Chapter 6
Reusing and Re-engineering Non-ontological Resources for Building Ontologies

Boris Villazón-Terrazas and Asunción Gómez-Pérez

Abstract With the goal of speeding up the ontology development process, ontology developers are reusing as much as possible available ontological and non-ontological resources such as classification schemes, thesauri, lexicons, and folksonomies, that have already reached some consensus. The reuse of such non-ontological resources necessarily involves their re-engineering into ontologies. Based on this new trend, this chapter presents a general method for re-engineering non-ontological resources into ontologies, taking into account that non-ontological resources are highly heterogeneous in their data model and contents. The method is based on the so-called re-engineering patterns, which define a procedure that transforms the non-ontological resource components into ontology representational primitives. This chapter also presents the description of a software library that implements the transformations suggested by the patterns. Finally, the chapter depicts an evaluation of the method.

6.1 Introduction and Motivation

Research on ontology engineering methodologies has provided methods and techniques for developing ontologies from scratch. Well-recognized methodological approaches such as METHONTOLOGY (Gómez-Pérez et al. 2003), On-To-Knowledge (Schnurr et al. 2001), and DILIGENT (Pinto et al. 2004) issue guidelines that help researchers to develop ontologies. However, researchers face

B. Villazón-Terrazas (✉) • A. Gómez-Pérez
Ontology Engineering Group, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo sn., 28660 Boadilla del Monte, Madrid, Spain
e-mail: bvillazon@fi.upm.es; asun@fi.upm.es

M.C. Suárez-Figueroa et al. (eds.), *Ontology Engineering in a Networked World*,
DOI 10.1007/978-3-642-24794-1_6,

an important limitation: no guidelines are provided for building ontologies by re-engineering some knowledge resources widely used within a particular community.

During the last decade, specific methods, techniques, and tools were proposed for building ontologies from available knowledge resources. First, ontology learning methods and tools were proposed to extract relevant concepts and relations from structured, semi-structured, and non-structured resources (Gómez-Pérez and Manzano-Macho 2004; Maedche and Staab 2001) in order to form a single ontology. One important constraint of these methods and tools is that they propose ad hoc solutions to transforming such resources, mainly texts, into ontologies. Hepp (2006), Hepp and de Brujin (2007), and Hepp (2007) stated that employing methods and techniques when transforming non-ontological resources to ontologies is key for the success of semantic technology for two main reasons: (1) if the use of semantic technologies for real-world data integration challenges is required, it is possible to refer to the original conceptual elements, and (2) for many domains, the existing category systems, XML schemas, and normative entity identifiers are the most efficient resources for engineering ontologies.

The literature presents a wide set of methods and tools for the ontologization of non-ontological resources. This ontologization of resources has led to the design of several specific methods, techniques, and tools (Hepp and de Brujin 2007; Hyvöonen et al. 2008; Gangemi et al. 2003; García and Celma 2005). These are mainly specific to a particular resource type, or to a particular resource implementation. Thus, every time ontology engineers are faced with a new resource type or implementation, they develop ad hoc solutions to transforming such resource into a single ontology.

The analysis of the ontologies developed by distinct research groups in different international and national projects have revealed that there are different alternative ways or possibilities to build ontologies by reusing and re-engineering the available knowledge resources used by a particular community. However, at this stage, we can state that all the projects perform an ad hoc transformation of the resources available for building ontologies.

Therefore, a new ontology development paradigm started approximately in 2007, whose emphasis was on the *reuse and possible subsequent reengineering of knowledge resources*, as opposed to custom-building new ontologies from scratch. However, in order to support and promote such reuse-based approach, new methods, techniques, and tools are needed.

The remainder of the chapter is organized as follows: Section 6.2 presents our categorization of non-ontological resources. Then, Sect. 6.3 describes the methodological guidelines for reusing non-ontological resources. Next, Sect. 6.4 provides the pattern-based method for re-engineering non-ontological resources into ontologies. Section 6.5 introduces the technological support for our re-engineering method. Then, Sect. 6.6 describes an example of the methodological guidelines presented here. Finally, Sect. 6.7 presents the conclusions and future work.

6.2 Types of Non-ontological Resources

The knowledge resources, reused in several projects for building ontologies, contain readily available a wealth of category definitions and reflect some degree of community consensus. In this chapter, we refer to *non-ontological resources (NOR)*[1]. Examples of NORs are classification schemes, thesauri, lexica, and folksonomies, among others. This type of resources encodes different types of knowledge and can be implemented in different ways.

Our analysis of the literature has revealed different ways of categorizing non-ontological resources. Thus, Maedche and Staab (2001) and Sabou et al. (2007) classify non-ontological resources into unstructured (e.g., free text), semi-structured (e.g., folksonomies), and structured (e.g., databases) resources, whereas Gangemi et al. (1998) distinguish catalogs of normalized terms, glossed catalogues, and taxonomies. Finally, Hodge (2000) proposes characteristics such as structure, complexity, relationships among terms, and historical functions for classifying them. However, an accepted and agreed-upon typology of non-ontological resources does not exist yet.

Therefore, one of the contributions of this chapter is the categorization of NORs, according to the following three features presented in Fig. 6.1: (1) type of NOR, which refers to the type of inner organization of the information; (2) data model, that is, the design data model used to represent the knowledge encoded by the resource; and (3) resource implementation.

According to the *type of NORs,* we classify them into:

- *Glossaries*: A glossary is an alphabetical list of terms or words found in or related to a specific topic or text. It may or may not include explanations, and its vocabulary may be monolingual, bilingual, or multilingual (Wright and Budin 1997). An example of glossary is the FAO Fisheries Glossary[2].
- *Lexicons*: In a restricted sense, a computational lexicon is considered as a list of words or lexemes hierarchically organized and normally accompanied by meaning and linguistic behavior information (Hirst 2004). A fine example is WordNet[3], the best known computational lexicon of English.
- *Classification schemes*: A classification scheme is the descriptive information of an arrangement or division of objects into groups according to the characteristics that the objects have in common (ISO/IEC FDIS 11179-1). A good example is the Fishery International Standard Statistical Classification of Aquatic Animals and Plants (ISSCAAP)[4].
- *Thesauri*: Thesauri are controlled vocabularies of terms in a particular domain with hierarchical, associative, and equivalence relations between terms.

[1] Along this chapter, we use either *NOR* or *non-ontological resource* without distinction

[2] http://www.fao.org/fi/glossary/default.asp

[3] http://wordnet.princeton.edu/

[4] http://www.fao.org/figis/servlet/RefServlet

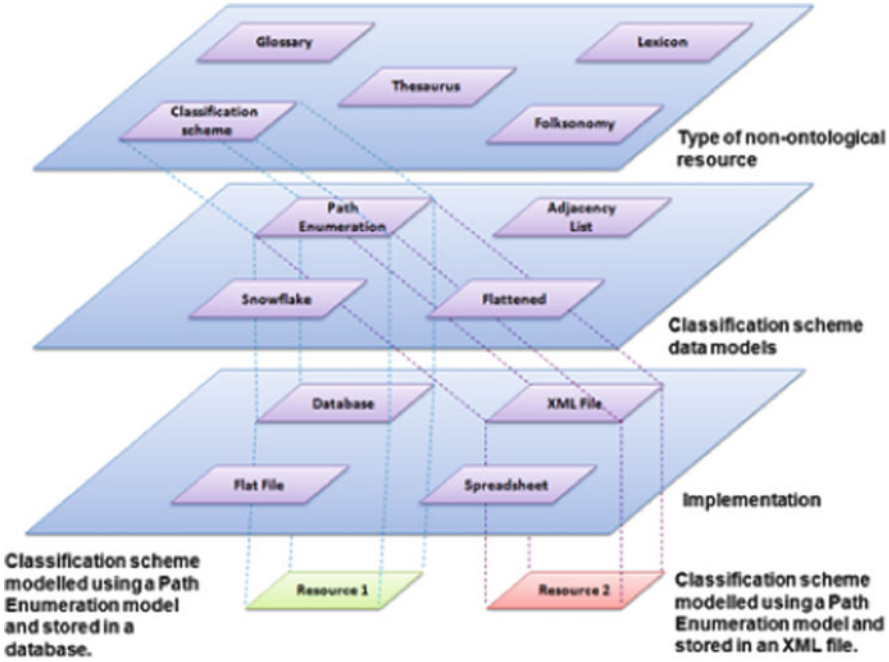


Fig. 6.1 Non-ontological resource categorization

Thesauri are mainly used for indexing and retrieving articles in large databases (ISO 2788). An example of thesaurus is the AGROVOC[5] thesaurus.

- *Folksonomies*: Folksonomies are Web 2.0 systems that users employ to upload and annotate their content effortlessly and without requiring any expert knowledge[6]. This simplicity has made folksonomies widely successful, and this success, in its turn, has resulted in a massive amount of user-generated and user-annotated web content. The main advantage of folksonomies is the implicit knowledge they contain. When users tag resources with one or more tags, they assign these resources the meaning of the tag. Furthermore, the co-occurrence of tags implies a semantic correlation among them. An example of how folksonomies are used can be seen in the *del.icio.us*[7] web site.

The knowledge encoded by the resource can be represented in different ways, known as data models. A data model (Carkenord 2002) is an abstract model that describes how data is represented and accessed. There are three types: (1) the conceptual data model, which presents the primary entities and relationships of

[5] http://www.fao.org/agrovoc/
[6] http://www.vanderwal.net/folksonomy.html
[7] http://del.icio.us/

concern to a specific domain; (2) the logical data model, which depicts the logical entity types, the data attributes describing those entities, and the relationships between entities; and (3) the physical data model, which is related to a specific implementation of the resource. In this chapter, we will use the term data model when referring to the logical data model. With regard to the *data model*, there are different ways of representing the knowledge encoded by the resource. In this chapter, we only focus in data models for classification schemes, thesauri, and lexica. The data models are described in detail in Villazón-Terrazas et al. (2010).

Next we present several *data models for classification schemes*, shown in Fig. 6.2.

- *Path enumeration* (Brandon 2005): A path enumeration model (see Fig. 6.2b) is a recursive structure for hierarchy representations and is defined as a model that stores, for each node, the path (as a string) from the root to the node. This string is the concatenation of the node code in the path from the root to the node.
- *Adjacency list* (Brandon 2005): An adjacency list model is a recursive structure for hierarchy representations comprising a list of nodes with a linking column to their parent nodes. Figure 6.2c shows this model.
- *Snowflake* (Malinowski and Zimányi 2006): A snowflake model is a normalized structure for hierarchy representations. For each hierarchy level, a table is created. In this model, each hierarchy node has a column linked to its parent node. Figure 6.2d shows this model.
- *Flattened* (Malinowski and Zimányi 2006): A flattened model is a denormalized structure for hierarchy representations. The hierarchy is represented by a table where each hierarchy level is stored in a different column. Figure 6.2e shows this model.

