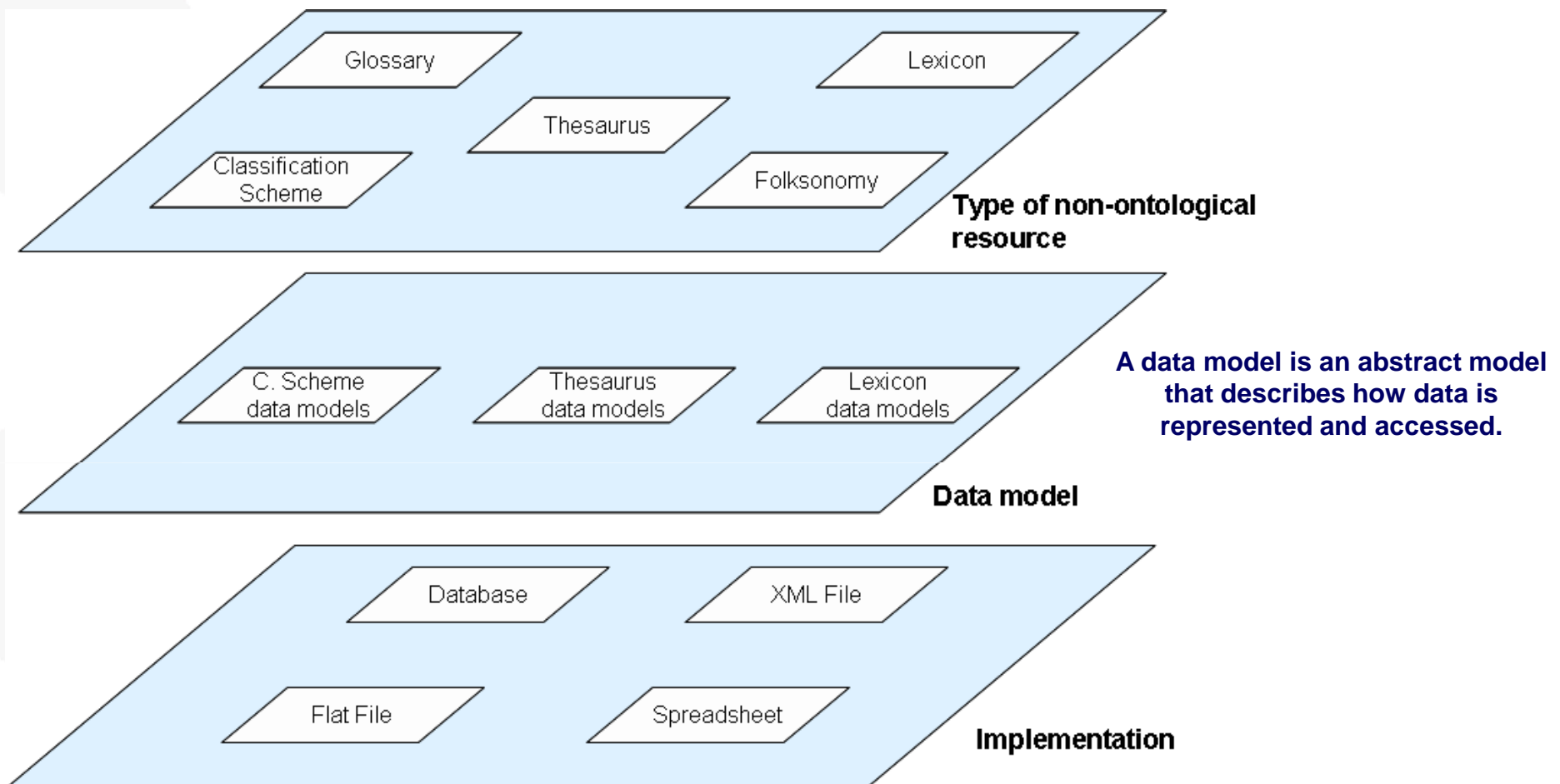


- **Non-Ontological Resource (NOR)** is an existing knowledge-aware resource whose semantics has not been formalized yet by an ontology.
- **Non-Ontological Resource Re-engineering** refers to the process of taking an existing non-ontological resource and transforming it into an ontology.

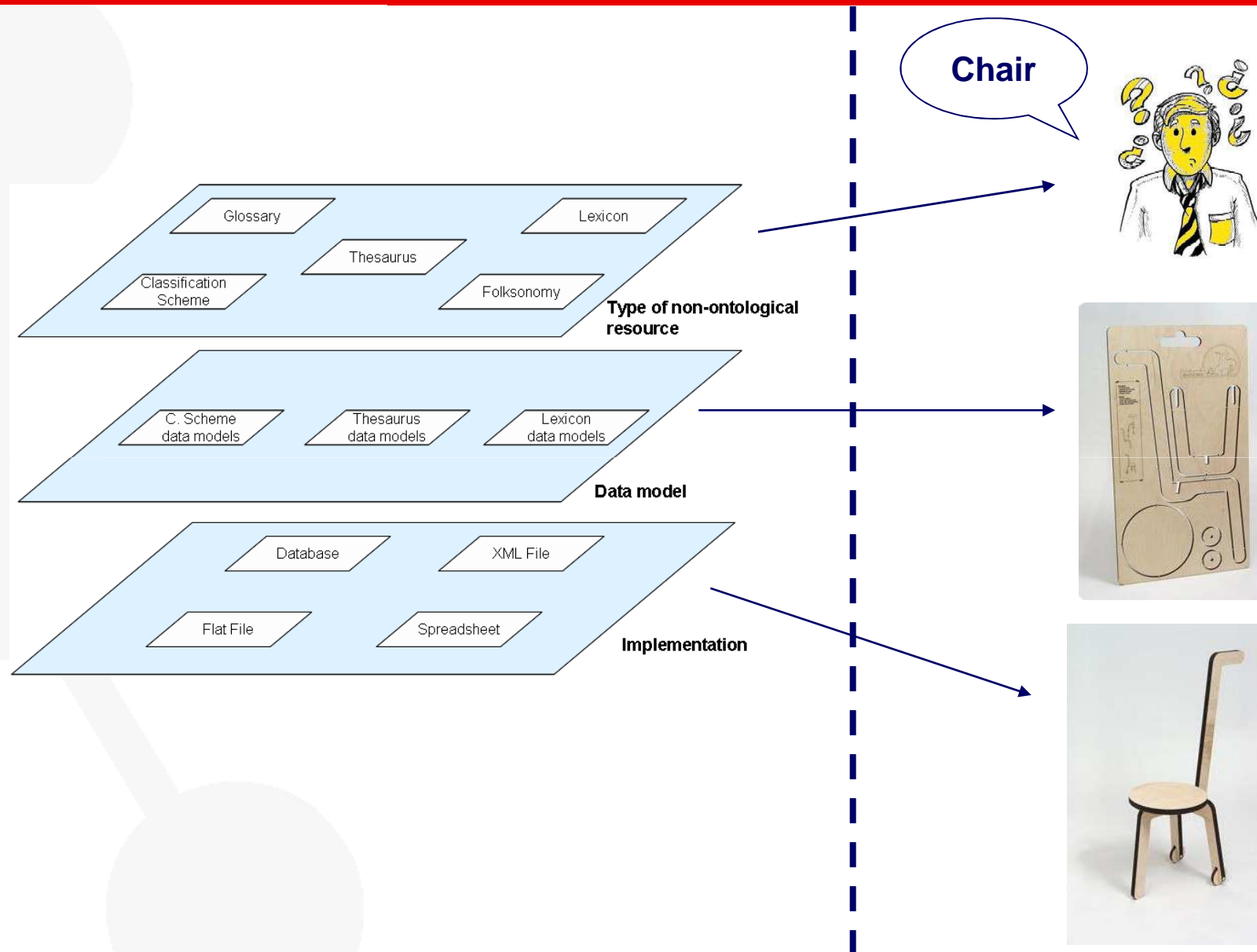


- Introduction
- A proposal for classifying/categorizing Non-Ontological Resources
  - Classification scheme
  - Thesaurus
- Method for Re-engineering Non-Ontological Resources
- PR-NORs

# Non-Ontological Resources Categorization



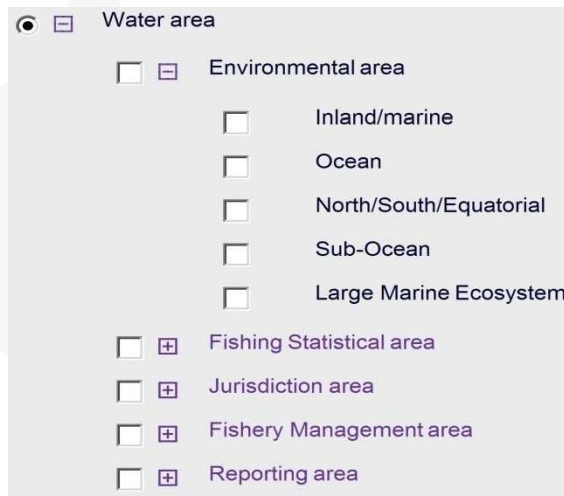
## Real life example





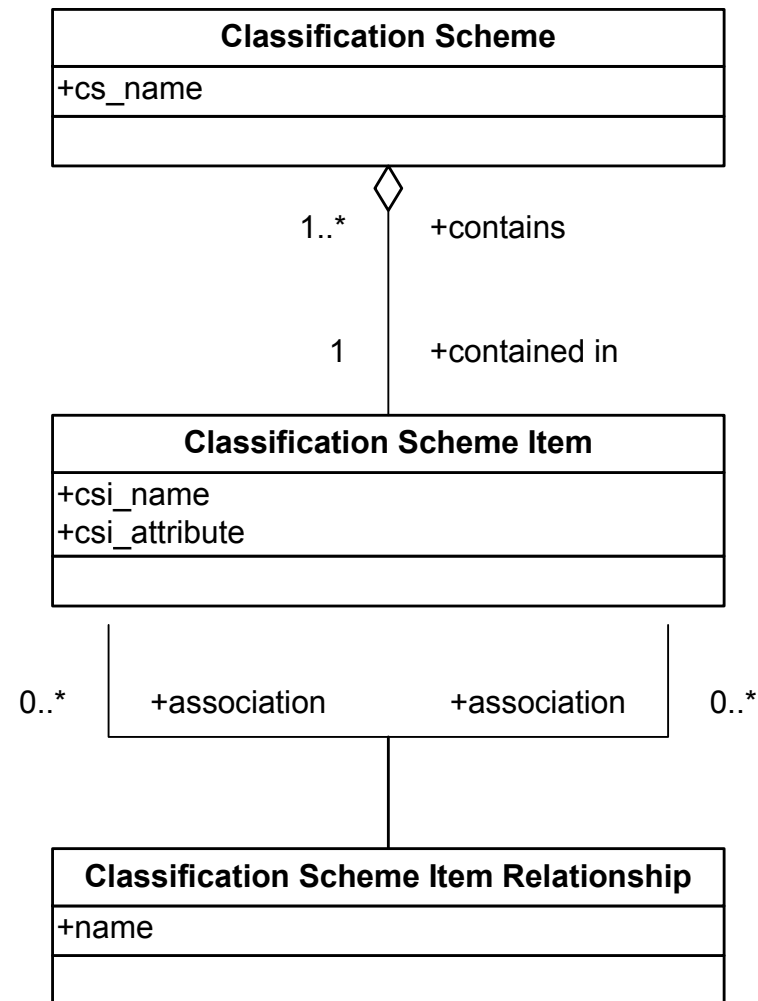
- Introduction
- A proposal for classifying/categorizing Non-Ontological Resources
  - Classification scheme
  - Thesaurus
- Method for Re-engineering Non-Ontological Resources
- PR-NORs

- A classification scheme<sup>1</sup> is the descriptive information for an arrangement or division of objects into groups based on characteristics, which the objects have in common. E.g. water area classification scheme<sup>2</sup>.



