NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D2.4.4 Modelling and re-engineering linguistic/terminological resources

---

Deliverable Co-ordinator: **Wim Peters**

Deliverable Co-ordinating Institution: **USFD**

Other Authors: **Aldo Gangemi (CNR); Boris Villazón-Terrazas (UPM)**

Document Identifier:	NEON/2010/D2.4.4/v1.0	Date due:	January 31st, 2010
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	January 31st, 2010
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – Kmi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta} @open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Peter Haase E-mail address: pha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person : Jérôme Euzenat E-mail address: glesi.euzenat@inrialpes.fr	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: glesi@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: gle@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 – 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Food and Agriculture Organization of the United Nations (FAO)** Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	**Laboratorios KIN, S.A. (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

USFD

UPM

CNR

Change Log

Version	Date	Amended by	Changes
0.1	17-12-2009	Boris Villazón-Terrazas	Initial TOC (according to Wim proposal)
0.2	18-12-2009	Wim Peters	Addition of chapter 2
0.3	5-1-2010	Boris Villazón-Terrazas	First draft of chapter 4
0.4	5-1-2010	Wim Peters	Summary, Chapter 1
0.5	12-1-2010	Boris Villazón-Terrazas	Update chapters 3 and 4
0.6	13-1-2010	Wim Peters	Update chapter 4
0.7	15-1-2010	Wim Peters	Extension chapter 2
0.8	26-1-2010	Boris Villazón-Terrazas	Update chapters 3 and 4
0.9	28-1-2020	Wim Peters	Chapter 5
2.0	1-1-2010	Wim Peters	Final remaining content issues and editorial work

Executive Summary

This document addresses the NeOn methodology for re-engineering, whose workflow integrates various aspects of WP2 tasks T2.2 and T.2.4: re-engineering patterns and standardized descriptions of the linguistic/terminological content of resources.

The main ingredients of this deliverable are the following:

- The creation of a set of networked ontologies on the basis of links between the LIR model (deliverables 2.4.1 to 2.4.3) and standard metamodels for linguistic/terminological description.
- The description of re-engineering patterns for lexicons in addition to the ones described in D2.2.2: Methods and Tools Supporting Re-engineering.
- The description and illustration of the full re-engineering method from linguistic resource to ontology, which integrates networked linguistic ontologies and re-engineering patterns in different stages of the re-engineering process.

Table of Contents

List of tables

List of figures

1. Introduction

This document integrates various aspects of WP2 tasks: the re-engineering methodology of task T2.2 and the T2.4 standardized descriptions of the linguistic/terminological content of resources.

To recapitulate, in D2.2.2 we presented the NeOn method for non-ontological resource re-engineering into ontologies. This method aims to perform a conversion, as completely as possible, of knowledge included in the resource into ontologies, using a pattern-based re-engineering approach. We described the application of this method to classification schemes and thesauri. Then, we presented the proposed template used to describe the patterns for re-engineering them.

In this deliverable, we extend the pattern-based approach to lexicons (chapter 3).

In deliverables 2.4.1 to 2.4.3, we described the creation of the Linguistic Information Repository (LIR) model, which captures multilingual linguistic information associated with class labels. This model incorporates elements from existing standard models for linguistic and terminological description, such as ISO12620 and LMF. Further, we described the integration of the LIR into a set of networked ontologies.

In this deliverable we present an implemented version of this network according to an extension of the mapping metamodel presented in deliverables D1.1.1 and D1.1.2 [Haase et al.,2007].

Re-engineering patterns and standardized modelling are combined into a re-engineering workflow. The re-engineering of resources makes use of the information captured by ontology elements in these networked ontologies. The re-use of elements from standard models guarantees interoperability between re-engineered resources. Also, adherence to standards and best practises for capturing linguistic knowledge increases the uniformity of re-engineering patterns and methods.

The main ingredients of this deliverable are the following:

- Section 2: the creation of a set of networked ontologies on the basis of links between the LIR model (deliverables 2.4.1 to 2.4.3) and standard metamodels for linguistic/terminological description.
- Section 3: the description of re-engineering patterns for lexicons in addition to the ones described in D2.2.2: Methods and Tools Supporting Re-engineering.
- Section 4: the description and illustration of the full re-engineering method from linguistic resource to ontology, which integrates networked linguistic ontologies and re-engineering patterns in different stages of the re-engineering process.

2. Networked ontologies for linguistic/terminological description: Creating the LingNet network.

In the previous deliverable D2.4.3 [Peters et al.,2009] we argued that the LIR is not a stand-alone model. It incorporates elements from, and is intricately related to a set of established standard models describing linguistic and terminological knowledge. We re-engineered some of these models, and described the embedding of LIR into this network. In summary, this establishes interoperability of LIR with:

- standard models for translation memory
- the LMF standard model
- the LMM metamodel
- the LexInfo model

Most of these are available from the web (see D2.4.3). Models for translation memory needed to be re-engineered from their xml source in an ad hoc fashion.

In this section we formalize the relations that pull the above ontologies together into a network.

In D2.4.3, the following mappings between ontology elements were informally expressed:

- IsEquivalentTo
- hasHypernym; hasHyponym
- hasPart; partOf
- PartialOverlap

They have now been integrated into a mapping model created for the networking of ontologies for linguistic/terminological description.

2.1 Mapping metamodel

This model is an extension of the mapping metamodel as described in D1.1.2 (section 6.1) [Haase et al.,2007], [Brockmans et al.,2006]. The mapping metamodel takes a number of different kinds of semantic relations that have been proposed in the literature into account [Brockmans et al.,2006]. Most common are the following kinds of semantic relations:

Equivalence states that the connected elements represent the same aspect of the real world according to some equivalence criteria. A very strong form of equivalence is identity, if the connected elements represent exactly the same real world object.

Containment states that the element in one ontology represents a more specific aspect of the world than the element in the other ontology. Depending on which of the elements is more specific, the containment relation is defined in the one or in the other direction.

Overlap states that the connected elements represent different aspects of the world, but have an overlap in some respect. In particular, it states that some objects described by the element in the one ontology may also be described by the connected element in the other ontology.

The mapping metamodel has the following properties:

- A mapping is a set of mapping assertions that consist of a semantic relation between mappable elements in different ontologies
- Mappings are first-class objects that exist independently of the ontologies. Mappings are directed and there can be more than one mapping between two ontologies
- mappings are independent on the concrete mapping formalism. The metamodeling approach of MDA and MOF, as described in the deliverable, allows to define the networked ontology model in an abstract form independent of the particularities of specific logical formalisms. This enables to be compatible with currently competing formalisms (e.g. in the case of mapping languages), for which no standard exists yet.

Figure 2.1 below illustrates this metamodel. It is available from:

http://www.gate.ac.uk/ns/ontologies/LingNet/MappingMetamodel.owl

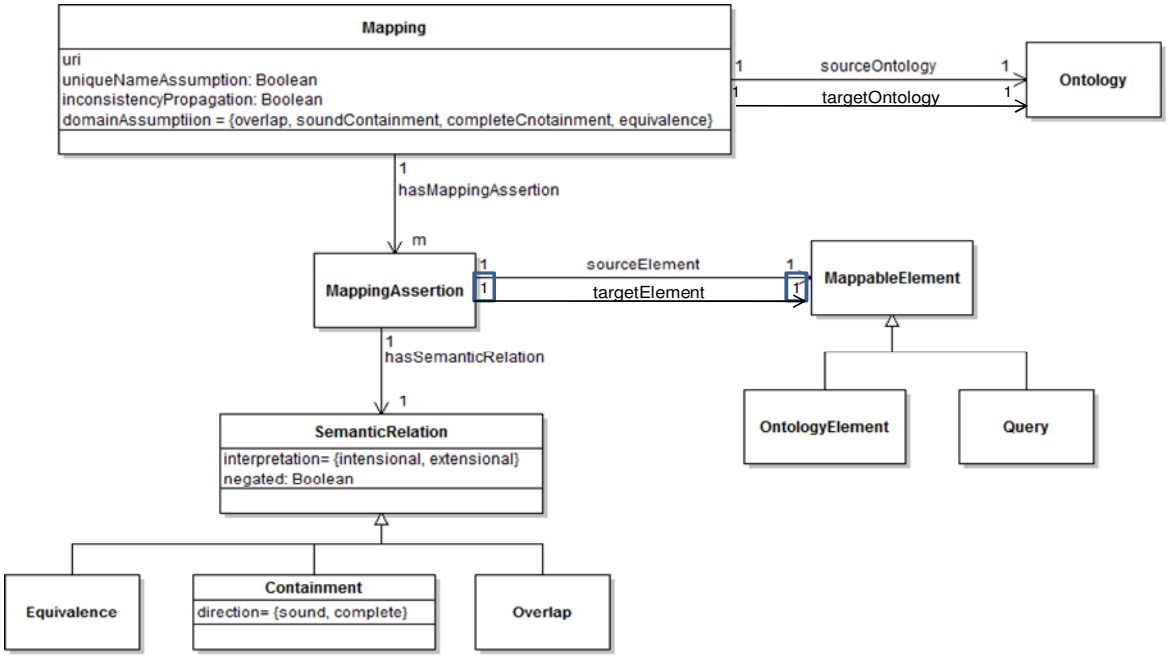


Figure 2.1. The metamodel for ontology mappings in D1.1.2

2.2 Extension and population of the metamodel: LingNet

For our purposes we need, for now, a more specific subcategorization than the metamodel described in figure 2.1 provides. We therefore have extended this model with the following subclasses of the Containment concept:

TaxonomicalContainment

Hypernymy

MeronymicalContainment

Part

Member

These classes make the specifics of the Containment relation explicit by means of a distinction between taxonomic and meronymic containment.