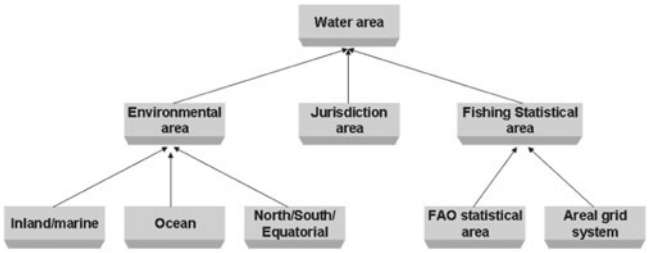
Next, we present two *data models for thesauri*.

- *Record-based model* (Soergel 1995): A record-based model is a denormalized structure that for every term uses a record with information about the term, such as synonyms, broader, narrower, and related terms. This model looks like the flattened model for classification scheme.
- *Relation-based model* (Soergel 1995): A relation-based model leads to a more elegant and efficient structure. Information is stored in individual pieces that can be arranged in different ways. Relationship types are not defined as fields in a record, they are simply data values in a relationship record, thus new relationship types can be introduced with ease. There are three entities: (1) a term entity, which contains the overall set of terms; (2) a term-term relationship entity, in which each record contains two different term codes and the relationship between them; and (3) a relationship source entity, which contains the overall resource relationships.

Next we present a *data model for lexica*.

- *Record-based model* (Soergel 1995): This model can also be used for lexicons because the use of a record for every lexical resource and information about that lexical resource is possible.

a



b

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

c

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

d

First Level	
ID	CSI_Name
20000	Water area

Second Level		
ID	First Level ID	CSI_Name
21000	20000	Environmental area
24020	20000	Jurisdiction area
22000	20000	Fishing Statistical area

Third Level		
ID	Second Level ID	CSI_Name
21001	21000	Inland/marine
21002	21000	Ocean
21003	21000	North/South/Equatorial
22001	22000	FAO statistical area
22002	22000	Areal grid system

e

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

f

```
<?xml version="1.0" encoding="UTF-8"?>
<WaterAreaClassificationScheme>
  <CSI>
    <ID>20000</ID>
    <Name>Water area</Name>
    <Parent></Parent>
  </CSI>
  <CSI>
    <ID>21000</ID>
    <Name>Environmental area</Name>
    <Parent>20000</Parent>
  </CSI>
```

g

	A	B
1	ID	CSI Name
2	20000	Water area
3	20000.21000	Environmental area
4	20000.24020	Jurisdiction area
5	20000.22000	Fishing Statistical area
6	20000.21000.21001	Inland/marine
7	20000.21000.21002	Ocean
8	20000.21000.21003	North/South/Equatorial
9	20000.22000.22001	FAO statistical area
10	20000.22000.22002	Areal grid system

Fig. 6.2 Example of classification scheme. (**a**) Excerpt of the Water Area classification scheme, (**b**) Path Enumeration data model, (**c**) Adjacency List data model, (**d**) Snowflake data model, (**e**) Flattened data model, (**f**) XML implementation for the Adjacency List data model, (**g**) Spreadsheet implementation for the Path Enumeration data model

- *Relation-based model* (Soergel 1995): It can also be used for lexicons because the storage of information about the lexicon in individual pieces is possible.

According to the *implementation*, we classify NORs into:

- *Databases*: A database is a structured collection of records or data stored in a computer system.
- *Spreadsheets*: An electronic spreadsheet consists of a matrix of cells where a user can enter formulas and values.
- *XML file*: EXtensible Markup Language is a simple, open, and flexible format used to exchange a wide variety of data on and off the web. XML is a tree structure of nodes and nested nodes of information where the user defines the names of the nodes.
- *Flat file*: A flat file is a file usually read or written sequentially. In general, a flat file is a file containing records with no structured interrelationships.

In summary, Fig. 6.1 shows how a given type of NOR can be modeled following one or more data models, each of which implemented in different ways at the implementation layer. Figure 6.1 shows, as an example, a classification scheme modeled following a path enumeration model. In this case, the classification scheme is implemented in a database and in an XML file.

To exemplify the non-ontological categorization presented with a real life classification scheme, we use an excerpt from the FAO water area classification presented in (Fig. 6.2a). This classification schema is modeled following a path enumeration model (Fig. 6.2b), an adjacency list model (Fig 6.2c), a snowflake model (Fig. 6.2d), and a flattened model (Fig. 6.2e). Figure 6.2f presents an XML implementation of the adjacency list model, and Fig. 6.2g presents a spreadsheet implementation of the path enumeration model of the same classification scheme.

It is worth mentioning that this first categorization of NORs is neither exhaustive nor complete. Currently, we are enriching it by adding examples taken from RosettaNet[8] and Electronic Data Interchange, EDI[9].

Moreover, we can map available non-ontological resources to our categorization. Next we present a brief list of them.

- The United Nations Standard Products and Services Code, UNSPSC[10], is a classification scheme, modeled with the path enumeration data model and stored in a relational database.
- WordNet[11], a lexical database for English, is a lexicon, modeled with the relation-based data model and stored in several implementations; a particular implementation of it is a relational database.

[8] http://www.rosettanet.org/

[9] http://www.edibasics.co.uk/

[10] http://www.unspsc.org/

[11] http://wordnet.princeton.edu/

- UMLS12[12] is a very large, multipurpose, multilingual thesaurus that contains information about biomedical and health-related concepts. It is modeled with the record-based model and stored in a flat file.
- MeSh[13], the Medical Subject Headings, is a classification scheme, modeled with the path enumeration data model.
- The Art and Architecture Thesaurus[14] is modeled with the record-based data model and implemented in XML.
- The ISCO-08 International Standard Classification of Occupations[15] is a classification scheme modeled with the path enumeration data model and implemented in a database and spreadsheet.
- The European Training Thesaurus, ETT[16], is modeled with the record-based data model and implemented in XML.
- The Classification of Fields of Education and Training, FOET[17], is a classification scheme modeled with path enumeration data model and implemented in XML and spreadsheet.
- The Aquatic Sciences and Fisheries Abstracts thesaurus, ASFA[18], is modeled with the record-based data model and implemented in XML.
- The AGROVOC thesaurus[19] is modeled with the relation-based data model and implemented in a database.
- The Fisheries Global Information System, FIGIS[20], is modeled with the adjacency list data model and implemented in a database.
- The Classification of Italian Education Titles published by the National Institute of Statistics, ISTAT[21], is a classification scheme modeled with the flattened data model and implemented in a spreadsheet.

[12] http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html
[13] http://www.nlm.nih.gov/mesh/
[14] http://www.getty.edu/research/tools/vocabularies/aat/index.html
[15] http://www.ilo.org/public/english/bureau/stat/isco/index.htm
[16] http://libserver.cedefop.europa.eu/ett/en/
[17] http://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=DSP_GEN_DESC_VIEW_NOHDR&StrNom=EDU_TRAINI&StrLanguageCode=EN
[18] http://www.fao.org/fishery/asfa/8/en
[19] http://aims.fao.org/website/AGROVOC-Thesaurus/sub
[20] http://www.fao.org/figis/servlet/RefServlet
[21] http://en.istat.it/

6.3 Methodological Guidelines for Reusing Non-ontological Resources

Once we have defined and categorized the non-ontological resources to be dealt with, we present the methodological guidelines for reusing them. The goal of the non-ontological resource reuse process is to choose the most suitable non-ontological resource for building ontologies. Domain experts, software developers, and ontology practitioners carry out this process by taking as input the ontology requirements specification document (ORSD)[22] to find the most suitable non-ontological resources for the development of ontologies. The output of the process is a set of non-ontological resources that, to some extent, covers the expected domain. Figure 6.3 shows the filling card used in the process of reusing non-ontological resources, which includes the definition, goal, input, output, performer of the process, and period of execution.

This process includes the activities and tasks presented in Fig. 6.4 and is explained next.

6.3.1 Activity 1. Search Non-ontological Resources

The goal of the activity is to search non-ontological resources from highly reliable web sites, domain-related sites, and resources within organizations. Domain experts, software developers, and ontology practitioners carry out this activity, taking as input the ORSD. They use the terms that have the highest frequency in the ORSD to search for the candidate non-ontological resources that cover the desired terminology. The activity output is a set of candidate non-ontological resources that may belong to any of the identified typologies described in Sect. 6.2.

6.3.2 Activity 2. Assess the Set of Candidate Non-ontological Resources

The goal of the activity is to assess the set of candidate non-ontological resources. Domain experts, software developers, and ontology practitioners carry out this activity, taking as input the set of candidate non-ontological resources. We propose to consider the following measurable criteria: (1) coverage, (2) precision plus two subjective criteria, (3) quality[23], and (4) consensus. These criteria are inspired on the work proposed in Gangemi et al. (2006).

[22] This document is the outcome of the ontology specification activity (Suárez-Figueroa et al. 2009) (see Chapter 5).

[23] A deep analysis of the quality of the resource is out of the scope of this chapter.

