Universidad Politécnica de Madrid
E.T.S.I. Montes

Multilingualism in Ontologies

Multilingual Lexico-Syntactic Patterns for Ontology Modeling and Linguistic Information Repository for Ontology Localization

Doctoral Thesis

Elena Montiel Ponsoda

2011

Departamento de Lingüística Aplicada a la Ciencia y a la Tecnología
E.T.S.I. Montes

Multilingualism in Ontologies

Multilingual Lexico-Syntactic Patterns for Ontology Modeling and Linguistic Information Repository for Ontology Localization

Doctoral Thesis

Author : Elena Montiel Ponsoda
Advisors : Dr. Guadalupe Aguado de Cea
Dr. Asunción Gómez Pérez

2011

Tribunal nombrado por el Sr. Rector Magfco. de la Universidad Politécnica de Madrid, el día.....de.....de 20....

Presidente : _____

Vocal : _____

Vocal : _____

Vocal : _____

Secretario : _____

Suplente : _____

Suplente : _____

Realizado el acto de defensa y lectura de la Tesis el día.....de.....de 20..... en la E.T.S.I. /Facultad.....

Calificación .....

EL PRESIDENTE LOS VOCALES

EL SECRETARIO

Als meus pares

Acknowledgments

Como no podía ser de otra forma, los agradecimientos de esta tesis también son multilingües, pues así es la realidad de su autora.

En primer lugar quiero expresar mi reconocimiento a mi primera directora de tesis, Guadalupe Aguado de Cea, a quien considero mi *Doktormutter* o madre académica. Su excelente dirección académica, su disponibilidad y atención absolutas, así como su enorme calidad profesional y humana, han hecho de este trabajo un viaje emocionante. Mi más profundo agradecimiento también a mi segunda directora de tesis, Asunción Gómez Pérez, por haberme brindado la oportunidad de trabajar en su grupo y haber confiado en mis capacidades. Su claridad de ideas, su seguimiento y guía han sido decisivas en la investigación llevada a cabo en esta tesis. Gracias a las dos.

Debo hacer extensivo este agradecimiento a Ricardo Mairal Usón por su gustosa disponibilidad y por trasmitirme su entusiasmo por la investigación lingüística. A Inmaculada Álvarez de Mon por sus sabios consejos, su coherencia y rigor en hacer bien las cosas. Y a Óscar Corcho por su tiempo y sus valiosas sugerencias.

Quisiera también agradecer a mis compañeros del OEG su apoyo y su paciencia. Sin ellos el mundo de la Inteligencia Artificial seguiría siendo un enigma. Me gustaría hacer una mención especial a aquellos con los que he tenido la oportunidad y la suerte de colaborar durante los últimos años, a José Ángel, Mari Carmen, Mauricio, Boris, Raúl García, Andrés, Luis, María, Miguel Ángel y Jorge.

I am also greatly indebted to many colleagues and friends I have come across the way and from whom I have learned so many things. Here I would like to mention Thierry Declerck, Paul Buitelaar, Philipp Cimiano, Wim Peters and Diana Maynard. The stimulating discussions we had during my research visits helped me clarify my thoughts on the topics dealt in this work. Thank you for make me feel at home when I was abroad!

De la mateixa forma, voldria agrair a la meua família i amics en Alacant i Astúries el seu suport i ànims constants, especialment a Ángel i Carmen, a Marcos i Eva, a Virgi, a Maribel, a Jaume, i a Rubén i Patri. Al meu germà, Joan Jesús,

per què sense ell saber-ho i malgrat la distància, m'ha tramés el seu coratge, determinació i esperit de superació. Gracies xiquet! Als meus iaios, Jesús i Francisca, per haver-me ensenyat a lluitar i a seguir endavant malgrat les dificultats. Als meus pares, Joan Vicent i Àngela, per estar sempre al meu costat i haver-me acompanyat en totes les aventures en què m'he embarcat. Sense el seu amor i suport aquest treball no haguera estat possible.

Letztenendlich möchte ich mich bei Dimas bedanken (ich kann mir keinen besseren Gefährten für das Leben vorstellen!). Danke für die Liebe, Unterstützung und ausdauernde Geduld. So ein Glück, dass wir uns gefunden habe. Zusammen können wir alles schaffen.

Finalmente, quisiera mencionar expresamente que el trabajo contenido en esta tesis doctoral ha sido co-financiado por una beca oficial de la Universidad Politécnica de Madrid (convocatoria 2006) y por la Comisión Europea en el contexto de los proyectos NeOn (FP6-027595) y Monnet (FP7-248458), y por el Ministerio de Ciencia y Tecnología a través del proyecto GeoBuddies (TSI-2007-65677-C02-01).

Madrid, noviembre de 2010.

Abstract

The objective of this PhD thesis is to face some of the problems that arise from the interaction between ontologies and natural language in a multilingual context. In particular, our work focuses on two activities of the ontology development process, namely, knowledge acquisition for ontology modeling from natural language expressions, on the one hand, and localization of ontologies to different natural languages, on the other hand. This work can be understood as a twofold process in which, in the first phase, linguistic expressions are transformed into ontological constructs, and, in the second phase, ontological constructs are associated to linguistic information in multiple languages. Along this process, we take into account multilingualism at both ends: the starting point and at the final result. Both approaches aim at bringing ontologies closer to average users coming from different linguistic and cultural communities, being this a fundamental requirement for the consolidation of the Semantic Web.