1. International Standard Organization (ISO). Information technology - Metadata registries – Part 1: Framework, 2004. Report ISO/IEC FDIS 11179-1.
2. <http://www.fao.org/figis/servlet/RefServlet>

# Classification Scheme Components

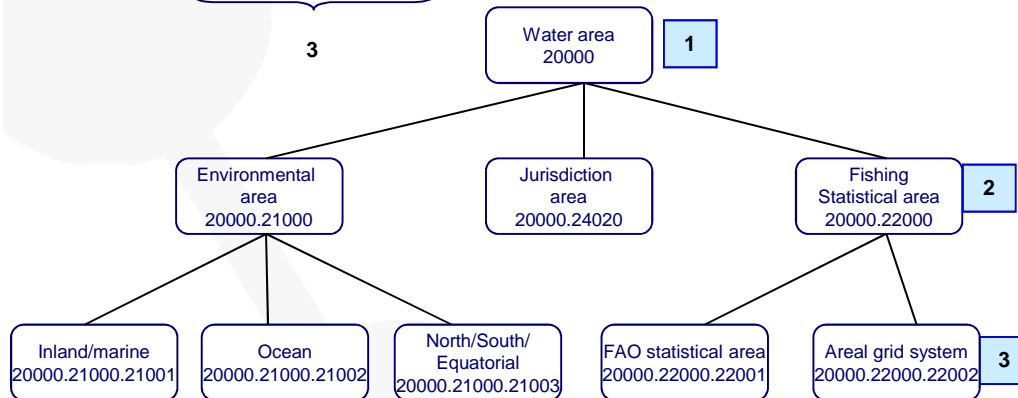
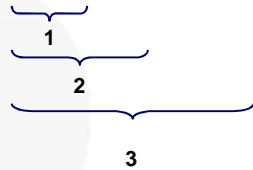


1. International Standard Organization (ISO). Information technology - Metadata registries – Part 1: Framework, 2004. Report ISO/IEC FDIS 11179-1.

# Classification Scheme Data Models (I)

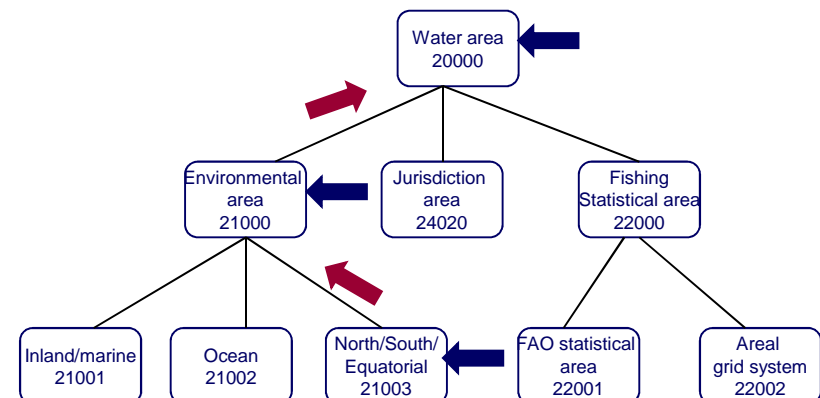
- Path Enumeration Data Model** is defined as a model that stores for each node the path (as a string) from the root to the node.

ID	CSI_Name
20000	Water area
20000.21000	Environmental area
20000.24020	Jurisdiction area
20000.22000	Fishing Statistical area
20000.21000.21001	Inland/marine
20000.21000.21002	Ocean
20000.21000.21003	North/South/Equatorial
20000.22000.22001	FAO statistical area
20000.22000.22002	Areal grid system



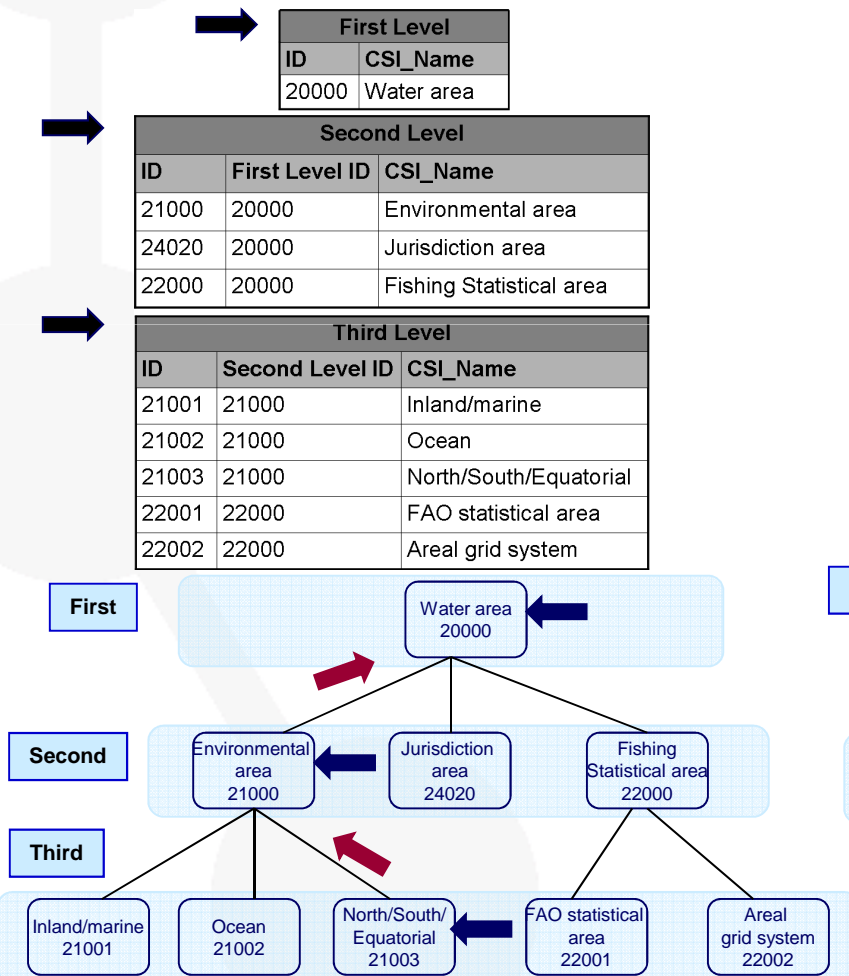
- Adjacency List** is a recursive structure for hierarchy representations that comprises a list of nodes with a linking column to their parent nodes.

ID	CSI_Name	Parent
20000	Water area	
21000	Environmental area	20000
24020	Jurisdiction area	20000
22000	Fishing Statistical area	20000
21001	Inland/marine	21000
21002	Ocean	21000
21003	North/South/Equatorial	21000
22001	FAO statistical area	22000
22002	Areal grid system	22000

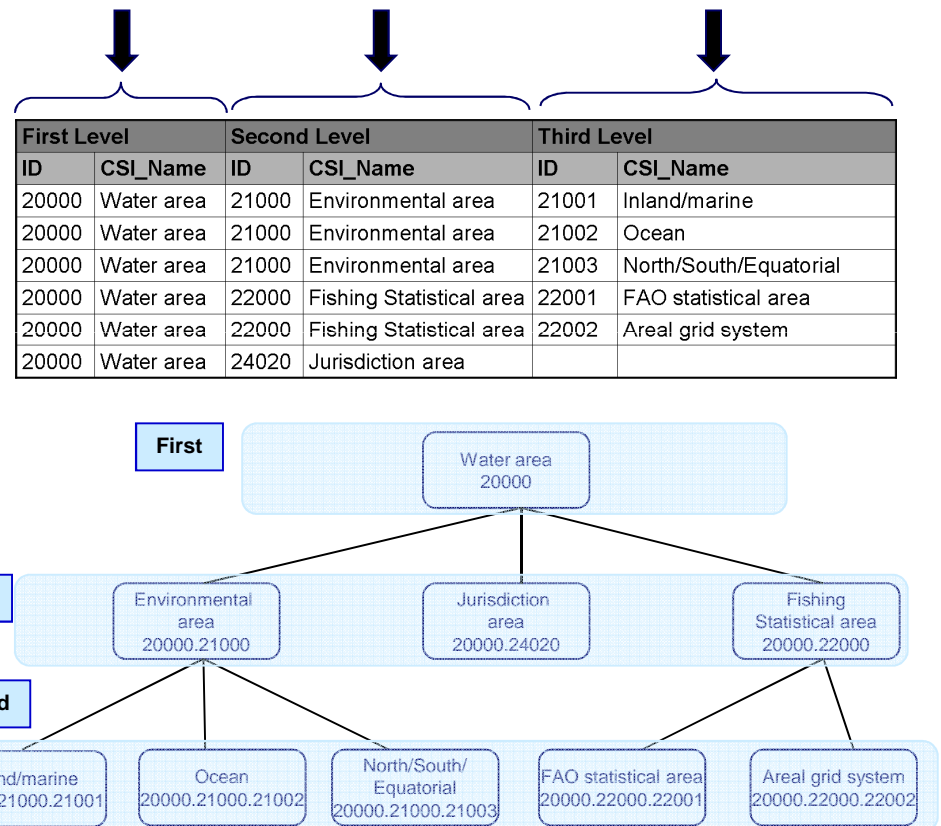


## Classification Scheme Data Models (II)

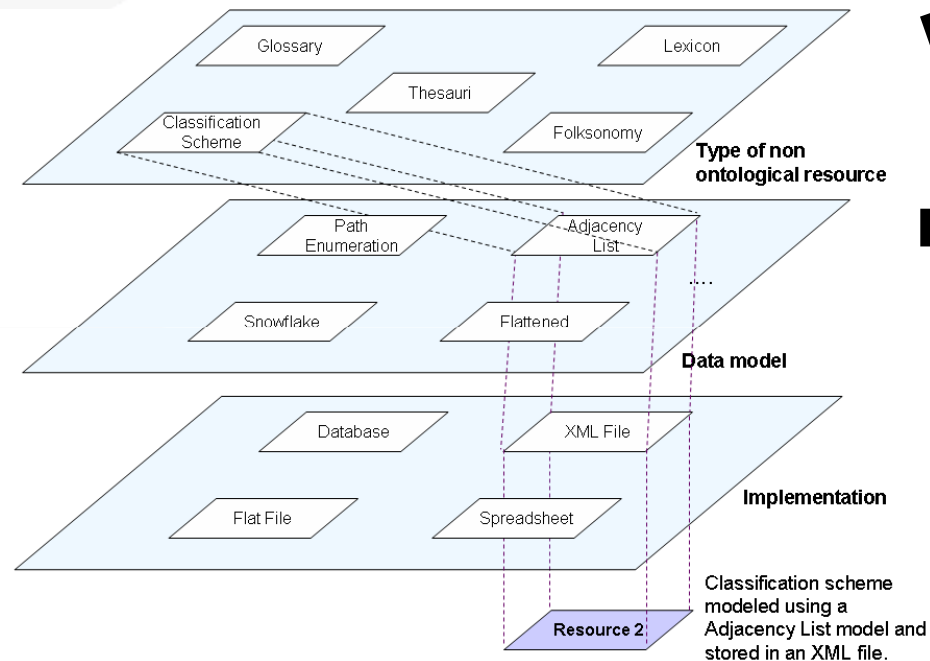
- Snowflake Data Model** is a normalized structure for hierarchy representations. For each hierarchy level a entity is created. In this model each hierarchy node has a column linked to its parent node.



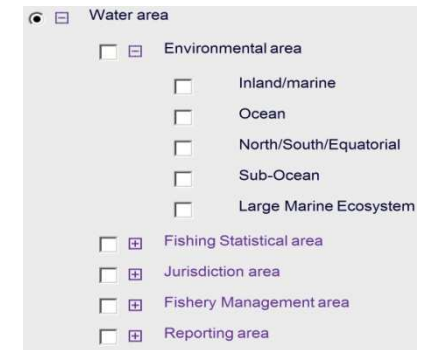
- Flattened Data Model**, is a denormalized structure. The hierarchy is represented with an entity where each hierarchy level is stored on a different column.