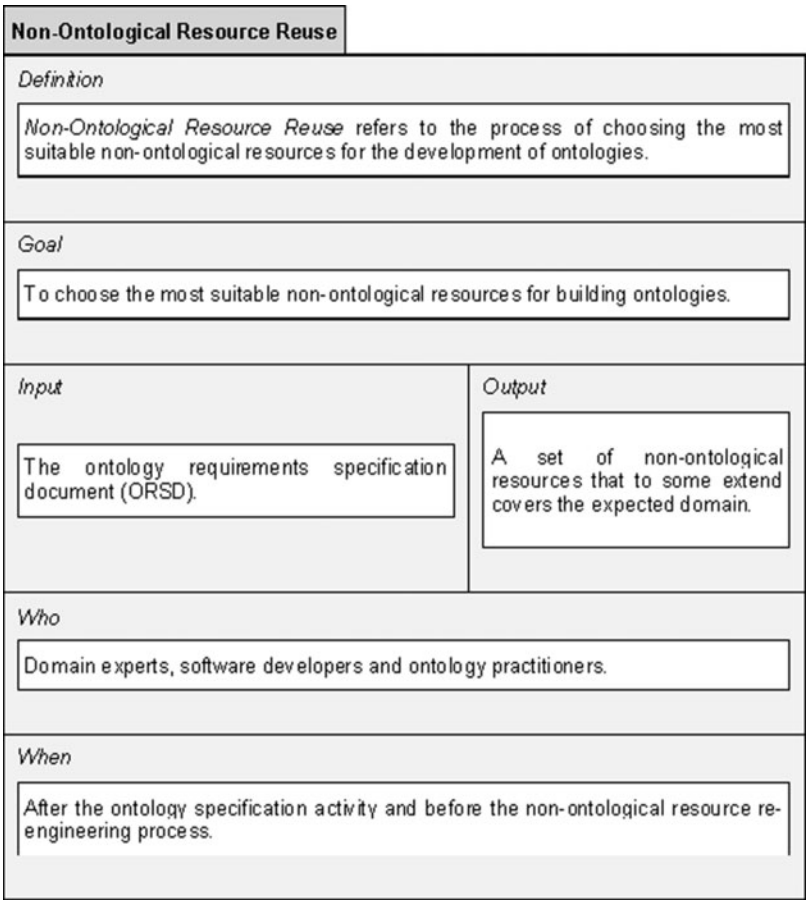


Fig. 6.3 Non-ontological resource reuse filling card

6.3.2.1 Task 2.1 Extract Lexical Entries

The goal of this task is to extract the lexical entries of the non-ontological resources. The task is carried out by software developers and ontology practitioners by taking as input the non-ontological resources and extracting their lexical entries with terminology extraction tools.

6.3.2.2 Task 2.2 Calculate Precision

The goal of this task is to calculate the precision of the candidate non-ontological resources. Precision is a measure widely used in information retrieval (Baeza-Yates and Ribeiro-Neto 1999) and is defined as the proportion of retrieved material that is

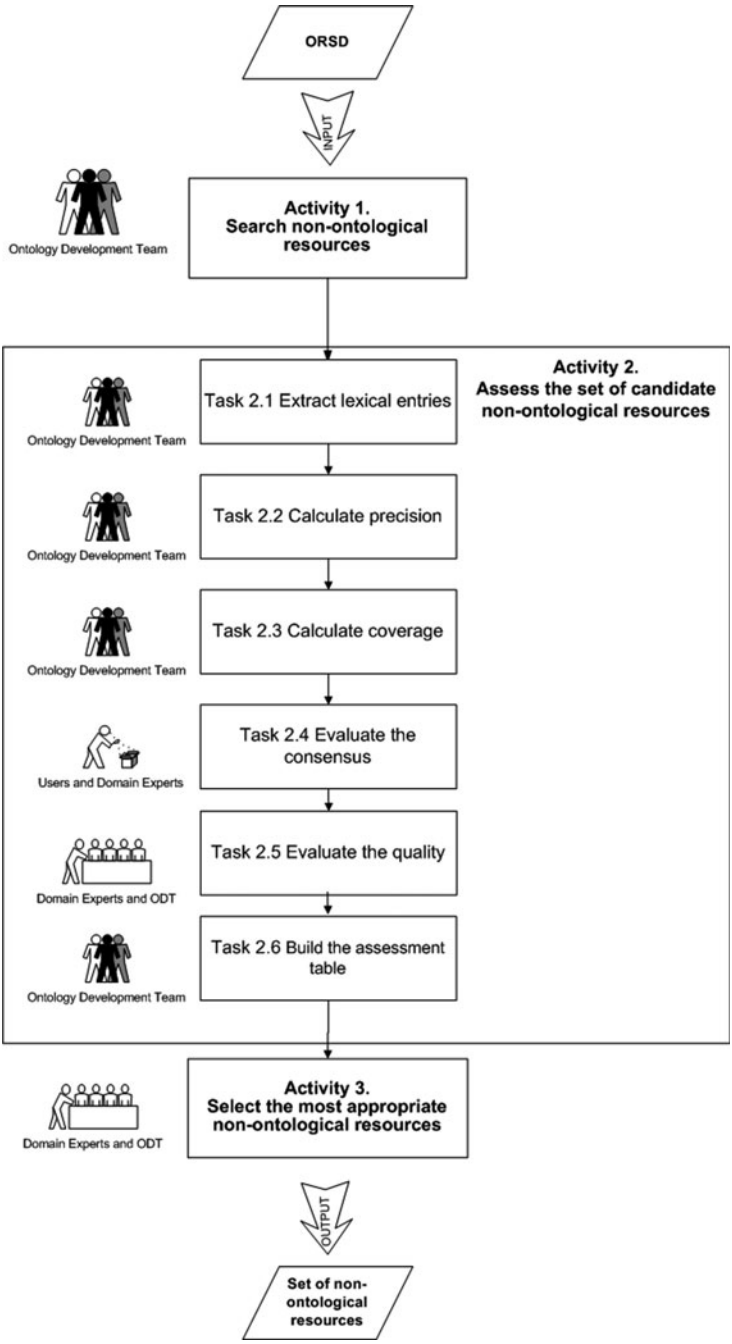


Fig. 6.4 Activities for the non-ontological resource reuse process

actually relevant. This task is carried out by software developers and ontology practitioners by taking as input the lexical entries extracted for the non-ontological resources and the terminology gathered in the ORSD. To adapt this precision measure into our context, we need to define:

- *NORLexicalEntries* as the set of lexical entries extracted from the non-ontological resource
- *ORSDTerminology* as the set of identified terms included in the ORSD

Now we can define the precision, in our context, as the proportion of the lexical entries of the non-ontological resource that are included in the identified terms of the ORSD over the lexical entries of the non-ontological resource. This is expressed as follows:

$$Precision = \frac{|\{NORLexicalEntries\} \cap \{ORSDTerminology\}|}{|\{NORLexicalEntries\}|}$$

6.3.2.3 Task 2.3 Calculate Coverage

The goal of this task is to calculate the coverage of the non-ontological resources. Coverage is based on the recall measure used in information retrieval (Baeza-Yates and Ribeiro-Neto 1999). Recall is defined as the proportion of relevant material actually retrieved in answer to a search request. This task is carried out by software developers and ontology practitioners by taking as input both the lexical entries extracted from the non-ontological resources and the terminology gathered in the ORSD. To adapt this measure into our context, we use the aforementioned definitions of *NORLexicalEntries* and *ORSDTerminology*. In this context, coverage is the proportion of the identified terms of the ORSD that are included in the lexical entries of the non-ontological resource over the identified terms of the ORSD. This is expressed as follows:

$$Coverage = \frac{|\{NORLexicalEntries\} \cap \{ORSDTerminology\}|}{|\{ORSDTerminology\}|}$$

6.3.2.4 Task 2.4 Evaluate the Consensus

The goal of this task is to evaluate the consensus of the non-ontological resources. Consensus is a subjective and not quantifiable criterion. This task is carried out by domain experts, taking as input the non-ontological resources for stating whether the non-ontological resources contain terminology agreed upon by the community

or not. We propose a preliminary starting point for this evaluation. Domain experts have to check whether the resource is coming from:

- A standardization body or any entity whose primary activity is to develop, coordinate, promulgate, revise, amend, reissue, or otherwise maintain standards; for example, the International Organization for Standardization (ISO), the American National Standards Institute (ANSI), and the World Wide Web Consortium (W3C)
- Large organizations across national governments, such as the Food and Agriculture Organization of the United Nations (FAO), the World Health Organization (WHO), the United Nations Educational, Scientific and Cultural Organization (UNESCO), and the International Olympic Committee (IOC)
- A large enough user community to make it profitable for developers to use it as a means of general interoperability

Either the resource is coming from any of the aforementioned parties or not, domain experts may state that the resource has reached some degree of consensus.

6.3.2.5 Task 2.5 Evaluate the Quality

The goal of this task is to evaluate the quality of the resource. We do not intend to provide a deep analysis of the quality of the resource but to offer some preliminary considerations about it. In this chapter, we propose to check the following quality attributes:

- Good documentation of the resource.
- Lack of anomalies of the non-ontological resource, such as redundancies or inconsistencies.
- Reliability of the non-ontological resource. This means analyzing whether we can trust the resource or not.

6.3.2.6 Task 2.6 Build the Assessment Table

The goal of this task is to create an assessment table of the non-ontological resources. Software developers and ontology practitioners carry out this task, taking as input the non-ontological resources with their respective values for precision, coverage, consensus, and quality criteria, for the construction of the assessment table. This table is shown in Table 6.1. The first column shows the non-ontological resources found. The

Table 6.1 Assessment table for the NORs

NOR	Precision	Coverage	Consensus	Quality
NOR 1	NOR 1 precision value	NOR 1 coverage value	(Yes/no)	(Yes/no)
NOR 2	NOR 2 precision value	NOR 2 coverage value	(Yes/no)	(Yes/no)
NOR 3	NOR 3 precision value	NOR 3 coverage value	(Yes/no)	(Yes/no)

precision column shows the precision value calculated for each non-ontological resource. Then, the coverage column shows the coverage value calculated for each non-ontological resource. Next, the consensus column depicts the domain experts' judgment about whether the non-ontological resource has been agreed on by the community or not (Yes/No). Finally, the quality column illustrates the domain experts, software developers, and ontology practitioners' judgment about whether the resource has an acceptable level of quality or not (Yes/No).

6.3.3 Activity 3. Select the Most Appropriate Non-ontological Resources

The goal of this activity is to select the most appropriate non-ontological resources to be transformed into an ontology. This activity is carried out by domain experts, software developers, and ontology practitioners, taking as input the non-ontological resource assessment table. The selection is performed manually and we recommend looking for resources with:

- Consensus. This criterion is taken into account in the first place because if the resource to be reused contains terminology agreed upon by the community, the effort and time spent in finding out the right labels for the ontology terms will decrease considerably.
- Quality. This criterion is taken into account in the second place because if the resource to be reused has an acceptable level of quality, then the resultant ontology should also have it.
- High value of coverage. This criterion is taken into account in the third place because our third concern is to consider most of the ORSD terms identified.
- High value of precision. This criterion is taken into account in the fourth place because our fourth concern is the proportion of non-ontological lexical entries over the identified terms of the ORSD.

The activity output is a ranked list of non-ontological resources that, to some extent, covers the expected domain. These resources will be ready for the re-engineering process.