An additional requirement is that the model should cover structural differences between ontologies. In D2.4.3 it was argued that, even if ontologies are conceptually equivalent, they often express their content in different ways, because they differ from each other in structural terms. For instance, a concept in one ontology can be expressed with a concept and an attribute in the other.

[Scharffe et al., 2008] defines a principled approach to the creation of a typology of alignment patterns The identified patterns have been collected into a library[1].

The pattern hierarchy looks as follows:

Pattern

AttributeCorrespondence

AttributeTransformation

EquivalentAttribute

Sub-Super-Attribute

ClassCorrespondence

ClassByAttribute

ClassByAttributeOccurence

ClassByAttributeType

ClassByAttributeValue

ClassIntersection

EquivalentClassIntersection

SubClassIntersection

SuperClassIntersection

ClassUnion

EquivalentClassUnion

SubClassUnion

SuperClassUnion

EquivalentClass

InstanceOfClass

Sub-Super-Class

RelationCorrespondence

[1] http://www.omwg.org/TR/d7/patterns-library/

EquivalentRelation

Sub-Super-Relation

To recapitulate, when comparing the empirically identified typology on the basis of the informal alignments described in D2.4.3, sections 4.2 to 4.4 with this typology, we find the following:

D2.4.3 empirical patterns	pattern library
1. Class to class	EquivalentClass
2. Class with attribute to class	ClassByAttributeValue
3. Class to union of classes	ClassUnionCorrespondence
4. Relation to relation	Sub-Super-RelationCorrespondence EquivalentRelationCorrespondence
5. Relation to class	
6. Relation to attribute	
7. Attribute to attribute	EquivalentAttributeCorrespondence SubSuperAttributeCorrespondence AttributeTransformationCorrespondence

The pattern library covers five out of seven patterns. In order to cater for correspondence patterns 5 and 6 we added the classes RelationClassCorrespondence and AttributeRelationCorrespondence respectively into the LingNet mapping metamodel.

The correspondence patterns have been intensionally used in this model. The logical consequences of their extensional use have not been taken into account.

Figure 2.2 below illustrates the resulting LingNet mapping model. It is available from:

http://www.gate.ac.uk/ns/ontologies/LingNet/LingNetMetamodel.owl

The advantage of the extended meta-model is that is formalism-independent [Brockmans et al, 2006]. The mappings are knowledge-based, i.e. they do not require language-specific constructs for mappings. The reification of mapping relations into classes allows us to describe them as ontological objects and model the relations in a detailed and extendable fashion.

The interoperability of the standard models allows a flexible choice of standard modelling for a resource, and the inherent potential for conversion of the resource model into any networked standard format.

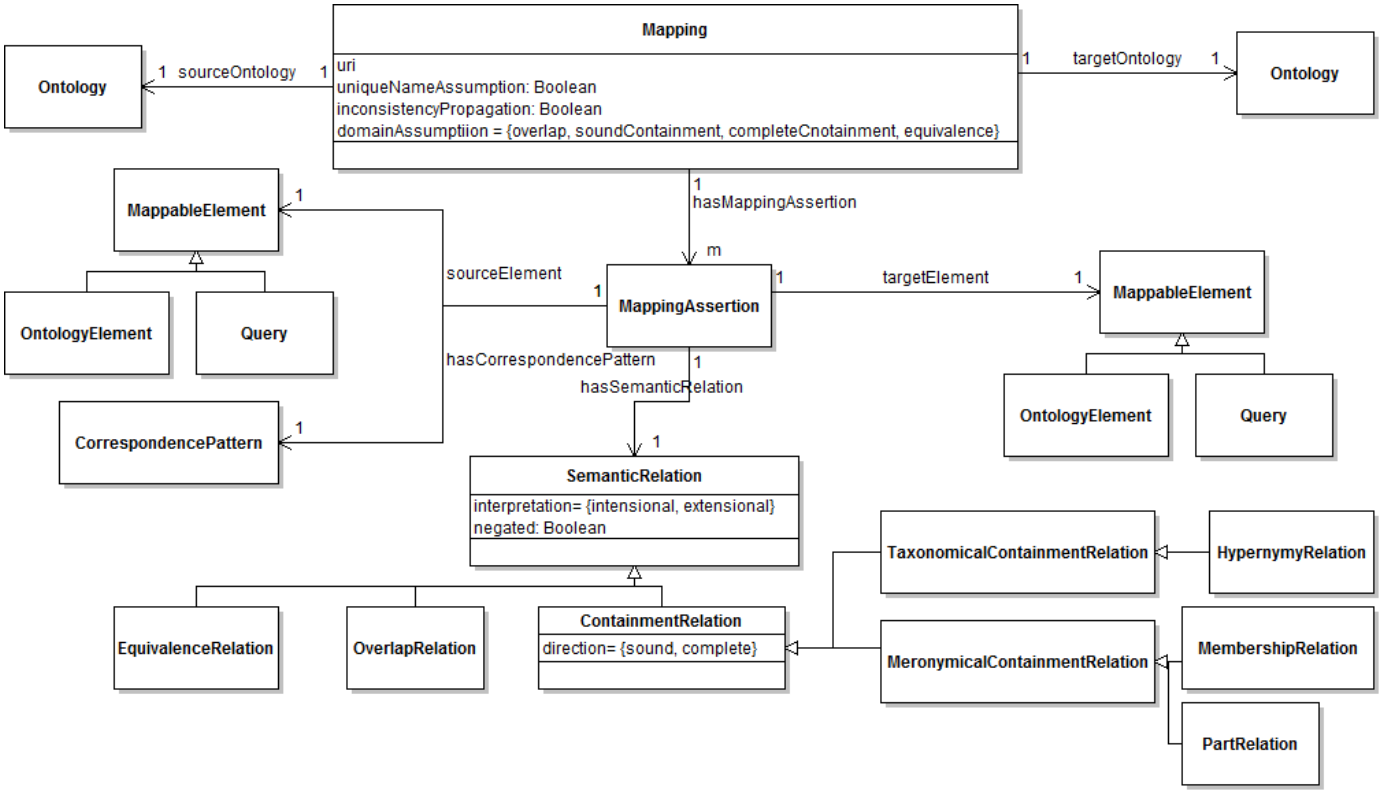


Figure 2.2. The mapping model for linguistic/terminological ontology mappings

2.3 Implemented correspondences

In total 6 binary mappings between LIR and five other ontologies have been implemented involving 72 ontology elements and 55 mapping assertions. The five networked ontologies are the following (see D2.4.3 for their descriptions):

TMX[2]**:** Translation Memory eXchange

XLIFF[3]**:** XML Localization Interchange File Format

MLIF: Multi Lingual Information Framework

LMF[4]**:** Lexical Markup Framework

LMM[5]**:** Linguistic Meta-Model

LexInfo[6]

[2] http://www.lisa.org/tmx/

[3] http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.pdf

[4] http://www.lexicalmarkupframework.org/

[5] http://www.ontologydesignpatterns.org/ont/lmm/LMM_L2.owl

[6] http://lexonto.ontoware.org/lexinfo

The populated model is available from the following URI:
http://www.gate.ac.uk/ns/ontologies/LingNet/LingNet-v0.1.owl

2.3.1 LIR-LMF

Class to class

- Lir:LexicalEntry isPartOf lmf:LexicalEntry
- Lir:Lexicalization isEquivalentTo lmf:WordForm
- Lir:Sense isEquivalentTo lmf:Sense
- Lir:Definition isEquivalentTto lmf:Definition
- Lir:UsageContext isEquivalentTo lmf:Context
- Lir:Source overlapsWith lmf:MultilingualExternalReference
- Lir:Source overlapsWith lmf:MonolingualExternalReference

Attribute to attribute

- Lir:grammaticalNumber isEquivalentTo lmf: grammaticalNumber

Relation to class

- Lir:isRelatedTo isEquivalentTo lmf:SenseRelation
- Lir:hasStemmedForm isEquivalentTo lmf:Stem
- Lir:hasSynonym isHyponymOf lmf:SenseRelation

Relation to class

- Lir:hasTranslation isEquivalentTo lmf:SenseAxis

Relation to attribute

- Lir:belongsToLanguage isEquivalentTo Language (LMF core module)
- Lir:haspartOfSpeech isEquivalentTo lmf:partOfSpeech

2.3.2 LIR-TMX

Class to class

- lir:UsageContext isEquivalentTo tmx:Context
- lmf-component:Component isEquivalentTo tmx:Segment
- lir:Note isEquivalentTo tmx:Note

- lir:LexicalEntry isEquivalentTo tmx:TranslationUnitVariant

Relation to class

- lir:hasTranslation overlapsWith tmx:TranslationUnit

Relation to attribute

- lir:belongsToLanguage isEquivalentTo tmx:srcLang

2.3.3 LIR-XLIFF

Class to class

- lir:LexicalEntry isEquivalentTo xliff:Source
- lir:LexicalEntry isEquivalentTo xliff:Target
- lir:LexicalEntry overlapsWith xliff:SegSource
- lir:hasTranslation overlapsWith xliff:AltTrans
- lmf-component-module:Component isEquivalentTo xliff:Mrk with mType=”seg”
- lmf-component-module:ComponentList isEquivalentTo xliff:SegSource