# Example - FAO Water Areas Classification Scheme



## Classification Scheme



## Adjacency List

ID	CSI_Name	Parent
20000	Water area	
21000	Environmental area	20000
24020	Jurisdiction area	20000
22000	Fishing Statistical area	20000
21001	Inland/marine	21000
21002	Ocean	21000
21003	North/South/Equatorial	21000
22001	FAO statistical area	22000
22002	Areal grid system	22000

## XML

```

<Classification>
  <Category>
    <NodeId>20000</NodeId>
    <WaterCategory>Water Area</WaterCategory>
    <parentNodeId>1</parentNodeId>
  </Category>
  <Category>
    <NodeId>21000</NodeId>
    <WaterCategory>Environmental area</WaterCategory>
    <parentNodeId>20000</parentNodeId>
  </Category>
  <Category>
    <NodeId>22000</NodeId>
    <WaterCategory>Fishing statistical area</WaterCategory>
    <parentNodeId>20000</parentNodeId>
  </Category>
  <Category>
    <NodeId>24020</NodeId>
    <WaterCategory>Jurisdiction area</WaterCategory>
    <parentNodeId>20000</parentNodeId>
  </Category>
  <Category>
    <NodeId>21001</NodeId>
    <WaterCategory>inland/marine</WaterCategory>
    <parentNodeId>21000</parentNodeId>
  </Category>
  ....
</Classification>
  
```

- Introduction
- A proposal for classifying/categorizing Non-Ontological Resources
  - Classification scheme
  - Thesaurus
- Method for Re-engineering Non-Ontological Resources
- PR-NORs

- Thesauri<sup>1</sup> are controlled vocabularies of terms in a particular domain with hierarchical, associative and equivalence relations between terms. Thesauri are mainly used for indexing and retrieving of articles in large databases. An example of thesaurus is the AGROVOC<sup>2</sup> thesaurus.

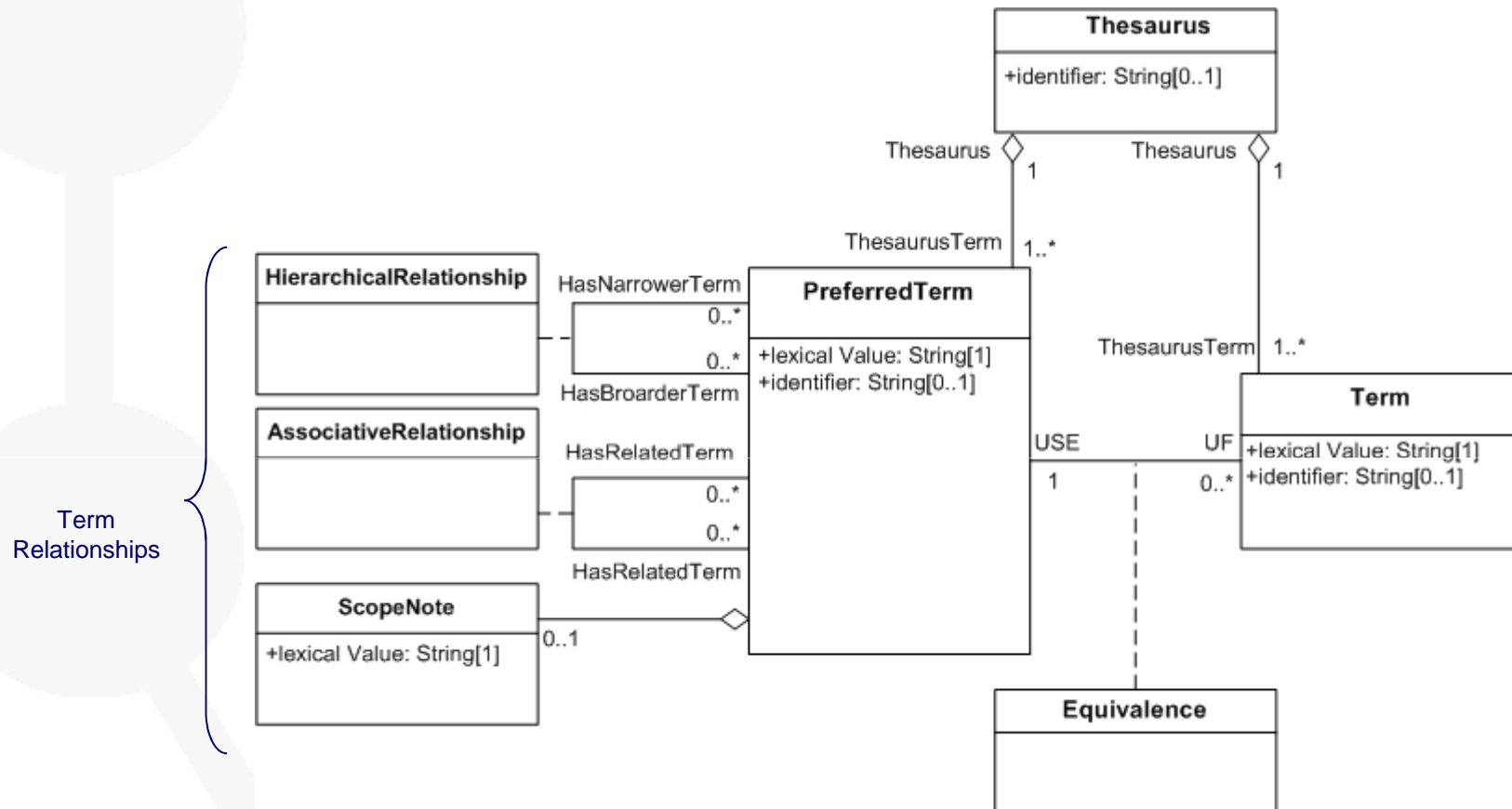
EN : Oryza	BT ( subclassOf ) : Poaceae
	NT ( hasSubclass ) : Oryza sativa
	NT ( hasSubclass ) : Oryza perennis
	NT ( hasSubclass ) : Oryza rufipogon
	NT ( hasSubclass ) : Oryza longistaminata
	NT ( hasSubclass ) : Wetland rice
	NT ( hasSubclass ) : Oryza glaberrima
	NT ( hasSubclass ) : Upland rice
	NT ( hasSubclass ) : Oryza punctata
	RT : Rice fields
	RT : Cereal crops
	RT : Rice
EN : Rice	BT ( subclassOf ) : Cereals
	NT ( hasSubclass ) : Broken rice
	NT ( hasSubclass ) : Basmati rice
	RT : Rice straw
	RT : Oryza
	RT : Rice flour
	UF : Paddy



1. International Standard Organization (ISO). Documentation – Guidelines for the establishment and development of monolingual thesaurus, 1986. Report ISO 2788.
2. <http://www.fao.org/agrovoc/>



# Term-based thesaurus



1. International Standard Organization (ISO). Documentation – Guidelines for the establishment and development of monolingual thesaurus, 1986. Report ISO 2788.

Example: European Training Thesaurus - ETT

## Record-based data model for a term-based thesaurus

EN : Oryza	BT ( subclassOf ) : Poaceae
	NT ( hasSubclass ) : Oryza sativa
	NT ( hasSubclass ) : Oryza perennis
	NT ( hasSubclass ) : Oryza rufipogon
	NT ( hasSubclass ) : Oryza longistaminata
	NT ( hasSubclass ) : Wetland rice
	NT ( hasSubclass ) : Oryza glaberrima
	NT ( hasSubclass ) : Upland rice
	NT ( hasSubclass ) : Oryza punctata
	RT : Rice fields
	RT : Cereal crops
	RT : Rice
EN : Rice	BT ( subclassOf ) : Cereals
	NT ( hasSubclass ) : Broken rice
	NT ( hasSubclass ) : Basmati rice
	RT : Rice straw
	RT : Oryza
	RT : Rice flour
	UF : Paddy

Term	BT	NT	RT	UF
Rice	Cereals	Broken rice Basmati rice	Rice straw Oryza	Paddy
Oryza	Poaceae	Oryza sativa Oryza perennis Oryza rufipogon Oryza longistaminata Wetland rice Oryza glaberrima Upland rice Oryza punctata	Rice fields Cereal crops Rice	

# Relation-based data model for a term-based thesaurus

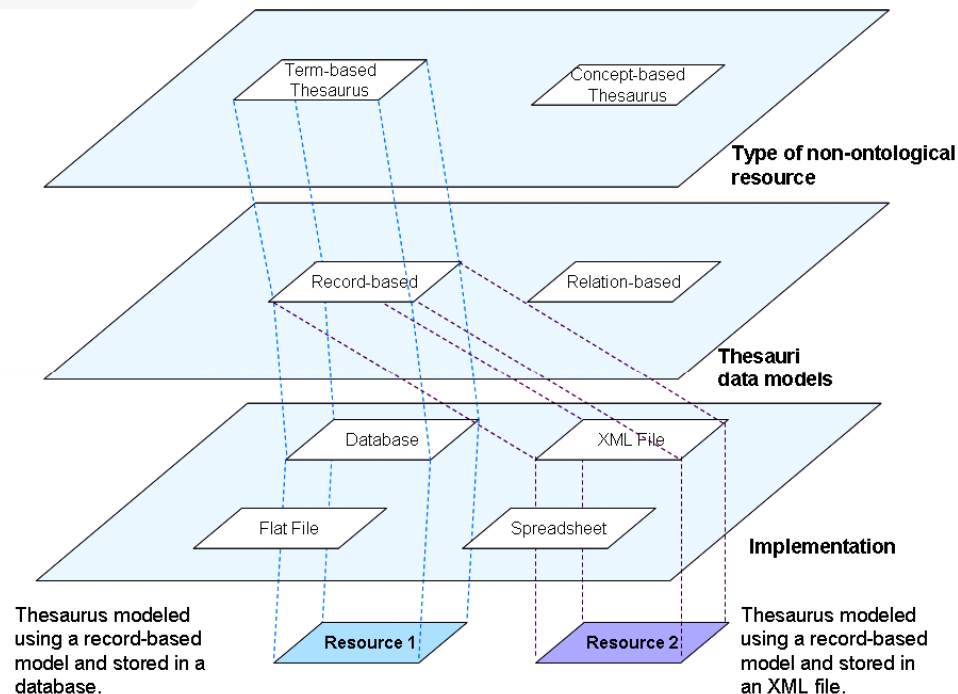
EN : Oryza	BT ( subclassOf ) : Poaceae
	NT ( hasSubclass ) : Oryza sativa
	NT ( hasSubclass ) : Oryza perennis
	NT ( hasSubclass ) : Oryza rufipogon
	NT ( hasSubclass ) : Oryza longistaminata
	NT ( hasSubclass ) : Wetland rice
	NT ( hasSubclass ) : Oryza glaberrima
	NT ( hasSubclass ) : Upland rice
	NT ( hasSubclass ) : Oryza punctata
	RT : Rice fields
	RT : Cereal crops
	RT : Rice
EN : Rice	BT ( subclassOf ) : Cereals
	NT ( hasSubclass ) : Broken rice
	NT ( hasSubclass ) : Basmati rice
	RT : Rice straw
	RT : Oryza
	RT : Rice flour
	UF : Paddy

(1) Term Entity	
TermCode	Term
1001	Term1
1002	Term2
1003	Term3
1004	Term4
1005	Term5

(2) Term-Term Relationship Entity		
TermCode1	TermCode2	RelID
1001	1003	10
1003	1004	20
1002	1005	10
1003	1005	30

(3) Relationship Entity		
RelID	RelDesc	RelAbr
10	Broader Term	BT
30	Related Term	RT
20	Used For	UF