6.4 Methodological Guidelines for Re-engineering NORs into Ontologies

In this section, we depict the prescriptive methodological guidelines for re-engineering NORs. The goal of the method for re-engineering non-ontological resources is to transform a non-ontological resource into an ontology. The output of the process is an ontology. Figure 6.5 shows the filling card of the non-ontological

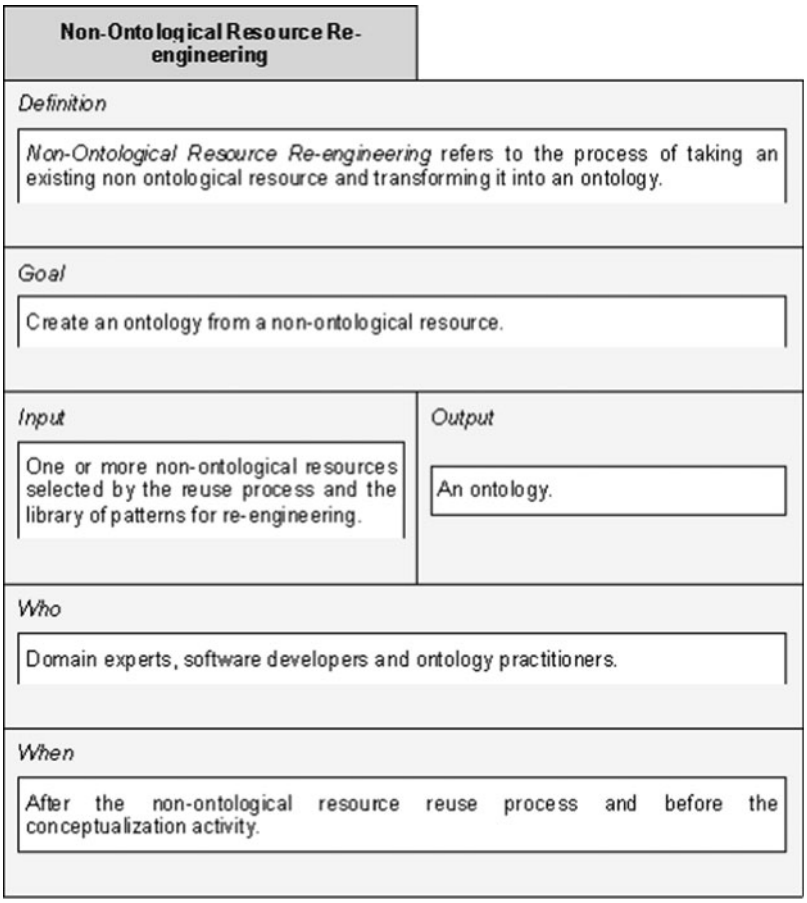


Fig. 6.5 Non-ontological resource re-engineering filling card

resource re-engineering process, which includes the definition, goal, input, output, performer of the process, and time execution.

The NOR re-engineering process consists of the three activities depicted in Fig. 6.6.

6.4.1 Activity 1. Non-ontological Resource Reverse Engineering

The goal of this activity is to analyze a non-ontological resource, to identify its underlying terms, and to create representations of the resource at the different levels of abstraction (design, requirements, and conceptual).

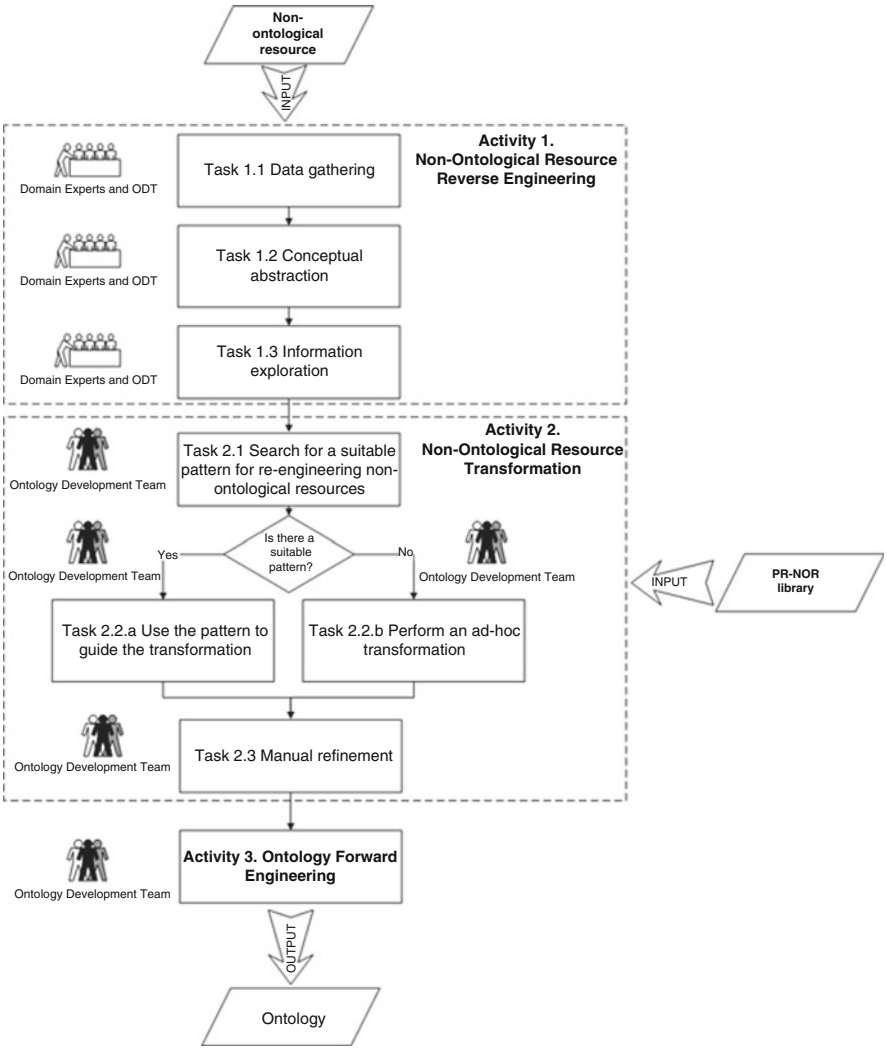


Fig. 6.6 Re-engineering process for non-ontological resources

6.4.1.1 Task 1.1 Data Gathering

The goal of this task is to search and compile all the available data and documentation about the non-ontological resource, including purpose, components, data model, and implementation details.

6.4.1.2 Task 1.2 Conceptual Abstraction

The goal of this task is to identify the schema of the non-ontological resource including the conceptual components and their relationships. If the conceptual schema is not available in the documentation, the schema should be reconstructed manually or with a data modeling tool.

6.4.1.3 Task 1.3 Information Exploration

The goal of this task is to find out how the conceptual schema of the non-ontological resource and its content are represented in the data model. If the non-ontological resource data model is not available in the documentation, the data model should be reconstructed manually or with a data modeling tool.

6.4.2 Activity 2. Non-ontological Resource Transformation

This activity has as a goal to generate a conceptual model from the non-ontological resource. We propose the use of patterns for re-engineering non-ontological resources (PR-NOR) to guide the transformation process.

6.4.2.1 Task 2.1 Search for a Suitable Pattern for Re-engineering Non-ontological Resource

The goal of this task is to find out if there is any applicable re-engineering pattern that transforms the non-ontological resource into a conceptual model. The search is performed in the ODP Portal24[24], which includes the PR-NOR library, and with the following criteria: (1) non-ontological resource type, (2) internal data model of the resource, and (3) the transformation approach.

6.4.2.2 Task 2.2.a Use Re-engineering Patterns to Guide the Transformation

The goal of this task is to apply the re-engineering pattern obtained in Task 2.1 (see Sect. 6.4.2.1) to transform the non-ontological resource into a conceptual model. If a suitable pattern for re-engineering non-ontological resources is found, then the conceptual model is created from the non-ontological resource following the procedure established in the pattern for re-engineering. Alternatively, the software

[24] http://ontologydesignpatterns.org

library, described later in Sect. 6.5, can be used for generating the ontology automatically.

6.4.2.3 Task 2.2.b Perform an Ad Hoc Transformation

The goal of this task is to set up an ad hoc procedure that transforms the non-ontological resource into a conceptual model when a suitable pattern for re-engineering cannot be found. This ad hoc procedure may be generalized to create a new pattern for re-engineering non-ontological resources.

6.4.2.4 Task 2.3 Manual Refinement

The goal of this task is to check whether any inconsistency appears after the transformation. Software developers and ontology practitioners, with the help of domain experts, can fix manually any inconsistencies generated from the transformation.

6.4.3 Activity 3. Ontology Forward Engineering

The goal of this activity is to generate the ontology. We use the ontology levels of abstraction to depict this activity because they are directly related to the ontology development process. The conceptual model obtained in Task 2.2.a (Sect. 6.4.2.2) or 2.2.b (Sect. 6.4.2.3) is transformed into a formalized model, according to a knowledge representation paradigm such as description logics, first order logic, or F-logic. Then, the formalized model is implemented in an ontology language.

6.5 Technological Support

Our technological support consists in (1) a PR-NOR pattern library that includes the set of patterns for re-engineering non-ontological resources and the implementation of (2) NOR_2O, a software library that implements the transformation process suggested by the patterns.