Relation to class

- lir:hasTranslation isEquivalentTo xliff:TransUnit
- lir:hasTranslation isEquivalentTo xliff:AltTrans

Relation to attribute

- lir:belongsToLanguage isEquivalentTo xliff:languageIdentifier
- lir:hasSynonym isEquivalentTo xliff: equivTrans

2.3.4 LIR-MLIF

Class to class

- Lir:LexicalEntry isEquivalentTo mlif:MonolingualComponent
- lmf-component-module:Component isEquivalentTo mlif:SegmentationComponent

Relation to class

- lir:hasTranslation isEquivalentTo mlif:MultiLingualComponent

2.3.5 LIR-LMM

Class to class

- Lir:LexicalEntry isEquivalentTo lmm2:Lexeme
- Lir:Lexicalization hasPartialOverlapWith lmm2:Grapheme
- Lir:UsageContext hasHypernym lmm2:CoText
- Lir:Sense hasHypernym lmm2:Meaning
- Lir:Definition hasHypernym lmm2:Description
- Lir:Note hasHypernym lmm2:Text
- Lir:Source hasHypernym lmm2:InformationObject
- Lir:Language isEquivalentTo lmm2:NaturalLanguage
- Lir:LanguageCode hasHypernym lmm2:Code
- lmf-component-module:Component hasHypernym lmm2:InformationObject

Class with attribute to class

- Lir: LexicalEntry with Lir:Lexicalization attribute multiWordExpression =”true” isEquivalentTo lmm:MultiWord
- Lir: LexicalEntry with Lir:Lexicalization attribute phrase =”true” isEquivalentTo lmm:Phrase

Relation to class

- lir:isRelatedTo isEquivalentTo lmm2:relatedMeaning
- lir:hasSynonym hasHypernym lmm2:relatedMeaning

2.3.6 LIR-LexInfo

In order to widen the scope of the network to a satisfactorily comprehensive level, we decided to incorporate LexInfo [Buitelaar et al.,2009]. Lexinfo imports Lmf. For this import the mappings established between LIR and LMF are valid.

In addition, given the lack of orthographic coverage in LexInfo, and the lack of syntactic information in LIR, only the following additional mappings have been created:

Class to class

Lir:LexicalEntry equivalence lexinfo:LexicalEntry

Lir:Sense equivalence lexinfo:Sense

Class with attribute to class

Lir:LexicalEntry classByAttribute lexinfo:Noun

Lir:LexicalEntry classByAttribute lexinfo:Verb

Lir:LexicalEntry classByAttribute lexinfo:Adjective

Lir:LexicalEntry classByAttribute lexinfo:Preposition

2.4 Conclusion and future work on LingNet

LingNet provides a knowledge-based, language-independent representation of alignments between ontologies that capture linguistic/terminological description.
It captures semantic as well as structural alignments.
The populated LingNet model constitutes a first step towards a full network in which ontology elements from all networked ontologies are connected to each other. At the moment the LIR functions as hub, and therefore the coverage of the network is restricted to the areas of linguistic description covered by LIR (orthography, morphosyntax, semantics, translation), which are covered to varying degrees by the different NLP application areas of linguistics, terminology and translation. This is illustrated in the figure below. Ontologies are now indirectly mapped through the LIR.

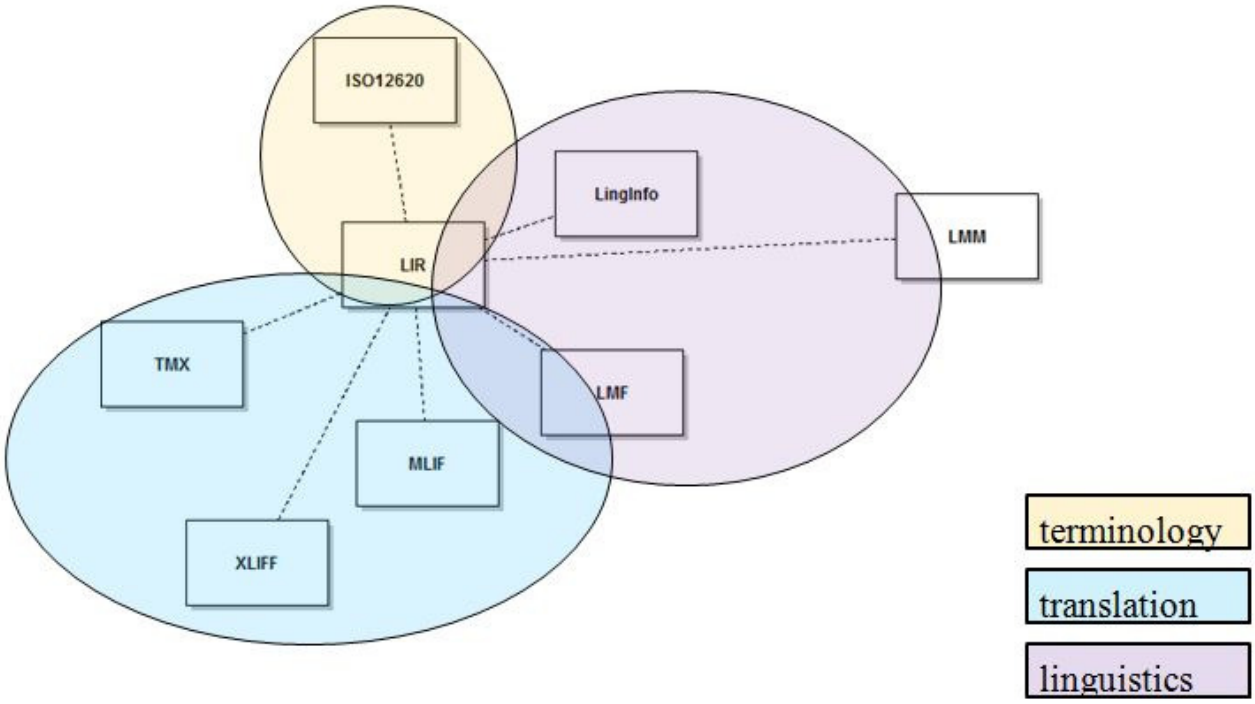


Figure 2.3. Networked ontologies grouped according to domains of linguistic description

By adding missing binary mappings between ontology element pairs contained in the as yet unmapped ontologies, the network will be completed. Eventually, all mappings will be defined explicitly by means of direct binary relations.

3. Patterns for re-engineering lexicons

3.1 Introduction

The term *lexicon* is used in many ways, including conventional print dictionaries in book form, CD-ROM editions, Web based versions of the same, but also computerized resources of similar structures to be used by applications. During de 1970's and 80's computational linguistics began to develop *computational lexicons* for natural language processing programs. Computational lexicons differ from dictionaries intended for human use in that they must contain much more explicit and specific linguistic information about phrases and words, and must be encoded in strictly formal structures operable by computer programs. In this chapter we present a definition of lexicon, the data models for representing lexicons and the patterns for re-engineering lexicons into ontologies.

3.2 Lexicon

According to [Hirst, 2004], a lexicon is a list of words in a language (a vocabulary) along with some knowledge of how to use each word. A lexicon may be general or domain-specific; we might have, for example, a lexicon of several thousand common words of English or German, or a lexicon of the technical terms of dentistry in some language. The words that are of interest are usually open-class or content words, such as nouns, verbs, and adjectives, rather than closed-class or grammatical function words, such as articles, pronouns, and prepositions, whose behaviour is more tightly bound to the grammar of the language. A lexicon may also include multi-word expressions such as fixed phrases (*by and large*), phrasal verbs (*tear apart*), and other common expressions (merry Christmas!; Elvis has left the building).

Also, Hirst [Hirst, 2004] points out that an ordinary dictionary is an example of a lexicon. However, a dictionary is intended for use by humans, and its style and format are unsuitable for computational use in a text or natural language processing system without substantial revision. A dictionary in a machine-readable format can serve as the basis for a computational lexicon, as in the ACQUILEX project[7], and it can also serve as the basis of a semantic hierarchy.

3.3 Lexicon data model

As we mentioned in D222 [Villazón-Terrazas et al., 2008] there are different ways of representing the knowledge encoded by a particular resource. In this section we present the data models we have found for lexicons. Next, we present brief descriptions of the models and standards that deal with lexical information

3.3.1 Lexical Markup Framework

The Lexical Markup Framework (LMF; ISO/CD 24613) [Francopoulo et al., 2006] is an abstract metamodel that provides a common, standardized framework for the construction of computational lexicons. LMF ensures the encoding of linguistic information in a way that enables reusability in different applications and for different tasks. LMF provides a common, shared representation of lexical objects, including morphological, syntactic, and semantic aspects. LMF provides mechanisms that allow the development and integration of a variety of electronic lexical resource types. It supports existing lexical resource models such as the Genelex [Antoni-Lay et al., 1994],

[7] http://www.cl.cam.ac.uk/research/nl/acquilex/

the EAGLES International Standards for Language Engineering (ISLE) [Calzolari et al., 1996] and Multilingual ISLE Lexical Entry (MILE) models [Calzolari et al., 2003].