The two approaches presented here are based on our conviction that language forms an integral part of human cognition, of our understanding and categorization of reality. This is indeed one of the basic tenets of the functional-cognitive tradition. Taking this assumption into account, the first contribution of this PhD relies on an analysis of the deep semantics of users' formulations in the ontology development process. Such an analysis allows us to establish a correspondence to the ontological constructs that better capture the semantics of users' expressions. Target users in this case are newcomers to ontological engineering. For this aim, we propose a repository of linguistic patterns associated to a specific type of ontological constructs, called Ontology Design Patterns, as well as methodological guidelines to guide users in the activities of knowledge acquisition and ontology modeling.

As for the second contribution of this work, we have designed a model of linguistic descriptions that is to be associated to ontologies in order to enrich them with multilingual information. The purpose of this model is to make the same conceptualization reusable in different linguistic and cultural settings. This research work also relies on functional-cognitive theories, specifically on experientialism, to face some of the issues regarding the validity of the same categorization of reality in different cultural settings. In this sense, we believe that ontologies represent interpretations of the extralinguistic world that reflect how certain groups of people perceive reality. Accordingly, ontologies may capture categorizations that are valid

and shared by several groups of users or others that present some disparities. This fact has been taken into account when proposing a model for the localization of ontologies.

The principal contributions of this work are summarized in the following:

1. We have created a repository of linguistic patterns in English and Spanish that are associated with ontological representations, considered good practices in ontology modeling, namely, the so-called Ontology Design Patterns. With the aim of establishing a reliable correspondence between linguistic patterns and Ontology Design Patterns, we have performed an analysis of the deep semantics of those linguistic structures characterized by a polysemous behavior.

2. We have defined a method to guide novice users in the formulation of linguistic expressions that are subsequently modeled in ontologies making use of Ontology Design Patterns.

3. We have provided an analysis of the dimensions involved in the ontology localization process, and devise some of the strategies to be followed according to the dimensions involved. We have also analyzed extant formalisms and models for the representation of multilingualism in ontologies.

4. We have designed a model of lexical and terminological descriptions that associated to ontologies allows for the representation of cultural mismatches, and the establishment of well-defined relations within descriptions both in the same language and across languages.

The validity of both approaches has been supported by a set of experiments relying on suitable test cases. Experimental results reveal the feasibility of the proposed approaches, models and techniques.

Resumen

El objetivo de esta tesis es abordar algunos de los problemas que surgen de la interacción entre las ontologias y el lenguaje natural en un contexto multilingüe. En concreto, nuestro trabajo se centra en las actividades de adquisición de conocimiento para el modelado de ontologías a partir de expresiones en lenguaje natural, y en la localización de ontologías a diversas lenguas. En este sentido, podemos entender este trabajo como un doble proceso en el que el punto de partida son expresiones lingüísticas que se transforman en representaciones ontológicas, y representaciones ontológicas a las que asociamos información lingüística. Todo ello teniendo en cuenta el multilingüismo en el punto de partida y en el resultado final. Ambas aproximaciones tienen como propósito acercar las ontologías a los usuarios provenientes de comunidades lingüísticas y culturales diversas, requisito fundamental para el progreso y consolidación de la Web Semántica.

Las dos vertientes que aquí presentamos se basan en nuestra convicción de que la capacidad lingüística es un elemento clave para la comprensión y categorización de la realidad, siendo éste uno de los principios básicos de las teorías cognitivo-funcionales. Tomando estas asunciones como punto de partida, la primera contribución de esta tesis se apoya en el análisis semántico de las oraciones producidas por un usuario en el proceso de desarrollo de una ontología. Dicho análisis semántico nos permite establecer una correspondencia con la estructura ontológica que mejor reproduce la intención del usuario. Nuestro planteamiento está pensado para usuarios con un bajo nivel de conocimiento en ingeniería ontológica. Con ese fin, proponemos un repositorio de patrones lingüísticos asociados a patrones de diseño ontológico, así como unas guías metodológicas. De esta forma proporcionamos el soporte necesario para las actividades de adquisición de conocimiento y modelado de ontologías de forma transparente para el usuario no experto.

En cuanto a la segunda contribución de esta tesis doctoral, hemos diseñado un modelo que, asociado a una ontología, permite describir la conceptualización representada en la ontología en múltiples lenguas. De esta manera se consigue que una misma conceptualización pueda ser utilizada en diversos contextos lingüísticos y culturales. Para esta investigación también nos hemos apoyado en las teorías cognitivo-funcionales, en particular en la concepción experiencialista, para abordar la cuestión de la validez de una misma categorización de la realidad en distintos contextos culturales. En este sentido, consideramos que las ontologías representan interpretaciones del mundo extralingüístico realizadas por distintos grupos de per-

sonas que reflejan una forma de entender o interpretar el mundo. Dichas ontologías pueden representar realidades compartidas u otras que no lo son tanto. Esto se ha tenido en cuenta a la hora de proponer un modelo para la localización de ontologías.