# Example - FAO Agrovoc thesaurus



## Term-based Thesaurus

EN : Oryza	BT ( subclassOf ) : Poaceae
	NT ( hasSubclass ) : Oryza sativa
	NT ( hasSubclass ) : Oryza perennis
	NT ( hasSubclass ) : Oryza rufipogon
	NT ( hasSubclass ) : Oryza longistaminata
	NT ( hasSubclass ) : Wetland rice
	NT ( hasSubclass ) : Oryza glaberrima
	NT ( hasSubclass ) : Upland rice
	NT ( hasSubclass ) : Oryza punctata
	RT : Rice fields
	RT : Cereal crops
	RT : Rice
EN : Rice	BT ( subclassOf ) : Cereals
	NT ( hasSubclass ) : Broken rice
	NT ( hasSubclass ) : Basmati rice
	RT : Rice straw
	RT : Oryza
	RT : Rice flour
	UF : Paddy

## Record-based model

Term	BT	NT	RT	UF
Rice	Cereals	Broken rice Basmati rice	Rice straw Oryza	Paddy
Oryza	Poaceae	Oryza sativa Oryza perennis Oryza rufipogon Oryza longistaminata Wetland rice Oryza glaberrima Upland rice Oryza punctata	Rice fields Cereal crops Rice	

## XML

```

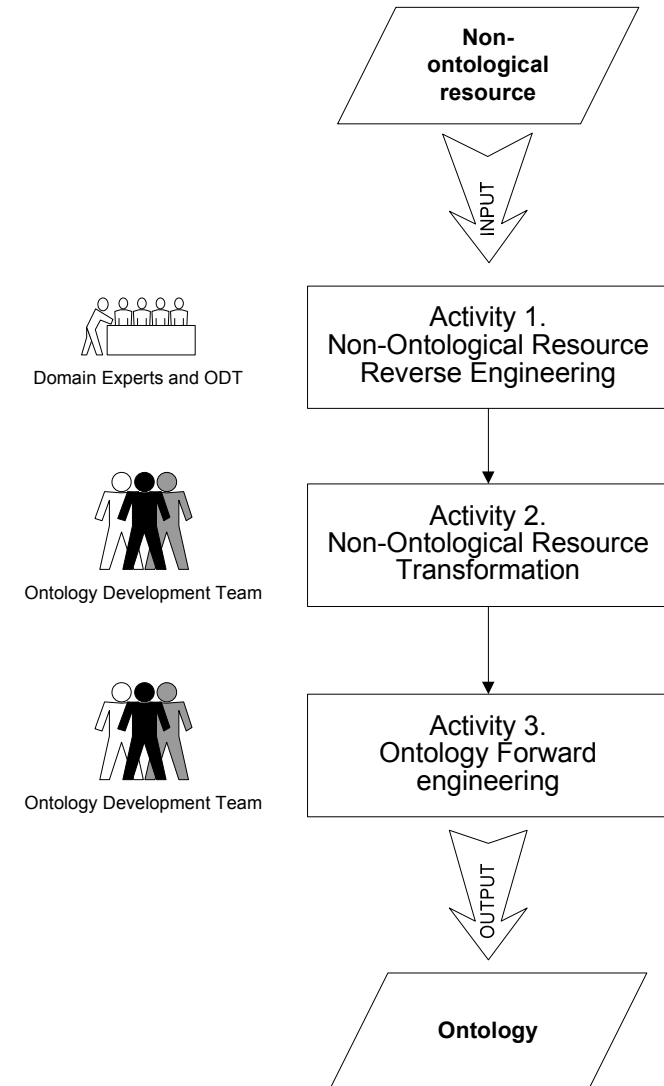
<thesaurus>
  <terms>
    <term name="Rice">
      <broadenerTerm>Cereals</broadenerTerm>
      <narrowerTerm>Broken rice</narrowerTerm>
      <narrowerTerm>Basmati rice</narrowerTerm>
      <relatedTerm>Rice straw</relatedTerm>
      <relatedTerm>Oryza</relatedTerm>
      <usedFor>Paddy</usedFor>
    </term>
    <term name="Oryza">
      <broadenerTerm>Poaceae</broadenerTerm>
      ...
    </term>
    ...
  </terms>
</thesaurus>

```

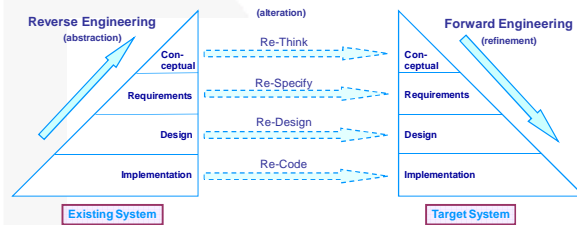
- Introduction
- A proposal for classifying/categorizing Non-Ontological Resources
  - Classification scheme
  - Thesaurus
- Method for Re-engineering Non-Ontological Resources
- PR-NORs

# Non-Ontological Resource Re-engineering (I)

- **Definition:** The process of taking an existing non-ontological resource and transforming it into ontology.
- **Input:** NOR
- **Output:** An ontology

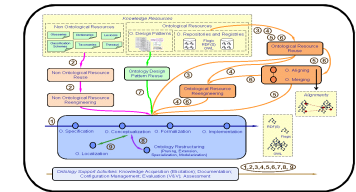


# Non-Ontological Resource Re-engineering (II)

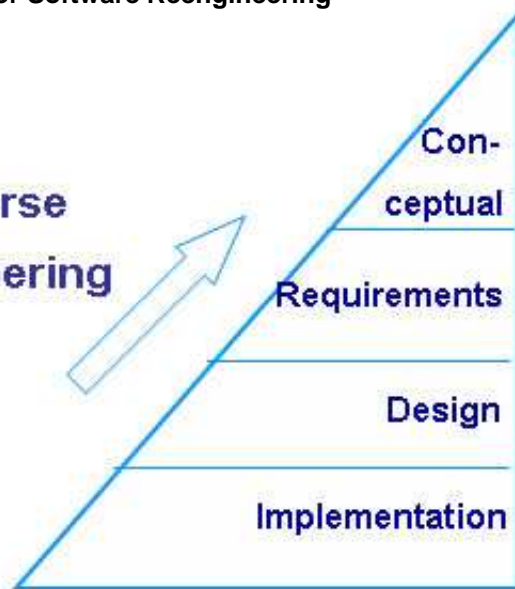


General Model for Software Reengineering

## Patterns for Reengineering Non Ontological Resources (PR-NOR)

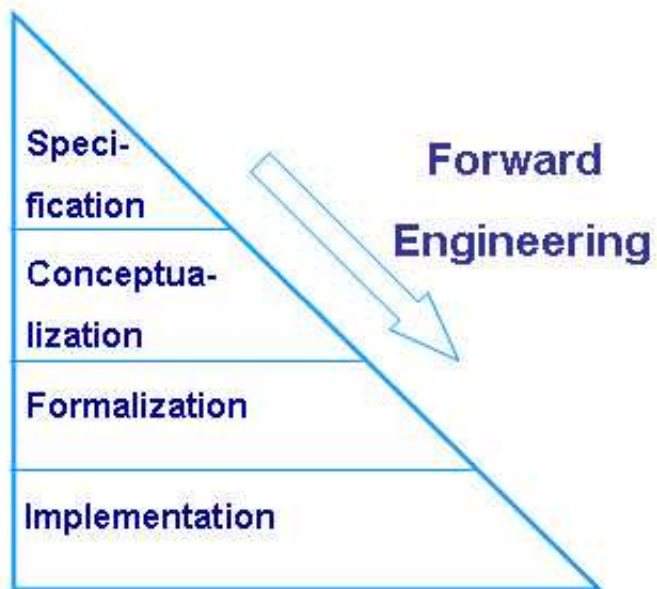


**Reverse Engineering**



**Non Ontological Resource**

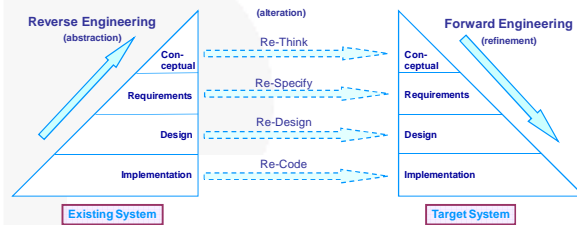
**Transformation**



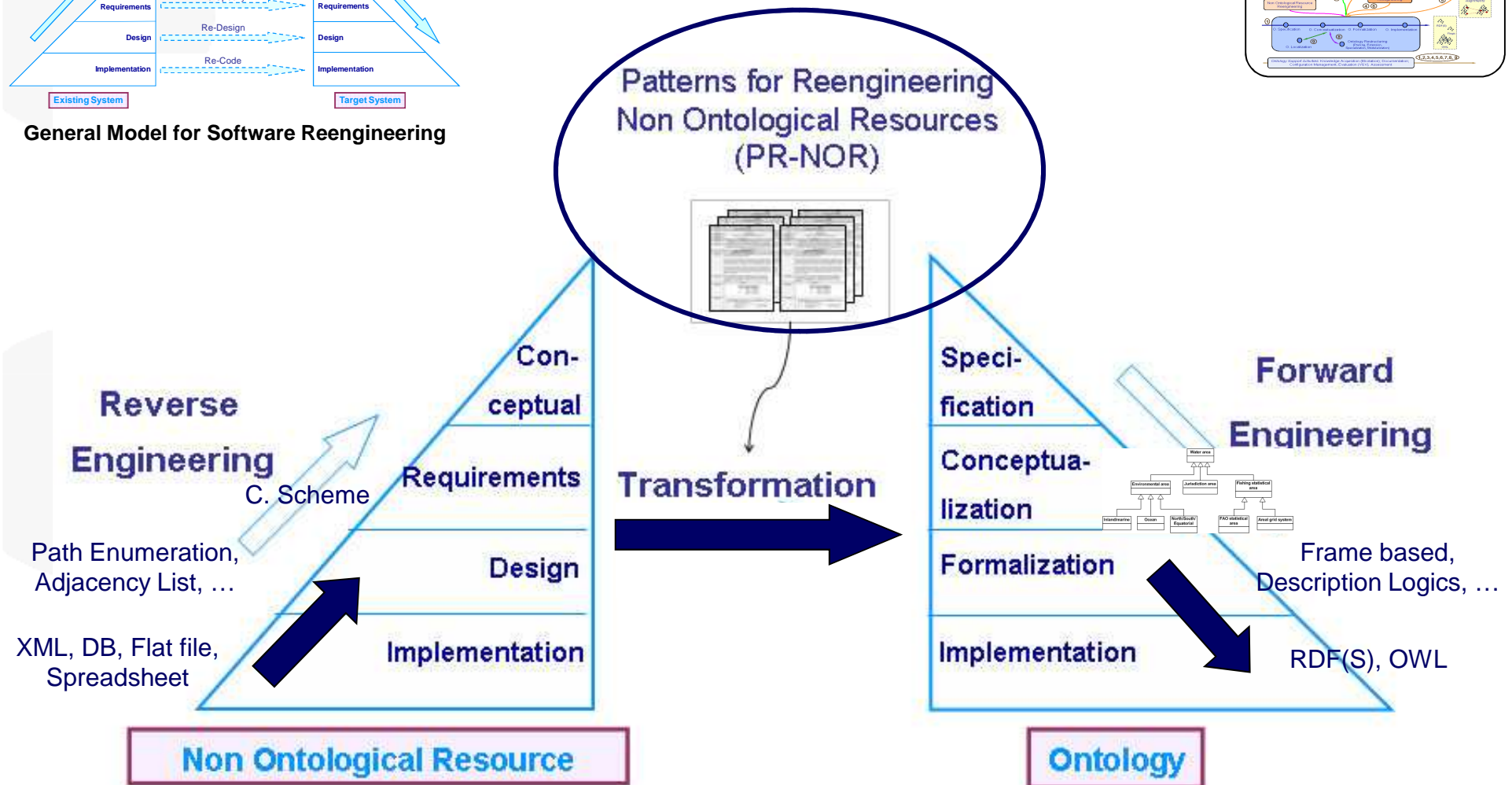
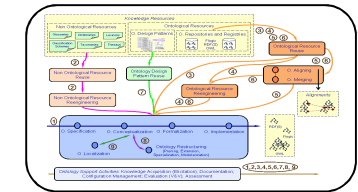
**Ontology**