Based on LMF [Francopoulo et al., 2006] we identify the following lexicon components (see Figure 1):

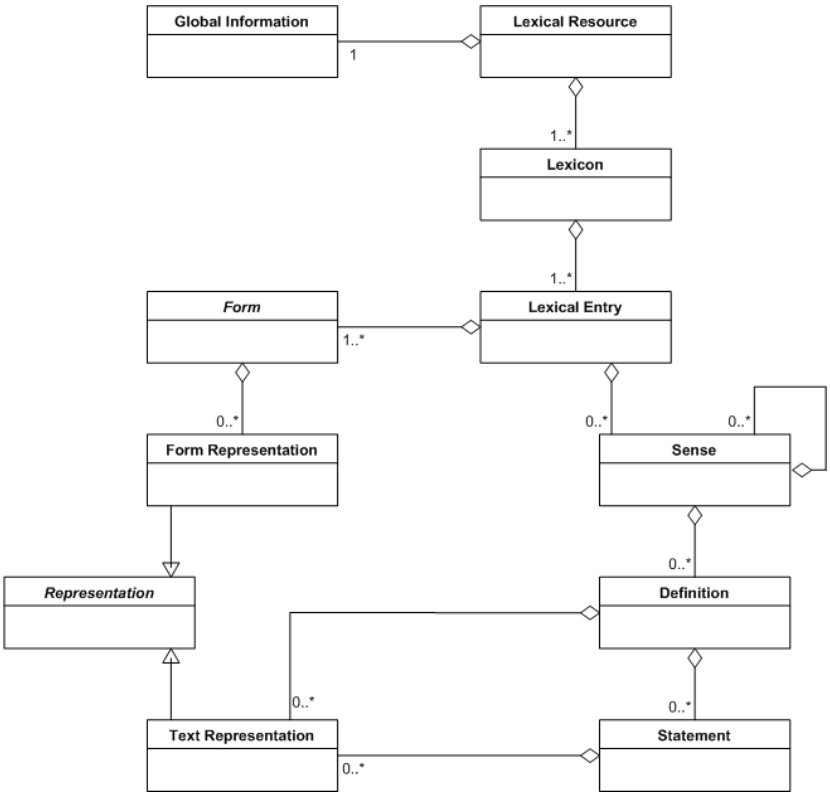


Figure 3.1. UML representation of the LMF Model components

- *Lexical Resource* component, which represents the entire resource. The Lexical Resource is a container for one or more lexicons.
- *Global Information* component, which constitutes the administrative information and other general attributes. There is an aggregation relationship between the *Lexical Resource* and the *Global Information* in that the latter describes the administrative information and general attributes of the entire resource.
- *Lexicon* component, which contains all the lexical entries of a given language within the entire resource. A *Lexicon* must contain at least one lexical entry.
- *Lexical Entry* component, which represents a lexeme in a given language. The *Lexical Entry* is a container for managing the *Form* and *Sense*. Therefore, the *Lexical Entry* manages the relationship between the forms and their related senses. A *Lexical Entry* can contain one to many different forms, and can have from zero to many different senses.
- *Form* component, which represents a lexeme, a morphological variant of a lexeme or a morph. The *Form* manages one or more orthographical variants of the data categories that describe the attributes of the word form (e.g., lemma, pronunciation, syllabification).
- *Form Representation* component, which constitutes one variant orthography of a *Form*. When there is more than one variant orthography, the *Form Representation* contains a Unicode string representing the *Form* as well as, if needed, the unique attribute-value pairs that describe the specific language, script, and orthography.

- *Representation* component, which represents a Unicode string as well as, if needed, the unique attribute-value pairs that describe the specific language, script, and orthography.
- *Sense* component, which represents one meaning of a lexical entry. It allows for hierarchical senses in that a sense may be more specific than another sense of the same lexical entry.
- *Definition* component, which represents a narrative description of a sense. It is displayed for human users to facilitate their understanding of the meaning of a *Lexical Entry* and is not meant to be processable by computer programs. A *Sense* can have zero to many definitions. Each *Definition* may be associated with zero to many *Text Representation* components in order to manage the text definition in more than one language or script. The narrative description can be expressed in a different language and/or script than the one of the *Lexical Entry* component.
- *Statement* component, which constitutes a narrative description and refines or complements *Definition*. A *Definition* can have zero to many *Statement* instances.
- *Text Representation* component, which represents one textual content of *Definition* or *Statement*. When there is more than one variant orthography, the *Text Representation* contains a Unicode string representing the textual content as well as the unique attribute-value pairs that describe the specific language, script, and orthography.

3.3.2 WordNet-LMF

WordNet-LMF [Soria et al., 2009] is a dialect of ISO Lexical Markup Framework that instantiates LMF for representing wordnets. The goal of WordNet-LMF is 1) to give a preliminary assessment of LMF, by large-scale application to real lexical resources and 2) to endow WordNet with a format representation that will allow easier integration among resources sharing the same structure (i.e. wordnets).

LMF specifications are fully compatible with the structural organization of lexical knowledge encoded in wordnet-like lexical resources. Starting from the meta-model provided by LMF, the additional packages used in WordNet-LMF are the semantics and the multilingual extension packages. These extensions packages are briefly described next and depicted in Figure 3.2 and Figure 3.3.

The representations of the semantic aspects of words is entrusted to objects related and aggregated to *Sense* class. This class represents lexical items as lexical semantic units. Each *Sense* instance describes one meaning of a *Lexical Entry*. *Synset* clusters synonymous *Sense* instances. *SenseRelation* and *SynsetRelation* classess encode (lexical) semantic relationships among instances of the *Sense* or *Synset* class.

The multilingual notation package can be used to represent bilingual and multilingual resources. The framework, based on the notion of Axis, accommodates transfer, *TransferAxis*, and interlingual pivot approaches, *SenseAxis*. This package comes equipped with the possibility to define connections between a node in a lexicon (e.g. a *SenseAxis* instance) and knowledge representation systems, such as ontologies or fact databases as well.

Wordnet-LMF wordnet format deviates from standard LMF only regarding the way data categories are instantiated. WordNet-LMF represents the information by means of XML attributes and values instead of nested elementes. By explicitly naming the attributes, it is possible to make a stronger claim about the features and properties of the structure of a wordnet. This enforces better compability and interoperability across many wordnets for different languages that are available.

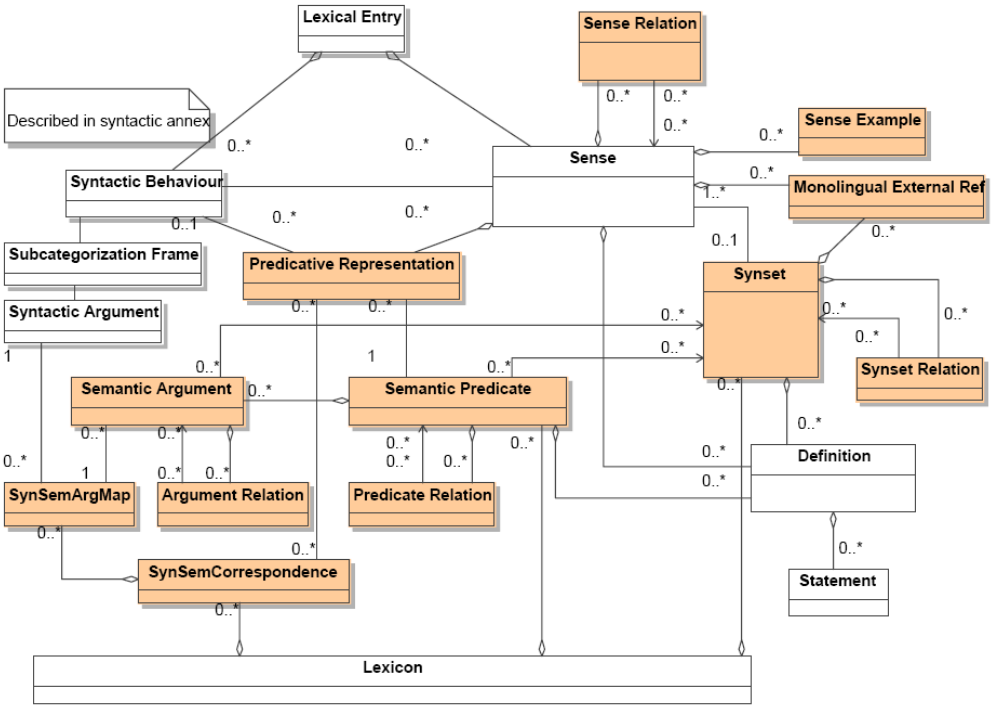


Figure 3.2. LMF Semantic extension package [Francopoulo et al., 2006]

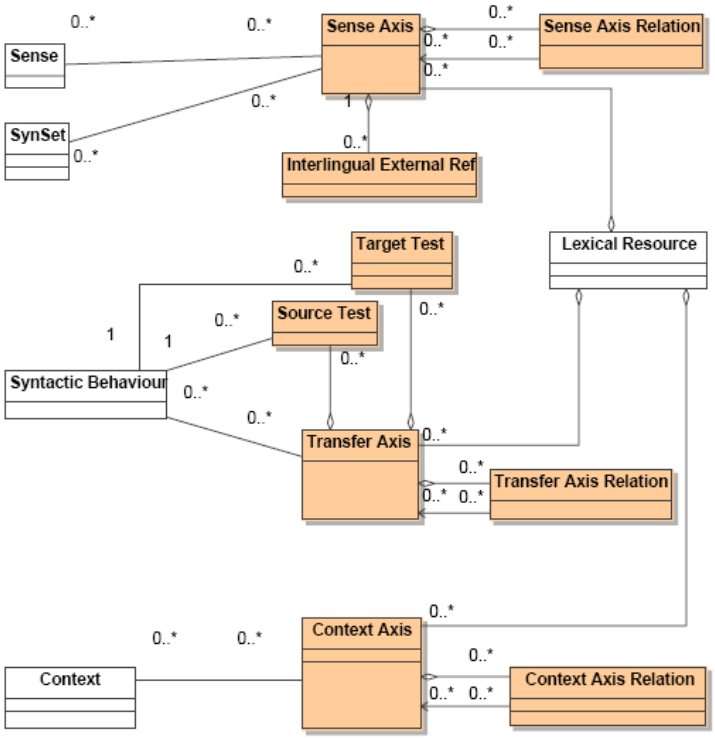


Figure 3.3. LMF Multilingual notation package [Francopoulo et al., 2006]

3.3.3 TMX

TMX[8] (Translation Memory eXchange) is the vendor-neutral open XML standard for the exchange of Translation Memory (TM) data created by Computer Aided Translation (CAT) and localization tools. The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process. In existence since 1998, TMX is a certifiable standard format. TMX is developed and maintained by OSCAR (Open Standards for Container/Content Allowing Re-use), a LISA Special Interest Group.