Table 6.2 Template of pattern for re-engineering non-ontological resource

Slot	Value
General information	
Name	Name of the pattern
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	
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 (Brockmans and Haase 2006), of the ontology created for the non-ontological resource being re-engineered
(UML) Example solution ontology	A graphical representation example, which uses the UML profile (Brockmans and Haase 2006), of the ontology created for the non-ontological resource being used
Process: how to re-engineer	
General	Algorithm for the re-engineering process
Example	Application of the algorithm to the non-ontological resource example
Time complexity	The time complexity of the algorithm
Additional notes	Additional notes of the algorithm
Formal transformation	
General	Formal description of the transformation made with the formal definitions of the resources
Relationships (optional)	
Relations to other modeling components	Description of any relation to other PR-NOR patterns or other ontology design patterns

Table 6.3 Set of patterns for re-engineering NORs

N	Identifier	Type of NOR	NOR data model	Target
1	PR-NOR-CLTX-01	Classification scheme	Path enumeration	Ontology schema (TBox)
2	PR-NOR-CLTX-02	Classification Scheme	Adjacency list	Ontology schema (TBox)
3	PR-NOR-CLTX-03	Classification scheme	Snowflake	Ontology schema (TBox)
4	PR-NOR-CLTX-04	Classification scheme	Flattened	Ontology schema (TBox)
5	PR-NOR-CLAX-10	Classification scheme	Path enumeration	Ontology (TBox + ABox)
6	PR-NOR-CLAX-11	Classification scheme	Adjacency list	Ontology (TBox + ABox)
7	PR-NOR-CLAX-12	Classification scheme	Snowflake	Ontology (TBox + ABox)
8	PR-NOR-CLAX-13	Classification scheme	Flattened	Ontology (TBox + ABox)
9	PR-NOR-TSTX-01	Thesaurus	Record based	Ontology Schema (TBox)
10	PR-NOR-TSTX-02	Thesaurus	Relation based	Ontology Schema (TBox)
11	PR-NOR-TSAX-10	Thesaurus	Record based	Ontology (TBox + ABox)
12	PR-NOR-TSAX-11	Thesaurus	Relation based	Ontology (TBox + ABox)
13	PR-NOR-LXTX-01	Lexicon	Record based	Ontology schema (TBox)
14	PR-NOR-LXTX-02	Lexicon	Relation based	Ontology schema (TBox)
15	PR-NOR-LXAX-10	Lexicon	Record based	Ontology (TBox + ABox)
16	PR-NOR-LXAX-11	Lexicon	Relation based	Ontology (TBox + ABox)

6.5.1 Patterns for Re-engineering Non-ontological Resources

In this section, we introduce the 16 patterns that perform the transformations of NORs into ontologies. Patterns for re-engineering NORs (PR-NOR) define a procedure that transforms the NOR terms into ontology representational primitives.

Next, we present the template proposed that describes the patterns for re-engineering non-ontological resources (PR-NOR). We have modified the tabular template used in Villazón-Terrazas et al. (2008) for describing the PR-NORs. The meaning of each field is shown in Table 6.2.

According to the NOR categorization presented in Sect. 6.2, we propose patterns for re-engineering classification schemes, thesauri, and lexicons (see Table 6.3). For every data model, we can define a process with a well-defined sequence of activities in order to extract the NOR terms and then to map these terms to a conceptual model of an ontology. This process is expressed as an algorithm. Moreover, it is worth mentioning that we refer to *ontology schema* as TBox, and just *ontology* as TBox and ABox. These patterns are included in the ODP Portal[25].

The re-engineering patterns take advantage of the use of the ontology design patterns[26] for creating the ontology code. So, most of the code generated follows the best practices already identified by the community (see section *Process* on Table 6.2).

[25] http://ontologydesignpatterns.org

[26] Ontology design patterns are included in the ODP portal. The ODP portal is a Semantic Web portal dedicated to ontology design best practices for the Semantic Web, emphasizing particularly ontology design patterns (OPs)

Although we have identified five types of NORs, here we just list patterns for re-engineering classification schemes, thesauri, and lexica (see Table 6.3).

6.5.1.1 Semantics of the Relations Among the NOR Terms

The TBox transformation approach converts the resource content into an ontology schema. TBox transformation tries to impose a formal semantics on the resource by making explicit the semantics hidden in the relations of the NOR terms. To this end, each NOR term is mapped to a class, and then, the semantics of the relations among those entities must be discovered and then made explicit. Thus, patterns that follow the TBox transformation approach must discover first the semantics of the relations among the NOR terms. To perform this task, we rely on WordNet, which organizes the lexical information into meanings (senses) and synsets. What makes WordNet remarkable is the existence of various relations explicitly declared between the word forms (e.g., lexical relations, such as synonymy and antonymy) and the synsets (meaning to meaning or semantic relations, e.g., hyponymy/hypernymy relation, meronymy relation). Here, we want to prove that we can rely on an external resource for making explicit the relations. For this purpose, first, we rely on WordNet, and then, as a future line of this work, we may rely on other information resources, such as DBpedia[27].

Algorithm 1 describes how to make explicit the semantics of the relations in the NOR terms. The abbreviation of the algorithm name is *getRelation*.

6.5.2 *NOR$_2$O*

This section presents NOR$_2$O, a Java library that implements the transformation process suggested by the patterns for re-engineering non-ontological resources (PR-NOR). The library performs the ETL process[28] for transforming the non-ontological resource components into ontology terms. A high-level conceptual architecture diagram of the modules involved is shown in Fig. 6.7.

Algorithm 1 Discovering the semantics of the relations – *getRelation*

```
1: Take two related terms from the NOR, ti and tj
2: defaultRelation ← userDefinedRelation
3: if contains(ti,tj) then
4:     relation ← ti.subClassOf.tj
```

(continued)

[27] http://www.dbpedia.org/

[28] Extract, transform, and load (ETL) of legacy data sources is a process that involves (1) extracting data from the outside resources, (2) transforming data to fit operational needs, and (3) loading data into the end target resources (Kimball and Caserta 2004).

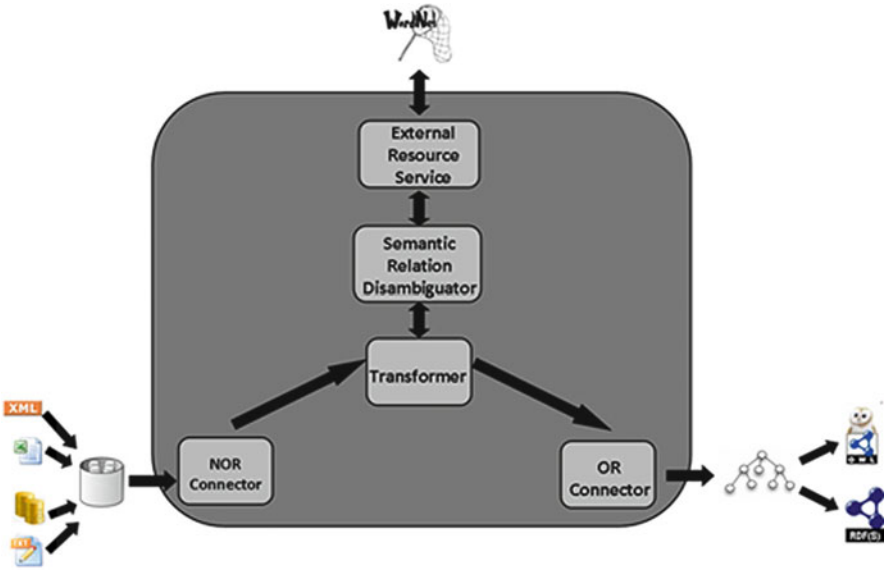


Fig. 6.7 Modules of the NOR_2O software library

Algorithm 1 Discovering the semantics of the relations – *getRelation*

```
5: else if contains(tj,ti) then
6:     relation ← tj.subClassOf.ti
7: else
8:     wordnetRelation ← WordNet(ti, tj)
9:     if wordnetRelation = = hyponym then
10:        relation ← ti.subClassOf.tj
11:    else if wordnetRelation = = hypernym then
12:        relation ← tj.subClassOf.ti
13:    else if wordnetRelation = = meronym then
14:        relation ← ti.partOf.tj
15:    else if wordnetRelation = = holonym then
16:        relation ← tj.partOf.ti
17:    else
18:        relation ← defaultRelation
19:    end if
20: end if
21: return relation
```

Figure 6.7 depicts the modules of the PR-NOR software library: NOR `Connector`, `Transformer`, `Semantic Relation Disambiguator`, `External Resource Service`, and `OR Connector`. In the following sections, these modules are described in detail. For illustrating the modules, the example of the transformation of the ASFA thesaurus[29] into an ontology schema[30] is provided.

6.5.2.1 NOR Connector

The NOR Connector loads classification schemes, thesauri, and lexicons modeled with their corresponding data models, and implemented in databases, XML, flat files, and spreadsheets.

This module utilizes an XML configuration file for describing the NOR. An example of the XML configuration file is presented in Listing 6.1. The Listing shows how the file describes a thesaurus. The thesaurus has two schema entities, *Term* and *NonPreferredTerm*, is modeled following the record-based data model, and is implemented in XML.

Listing 6.1 NOR Connector configuration file example

```
<Nor type="Classification Scheme" name="cepa94">
 <Schema>
  <SchemaEntities>
        <SchemaEntity name="CSItem">
         <Attribute name="CSIdentifier"
                                          valueFrom="cepa.CodeNumber"
                                          type="string"/>
               <Attribute name="CSName"
                                          valueFrom="cepa.DescriptionEnglish"
                                          type="string"/>
               <Relation name="subType"
                                          using="PathEnumeration"
                                          destination="CSItem"/>
               <Relation name="superType"
                                          using="PathEnumeration"
                                          destination="CSItem"/>
        </SchemaEntity>
       </SchemaEntities>
 </Schema>
 <DataModel>
  <PathEnumeration>
        <PathEntity>cepa</PathEntity>
        <PathSeparator>.</PathSeparator>
         <PathField>CodeNumber</PathField>
        </PathEnumeration>
 </DataModel>
 <Implementation>
  <Database>
        <Dbms>MSACCESS</Dbms>
        <Name>cepa94</Name>
        <Username></Username>
        <Password></Password>
        <Host></Host>
        <Port></Port>
       </Database>
 </Implementation>
</Nor>
```

[29] http://www4.fao.org/asfa/asfa.htm

[30] http://mccarthy.dia.fi.upm.es/ontologies/asfa.owl

6.5.2.2 Transformer

This module performs the transformation suggested by the patterns by implementing the sequence of activities included in the patterns. The module transforms the NOR elements, loaded by the `NOR Connector` module, into internal model representation elements. It also interacts with the `Semantic Relation Disambiguator` module for obtaining the suggested semantic relations of the NOR elements.

The `Transformer` also utilizes an XML configuration file, called prnor.xml, for describing the transformation between the NOR elements and the ontology elements. This XML configuration file has only one section, *PRNOR*, which includes the description of the transformation from the NOR schema components (e.g., schema entities, attributes, and relations) into the ontology elements (e.g., classes, object properties, data properties, and individuals). Additionally, it indicates the transformation approach (e.g., TBox, ABox, or Population).

Two examples of the XML configuration file are shown in Listings 6.2 and 6.3.

Listing 6.2 indicates that the pattern follows the TBox transformation approach and that it transforms the elements of the *CSItem* schema component into ontology classes. Also, by default, it transforms the *subType* schema relation into a *subClassOf* relation and the *superType* schema relation into a *superClassOf* relation, unless the `Semantic Relation Disambiguator` module suggests another relation.