- 1) The conceptual level describes in general terms the functional characteristics of the system;
- 2) The requirement level is the specification of the problem being solved;
- 3) The design level is the specification of the solution; and
- 4) The implementation level refers to the coding, testing and delivery of the operational system

# Non-Ontological Resource Re-engineering (III)

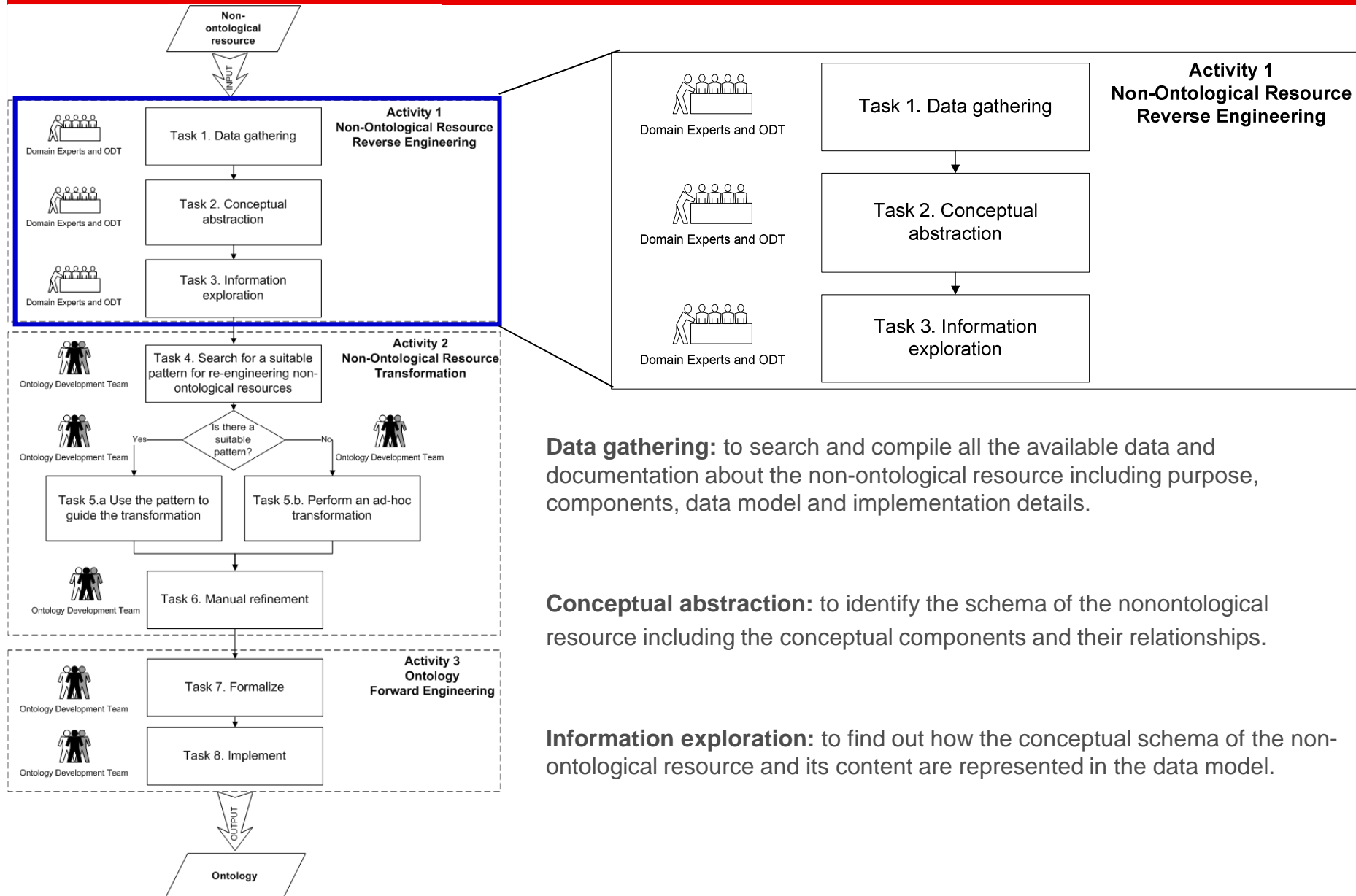


General Model for Software Reengineering





# Non-Ontological Resource Re-engineering (IV)

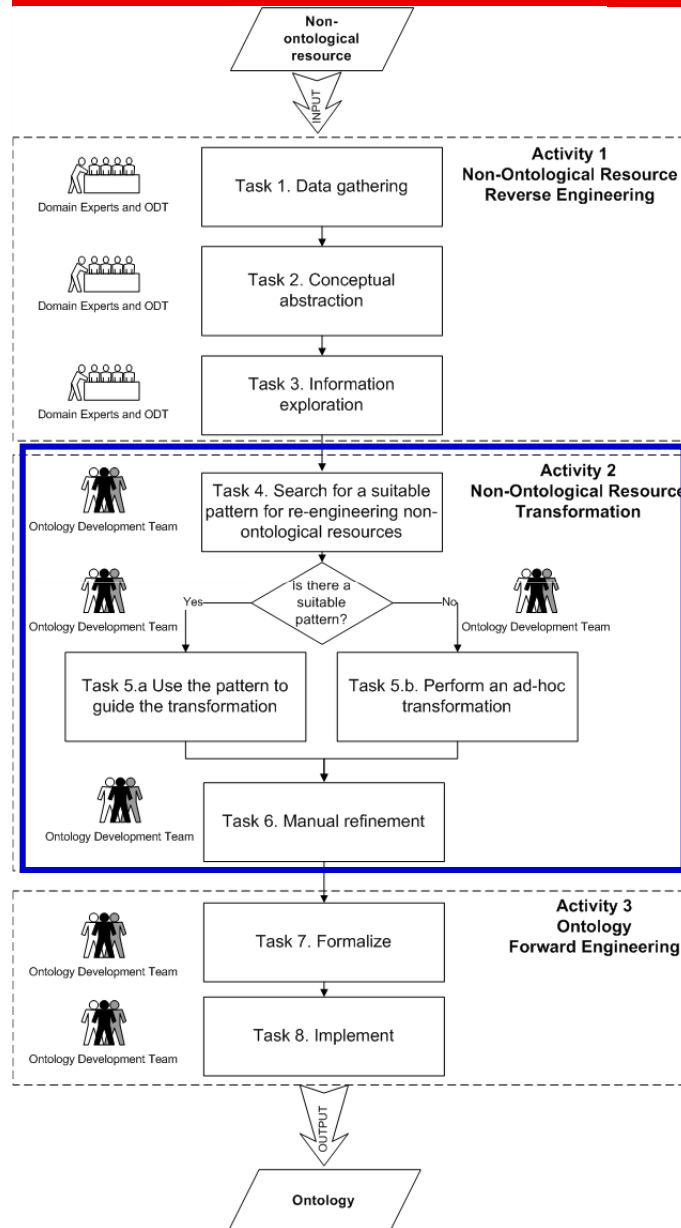


**Data gathering:** to search and compile all the available data and documentation about the non-ontological resource including purpose, components, data model and implementation details.

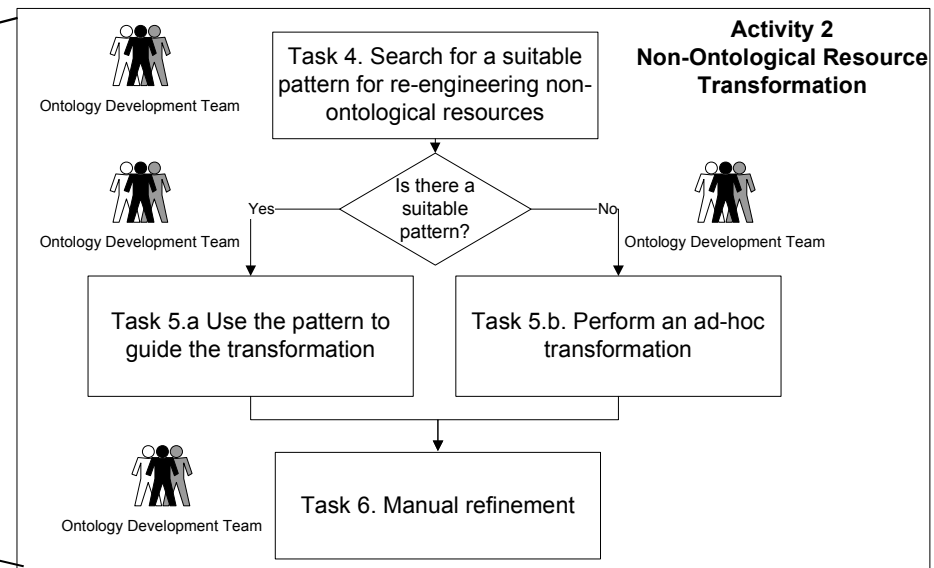
**Conceptual abstraction:** to identify the schema of the nonontological resource including the conceptual components and their relationships.

**Information exploration:** to find out how the conceptual schema of the non-ontological resource and its content are represented in the data model.

# Non-Ontological Resource Re-engineering (V)



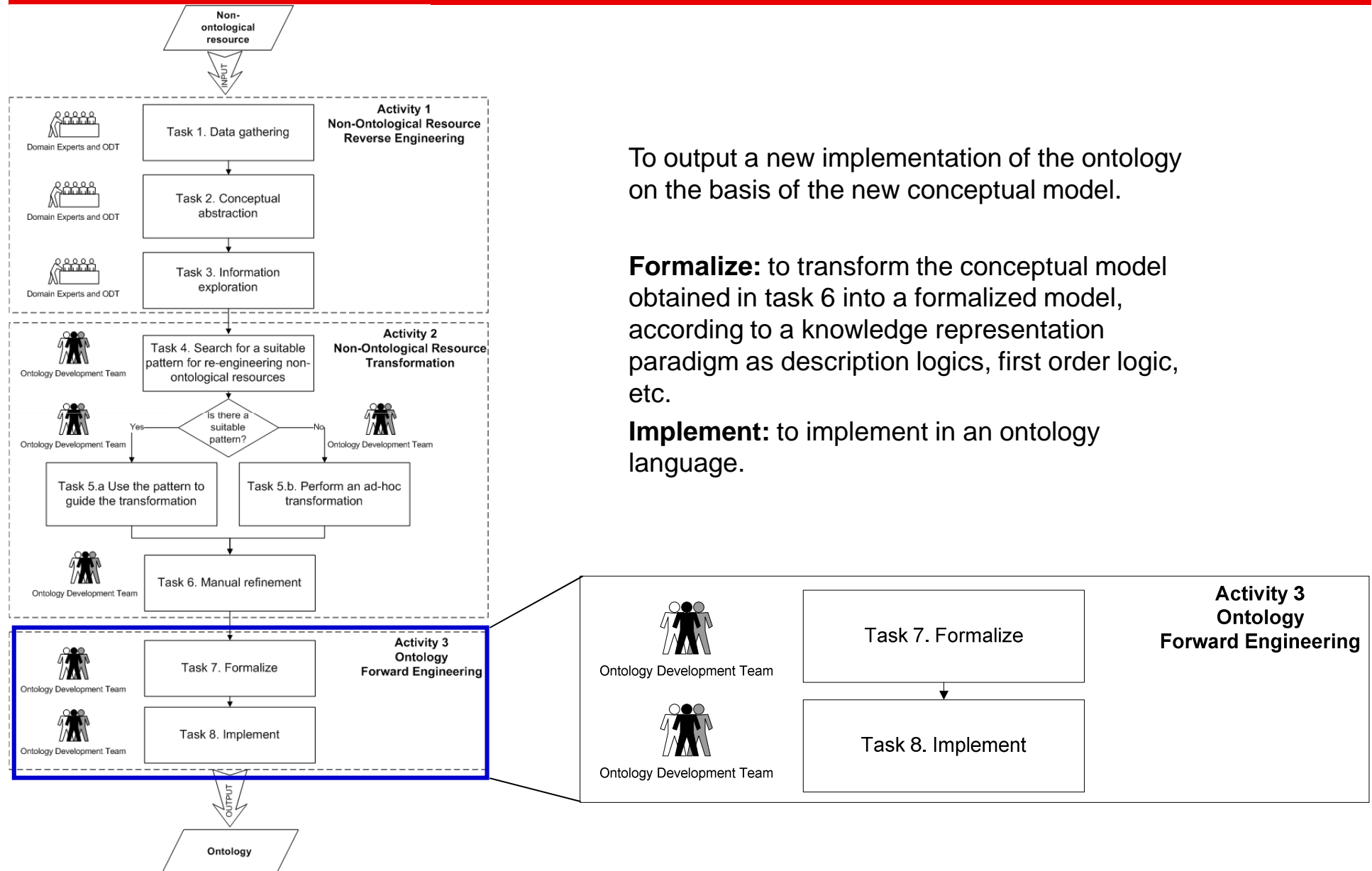
The goal is to generate a conceptual model from the NOR