Figure 3.4 illustrates the organization of the subset of TMX classes that are most relevant to the modelling of translation relations. These are the following:

1. **TranslationUnit** (tu) contains the data for a given translation unit.
2. Attribute segType: "block", "paragraph", "sentence", or "phrase".
3. Attribute srcLang: Source language: specifies the language of the source text
4. **TranslationUnitVariant** (tuv) specifies text in a given language
5. Required attribute: xml:lang
6. **Context** describes the context of a TranslationUnit. The purpose of this context information is to allow certain pieces of text to have different translations depending on where they came from. The translation of a piece of text may differ if it is a web form or a dialog or an Oracle form or a Lotus form for example. This information is thus required by a translatorwhen working on the file. Likewise, the information may be used by any tool proposing to automatically leverage the text successfully.
7. **Segment** an individual segment of translation-memory text in a particular language. It contains the text of the given segment. There is no length limitation to the content of a Segment element. All spacing and line-breaking characters are significant within a Segment element.
8. **Note** is used for comments. It has the attribute xml:lang.

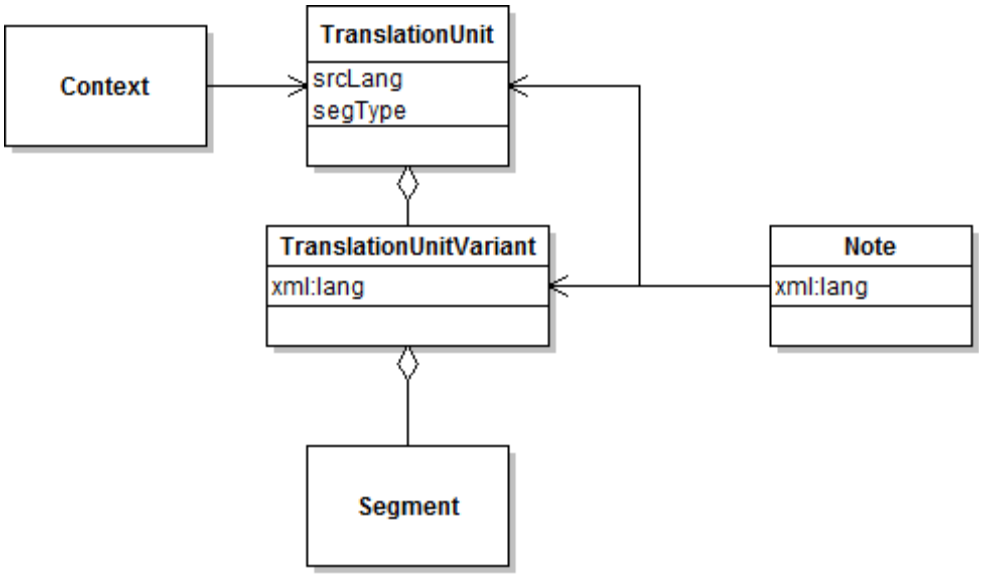


Figure 3.4. Structure of TMX fragment

[8] http://www.lisa.org/Translation-Memory-e.34.0.html

3.3.4 XLIFF

The purpose of the XLIFF[9] (XML Localization Interchange File Format) is to store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools.

XLIFF should be able to mark-up and capture localization information and interoperate with different processes and phases without loss of information. It should fulfil specific requirements of being tool-neutral. It should support the localization related aspects of internationalization and entire localization process. It also needs to support common software and content data formats. This should also provide an extensibility mechanism to allow the development of tools compatible with an implementer's proprietary data formats and workflow requirements.

1. **TransUnit**: contains (a set of) translational equivalences
2. **Source**: the source of the translation pair
3. **SegSource**: the translatable text, divided into segments
4. **Mrk**: Each segment is marked by means of the <Mrk> element with attribute **mType** set to the value "seg".
5. **Target**: the target of the translation pair; the attribute **Equiv-trans** indicates if the target language translation is a direct equivalent of the source text.
6. **Alt-Trans:** possible translations as Target instances

Figure 3.5 describes a selection from the XLIFF structural elements section in diagrammatical form.

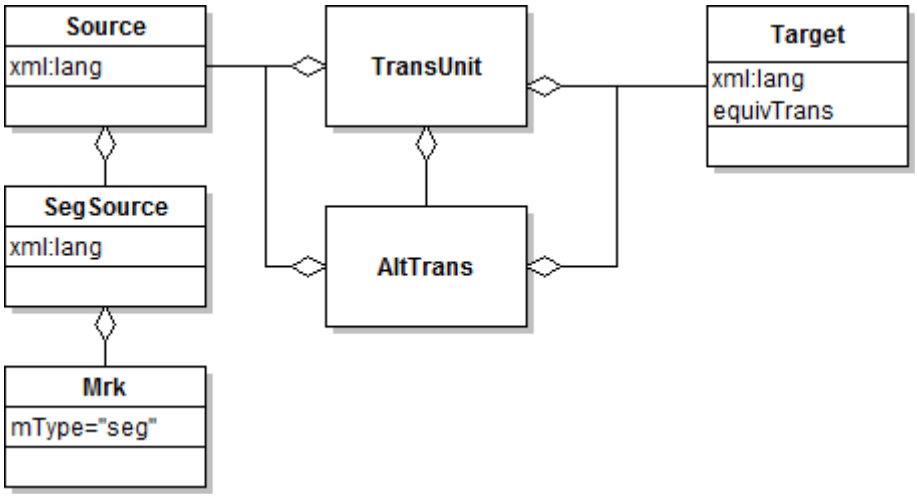


Figure 3.5. Structure of XLIFF fragment

3.3.5 MLIF

The Multi Lingual Information Framework (MLIF) [Cruz-Lara et al., 2004] [ISO 2006] introduces a metamodel for ensuring interoperability between several multilingual applications and corpora. MLIF promotes the use of a common framework for the future development of several different formats: TMX, XLIFF, etc. MILF can be considered as a parent for all these formats, since all of them deal with multilingual data expressed in the form of segments or text units. They all can be stored, manipulated and translated in a similar way.

According to the latest specifications, the MLIF core model has the following elements:

1. **MultiC** (Multilingual Component): groups together all variants of a given textual content.
2. **MonoC** (Monolingual Component): part of a multilingual component, containing information related to one language. Its attributes are the following: **languageIdentifier**

[9] http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html

contains an ISO639 code; **translationRole** determines whether the encompassing MonoC component corresponds to a source language or a target language in a translation process.
3. **SegC** (Segmentation Component): a recursive component allowing any level of segmentation for textual information. It has the following attributes: **segment** contains the segment string; **pos** denotes part of speech and **lemma** contains the citation/canonical form of the segment.

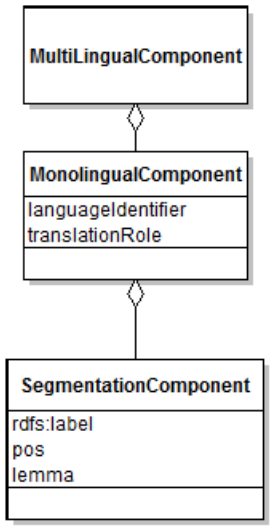


Figure 3.6. Structure of core MLIF fragment

3.4 Lexicon data models

As we mentioned in D222 [Villazón-Terrazas et al., 2008] there are different ways of representing the knowledge encoded by a particular resource. After we study several lexicons, we have identified the same data models we identify for thesauri. In this section we present these data models, which are independent of the standards described in section 3.2. In order to exemplify the data models for lexicons, we use an excerpt of WordNet presented in Figure 3.7.

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Noun

- {09411430} S: (n) **river** (a large natural stream of water (larger than a creek)) *"the river was navigable for 50 miles"*
  - ***part meronym***
    - {09274500} S: (n) estuary (the wide part of a river where it nears the sea; fresh and salt water mix)
    - {09405396} S: (n) rapid (a part of a river where the current is very fast)
    - {09475292} S: (n) waterfall, falls (a steep descent of the water of a river)
  - *domain term category*
  - *has instance*
  - *direct hypernym* / *inherited hypernym* / *sister term*
    - {09448361} S: (n) stream, watercourse (a natural body of running water flowing on or under the earth)
  - *part holonym*
    - {09476011} S: (n) water system (a river and all of its tributaries)

WordNet home page

Figure 3.7. Excerpt of WordNet

3.4.1 Record-based model

The record-based model, which is a denormalized structure, uses a record for every element of the lexicon with the information about the element, such as synonyms, antonyms, hypernyms, hyponym, etc. In this model, the information is stored in large packages, and to access or change any piece of information we must get into the appropriate package.