Las principales contribuciones de este trabajo se resumen como sigue:

1. Hemos creado un repositorio de patrones lingüísticos en inglés y español asociados a representaciones ontológicas consideradas "buenas prácticas" en el modelado de ontologías, a saber, los patrones de diseño ontológico. Hemos llevado a cabo un análisis profundo de la semántica de aquellos patrones lingüísticos que presentan usos polisémicos, para un correcto establecimiento de las correspondencias entre los patrones lingüíticos y los patrones de diseño.

2. Hemos definido un método para guiar a usuarios no expertos en la tarea de formulación de expresiones lingüísticas para su consecuente modelado en una ontología, haciendo uso de los patrones de diseño ontológico.

3. Hemos proporcionado un análisis de las dimensiones que intervienen en el proceso de localización de ontologías, así como de las distintas estrategias de traducción a seguir en cada caso. Del mismo modo, hemos analizado las diferentes modalidades de representación de descripciones multilingües en ontologías de acuerdo con los formalismos de representación existentes.

4. Hemos diseñado un modelo de descripciones léxicas y terminológicas, que asociado a ontologías, permite la representación de discrepancias culturales, así como de relaciones entre descripciones en un misma lengua y entre distintas lenguas.

La validez de ambas aproximaciones ha sido respaldada por una serie de experimentos realizados utilizando casos de prueba adecuados. Los resultados experimentales apuntan a la viabilidad de los enfoques, los modelos y las técnicas propuestas.

Contents

1 Introduction **1**
- 1.1 Thesis Context . . . 3
  - 1.1.1 Ontologies as Underpinnings of the Semantic Web . . . 6
  - 1.1.2 Multilingualism in the Semantic Web . . . 8
- 1.2 Goals and Contributions . . . 10
- 1.3 Methodology . . . 13
- 1.4 Structure of the Document . . . 14

2 Theoretical Framework **17**
- 2.1 Theoretical Assumptions . . . 26
- 2.2 The Lexical Constructional Model . . . 29
  - 2.2.1 Role and Reference Grammar . . . 35
  - 2.2.2 The Generative Lexicon . . . 37
  - 2.2.3 LCM Lexical Templates . . . 42
- 2.3 Summary . . . 45

I Multilingual Lexico-Syntactic Patterns for Ontology Modeling **47**

3 Knowledge Acquisition for Ontology Modeling **49**
- 3.1 Knowledge Acquisition from Text . . . 50
  - 3.1.1 Verb-centred Patterns for Knowledge Acquisition . . . 53
  - 3.1.2 Main Limitations of Pattern Approaches for Knowledge Acquisition from Text . . . 60
  - 3.1.3 Open Research Problems and Work Assumptions . . . 61
- 3.2 Knowledge Acquisition from Experts . . . 63
  - 3.2.1 Controlled Languages in Ontology Engineering . . . 64
  - 3.2.2 Main limitations of CLs in Knowledge Acquisition for Ontology Modeling . . . 68
  - 3.2.3 Open Research Problems and Work Assumptions . . . 70
- 3.3 Summary . . . 71

4 Ontology Design Patterns **73**
4.1 Design Patterns . . . 74
4.2 Design Patterns in Ontology Engineering . . . 76
4.2.1 Templates for ODPs . . . 78
4.2.2 ODPs Repositories . . . 78
4.2.3 ODPs Reuse Methods . . . 81
4.2.4 Tools for supporting ODPs Reuse . . . 84
4.3 NeOn Methodology as Framework for the Reuse of ODPs . . . 85
4.4 Open Research Problems and Work Assumptions . . . 89
4.5 Summary . . . 90

5 Multilingual LSPs-ODPs Pattern Repository **93**
5.1 Selection of Logical and Content ODPs . . . 94
5.2 Strategies for the Identification of *candidate verbal patterns* . . . 98
5.3 LSPs on the light of the Lexical-Constructional Model . . . 103
5.4 Multilingual LSPs-ODPs Pattern Repository . . . 121
5.4.1 English LSPs-ODPs Pattern Repository . . . 125
5.4.2 Spanish LSPs-ODPs Pattern Repository . . . 139
5.5 Summary . . . 148

6 ODPs Reuse Method for Novice Users **149**
6.1 Methodological Guides . . . 151
6.2 Example of Use . . . 153
6.3 Methodological and Technological Interaction . . . 155
6.4 Strategies for solving NL Ambiguities in LSPs . . . 156
6.5 Concluding Remarks . . . 160

7 LSPs Implementation and Evaluation **161**
7.1 LSPs Implementation in GATE . . . 162
7.2 LSPs Publication in the ODPs Portal . . . 170
7.3 Evaluation . . . 173
7.3.1 Experiment Setting . . . 173
7.3.2 Analysis of Results . . . 175
7.3.3 Concluding Remarks . . . 182
7.4 Summary . . . 183

II Linguistic Information Repository for Ontology Localization **185**

8 Ontology Localization **187**
8.1 Ontology Localization: Definition and Baselines . . . 188
8.2 Translation Theories in Ontology Localization . . . 189
8.3 Dimensions in Ontology Localization: Function and Domain Type 191
8.4 Characterization of the Localization Problem in Ontologies . . . 194