Criteria for searching for a suitable pattern:

- **NOR Type:** classification scheme, thesauri, lexicon, glossary, or folksonomy
- **Data model:** C.Scheme (path enumeration, adjacency list, snowflake, or flattened), Thesaurus (record-based, relation-based)
- **Transformation approach:** TBox, ABox, or population

# Non-Ontological Resource Re-engineering (VI)



- Introduction
- A proposal for classifying/categorizing Non-Ontological Resources
  - Classification scheme
  - Thesaurus
- Method for Re-engineering Non-Ontological Resources
- PR-NORs

## Patterns for Re-engineering Non-Ontological Resources (PR-NOR)

- PR-NORs define a procedure that transforms the non-ontological resource components into ontology representational primitives.
- These patterns will generate the ontologies at a conceptualization level, independent of the ontology implementation language.
- Advantages:
  - embody expertise about how to guide a re-engineering process,
  - improve the efficiency of the re-engineering process,
  - make the transformation process easier for both ontology engineers and domain experts
  - improve the reusability of non-ontological resources.

# 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 <a href="#">[BHO6]</a> , 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 <a href="#">[BHO6]</a> , 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

INPUT: 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.

OUTPUT: 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.

PROCESS: 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.

# PR-NOR Transformation Approaches

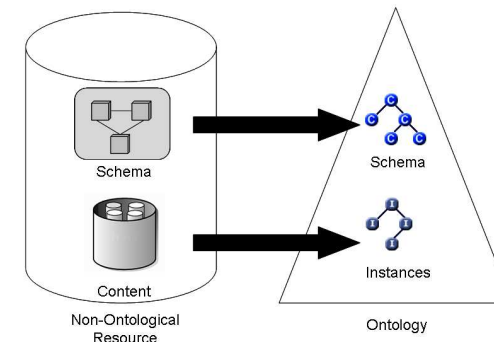
- ABox Transformation

This transformation approach leaves the informal semantics of the re-engineered resources mostly untouched

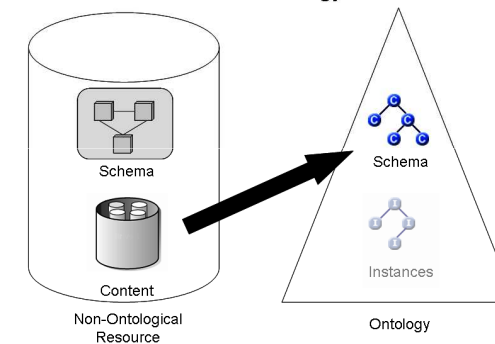
- TBox Transformation

This transformation approach tries to enforce a formal semantics to the re-engineered resources, even at the cost of changing their structure.

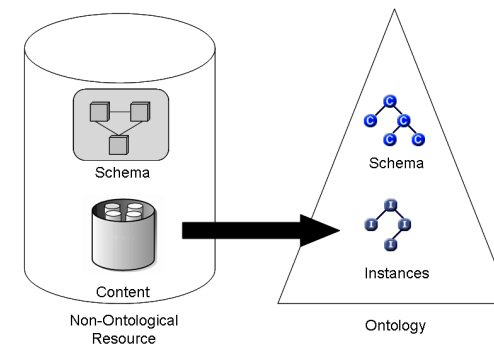
- Population



Transforming resource schema into an ontology schema, and resource content into ontology instances



Transforming resource content into an ontology schema

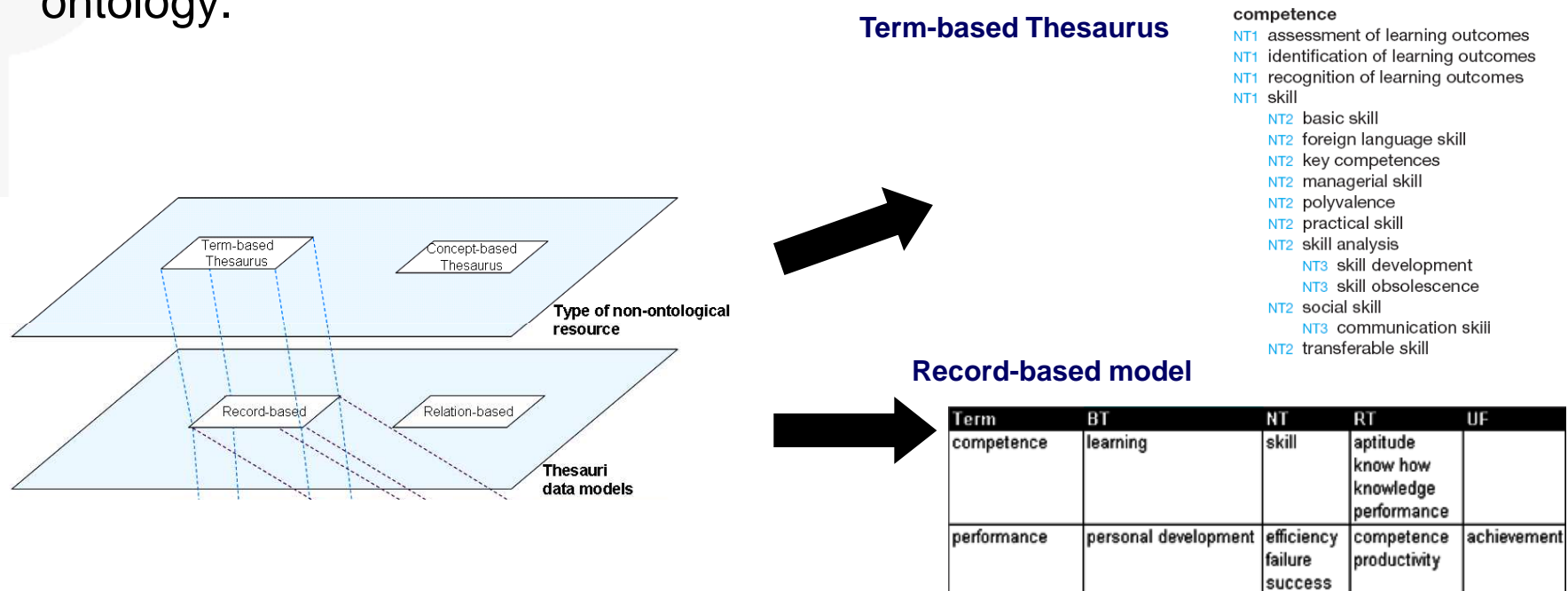


Transforming resource content into instances of an existing ontology



# Problem

Re-engineering a term-based thesaurus, which follows the record-based model, to design an ontology.



## Applicability

- The semantics of the relation between narrower and broader terms: *subClassOf*.

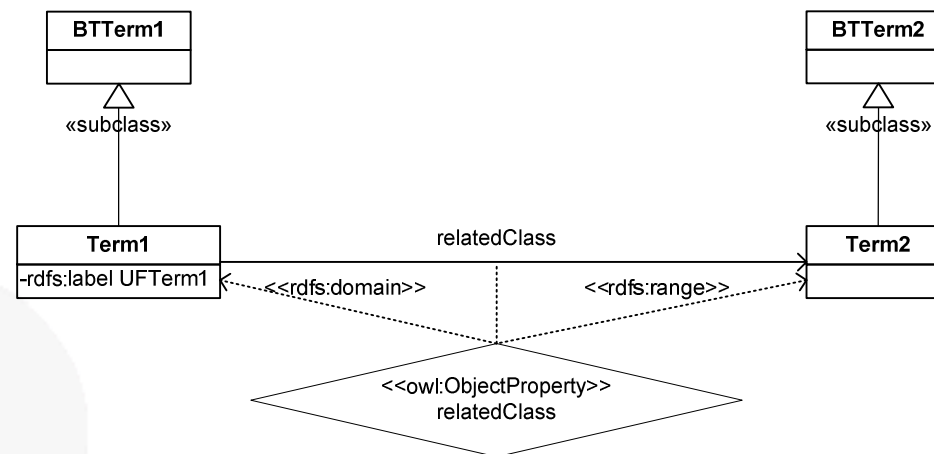
## Input: Non-Ontological Resource

- A non-ontological resource holds a term-based thesaurus which follows the record-based model.
- A thesaurus represents the knowledge of a domain with a collection of terms and a limited set of relations between them.
- The record-based data model is a denormalized structure, it uses a record for every term with the information about the term, such as synonyms, broader, narrower and related terms.

Term	BT	NT	RT	UF
Term1	BTterm1	NTTerm1 NTTerm2	Term2	UFTerm1
Term2	BTterm2	NTTerm3 NTTerm4	RTerm3 RTerm4 RTerm5	

# Output: Ontology Generated

- The ontology generated will be based on the lightweight ontology architectural pattern (AP-LW-01). Each thesaurus term is mapped to a class.
- A *subClassOf* relation is defined between the new classes for the BT/NT relation.
- A *relatedClass* relation is defined between the new classes for the RT relation.
- For the UF/USE relations we use the logical pattern proposed by Corcho et al. suggested as best practice in the context of this antipattern: the tendency to declare two classes equivalent when in fact their labels simply express synonym.



# Process

1. Identify the records that contain thesaurus terms without a broader term.
2. For each one of the above identified thesaurus terms  $t_i$ :
  - 2.1 Create the corresponding ontology class,  $C_i$  class, if it is not created yet.

Term	BT	NT	RT	UF
Term1	BTTerm1	NTTerm1 NTTerm2	Term2	UFTerm1
Term2	BTTerm2	NTTerm3 NTTerm4	RTerm3 RTerm4 RTerm5	

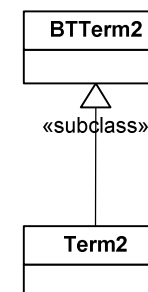
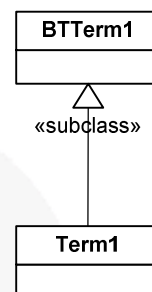
BTTerm1

BTTerm2

# Process

- 2.2 Identify the thesaurus terms,  $t_j$ , which are narrower terms of  $t_i$ . They are referenced in the same record that contains  $t_i$ .
- 2.3 For each one of the above identified thesaurus terms  $t_j$ :
- 2.3.1 Create the corresponding ontology class,  $C_j$  class, if it is not created yet.
  - 2.3.2 Set up the *subClassOf* relation between  $C_j$  and  $C_i$
  - 2.3.3 Repeat from step 2.2 for  $t_j$  as a new  $t_i$ .