Word	Gloss	POS	Part Meronym	Part Holonym	Hypernym	Hyponym	…
river	a large natural stream of water (larger than a creek); "the river was navigable for 50 miles"	N	estuary rapid waterfall	water system	stream		
	….						

Figure 3.8. Record-based model

3.4.2 Relation-based model

The 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, but they are simply data values in a relationship record, thus new relationship types can be introduced with ease. In this case, Figure 3.9, there are three entities: (1) an element entity, which contains the overall set of lexicon elements, (2) an element-element relationship entity, in which each record contains two different element codes and the relationship between them, and (3) a relationship source entity, which contains the overall lexicon relationships.

Synsetid	Word	POS	Gloss	…
108614198	river	n	a large natural stream of water (larger than a creek); "the river was navigable for 50 miles"	…
108814882	rapid	n	a part of a river where the current is very fast	…
108696219	stuary	n	the wide part of a river where it nears the sea; fresh and salt water mix	…
108854154	stream	n	a natural body of running water flowing on or under the earth	…
…	…	…	…	…

Synset1id	Synset2id	Linkid
108614198	108696219	11
108614198	108854154	1
…	…	…

Linkid	Link
1	hypernym
11	part holonym
12	part meronym
…	…

Figure 3.9. Relation-based model

3.5 Lexicon Implementations

Finally these data models can be implemented as any of the identified types on D222, i.e. databases, XML files, flat files, and spreadsheets. A direct implementation would be as tables in a relational database or in a spreadsheet. Figure 3.10 presents a database implementation of the relation-based model of a WordNet, specifically the linktype table.

linkid	link	recurses
92	domain member categor	0
1	hypernym	1
4	instance hyponym	1
80	pertainym	0
50	also	0
93	domain region	0
30	antonym	0
40	similar	0
70	verb group	0
21	entail	1
15	substance holonym	1
14	member meronym	1
81	derivation	0
11	part holonym	1
95	domain usage	0
12	part meronym	1
98	member	0
13	member holonym	1
94	domain member region	0
16	substance meronym	1
97	domain	0
2	hyponym	1
3	instance hypernym	1
96	domain member usage	0
23	cause	1
60	attribute	0
71	participle	0
91	domain category	0

Figure 3.10 Excerpt of a WordNet database implementation

3.6 PR-NOR Library

In this section we present the re-engineering patterns (PR-NOR) for re-engineering lexicons into ontologies. We include the patterns for the TBox and ABox transformations[10].

- Patterns for the TBox transformations
  - PR-NOR-LXLO-01. Pattern for re-engineering a wordnet lexicon, created with the WordNet-LMF standard and modelled with the record-based data model, into an ontology schema.
  - PR-NOR-LXLO-02. Pattern for re-engineering a wordnet lexicon, created with the WordNet-LMF standard and modelled with the relation-based data model, into an ontology schema.
- Patterns for the ABox transformations
  - PR-NOR-LXLO-10. Pattern for re-engineering a wordnet lexicon, created with the WordNet-LMF standard and modelled with the record-based data model, into an ontology
  - PR-NOR-LXLO-10. Pattern for re-engineering a wordnet lexicon, created with the WordNet-LMF standard and modelled with the relation-based data model, into an ontology.

We will include these patterns in the NeOn library of patterns[11]. In this deliverable we present as an example the PR-NOR-LXLO-01 pattern.

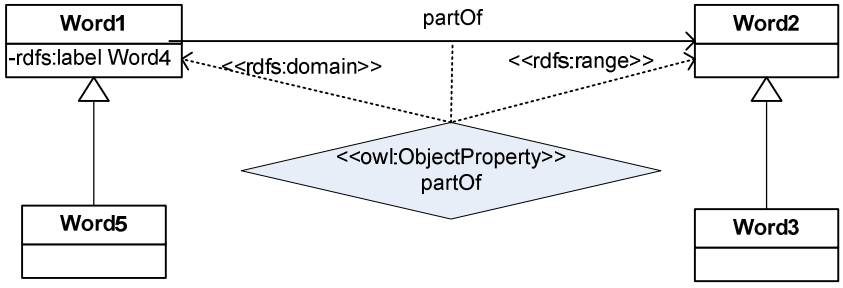
[10] The transformation approaches are described in D224. Final version of methods for re-engineering and evaluation.

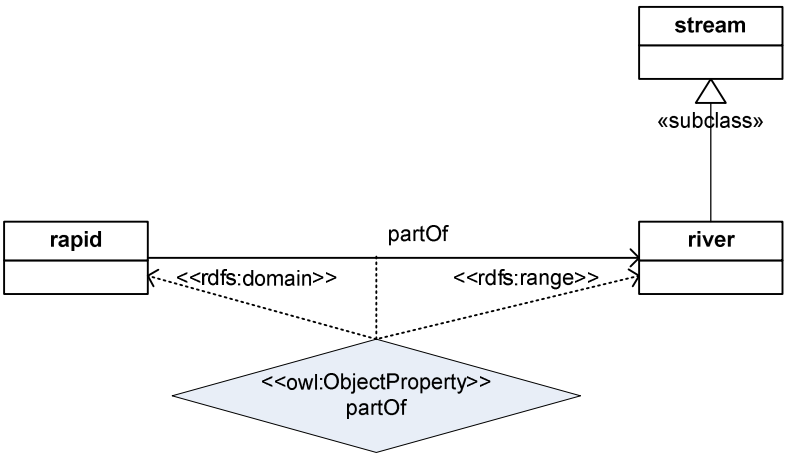
[11] http://www.ontologydesignpatterns.org

PR-NOR-LXLO-01. Pattern for re-engineering a wordnet lexicon, created with the WordNet-LMF standard and modelled with the record-based data model, into an ontology schema.

<table>
<tr><th>Slot</th><th>Value</th></tr>
<tr><td colspan="2">General Information</td></tr>
<tr><td>Name</td><td>Pattern for re-engineering a wordnet lexicon, which follows the WordNet-LMF, into an ontology schema</td></tr>
<tr><td>Identifier</td><td>PR-NOR-LXLO-01</td></tr>
<tr><td>Type of Component</td><td>Pattern for Reengineering Non Ontological Resource (PR-NOR)</td></tr>
<tr><td colspan="2">Use Case</td></tr>
<tr><td>General</td><td>Re-engineering a wordnet lexicon which follows the WordNet-LMF, into an ontology schema</td></tr>
<tr><td>Example</td><td>Suppose that someone wants to build an ontology based on the Princeton WordNet[12]. The Princeton WordNet follows the WordNet-LMF.</td></tr>
<tr><td colspan="2">Pattern for Re-engineering Non-Ontological Resource</td></tr>
<tr><td colspan="2">INPUT: Resource to be Re-engineered</td></tr>
<tr><td>General</td><td>A non-ontological resource holds a wordnet lexicon which follows the WordNet-LMF.
A lexicon is a list of words in a language along with some knowledge of how to use each word.
WordNet-LMF [Soria et al., 2009] is a dialect of ISO Lexical Markup Framework that instantiates LMF for representing wordnets.</td></tr>
<tr><td>Example</td><td>The Princeton WordNet is the best known computational lexicon of English. This lexicon is available at: http://wordnet.princeton.edu/</td></tr>
<tr><td colspan="2">Graphical Representation</td></tr>
<tr><td>General</td><td>
<table>
<tr><th>Synsetid</th><th>Word</th><th>POS</th><th>Gloss</th><th>…</th></tr>
<tr><td>1</td><td>word1</td><td>pos1</td><td>gloss1</td><td>…</td></tr>
<tr><td>2</td><td>word2</td><td>pos2</td><td>gloss2</td><td>…</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
</table>
<table>
<tr><th>Synset1id</th><th>Synset2id</th><th>Linkid</th></tr>
<tr><td>1</td><td>2</td><td>11</td></tr>
<tr><td>1</td><td>3</td><td>1</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
</table>
<table>
<tr><th>Linkid</th><th>Link</th></tr>
<tr><td>1</td><td>description1</td></tr>
<tr><td>2</td><td>description2</td></tr>
<tr><td>3</td><td>description3</td></tr>
<tr><td>…</td><td>…</td></tr>
</table>
</td></tr>
<tr><td>Example</td><td></td></tr>
</table>