Listing 6.2 PR-NOR Connector configuration file example – Classification Scheme

```
<Prnor identifier="PR-NOR-CLTX-01" transformationApproach="TBox"
topLevelClass="Protection_Activities" externalResource="WordNet">
        <Class from="CSItem" identifier="[CSName]...[CSIdentifier]">
                <ObjectProperty from="subType" to="subClassOf"/>
                <ObjectProperty from="superType" to="superClassOf"/>
        </Class>
</Prnor>
```

Listing 6.3 indicates that the pattern follows the TBox transformation approach and that it transforms the elements of the *Term* schema component into ontology classes. Also, by default, it transforms the *NT* schema relation into a *superClassOf* relation, the *RT* schema relation into a *relatedTerm* relation, and the *BT* schema relation into a *subClassOf* relation, unless the `Semantic Relation Disambiguator` module suggests another relation. Finally, the *UF* schema relation is transformed into a *rdfs:label*, and the module uses WordNet as external resource for disambiguation.

Listing 6.3 PR-NOR Connector configuration file example – Thesaurus

```
<Prnor identifier="PR-NOR-TSTX-01" transformationApproach="TBox"
externalResource="WordNet">
        <Class from="Term" identifier="[Identifier]">
                <ObjectProperty from="NT" to="superClassOf"/>
                <ObjectProperty from="RT" to="relatedTerm"/>
                <ObjectProperty from="BT" to="subClassOf"/>
                <ObjectProperty from="UF" to="rdfs:label"/>
        </Class>
</Prnor>
```

6.5.2.3 Semantic Relation Disambiguator

This module is in charge of obtaining the semantic relation between two NOR elements. Basically, the module receives two NOR elements from the `Transformer` module and returns the semantic relation between them. First, the module verifies whether it can obtain the *subClassOf* relation by identifying attribute adjectives[31] within the two given elements of the resource. If this is not the case, then the module connects the external resource through the `External Resource Service` module to get the relation.

The TBox transformation approach converts the resource content into an ontology schema. To this end, each NOR term is mapped to a class, and then the semantics of the relations among those entities is made explicit. Thus, patterns that follow the TBox transformation approach must make explicit the semantics of the relations among the NOR terms. To perform this task, we rely on WordNet, which organizes the lexical information into meanings (senses) and synsets.

Algorithm 1, presented in Sect. 6.5.1.1, describes how to make explicit the semantics of the relations in the NOR terms.

It is worth mentioning that, when asserting the *partOf* relation the algorithm takes advantage of the use of the `PartOf content pattern`[32] to guarantee that the OWL code generated follows common practices in ontological engineering.

6.5.2.4 External Resource Service

The `External Resource Service` is in charge of interacting with external resources for obtaining the semantic relations between two NOR elements. At this

[31] Attributive adjectives are part of the noun phrase headed by the noun they modify, for example, happy is an attributive adjective in "happy people." In English, the attributive adjective usually precedes the noun in simple phrases but often follows the noun when the adjective is modified or qualified by a phrase acting as an adverb.

[32] http://ontologydesignpatterns.org/wiki/Submissions:PartOf

moment, the module interacts with WordNet. We are now implementing the access to DBpedia[33] due to the reasons explained in Sect. 6.5.1.1.

6.5.2.5 OR Connector

The `Ontological Resource (OR) Connector` generates the ontology in OWL Lite. To this end, this module relies on the OWL API[34]. It also utilizes an XML configuration file for describing the ontology to be generated.

An example of the XML configuration file is shown in Listing 6.4. The listing indicates that the ontology generated will be stored in the *asfa.owl* file, that its name will be *asfa ontology*, and that it will be implemented in OWL.

Listing 6.4 OR Connector configuration file example

```
<Or name="asfa ontology"
ontologyURI="http://mccarthy.dia.fi.upm.es/ontologies/asfa.owl"
ontologyFile="asfa.owl" implementation="OWL"
alreadyExist="no" separator="#">
</Or>
```

Finally, to conclude the description of the software library, it is worth mentioning that the implementation of this library follows a modular approach; therefore, it is possible to extend it and include other types of NORs, data models, and implementations in a simple way, as well as to exploit other external resources for making explicit the hidden semantics in the relations of the NOR terms.

6.6 Example

In order to evaluate the methodological guidelines proposed in this chapter, we conducted two experiments in real case scenarios within the SEEMP[35] and mIO![36] projects.

6.6.1 SEEMP Project

The main objective of this project was to develop an interoperable architecture for public employment services (PES). The resultant architecture consisted of (1) a reference ontology, the core component of the system, that acts as a common "language" in the form of a set of controlled vocabularies that describes the details

[33] http://dbpedia.org/
[34] http://owlapi.sourceforge.net/
[35] http://www.seemp.org/
[36] http://www.cenitmio.es/

of a job posting; (2) a set of local ontologies, each PES uses its own local ontology, which describes the employment market in its own terms; (3) a set of mappings between each local ontology and the reference ontology; and (4) a set of mappings between the PES schema sources and the local ontologies.

In the following sections, we describe the application of our methodological guidelines for reusing and re-engineering non-ontological resources when building an occupation ontology.

6.6.1.1 Reusing Non-ontological Resources

This section presents the application of the method for reusing non-ontological resources within the SEEMP project. It shows the process we followed for selecting the non-ontological resources to be reused when building the occupation domain ontology.

Activity 1. Search Non-ontological Resources

Following the suggestions of some domain experts, we searched for the occupation classifications at (1) the Ramon Eurostat Portal[37], (2) the ONET web site[38], and (3) the companies the project partners. Thus, we found the following classifications:

- Standard Occupational Classification System (SOC)
- International Standard Classification of Occupations (ISCO-88)
- International Standard Classification of Occupations, for European Union purposes, ISCO-88 (COM)
- Occupational Information Network (ONET)
- EURES[39] proprietary occupation classification

Activity 2. Assess the Set of Candidate Non-ontological Resources

The goal of this activity was to assess the set of candidate non-ontological resources. Experts of the occupation domain, software developers, and ontology practitioners carried out this activity taking as input the set of candidate non-ontological resources.

Task 1. Extract Lexical Entries
Within this task, we extracted the lexical entries of the aforementioned occupation classifications. We developed an ad hoc extraction tool for performing automatically the extraction task.

[37] http://ec.europa.eu/eurostat/ramon/
[38] http://online.onetcenter.org/
[39] http://www.eurodyn.com/

Task 2. Calculate Precision
Since we were dealing with occupations related to the IT domain, it was impossible to cover all the IT domain occupations already identified in the ontology requirements specification document. Thus, we used a constant that represents the complete set of IT domain occupations. In this case, the cardinality of the complete set is K. Therefore, the intersection of the complete set with the set of terms available in the ORSD is the set of terms of the ORSD. Next, we present the precision for each occupation classification:

$$Precision = \frac{card\{\{NORLexicalEntries\} \cap \{ORSDTerminology\}\}}{card\{NORLexicalEntries\}}$$

- $SOCPrecision = \frac{6 \cap K}{26162} = \frac{6}{26162} = 0.0002$
- $ISCO-88Precision = \frac{9 \cap K}{544} = \frac{9}{544} = 0.0165$
- $ISCO-88COMPrecision = \frac{9 \cap K}{520} = \frac{9}{520} = 0.0173$
- $ONETPrecision = \frac{21 \cap K}{1167} = \frac{21}{1167} = 0.0179$
- $EURESPrecision = \frac{89 \cap K}{355} = \frac{89}{355} = 0.2507$

Task 3. Calculate Coverage
Again, since we were dealing with the occupations related to the IT domain, it was impossible to cover all the IT domain occupations in the ORSD. Thus, we used a constant K that represents the complete set of IT domain occupations. Next, we present the coverage for each occupation classification:

$$Coverage = \frac{card\{\{NORLexicalEntries\} \cap \{ORSDTerminology\}\}}{card\{ORSDTerminology\}}$$

- $SOCPrecision = \frac{6 \cap K}{K} = \frac{6}{K}$
- $ISCO-88Precision = \frac{9 \cap K}{K} = \frac{9}{K}$

Table 6.4 Assessment table for SEEMP occupation standards

NOR	Precision	Coverage	Consensus
SOC	0.0002	6/K	No
ISCO-88	0.0165	9/K	No
ISCO-88 COM	0.0173	9/K	Yes
ONET	0.0179	21/K	No
EURES	0.2507	89/K	Yes

- $ISCO-88COMPrecision = \frac{9 \cap K}{K} = \frac{9}{K}$
- $ONETPrecision = \frac{21 \cap K}{K} = \frac{21}{K}$
- $EURESPrecision = \frac{89 \cap K}{K} = \frac{89}{K}$

Task 4. Evaluate the Consensus
It was important for the project that resources focused on the current European reality because the user partners involved in SEEMP are European, and the outcoming prototype has to be validated in European scenarios. Thus, domain experts confirmed whether the resources were built with the consensus of the European community or not. They also explained that ISCO-88(COM) and EURES proprietary occupation classification contains terminology that had already reached a consensus.

Table 6.4 summarizes all the information of each non-ontological resource.

Activity 3. Select the Most Appropriate Non-ontological Resources

Following Table 6.1 we selected a non-ontological resource, the EURES proprietary occupation classification.

We followed the same process for selecting the non-ontological resources when building the remaining ontologies. We provide a table (see Table 6.5) that summarizes the selection of standards, codes, and classifications accomplished for building every domain ontology.

6.6.1.2 Re-engineering Non-ontological Resources

In this section, we present the application of the method for re-engineering non-ontological resources within the SEEMP project. Once we select the non-ontological resource, we have to transform it into an ontology. Next, we describe the process of generating an occupation ontology from the EURES proprietary occupation classification.

Table 6.5 Standards, codes, and classifications reused

Domain	Candidate standards/classifications	Selected standards/ classifications	Justification
Economic sector	ISIC, NACE, NAICS	NACE	Best coverage and European scope
Education fields	ISCED 97, FOET	FOET	Best coverage and European scope
Education levels	ISCED 97	ISCED 97	Worldwide scope, widely accepted
Currency	Pacific exchange, ISO 4217, WordAtlas	ISO 4217	Worldwide scope, widely accepted
Geographic	ISO 3166, Regions of the World	ISO 3166	Worldwide scope, widely accepted
Language	ISO 639	ISO 639	Worldwide scope, widely accepted
Language levels	CEFR	CEFR	European scope, widely accepted
Driving licence	EU driving licence	EU driving licence	European legislation
Skills	EURES	EURES	Coverage and European scope
Contract types	LE FOREM proprietary classification, ARL proprietary classification	Mix of both classifications	Acceptable coverage in SEEMP scope
Work condition	LE FOREM proprietary classification	LE FOREM proprietary classification	Acceptable coverage in SEEMP scope

Activity 1. Non-ontological Resource Reverse Engineering

In this activity, we gathered documentation on the EURES occupation classification from the European Dynamics SEEMP user partner. From this documentation, we extracted the schema of the classification scheme, which consists of two tables, *CVO OCCGROUP* and *CVO OCCUGROUP NAME*. Since the data model was not available in the documentation, it was necessary to extract it for the resource implementation itself. The EURES occupation classification is modeled following the snowflake data model and is implemented in a MS Access database.