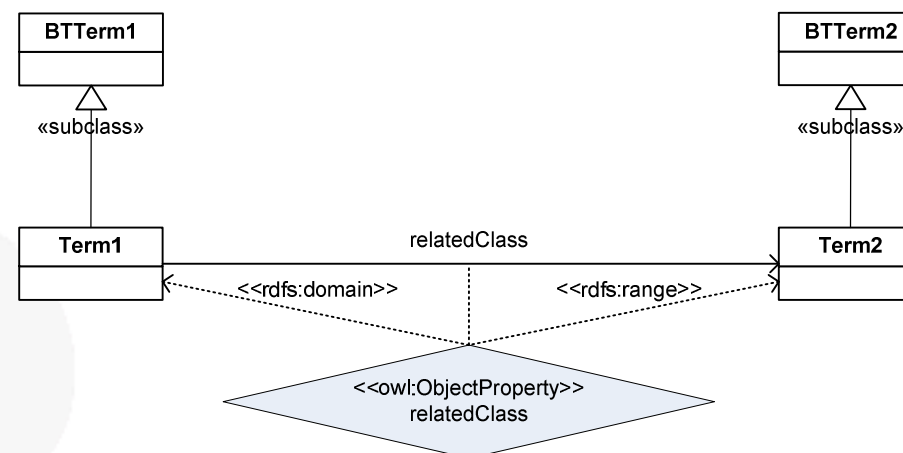
Term	BT	NT	RT	UF
Term1	BTTerm1	NTTerm1 NTTerm2	Term2	UFTerm1
Term2	BTTerm2	NTTerm3 NTTerm4	RTerm3 RTerm4 RTerm5	



# Process

- 2.4 Identify the thesaurus terms, *tr*, which are related terms of *ti*. They are referenced in the same record that contains *ti*.
- 2.5 For each one of the above identified thesaurus terms *tr*:
  - 2.5.1 Create the corresponding ontology class, *Cr* class, if it is not created yet.
  - 2.5.2 Set up the *relatedClass* relation between *Cr* and *Ci*.
  - 2.5.3 Repeat from step 2.4 for *tr* as a new *ti*.

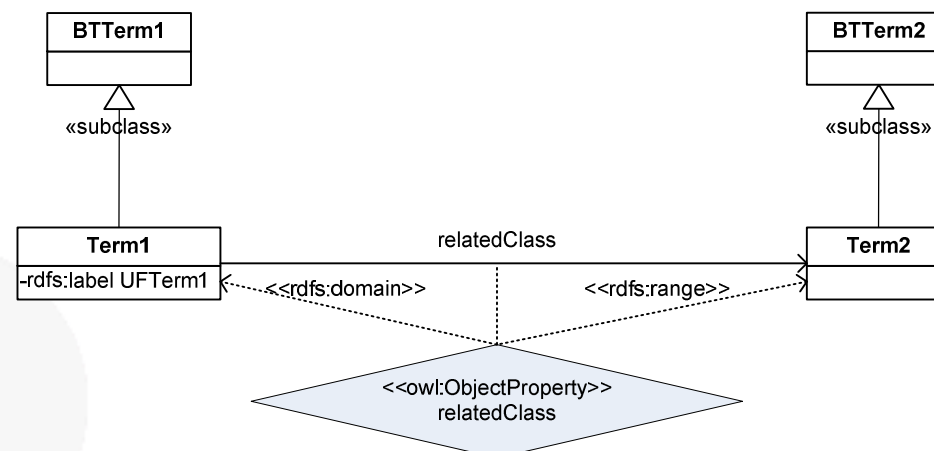
Term	BT	NT	RT	UF
Term1	BTTerm1	NTTerm1 NTTerm2	Term2	UFTerm1
Term2	BTTerm2	NTTerm3 NTTerm4	RTerm3 RTerm4 RTerm5	



# Process

- 2.6 Identify the thesaurus terms, *tq*, which are equivalent terms of *ti*. They are referenced in the same record that contains *ti*.
- 2.7 For each one of the above identified thesaurus terms *tq*:
- 2.7.1 Use the logical pattern proposed by Corcho et al.

Term	BT	NT	RT	UF
Term1	BTerm1	NTTerm1 NTTerm2	Term2	UTerm1
Term2	BTerm2	NTTerm3 NTTerm4	RTerm3 RTerm4 RTerm5	



# Example

- Suppose that someone wants to build a lightweight ontology based on the European Training Thesaurus (ETT), which is a term-based thesaurus and it follows the record-based model.

**competence**

- NT1 assessment of learning outcomes
- NT1 identification of learning outcomes
- NT1 recognition of learning outcomes
- NT1 skill
  - NT2 basic skill
  - NT2 foreign language skill
  - NT2 key competences
  - NT2 managerial skill
  - NT2 polyvalence
  - NT2 practical skill
  - NT2 skill analysis
    - NT3 skill development
    - NT3 skill obsolescence
  - NT2 social skill
    - NT3 communication skill
  - NT2 transferable skill



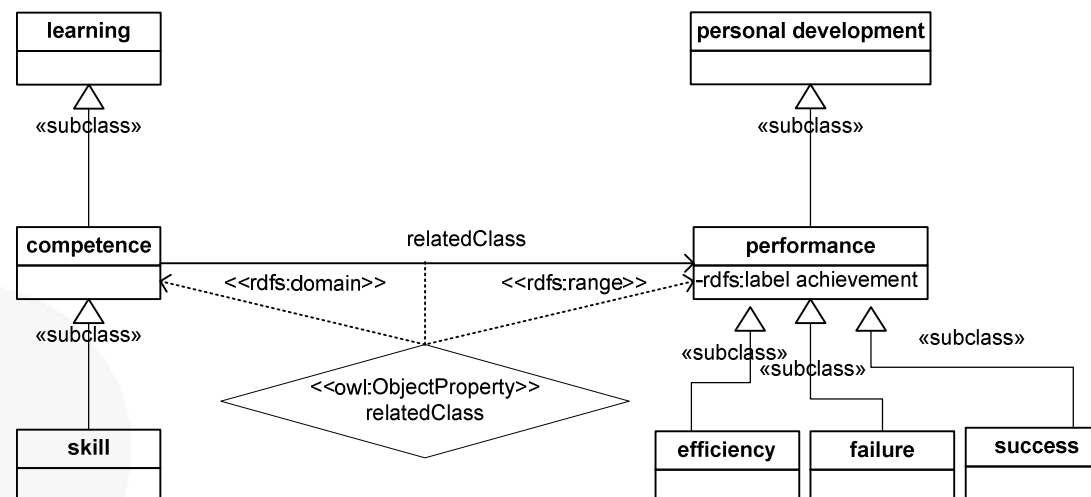
## Input: Non-Ontological Resource

- The European Training Thesaurus (ETT) constitutes the controlled vocabulary of reference in the field of vocational education and training (VET) in Europe. The relation semantics between the sub-ordinate and the super-ordinate concepts is *subClassOf*.
- This thesaurus is available at <http://libserver.cedefop.europa.eu/ett/en/>

Term	BT	NT	RT	UF
competence	learning	skill	aptitude know how knowledge performance	
performance	personal development	efficiency failure success	competence productivity	achievement

## Output: Ontology Generated

- The ontology generated will be based on the lightweight ontology architectural pattern (AP-LW-01). Each thesaurus term is mapped to a class.
- A *subClassOf* relation is defined between the new classes for the BT/NT relation.
- A *relatedClass* relation is defined between the new classes for the RT relation.
- For the UF/USE relations we use the logical pattern proposed by Corcho et al. suggested as best practice in the context of this antipattern: the tendency to declare two classes equivalent when in fact their labels simply express synonym.



# Process

1. Identify the records that contain thesaurus terms without a broader term.
2. For each one of the above identified thesaurus terms  $t_i$ :

2.1 Create the corresponding ontology class,  $C_i$  class, if it is not created yet.

Term	BT	NT	RT	UF
competence	learning	skill	aptitude know how knowledge performance	
performance	personal development	efficiency failure success	competence productivity	achievement

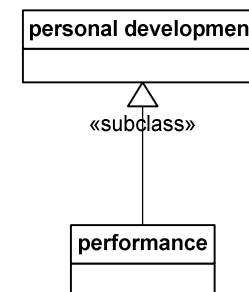
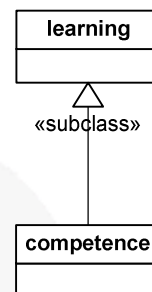
learning

personal development

# Process

- 2.2 Identify the thesaurus terms,  $t_j$ , which are narrower terms of  $t_i$ . They are referenced in the same record that contains  $t_i$ .
- 2.3 For each one of the above identified thesaurus terms  $t_j$ :
- 2.3.1 Create the corresponding ontology class,  $C_j$  class, if it is not created yet.
  - 2.3.2 Set up the *subClassOf* relation between  $C_j$  and  $C_i$
  - 2.3.3 Repeat from step 2.2 for  $t_j$  as a new  $t_i$ .

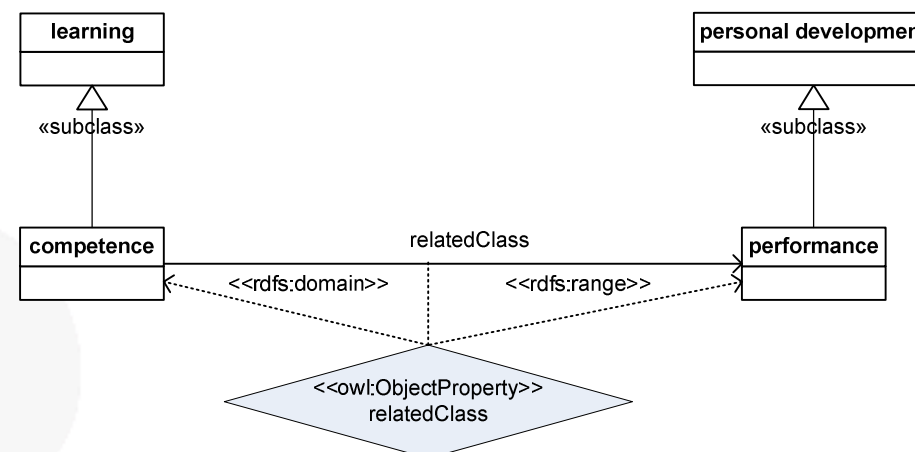
Term	BT	NT	RT	UF
competence	learning	skill	aptitude know how knowledge performance	
performance	personal development	efficiency failure success	competence productivity	achievement



# Process

- 2.4 Identify the thesaurus terms, *tr*, which are related terms of *ti*. They are referenced in the same record that contains *ti*.
- 2.5 For each one of the above identified thesaurus terms *tr*:
  - 2.5.1 Create the corresponding ontology class, *Cr* class, if it is not created yet.
  - 2.5.2 Set up the *relatedClass* relation between *Cr* and *Ci*.
  - 2.5.3 Repeat from step 2.4 for *tr* as a new *ti*.