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

	<table><tr><th>Synsetid</th><th>Word</th><th>POS</th><th>Gloss</th><th>…</th></tr><tr><td>108614198</td><td>river</td><td>n</td><td>a large natural stream of water (larger than a creek); "the river was navigable for 50 miles"</td><td>…</td></tr><tr><td>108814882</td><td>rapid</td><td>n</td><td>a part of a river where the current is very fast</td><td>…</td></tr><tr><td>108696219</td><td>stuary</td><td>n</td><td>the wide part of a river where it nears the sea; fresh and salt water mix</td><td>…</td></tr><tr><td>108854154</td><td>stream</td><td>n</td><td>a natural body of running water flowing on or under the earth</td><td>…</td></tr><tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr></table><table><tr><th>Synset1id</th><th>Synset2id</th><th>Linkid</th></tr><tr><td>108614198</td><td>108696219</td><td>11</td></tr><tr><td>108614198</td><td>108854154</td><td>1</td></tr><tr><td>…</td><td>…</td><td>…</td></tr></table><table><tr><th>Linkid</th><th>Link</th></tr><tr><td>1</td><td>hypernym</td></tr><tr><td>11</td><td>part holonym</td></tr><tr><td>12</td><td>part meronym</td></tr><tr><td>…</td><td>…</td></tr></table>
OUTPUT: Designed Ontology	
General	The generated ontology will be based on the lightweight ontology architectural pattern (AP-LW-01) [Suarez-Figueroa et al., 2007]. Each WordNet synset is mapped to a class. The hyponymy/hypernym relations are mapped to *subClassOf/superClassOf* relations. The member meronym/holonym relations are mapped to *partOf/hasPart*. For *Synonyms* we use the logical pattern proposed by Corcho et al. [Corcho et al., 2009] suggested as best practice in the context of this antipattern: the tendency to declare two classes equivalent when in fact their labels simply express synonymy.
Graphical Representation	
(UML) General Solution Ontology	
(UML) Example Solution Ontology	

	
PROCESS: How to Re-engineer	
General	1. For all the synsets, e_i, in the element entity of the lexicon 1.1. Create a class *CE* for the synset e_i. 1.2. For each synset *Y* which is a hyponym of e_i 1.2.1. Create a class *CY* and set the *subClassOf* relation between *CY* and *CE.* 1.3. For each synset *Z* which is a hypernym of e_i 1.3.1. Create a class *CZ* and set the *superClassOf* relation between *CZ* and *CE.* 1.4. For each synset *P* which is a member meronym of e_i 1.4.1. Create a class *CP* and set the *partOf* relation between *CP* and *CE.* 1.5. For each synset *Q* which is a member holonym of e_i 1.5.1. Create a class *CQ* and set the *hasPart* relation between *CQ* and *CE.* .
Example	1. Create the `river` class. 2. Create the `stream` class and assert that `stream` is *subClassOf* `river`. 3. Create the `rapid` class and assert that `rapid` is *partOf* `river`.
Relationships	
Relations to other modeling components	Use the AP: TX-AP-01 [Suarez-Figueroa et al., 2007].

4. Method for re-engineering

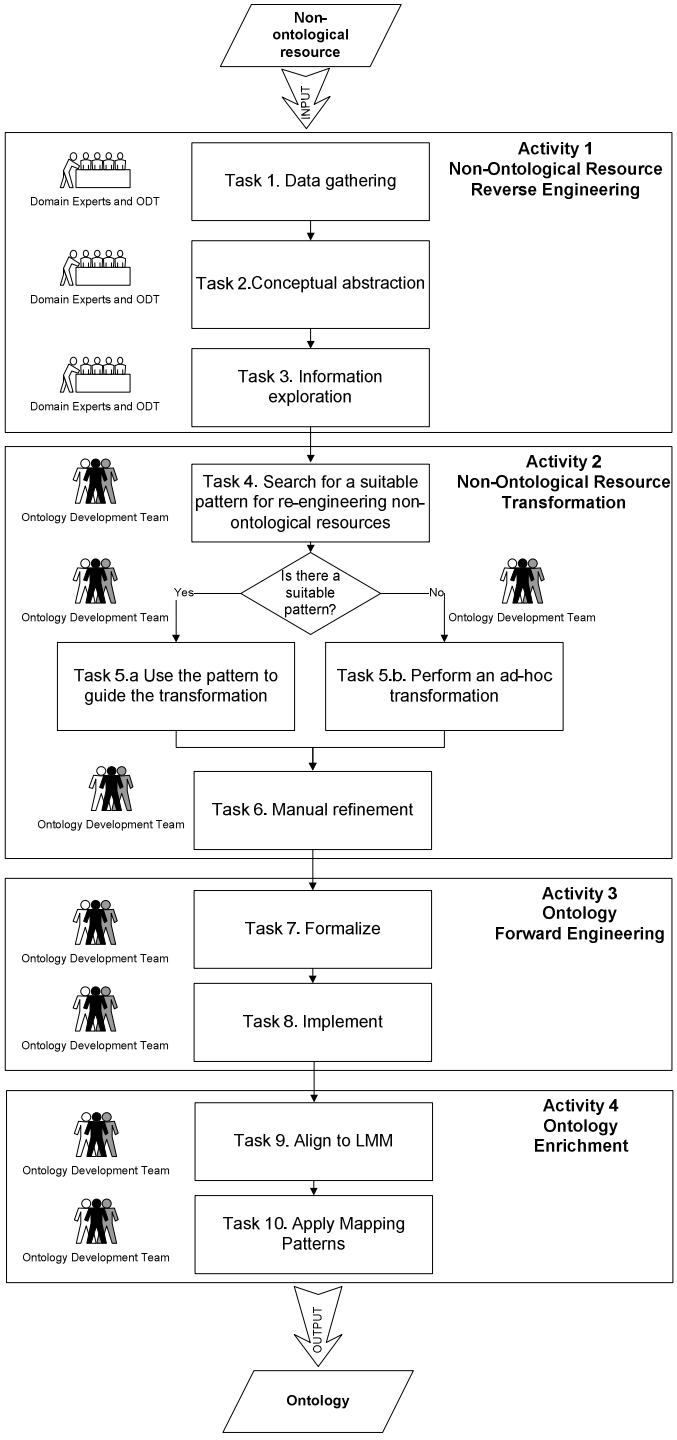


Figure 4.1 Method for Re-engineering

In this section we present a method for re-engineering existing resources. This method is based on the Method for Re-engineering non-ontological resources into ontologies described in D222

[Villazón-Terrazas et al., 2008], and D224 [Angeletou et al., 2009]. Figure 4.1 depicts the activities and tasks of the method.

In this deliverable we just describe briefly activities 1, 2, and 3, and we present a detailed description of activity 4.

4.1. Activity 1. Non-Ontological Resource Reverse Engineering.

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

4.1.1. Task 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.

4.1.2. Task 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 by using a data modelling tool.

4.1.3. Task 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 by using a data modelling tool.

4.2. Activity 2. Non-Ontological Resource Transformation.

The goal is 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.

4.2.1. Task 4. Search for a suitable pattern for re-engineering non-ontological resources.

The goal of this task is to find out if there is any applicable re-engineering pattern to transform the non-ontological resource into a conceptual model. To search for a suitable pattern for re-engineering non-ontological resource the NeOn library of patterns[13] can be used. First, the non-ontological resource type has to be identified. Second, the internal data model of the non-ontological resource has to be identified as well. Third, the transformation approach has to be selected.

4.2.2. Task 5.a Use the pattern to guide the transformation.

[13] http://ontologydesignpatterns.org

The goal of this task is to apply the re-engineering pattern obtained in task 4 to transform the non-ontological resource into a conceptual model. If a suitable pattern for re-engineering non-ontological resource is found then the conceptual model is created from the non-ontological resource following the procedure established in the pattern for re-engineering. We have developed a software library, that implements the transformations suggested by the patterns, and it is described in deliverable D2.5.2 Pattern based ontology design: methodology and software support.

4.2.3. Task 5.b Perform an ad-hoc transformation.

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

4.2.4. Task 6. Manual refinement.

The goal of this task is to check if some inconsistency is present after the transformation. Software developers and ontology practitioners with the support of domain experts can fix manually some generated inconsistencies after the transformation.

4.3. Activity 3. Ontology Forward Engineering.

The goal 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.

4.3.1. Task 7. Formalize.

The goal of this task is to transform the conceptual model obtained in task 5.a or 5.b into a formalized model, according to a knowledge representation paradigm as description logics, first order logic, etc.

4.3.2. Task 8. Implement.

The goal of this task is the ontology implementation in an ontology language.

4.4. Activity 4. Ontology Enrichment.

4.4.1. Task 9. Alignment to LMM, either through LingNet or ad hoc

Alignment through LingNet is possible because it contains direct mappings between LIR and LMM. Also, it contains alignments between LIR and other standard models for linguistic/terminological description, which have been selected as the basis for re-engineering patterns in section 3. This will, for the moment, at least enable partial alignment to LMM, and full alignment once LingNet will have been extended to cover all alignments between all elements from the models it captures.