Activity 2. Non-ontological Resource Transformation

Within this activity, we carried out the following tasks:

1. We identified the transformation approach, the TBox transformation, i.e., transforming the resource content into an ontology schema.
2. Then, we searched our local pattern repository for a suitable pattern to re-engineer NORs, taking into account the transformation approach (TBox

Table 6.6 Resources transformed in the SEEMP project

Resource	Type	Data model	Implementation	Pattern used
NACE	Classification scheme	Path enumeration	Database	PR-NOR-CLTX-01
FOET	Classification scheme	Path enumeration	Database	PR-NOR-CLTX-01
ISCED 97	Classification scheme	Adjacency list	Database	PR-NOR-CLTX-02
ISO 4217	Classification scheme	Snowflake	XML	PR-NOR-CLAX-12
ISO 3166	Classification scheme	Snowflake	XML	PR-NOR-CLAX-12
ISO 639	Classification scheme	Snowflake	XML	PR-NOR-CLAX-12
CEFR	Classification scheme	Proprietary model	Proprietary format	
EU driving licence	Classification scheme	Snowflake	Proprietary format	
EURES skill	Classification scheme	Path enumeration	Database	PR-NOR-CLTX-01
LE FOREM contracts	Proprietary classification	Proprietary model	Proprietary format	

transformation), the non-ontological resource type (classification scheme), and the data model (snowflake data model) of the resource.

3. The most appropriate pattern found for this case was the PR-NOR-CLTX-03 pattern. This pattern takes as input a classification scheme modeled with a snowflake data model and produces an ontology schema.

Activity 3. Ontology Forward Engineering

WSML[40] is the ontology implementation language used in the SEEMP project. Because of the number of occupations of the EURES classification, it was not practical to create the ontology manually. Therefore, we created an ad hoc wrapper, implemented in Java, that reads the data from the resource implementation and automatically creates the corresponding classes and relations of the new ontology following the suggestions given by the pattern for re-engineering NORs and the conceptual model.

We followed this process for all the resources identified, being the patterns used those presented in Table 6.6.

[40] http://www.wsmo.org/wsml/

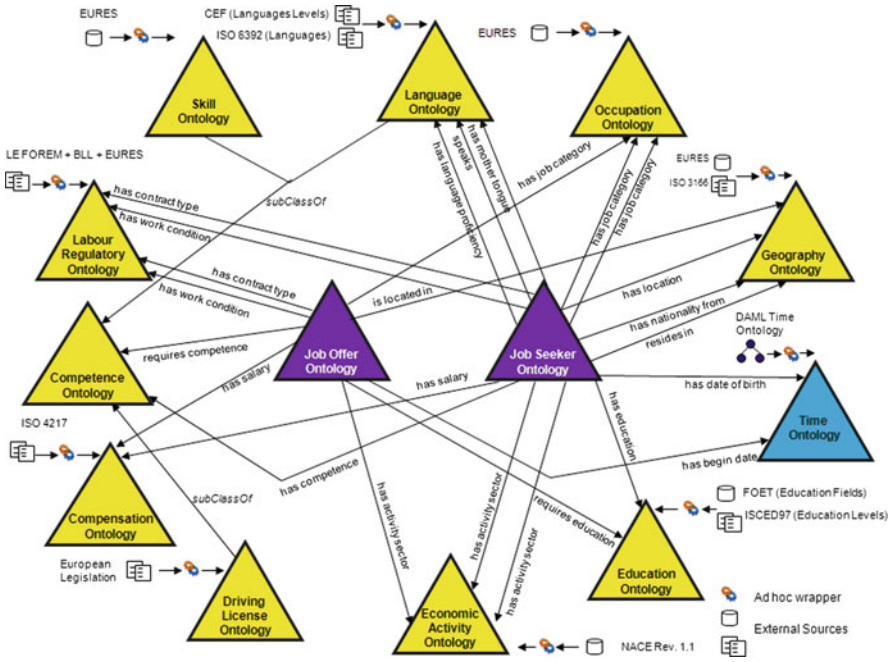


Fig. 6.8 SEEMP reference ontology

Table 6.7 SEEMP reference ontology statistical data

Ontology	Concepts	Attributes	Axioms	Instances	Efforts (man.months)
SEEMP RO	1,985	315	1,037	1,449	6

6.6.1.3 Analysis of the Applicability of the Method

The SEEMP Reference Ontology (SEEMP RO) was developed following the method for reusing and re-engineering non-ontological resources. It is composed of 13 modular ontologies: *competence, compensation, driving licence, economic activity, education, geography, job offer, job seeker, labur regulatory, language, occupation, skill, and time*. The main sub-ontologies are the *job offer* and *job seeker*, which are intended to represent the structure of a job posting and a CV, respectively. While these main two sub-ontologies were built taking as a starting point some HRXML recommendations, the others derived from some available international standards (like NACE, ISCO-88 (COM), FOET, etc.), employment services classifications, and international codes (like ISO 3166, ISO 6392, etc.) that best fitted the European requirements. Figure 6.8 presents these 13 modular ontologies (each ontology is represented by a triangle), 10 of which were obtained

after re-engineering the standard/classification. The SEEMP Reference Ontology is available at http://oeg-upm.net/index.php/en/ontologies/99-hrmontology .

In order to illustrate the dimension of the ontology and the ontological engineers' efforts required to build it, some statistical data are shown in Table 6.7.

Our experience in SEEMP has shown us that the approach of building ontologies by reusing and re-engineering non-ontological resources already agreed upon allows building ontologies faster, with less resources, and with an immediate consensus. This approach permits making explicit the knowledge implicitly coded in organization models and standards. By building ontologies in this fashion, we facilitate that ontologies become reference ontologies in their respective domains.

With respect to the application of the method for reuse and re-engineering, this was especially useful for guiding the steps of the ontological engineers involved since this method provides detailed and sufficient guidelines. In addition, the existence of a well-defined and structured process for building the ontology network in the e-employment domain eased the planning, coordination, and communication with other non–Semantic Web members of the development team, which in turn helped to convey reliability to the final result.

6.6.2 mIO! *Project*

The main objective of the *mIO!* project is to develop ubiquitous services in an intelligent environment, adapted to every user and its context by means of mobile interfaces. The project relies on ontologies for modeling the knowledge.

The following sections describe the application of our methodological guidelines for reusing and re-engineering non-ontological resources when building a geographical ontology, which includes continents, countries, and regions.

6.6.2.1 Reusing Non-ontological Resources

This section describes the activities carried out for reusing non-ontological resources.

Activity 1. Search Non-ontological Resources

Following some of the suggestions made by the domain experts, we searched geographical location resources on highly reliable web sites. Next, we list the geographic location classifications:

- ISO 3166[41] Maintenance Agency (ISO 3166/MA) ISO's focal point for country codes

[41] http://www.iso.org/iso/en/prods-services/iso3166ma/index.html

- Guide to regions of the world[42]
- Regions of the world[43]

Activity 2. Assess the Set of Candidate Non-ontological Resources

Once we had the set of candidate non-ontological resources, we needed to assess them according to the following criteria: precision, coverage, consensus, and quality of the resources.

Task 2.1 Extract Lexical Entries
Within this task, we extracted the lexical entries of the aforementioned geographic location classifications. For this purpose, we used TreeTagger[44], a syntactic annotator.
Task 2.2 Calculate Precision
It was impossible to cover all the geographic locations in the ORSD. Thus, we used a constant K that represents the cardinality of the complete set of geographical locations. Next, we present the precision for each geographic location classification:

$$Precision = \frac{card\{\{NORLexicalEntries\} \cap \{ORSDTerminology\}\}}{card\{NORLexicalEntries\}}$$

- $ISO3166 = \frac{195 \cap K}{200} = \frac{195}{200} = 0.975$
- $GuidetoregionsoftheWorld = \frac{102 \cap K}{193} = \frac{102}{193} = 0.528$
- $RegionsoftheWorld = \frac{110 \cap K}{154} = \frac{110}{154} = 0.714$

Task 2.3 Calculate Coverage
Again, it was impossible to cover all the geographic locations in the ORSD. Thus, we used a constant K that represents the cardinality of the complete set of

[42] http://www.countriesandcities.com/regions/
[43] http://park.org/Regions/
[44] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

geographic locations. Next, we present the coverage for each geographic location classification:

$$Coverage = \frac{card\{\{NORLexicalEntries\} \cap \{ORSDTerminology\}\}}{card\{ORSDTerminology\}}$$

- $ISO3166 = \frac{195 \cap K}{K} = \frac{195}{K}$
- $GuidetoregionsoftheWorld = \frac{102 \cap K}{K} = \frac{102}{K}$
- $RegionsoftheWorld = \frac{110 \cap K}{K} = \frac{110}{K}$

Task 2.4 Evaluate the Consensus
It was important for the project that resources focused on the current worldwide reality because the outcoming prototype will be validated by users.

Thus, domain experts evaluated whether the resource was built with the consensus of the worldwide community or not. They confirmed that ISO 3166 has the full consensus of the community, whereas the other resources have not.

Task 2.5 Evaluate the Quality
In this case, domain experts evaluated whether the resource was built with an acceptable level of quality. They confirmed that ISO 3166 has an acceptable level of quality.

Task 2.6 Build the Assessment Table
Table 6.8 summarizes all the information related to each non-ontological resource.

Activity 3. Select the Most Appropriate Non-ontological Resources

According to Table 6.8, we selected the following non-ontological resource: ISO 3166.

Table 6.8 Assessment table for the *mIO!* geographical locations

NOR	Precision	Coverage	Consensus	Quality
ISO 3166	0.975	195/K	Yes	Yes
Guide to regions of the World	0.528	102/K	No	No
Regions of the World	0.714	110/K	No	No

6.6.2.2 Re-engineering Non-ontological Resources

This section presents the application of the method for re-engineering non-ontological resources within the *mIO!* project. Once we have the non-ontological resource selected, the ISO 3166, we had to transform it into an ontology. Next, we describe the process of generating a geographical location ontology.