8.5 Ontology Layers involved in the Localization Activity 197
8.6 Translation Strategies in Ontology Localization 199
8.7 Summary 201

9 Modeling Multilingualism in Ontologies 203
9.1 Including Multilingual Labels in the Ontology 204
9.2 Combining the Ontology with a Mapping Model 206
9.3 Associating the Ontology with an External Linguistic Model . . . 208
9.4 Open Research Problems and Work Assumptions 211

10 Requirements for an Ontology Localization Model 215
10.1 Resource Interoperability 216
10.2 Localization Requirements 222
10.3 Accessibility Requirements 228
10.4 Summary 229

11 Linguistic Information Repository: a Model for Ontology Localization 231
11.1 Description of the LIR Model 233
11.2 LIR Technological Support 244
11.3 LexOMV: Multilingualism at the Metadata Level 247
11.3.1 Closing the circle: multilingualism at data, knowledge representation and metadata levels 251
11.4 Summary 253

12 LIR Validation 255
12.1 Compliance of the LIR against FAO Requirements 255
12.2 Comparison of the LIR against the RDF(S) and OWL Modeling Option 261
12.3 Summary 266

13 Conclusions and Future Research Lines 269
13.1 Main Contributions 269
13.1.1 Multilingual LSPs-ODPs Pattern Repository 270
13.1.2 Method for the Reuse of ODPs 270
13.1.3 Ontology Localization 271
13.1.4 LIR Model 271
13.2 Evaluation Results 272
13.2.1 Method for the Reuse of ODPs 273
13.2.2 Multilingual LSPs-ODPs Pattern Repository 274
13.2.3 Ontology Localization 275
13.2.4 LIR Model 276
13.3 Future Lines of Work 276
References 281

Appendix 297

List of Figures

1.1 Interaction between natural languages and ontologies in the semantic web . . . 3
1.2 Simplified representation of an ontology of cartoon animals . . . 5

2.1 Ogden and Richards' semiotic triangle . . . 19
2.2 Redefinition of the semiotic triangle . . . 21
2.3 Levels of the Lexical Constructional Model . . . 33
2.4 RRG linking algorithm . . . 35

3.1 Hearst's patterns . . . 52
3.2 Verbal patterns in French included in CAMÉLÉON . . . 54
3.3 Feliu and Cabré's verbal patterns in Catalan . . . 56
3.4 Sierra et al.'s definitional patterns in Spanish . . . 56
3.5 Aguado de Cea and Álvarez de Mon's classification patterns in Spanish . . . 57
3.6 Soler and Alcina's meronymy verbal patterns in Spanish . . . 57
3.7 Cimiano and Wenderoth's verbal patterns for *qualia* in English . . . 58
3.8 Sánchez and Moreno's *ad-hoc* verbal patterns in English . . . 58
3.9 Summary of knowledge acquisition approaches . . . 59
3.10 Summary of CLs for building ontologies . . . 68

4.1 Template describing the *logical pattern for disjoint classes* . . . 79
4.2 Ontology Design Patterns Portal screenshot . . . 80
4.3 *Agent role pattern* from the Ontology Design Patterns Portal . . . 81
4.4 *Defined class pattern* from GENE ONTOLOY project . . . 81
4.5 NeOn Methodology scenarios for building ontology networks . . . 86
4.6 Examples of CQs in the e-employment domain (SEEMP project) . . . 88

5.1 Steps in the development of the multilingual LSPs-ODPs pattern repository . . . 94
5.2 Examples of conceptually related terms in nearby context . . . 99
5.3 Search for "classified" concordances in the BNC . . . 100
5.4 Summarizing table: from ODPs to the English LSPs-ODPs pattern repository . . . 101
5.5 Summarizing table of LSPs-ODPs correspondences . . . 122

6.1 Filling card for the ODPs reuse activity aimed at novice users . . . 151
6.2 Method for the reuse of ODPs aimed at novice users 153
6.3 Overview of the proposed approach for the reuse of ODPs 156
6.4 Dependencies between ODPs: subclass-of relation, disjoint classes and exhaustive classes 159
6.5 Example of an instantiated UML diagram 160

7.1 GATE's main interface 163
7.2 Sequential order in the execution of the processing resources in GATE 164
7.3 Snapshot of a gazetteer in GATE 165
7.4 The two phases of a JAPE rule 166
7.5 Annotations generated from the JAPE rule SC1_1 169
7.6 LexicoSyntacticODPs repository at Ontology Design Patterns Portal 170
7.7 Description section at Ontology Design Patterns Portal 172
7.8 Cases section at Ontology Design Patterns Portal 173
7.9 Example of a wrong annotation provided by GATE's noun chunker 178
7.10 Annotations on the sentences provided by participant 7 180

8.1 Interaction between the activities and components dealt in this thesis 187
8.2 Example of near-equivalence relation between concepts 196
8.3 Example of subsumption relation among concepts 196
8.4 Example of many-to-many equivalence relation among concepts . 197