4.4.2. Task 10. Apply Mapping Patterns

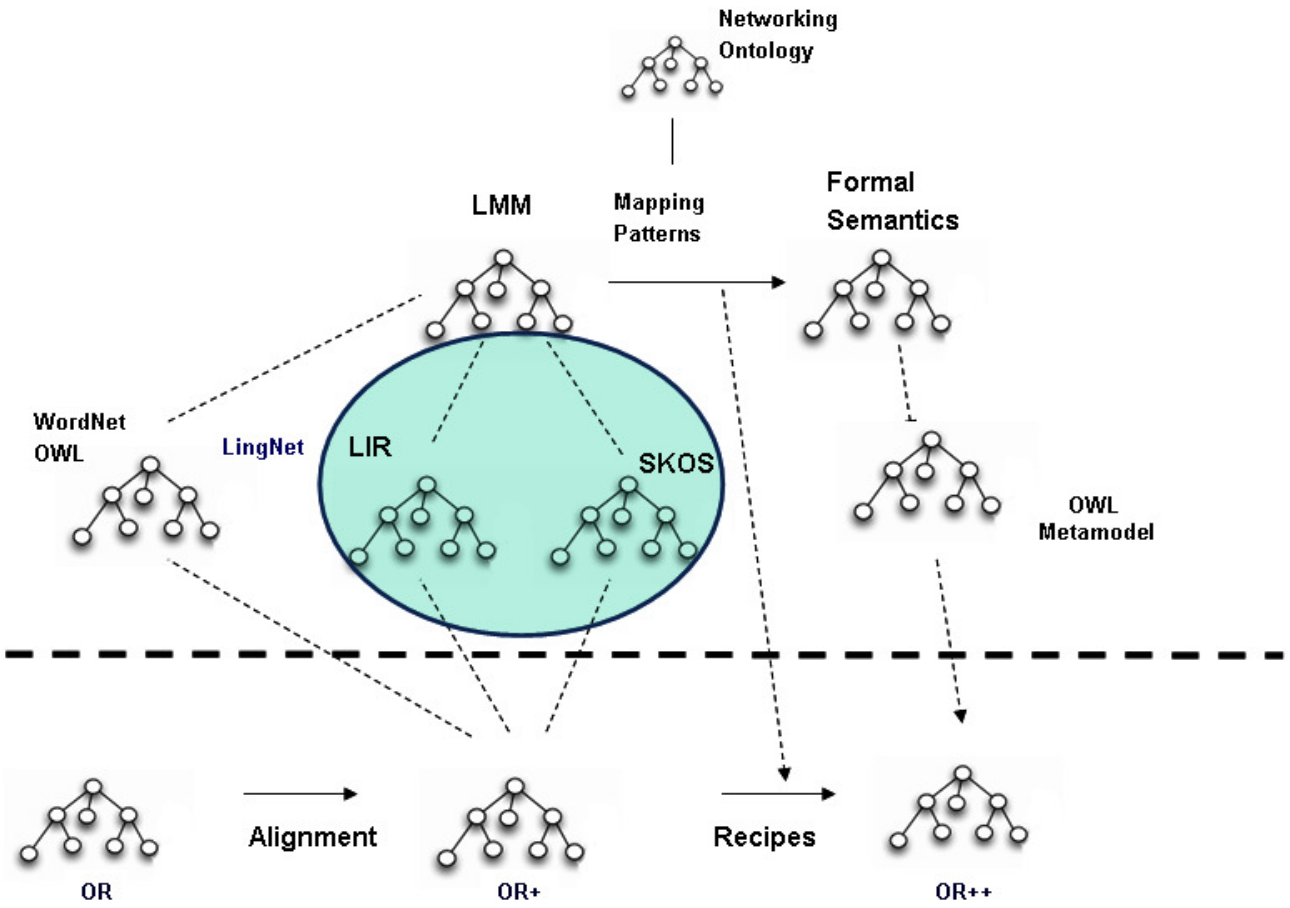


Figure 4.2 Ontology Enrichment

4.5 Use Case 1. ASFA Thesaurus

In this section we present the re-engineering process of the ASFA thesaurus[14] following the proposed guidelines of our method. Next we describe the steps we followed:

Activity 1. Non-Ontological Resource Reverse Engineering: We gathered all the data and documentation of the ASFA thesaurus. We have identified that it is a term-based theasaurus which follows the record-based model, and it is implemented in XML.

Activity 2. Non-Ontological Resource Transformation: Within this activity we searched for a suitable pattern for re-engineering non-ontological resources. The selected pattern was the PR-NOR-

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

TSLO-01. Pattern for re-engineering a term-based thesaurus, which follows the record-based data model, into an ontology schema.

Activity 3. Ontology Forward Engineering: This activity was carried out automatically by the PR-NOR software library. The resultant ontology is available at: http://droz.dia.fi.upm.es/ontologies/asfaskos.owl

Activity 4. Ontology Enrichment: the embedding of the label information of the resulting ontology into a network of standard terminological/linguistic description.

4.6 Use Case 2. WordNet

In this section we present the re-engineering process of WordNet[15] following the proposed guidelines of our method. We worked with a WordNet version implemented in MySQL[16]. Next we describe the steps we followed:

Activity 1. Non-Ontological Resource Reverse Engineering: We gathered all the data and documentation of WordNet. We have identified that it is a lexicon which follows the WordNet-LMF, and it is implemented in a MySQL database.

Activity 2. Non-Ontological Resource Transformation: Within this activity we searched for a suitable pattern for re-engineering non-ontological resources. The selected pattern was the PR-NOR-LXLO-01.Pattern for re-engineering a wordnet lexicon, which follows the WordNet-LMF, into an ontology schema.

Activity 3. Ontology Forward Engineering: This activity was carried out automatically by the PR-NOR software library.

Activity 4. Ontology Enrichment.

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

[16] http://www.androidtech.com/html/wordnet-mysql-20.php

5. Conclusion

In this deliverable, we have integrated various aspects of WP2 tasks T2.2 and T.2.4: re-engineering patterns and standardized descriptions of the linguistic/terminological content of resources.

The ontology network ontologies initiated in D2.4.3 unifies standard descriptions for linguistic information associated with lexicalizations of ontology concepts. It forms an enrichment of the re-engineering patterns for lexicons, translation memories and thesauri originating from D2.2.2.

Together they fully capture the conceptual and linguistic issues involved in the process of re-engineering the content of a non-ontological resource into a formal ontological representation.

Due to the fact that the deadline of this deliverable is month 48, not all phases have been fully described yet, nor have the use cases been worked out. For example, alternative patterns can be described for reengineering WordNets into OWL.

These will be our tasks in month 48.

References

[Angeletou et al., 2009] S. Angeletou, H. Lewen, and B. Villazón-Terrazas (2009). NeOn Deliverable D2.2.4 Final version of methods for re-engineering and evaluation. Technical report, NeOn.

[Antoni-Lay et al., 1994] M. Antoni-Lay, G. Francopoulo, and L. Zaysser (1994). A generic model for reusable lexicons: The GENELEX project. Literary and Linguistic Computing, 9(1):47–54.

[Brockmans et al.,2006] Brockmans, S., Haase, P.Stuckenschmidt, H. (2006) *Formalism-Independent Specification of Ontology Mappings - A Metamodeling Approach.* In: Meersman, R., Tari, Z. et al. (eds), OTM 2006 Conferences, Springer Verlag, Montpellier, France, October 2006.

[Buitelaar et al.,2009] Buitelaar, P., Cimiano, P., Haase, P. and Sintek, M. (2009), Towards Linguistically Grounded Ontologies, In: Proceedings of the 6th Annual European Semantic Web Conference (ESWC), 2009

[Calzolari et al., 1996] N. Calzolari, M. Naught, and J. Zampolli (1996). EAGLES Editor's Introduction. http://www.ilc.cnr.it/EAGLES96/edintro/edintro.html.

[Calzolari et al., 2003] N. Calzolari, F. Bertanga, A. Lenci, and M. Monachini (2003). Standards and best practice for Multilingual Computational Lexicons. MILE (The Multilingual ISLE Lexical Entry. Technical report, ISLE CLWG Deliverable D2.2 & 3.2.

[Cruz-Lara et al., 2004] S. Cruz-Lara, S. Gupta, & L. Romary (2004), Handling Multilingual content in digital media: The Multilingual Information Framework. Paper presented at the European Workshop on the Integration of Knowledge EWIMT 2004, Semantics and Digital Media Technology. London, UK.

[Francopoulo et al., 2006] G. Francopoulo, M. George, N. Calzolari, M. Monachini, N. Bel, M. Pet, and C. Soria (2006). Lexical Markup Framework (LMF). In Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy.

[Haase et al,2007] Haase, P., Brockmans, S., Palma, R., Euzenat, J., d'Aquin, M.(2007), NeOn Project deliverable D1.1.2: *Updated Version of the Networked Ontology Model*

[Hirst, 2004] G. Hirst (2004). Ontology and the lexicon. In Handbook on Ontologies in Information Systems, pages 209–230. Springer.

[Peters et al.,2009] Peters, W., Espinoza, M., Montiel-Ponsoda, E., Sini, M. (2009), NeOn Project deliverable D2.4.3: Multilingual and Localization Support for Ontologies (v3)

[Picca et al.,2008] Picca, D., Gangemi, A., and Gliozzo, A. (2008). LMM: an OWL Metamodel to Represent Heterogeneous Lexical Knowledge. In Proc. of the International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco. ACL.

[Scharffe et al., 2008] F. Scharffe, J. Euzenat, and D. Fensel (2008). *Towards design patterns for ontology alignment*. In R.L. Wainwright and H. Haddad (eds.): Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 2008: 2321-2325.

[Soria et al., 2009] Soria, C., Monachini, M., and Vossen, P. 2009. Wordnet-LMF: fleshing out a standardized format for wordnet interoperability. In *Proceeding of the 2009 international Workshop on intercultural Collaboration* (Palo Alto, California, USA, February 20 - 21, 2009). IWIC '09. ACM, New York, NY, 139-146. DOI= http://doi.acm.org/10.1145/1499224.1499246

[Villazón-Terrazas et al., 2008] B. Villazón-Terrazas, S. Angeletou, A. García-Silva, A. Gómez-Pérez, D. Maynard, M. C. Suárez-Figueroa, and W. Peters (2008). NeOn Deliverable D2.2.2 Methods and Tools for Supporting Re-engineering. Technical report, NeOn.