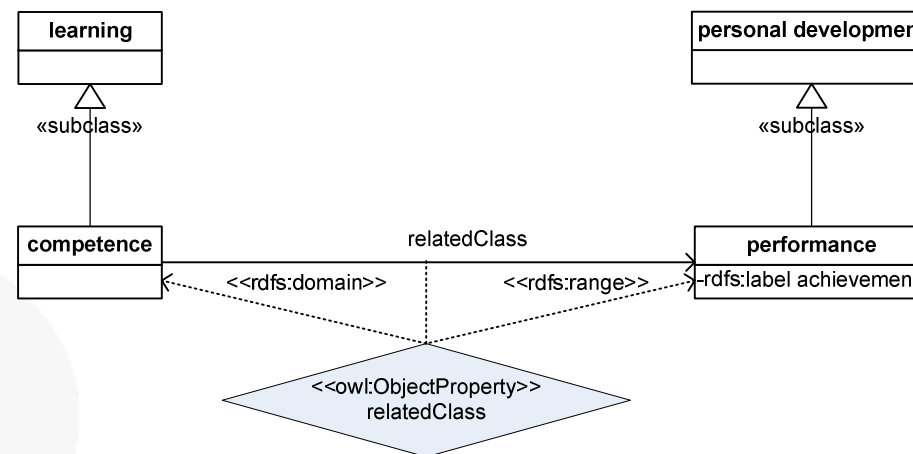
Term	BT	NT	RT	UF
competence	learning	skill	aptitude know how knowledge performance	
performance	personal development	efficiency failure success	competence productivity	achievement



# Process

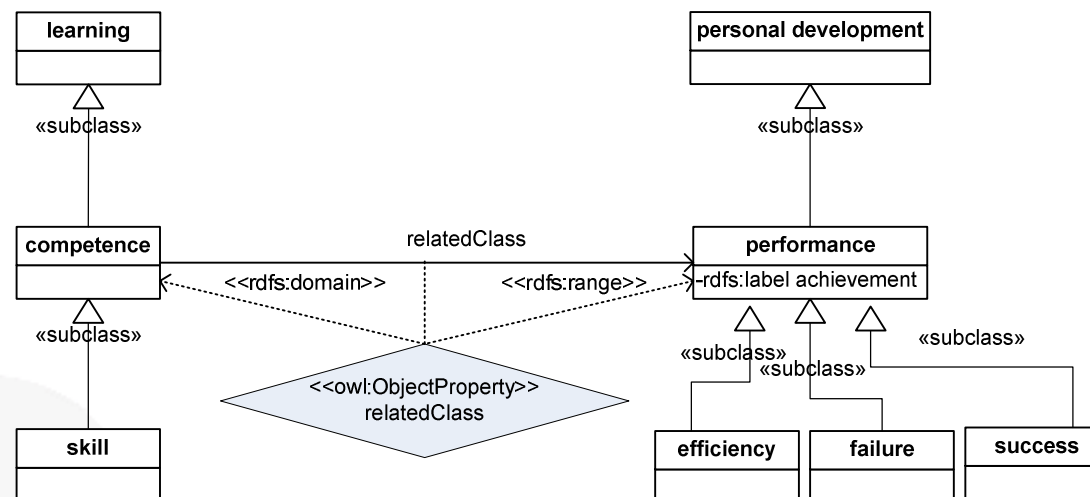
- 2.6 Identify the thesaurus terms, *tq*, which are equivalent terms of *ti*. They are referenced in the same record that contains *ti*.
- 2.7 For each one of the above identified thesaurus terms *tq*:
- 2.7.1 Use the logical pattern proposed by Corcho et al.

Term	BT	NT	RT	UF
competence	learning	skill	aptitude know how knowledge performance	
performance	personal development	efficiency failure success	competence productivity	achievement

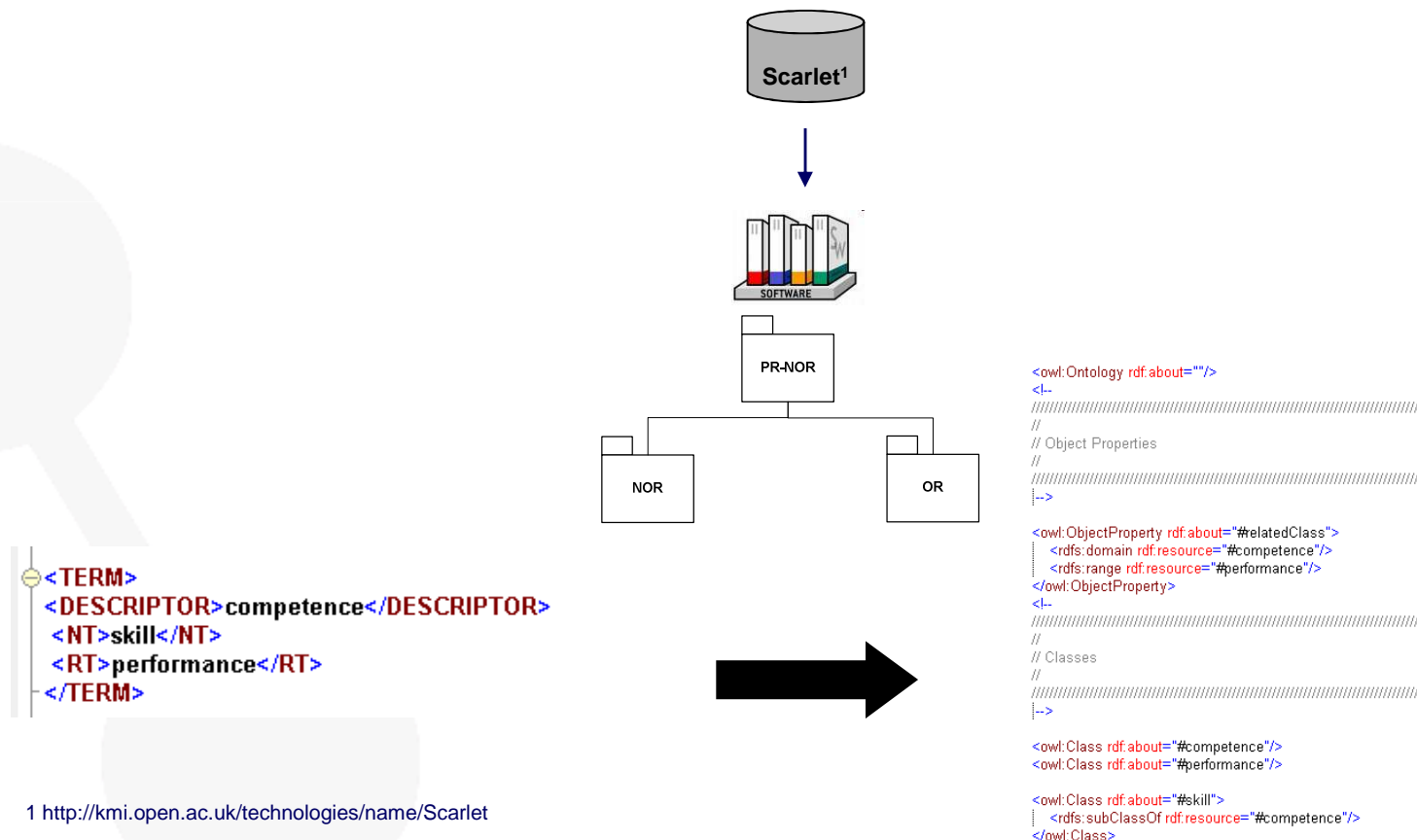


# Process

Term	BT	NT	RT	UF
competence	learning	skill	aptitude know how knowledge performance	
performance	personal development	efficiency failure success	competence productivity	achievement



- We include software libraries in a framework which implements the transformation process.
- Addition of external resources to disambiguate the semantic relationships among NOR elements.



1 <http://kmi.open.ac.uk/technologies/name/Scarlet>



- There are 3 configuration files:
  - nor.xml, for describing the non-ontological resource.
  - prnor.xml, for describing the pattern for re-engineering.
  - or.xml, for describing the ontology generated.
- There is a batch file
  - prnor.bat, for performing the transformation
- Requirements
  - Java runtime environment (tested in 1.5)
  - You should set the JAVA\_HOME to the location of the installation directory.

## Example: Transform the ASFA thesaurus

- non-ontological resource: asfa
  - Type: term-based thesaurus
  - Datamodel: record-based
  - Implementation: XML

 asfa\_xml\_20060522 xml

2.780.181 21/10/2009 01:36

```
<THESAURUS>
<TERM>
<NON-DESCRIPTOR>AAS</NON-DESCRIPTOR>
<USE>Absorption spectroscopy</USE>
<STA>Approved</STA>
<TYP>Non-descriptor</TYP>
<INP>2006-05-22</INP>
<UPD>2006-05-22</UPD>

</TERM>

<TERM>
<NON-DESCRIPTOR>Abalone fisheries</NON-DESCRIPTOR>
<USE>Gastropod fisheries</USE>
<STA>Approved</STA>
<TYP>Non-descriptor</TYP>
<INP>2006-05-22</INP>
<UPD>2006-05-22</UPD>

</TERM>

<TERM>
<DESCRIPTOR>Abdomen</DESCRIPTOR>
<UF>Peritoneum</UF>
<BT>Body regions</BT>
<RT>Digestive system</RT>
<STA>Approved</STA>
<TYP>Descriptor</TYP>
<INP>2006-05-22</INP>
<UPD>2006-05-22</UPD>

</TERM>
```

## Example: Transform the ASFA thesaurus

- nor.xml, description of the non-ontological resource

```
<Nor type="Thesaurus" name="ASFA">
  <Schema>
    <SchemaEntities>
      <SchemaEntity name="Term">
        <Attribute name="Identifier" valueFrom="DESCRIPTOR" type="string"/>
        <Relation name="NT" using="RecordBased" valueId="NT" destination="Term"/>
        <Relation name="BT" using="RecordBased" valueId="BT" destination="Term"/>
        <Relation name="RT" using="RecordBased" valueId="RT" destination="Term"/>
        <Relation name="UF" using="RecordBased" valueId="UF" destination="NonPreferredTerm"/>
      </SchemaEntity>
      <SchemaEntity name="NonPreferredTerm">
        <Attribute name="Identifier" valueFrom="NON-DESCRIPTOR" type="string"/>
      </SchemaEntity>
    </SchemaEntities>
  </Schema>
  <DataModel>
    <RecordBased>
      <Entity>TERM</Entity>
    </RecordBased>
  </DataModel>
  <Implementation>
    <Xml xmlFile="nors/thesauri/recordbased/asfa_xml_20060522.xml" xsdFile="nors/thesauri/recordbased/asfa.xsd"/>
  </Implementation>
</Nor>
```

## Example: Transform the ASFA thesaurus

- prnor.xml, description of the pattern
  - Transformation approach: TBox
    - Term -> Class

```
<Pnor identifier="PR-NOR-CLLO-10" transformationApproach="TBox">
  <Class from="Term" identifier="[Identifier]">
    <ObjectProperty from="NT" to="NT"/>
    <ObjectProperty from="RT" to="RT"/>
    <ObjectProperty from="BT" to="BT"/>
    <ObjectProperty from="UF" to="rdfs:label"/>
  </Class>
</Pnor>
```

- or.xml, description of the ontology generated

```
<Or name="ASFA Ontology" ontologyURI="http://droz.dia.fi.ump.es/ontologies/asfa.owl" ontologyFile="asfa.owl" implementation="OWL" alreadyExist="no">
</Or>
```

## Example: Transform the ASFA thesaurus

- Run the transformation

```
C:\prnor>prnor
Starting ...
Running ...
Done
C:\prnor>_
```

- Wait until it finishes, and check the ontology.

 asfa

owl

3.170.728 03/11/2009 00:05a--

```
<!-- http://droz.dia.fi.ump.es/ontologies/asfa.owl#Absorption_spectroscopy -->
<owl:Class rdf:about="#Absorption_spectroscopy"/>
<!-- http://droz.dia.fi.ump.es/ontologies/asfa.owl#Abalone_fisheries -->
<owl:Class rdf:about="#Abalone_fisheries"/>
<!-- http://droz.dia.fi.ump.es/ontologies/asfa.owl#Abdomen -->
<owl:Class rdf:about="#Abdomen"/>
<!-- http://droz.dia.fi.ump.es/ontologies/asfa.owl#Enzymes -->
<owl:Class rdf:about="#Enzymes"/>
```



# A Method for Re-engineering Non-Ontological Resources for Building Ontologies

Boris Villazón-Terrazas, Mari Carmen Suárez Figueroa

{bvillazon,mcsuarez}@fi.upm.es

Ontology Engineering Group. Laboratorio de Inteligencia Artificial

Departamento de Inteligencia Artificial

Facultad de Informática

Universidad Politécnica de Madrid