9.1 Multilingual labels included in the ontology 206
9.2 Binary mapping in a radial graph 207
9.3 Ontology associated with external linguistic model 209
9.4 Appropriateness of modeling option according to domain type and function 211

10.1 TMF structural representation 217
10.2 Multilingual term entries in TMF 218
10.3 RDF graph illustrating terminological and semantic relations in SKOS 219
10.4 Dependencies between the LMF core and extension packages . . . 220
10.5 LMF core package 221
10.6 Instantiation of sense axis and sense axis relation 222
10.7 Snapshot of the TermBase editor view in OntoTerm 225
10.8 Architecture of the GENOMA-KB implemented in OntoTerm . . 226
10.9 LingInfo model with multilingual instances 227
10.10 Main elements of the LexOnto lexicon model 228
10.11 Summary of requirements for an ontology localization model . . . 230

11.1 The LIR model 233
11.2 Link between ontological and lexical knowledge 240

11.3 LabelTranslator linguistic information entity properties view . . . 245
11.4 Instantiation of the LIR in LabelTranslator 247
11.5 OMV core v1.1 249
11.6 LexOMV 250
11.7 Ontology structure levels affected by multilingualism 252

12.1 Representation of acronyms and full forms within a language . . . 257
12.2 Representation of scientific names and common names across languages 258
12.3 Representation of conceptualization mismatches 260
12.4 Representation of non-native language expressions 260
12.5 Snapshot of the *hydrOntology* hierarchy and class annotation properties in Protégé 262
12.6 Linguistic information associated with *Río* in the LIR model . . . 264
12.7 Linguistic information associated with the lexical entry *Rivière* . . 266
12.8 Relations of synonymy and translation among labels 266

13.1 CQs about the olympic games used in LSPs experiment 298
13.2 Questionaire about the hands-on activity with ATHENS students . 299

List of Tables

2.1 NSM semantic primes 31
2.2 Lexical Functions and their meaning 32
2.3 RRG logical structures 36
2.4 Examples and instances of RRG logical structures 37
2.5 *Qualia* structure of the noun *novel* 41
2.6 Event, argument and *qualia* structures of the verb *build* 42
2.7 LCM lexical template for the verb *realize* 44
2.8 Lexical template proposed for the analysis of *candidate verbal patterns* 45

3.1 Keywords for symbols in Manchester Syntax 65
3.2 Examples of OWL ACE, Rabbit and Sydney OWL Syntax 67
3.3 Examples of CLOnE 67

5.1 Subset of Logical ODPs selected for the LSPs-ODPs pattern repository 95
5.2 Subset of Content ODPs selected for the LSPs-ODPs pattern repository 97
5.3 LCM lexical template 104
5.4 Lexical template for *be a(n)* 105
5.5 Lexical template for *be either... or...* 106
5.6 Lexical template for *classify* 107
5.7 Lexical template for *classify into* 109
5.8 Lexical template for *classify as* 111
5.9 Lexical template for *divide into* 112
5.10 Lexical template for *include* 116
5.11 Lexical template for *belong to* 117
5.12 Lexical template for *belong to the class of...* 118
5.13 Lexical template for *have (as part)* 118
5.14 Lexical template for *have (as property)* 119
5.15 Lexical template for *contain* 119
5.16 LSPs-ODPs pattern repository template 123
5.17 LSPs Symbols and Abbreviations 124
5.18 LSPs corresponding to *subclass-of relation* ODP 127

5.19 LSPs corresponding to *multiple inheritance* ODP 128
5.20 LSPs corresponding to *equivalence relation between classes* ODP 128
5.21 LSPs corresponding to *object property* ODP 129
5.22 LSPs corresponding to *datatype property* ODP 129
5.23 LSPs corresponding to *disjoint classes* ODP 130
5.24 LSPs corresponding to *specified values* ODP 130
5.25 LSPs corresponding to *participation* ODP 130
5.26 LSPs corresponding to *co-participation* ODP 131
5.27 LSPs corresponding to *location* ODP 131
5.28 LSPs corresponding to *object-role* ODP 132
5.29 LSPs corresponding to *defined classes* and *subclass-of relation* ODPs 133
5.30 LSPs corresponding to *subclass-of relation*, *disjoint classes* and *exhaustive classes* ODPs 134
5.31 LSPs corresponding to *object property* and *universal restriction* ODPs 135
5.32 LSPs corresponding to *subclass-of relation*, or *simple part-whole relation* ODPs 136
5.33 LSPs corresponding to *object property* or *datatype property* or *simple part-whole relation* ODPs 137
5.34 LSPs corresponding to *simple part-whole relation* or *constituency* or *componency* or *collection-entity* ODPs 138
5.35 LSPs corresponding to *subclassOf relation* ODP 139
5.36 LSPs corresponding to *multiple inheritance* ODP 140
5.37 LSPs corresponding to *equivalence relation between classes* ODP 141
5.38 LSPs corresponding to *object property* OP 141
5.39 LSPs corresponding to *datatype property* ODP 141
5.40 LSPs corresponding to *disjoint classes* ODP 142
5.41 LSPs corresponding to *specified values* ODP 142
5.42 LSPs corresponding to *participation* ODP 142
5.43 LSPs corresponding to *co-participation* ODP 143
5.44 LSPs corresponding to *location* ODP 143
5.45 LSPs corresponding to *object-role* ODP 143
5.46 LSPs corresponding to *defined classes* and *subclass-of relation* ODPs 144
5.47 LSPs corresponding to *subclassOf relation, disjoint classes and exhaustive classes* ODPs 145
5.48 LSPs corresponding to *object property* and *universal restriction* ODPs 145
5.49 LSPs corresponding to *subclass-of relation*, or *simple part-whole relation* ODPs 146
5.50 LSPs corresponding to *object property* or *datatype property* or *simple part-whole relation* ODPs 147
5.51 LSPs corresponding to *simple part-whole relation* or *constituency* or *componency* or *collection-entity* ODPs 147