Activity 1. Non-ontological Resource Reverse Engineering

In this activity, we gathered documentation about ISO 3166 from its web site. From this documentation, we extracted the schema of the classification scheme, which consists of one entity *ISO 31661 Entry*. Since the data model was not available in the documentation, it was necessary to extract it for the resource implementation itself. ISO 3166 is modeled following the snowflake data model and implemented in XML.

Activity 2. Non-ontological Resource Transformation

In this activity, we carried out the following tasks:

1. We identified the transformation approach, the ABox transformation, i.e., the transformation of the resource schema into an ontology schema, and the resource content into ontology instances.
2. Then we searched our local pattern repository for a suitable pattern to re-engineer NORs, taking into account the transformation approach (ABox transformation), the non-ontological resource type (classification scheme), and the data model (snowflake data model) of the resource.
3. The most appropriate pattern for this case is the PR-NOR-CLAX-12 pattern. This pattern takes as input a classification scheme modeled with a snowflake data model.
4. Finally, we followed the procedure defined by the pattern selected for transforming the resource components into ontology elements.

Activity 3. Ontology Forward Engineering

In this activity, we formalized and implemented the ontology in OWL. The ontology is available at http://mccarthy.dia.fi.upm.es/ontologies/.

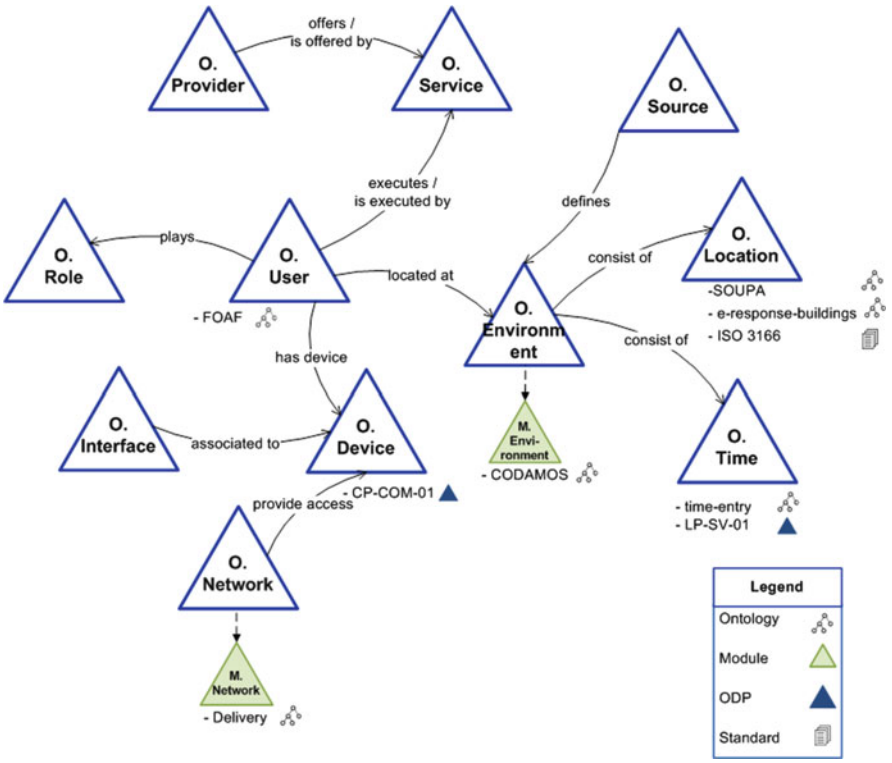


Fig. 6.9 *mIO!* Ontology network

Table 6.9 *mIO!* Ontology statistical data

Ontology	Concepts	Attributes	Axioms	Instances	Efforts (man.months)
mIO! ontology	432	276	154	120	3

6.6.2.3 Analysis of the Applicability of the Method

The network of ontologies of the *mIO!* project was developed following the NeOn Methodology (Suárez-Figueroa et al. 2008). This ontology is composed of 11 modular ontologies: *provider*, *service*, *source*, *geographical location*, *environment*, *time*, *device*, *user*, *network*, *interface*, and *role*. Only the geographical location ontology was built according to the method for reusing and re-engineering non-ontological resources. The other ontologies were built by reusing available ontologies or modules.

Figure 6.9 presents the *mIO!* ontology network and includes the location sub-ontology. The ontology network is available at http://oeg-upm.net/ index.php/en/ ontologies/82-mio-ontologies

In order to illustrate the dimension of the ontology and the efforts required by the ontological engineers to build it, we outline some data in Table 6.9.

Our experience in *mIO!* has served us to demonstrate that the approach of building ontologies by reuse and re-engineering non-ontological resources already agreed upon allows building ontologies faster, with less resources, and with consensus. With respect to the application of the method for reuse and re-engineering, this was especially useful for guiding the steps of the ontological engineers involved since the method provides detailed and sufficient guidelines.

6.7 Conclusions

In this chapter, we have provided a method and its technological support that rely on re-engineering patterns in order to speed up the ontology development process by reusing and re-engineering as much as possible available non-ontological resources. Moreover, we have introduced a three-level categorization of NORs according to three different features: type of NOR, data model, and implementation. Finally, we have presented two use cases of the proposed approach.

References

Baeza-Yates Ricardo, Ribeiro-Neto Berthier (1999) Modern information retrieval, 1st edn. Addison Wesley, Harlow. ISBN 020139829X

Brandon D (2005) Recursive database structures. J Comput Sci Coll 1:295–304

Brockmans S, Haase P (2006) A metamodel and UML profile for networked ontologies. A complete reference. Technical report, University Karlsruhe, 2006

Carkenord B (2002) Why build a logical data model. http://www.embarcadero.com/resources/tech papers/datamodel.pdf

Gangemi A, Pisanelli D, Steve G (1998) Ontology integration: experiences with medical terminologies. Ontol Inf Syst 1:163–178

Gangemi A, Guarino N, Masolo C, Oltramari A (2003) Sweetening WORDNET with DOLCE. AI Mag 24(3):13–24, ISSN 0738–4602

Gangemi A, Catenacci C, Ciaramita M, Lehmann J (2006) Modelling ontology evaluation and validation. In: Proceedings of the 3rd European Semantic Web Conference (ESWC2006), LNCS, vol 4011. Springer, Budva, 2006

García R, Celma O (2005) Semantic integration and retrieval of multimedia metadata. In: Proceedings of the ISWC 2005 workshop on knowledge markup and semantic annotation (Semannot'2005), Galway, Ireland

Gómez-Pérez A, Manzano-Macho D (2004) An overview of methods and tools for ontology learning from texts. Knowl Eng Rev 19(3):187–212. ISSN 0269–8889, doi: http://dx.doi.org/ 10.1017/ S0269888905000251

Gómez-Pérez A, Fernández-López M, Corcho O (2003) Ontological engineering, Advanced information and knowledge processing. Springer, New York/London. ISBN 1–85233–551–3

Hepp M (2006) Products and services ontologies: a methodology for deriving owl ontologies from industrial categorization standards. Int J Semant Web Inf Syst 2(1):72–99

Hepp M (2007) Possible ontologies: how reality constrains the development of relevant ontologies. IEEE Internet Comput 11(1):90–96

Hepp M,de Brujin J (2007) GenTax: a generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In: Proceedings of the 4th European Semantic Web Conference (ESWC2007). Springer, Innsbruck

Hirst G (2004) Ontology and the lexicon. In: Handbook on ontologies in information systems. Springer, Berlin, pp 209–230

Hodge G (2000) Systems of knowledge organization for digital libraries: beyond traditional authority files. http://www.clir.org/pubs/reports/pub91/contents.html

Hyvöonen E, Viljanen K, Tuominen J, Seppöalöa K (2008) Building a national semantic web ontology and ontology service infrastructure -the FinnONTO approach. In: ESWC, vol 1. Springer, Heidelberg, pp 95–109

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

ISO/IEC FDIS 11179–1 (2004) Information technology – metadata registries – part 1: framework. International Standard Organization (ISO), Report ISO/IEC FDIS 11179–1

Kimball R, Caserta J (2004) The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data. Wiley, New York. ISBN 0764567578

Maedche A, Staab S (2001) Ontology learning for the semantic web. IEEE Intell Syst 16:72–79

Malinowski E, Zimányi E (2006) Hierarchies in a multidimensional model: from conceptual modeling to logical representation. Data Knowl Eng 59:348–377

Pinto S, Tempich C, Staab S (2004) DILIGENT: towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004). IOS Press, Amsterdam, Washington, DC, pp 393–397, ISBN 1–58603–452–9

Sabou M, Angeletou S, d'Aquin M, Barrasa J, Dellschaft K, Gangemi A, Lehman J, Lewen H, Maynard D, Mladenic D, Nissim M, Peters W, Presutti V, Villazón-Terrazas B (2007) Selection and integration of reusable components from formal or informal specifications. Technical report, NeOn project deliverable D2.2.1, 2007

Schnurr H-P, Studer R, Sure Y (2001) Knowledge processes and ontologies. IEEE Intell Syst 1(16):26–34

Soergel D (1995) Data models for an integrated thesaurus database. Compat Integr Order Syst 24(3):47–57

Suárez-Figueroa MC, Aguado de Cea G, Buil C, Dellschaft K, Fernández-López M, García-Silva A, Gómez-Pérez A, Herrero G, Montiel-Ponsoda E, Sabou M, Villazón-Terrazas B, Yufei Z (2008) NeOn Methodology for building contextualized ontology networks. Technical report, NeOn project deliverable D5.4.1, 2008

Suárez-Figueroa MC, Gómez-Pérez A, Villazón- Terrazas B (2009) How to write and use the ontology requirements specification document. In: OTM Conferences (2), pp 966–982, 2009

Villazón-Terrazas B, Angeletou S, García-Silva A, Gómez-Pérez A, Maynard D, Suárez-Figueroa MC, Peters W (2008) NeOn deliverable D2.2.2 methods and tools for supporting reengineering. Technical report, NeOn, 2008

Villazón-Terrazas B, Suárez-Figueroa MC, Gómez-Pérez A (2010) A pattern-based method for re-engineering non-ontological resources into ontologies. Int J Semant Web Inf Syst 6(4):27–63

Wright SE, Budin G (eds) (1997) Handbook of terminology management, basic aspects of terminology management. John Benjamins Publishing Company, Amsterdam