6.1 Recommendations for task 1. 152
6.2 Example of CQs of the Health Care Domain 154

7.1 Number of resulting annotations from the experiment with the LSPs application 176

8.1 Combination options between function and domain type 193

Abbreviations

BNC	British National Corpus
BNF	Backus-Naur Form
CL	Controlled Language
CQs	Competency Questions
CREA	Corpus de Referencia del Español Actual
DL	Description Logics
FAO	Food and Agriculture Organization
GT	Generative Lexicon
HTML	HyperText Markup Language
ISO	International Organization for Standardization
JAPE	Java Annotation Pattern Language
LCM	Lexical Constructional Model
LHS	Left-hand-side
LIR	Linguistic Information Repository
LMF	Lexical Markup Framework
LSP	Lexico-Syntactic Pattern
NE	Named Entity

NL	Natural Language
NLP	Natural Language Processing
NSM	Natural Semantic Metalanguage
ODP	Ontology Design Pattern
OMV	Ontology Metadata Vocabulary
ORSD	Ontology Requirements Specification Document
OWL	Web Ontology Language
POS	Part of Speech
RDF	Resource Description Framework
RDF(S)	Resource Description Framework Schema
RHS	Right-hand-side
RRG	Role and Reference Grammar
SKOS	Simple Knowledge Organization Systems
TMF	Terminological Markup Framework
UML	Unified Modeling Language
URI	Universal Resource Identifier
W3C	World Wide Web Consortium

Chapter 1

Introduction

The unavoidable symbiosis between ontologies and natural language has proven more and more relevant on the light of the growing interest and application of Semantic Web technologies. Ontologies that are well-documented in a natural language not only provide humans with a better understanding of the world model they represent, but also a better exploitation by the systems that may use them. If ontologies are models that aim at reproducing how humans intelligently organize knowledge in their minds, they will inevitable reflect the world as captured by a certain culture, and in its turn, by a certain language. It may be argued that the objective of ontologies is to represent an *a priori* nature of the world, avoiding any type of arbitrariness or partial view imposed by languages. However, what is finally decided to be included in an ontology does indeed capture what is of interest for a certain group of people, or what will better meet the purposes of a final application. Otherwise, the models represented in ontologies would find no matching or correspondences in the applications they have been thought for.

In this dissertation work we raise some issues regarding the relationship between ontologies and natural languages[1] in a multilingual scenario. Particularly, we focus on two stages of the ontology development process in which natural languages are determinant: 1) in the acquisition of knowledge for ontology modeling, and 2) in the localization of ontologies to different natural languages.

Knowledge acquisition for ontologies comprises several activities for capturing knowledge that is to be modeled in an ontology from a variety of sources, such as domain experts, background documentation or data bases (M. C. Suárez-Figueroa, 2010).

In this context, our approach for the acquisition of knowledge and its modeling in ontologies is grounded on the analysis of the semantics conveyed by linguistic structures, specifically verbal phrases. We argue that it is feasible to establish a correspondence between the semantics of recurrent linguistic expressions, which

[1] We will use the term *natural language* to refer to human language as opposed to *formal language* used in Computer Science.

we name Lexico-Syntactic Patterns, and its representation in an ontological model. In this way we move from a language-based representation of reality to a language-independent knowledge representation in ontologies. In the research conducted in this work we establish a correspondence between a subset of verbal structures and a specific type of ontological constructs called Ontology Design Patterns, which are regarded as consensual modeling components. The analysis is performed in two languages, English and Spanish, based on our conviction that for ontologies to become the foundations of the Semantic Web, support has to be provided to users from different linguistic communities. Moreover, in our approach for knowledge acquisition and ontology modeling, target users are newcomers to Ontological Engineering rather than experts.

The second activity in the ontology development process dealt in this thesis is the so-called Ontology Localization. This activity is defined in (M. C. Suárez-Figueroa, 2010) as "the adaptation of an ontology to a particular language and culture". Ontology Localization is an activity that takes place once an ontology has been modeled and is available for reuse. We understand that the ontology has been modeled by a certain community of experts and for specific purposes, and it is localized to satisfy the needs of a different community of users. In most cases, localization involves the translation of the lexical layer in the ontology from an original natural language into a target natural language. Notwithstanding, we do not rule out the possibility of an "intra-linguistic" localization to satisfy the needs of a certain community of users with an expertise level different from the one of the community that developed the ontology.

Our approach concerning the localization of ontologies relies on the assumption that the representation of knowledge in ontologies can be considered language independent, and can be reused for the purposes of a different linguistic community. In this sense, we make a distinction between the knowledge representation layer (the ontology) and the lexical layer. Nevertheless, in certain domains of knowledge, the language-independent representation captured in the ontology requires some adaptations in order to satisfy the requirements of the target community. This is the result of the different conceptualizations that linguistic and cultural communities make of the same knowledge parcel. We argue that some of these discrepancies may be captured in the lexical layer, whereas others will need some modifications of the knowledge representation layer. A functional analysis of the localization requirements will allow us to identify the most appropriate strategy in each localization process.

These two approaches, though dealing with different aspects of ontologies, aim at achieving a common purpose, namely, **human interaction with semantically structured information in ontologies**. Although assuming that ontologies and, by extension, knowledge processing on the Semantic Web, is inherently language-independent, knowledge generation and access will remain language-based, and, consequently, multilingual. Multilingualism is therefore an emerging challenge to the Semantic Web development and to its global acceptance across language communities around the world. The interaction between multilingualism and knowl-

edge representation systems is illustrated in figure 1.1. There we see that production and consumption of knowledge are user-centered and, therefore, rely on the language of the final user, whereas the representation of knowledge is based on language-independent formalisms that facilitate reuse and interoperability in the Web.

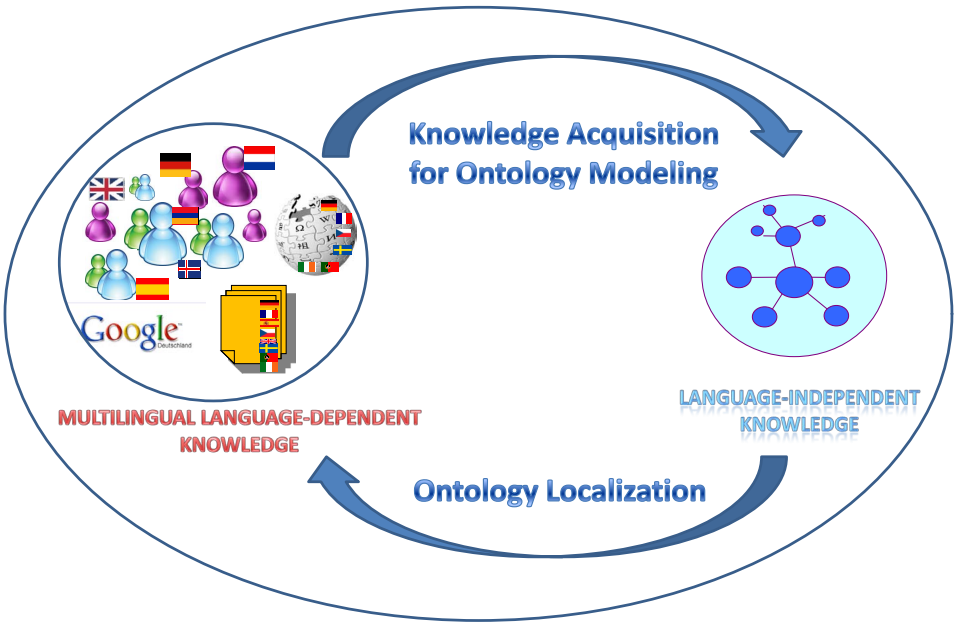


Figure 1.1: Interaction between natural languages and ontologies in the semantic web

In this chapter, we start with a brief description of the thesis context, in which we introduce the concepts of Ontology, Semantic Web, and Multilingualism in the Semantic Web. This is followed by an overview of the goals and contributions of the thesis. Then, we present the methodology adopted in this work, and finally, the thesis structure.

1.1 Thesis Context

The work presented in this thesis belongs to the domain of Ontological Engineering, overlapping other related fields such as Semantic Web, Terminology, Translation, and Natural Language Processing. This work tries to give response both to the need of considering multiple linguistic communities in the modeling of ontologies, focusing on novice users, and to the need of reusing the resulting ontologies in a multilingual scenario.

Ontological Engineering refers to the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and

languages for building ontologies. The term *ontology* has its origins in the Greek Philosophy, where it meant "systematic explanation of being" (Aristotle[2]). In the field of Philosophy, ontology is the theory of things or objects and their relationships. The Knowledge Engineering and Artificial Intelligence communities saw in this concept the core principle of the organization or structure they wanted to apply to parcels of knowledge in order to allow information interchange between both humans and computers. From this perspective, ontologies are the outcome of the activity of ontological analysis and modeling, rather than a discipline.

The 90s was the decade that witnessed the birth of the first ontologies. In 1993, Gruber provided one of the most quoted definitions of ontology in the Artificial Intelligence literature. For this author an ontology represents "an explicit specification of a conceptualization". Studer et al. (1998: 185) expanded this definition and explained the main notions involved in the concept of ontology:

> "**Conceptualization** refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. **Explicit** means that the type of concepts used, and the constraints on their use are explicitly defined. **Formal** refers to the fact that the ontology should be machine-readable. **Shared** reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted group".

To put it in simple words, we could define ontologies as models that structure knowledge by a) identifying the set of concepts that describe that domain, b) establishing relations among them, and c) listing the main properties of those concepts. Broadly speaking, an ontology consists of four main components: classes, properties, instances, and axioms[3].

Classes identify specific or abstracts concepts. *Properties* are divided into data type properties and object properties. *Data type properties* refer to features or characteristics that define concepts. *Object properties* represent dependencies between concepts, or how concepts relate to each other. *Individuals* are specific, real objects that belong to a certain class of objects. Finally, we should refer to a specific type of properties called axioms. *Axioms* can be broadly defined as restrictions imposed on classes or relations.

Consider, for example, an ontology of cartoon animals, where *cartoon mouse* would be a type or subclass of *cartoon animal*, i.e., a class in the ontology; *gender*, *size* and *colour* of cartoon mouse would be data type properties; *cartoon mouse* could be related to *cartoon cheese* by means of the relation or object property

[2]Aristotle's Metaphysics, Standford Encyclopaedia of Philosophy http://plato.stanford.edu/entries/aristotle-metaphysics/ [Accessed in June 2007].

[3]Depending of the paradigm followed, the terminology used to name ontology components will differ. In the Frames paradigm, ontology components are defined as concepts, attributes, relations and instances. In the Description Logics paradigm, ontologies consist of classes, properties (object properties and data type properties) and instances or individuals. For the sake of consistency, we will stick to the Description Logics vocabulary when referring to ontology components, and will use the term concept in the sense of cognitive "unit of meaning" (Croft and Cruse, 2004).

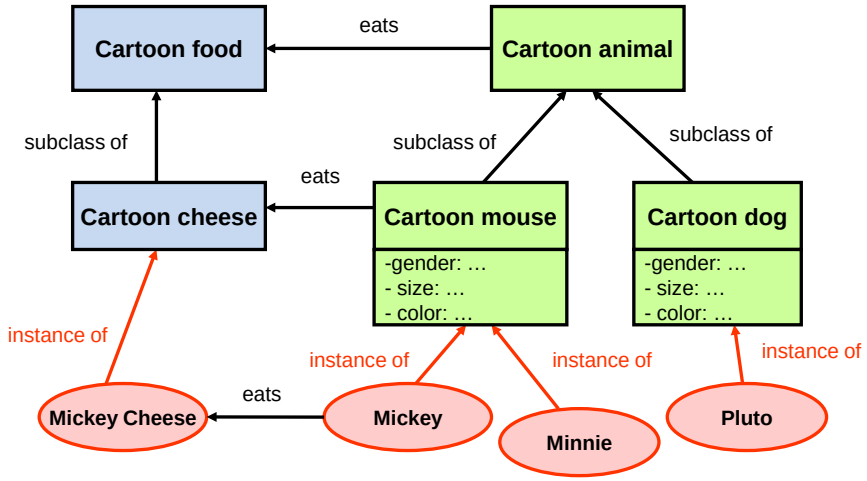


Figure 1.2: Simplified representation of an ontology of cartoon animals

eats; an axiom could be imposed on this relation saying that the relation *eats* can only established with *cartoon cheese* and no other type of cartoon food; and a certain mouse called *Mickey* could be an instance of the class *cartoon mouse*. A common representation of an ontology has been included in figure 1.2 with this highly simplified extract of a cartoon animals ontology.

In a sense, ontologies can be said to resemble other forms of knowledge organization such as conceptual maps or terminological resources, but the big contribution of ontologies is that the knowledge they structure can be made processable by computers, so that computers can reason over it and infer information. In addition, ontologies are defined as capturing knowledge consensually agreed by a community of users. The purpose of this is to enable sharing and reuse by other user communities so that cooperation, interchange of information and interoperability among systems is made easier. For a complete description of the notion of *ontology* from different paradigms see Vossen (2003).

Finally, it should be pointed out that not all ontologies are restricted to a specific domain of knowledge, the so-called *domain ontologies* (e.g., UMLS in the medical domain[4]), but some of the most relevant ones (e.g., SUMO[5], CYC[6]) organize "general concepts that are common across the domains and give general notions under which all the terms in existing ontologies should be linked to" (Gómez-Pérez et al., 2003: 71). These have been termed *fundamental ontologies* or *upper-level ontologies*.

All in all, ontologies are the knowledge representation systems that have shown most appropriate to model knowledge in the Web, and no efforts are being spared to

[4]http://www.nlm.nih.gov/research/umls/
[5]http://www.ontologyportal.org/
[6]http://www.cyc.com/

bring them closer to the average user. The challenge now is to consider ontologies in a broader scenario in which users come from different linguistic communities and have different expertise levels in ontological engineering. These and other issues will be dealt in the next sections.

1.1.1 Ontologies as Underpinnings of the Semantic Web

The Semantic Web has represented a turning point in the evolution of ontologies. Commonly defined as "an extension of the *traditional Web* in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee et al., 2001), the Semantic Web relies on the assignment of ontological classes and properties to unstructured or semi-structured data in the Web. This annotation of web documents -available in natural languages- with the vocabularies and the semantics made explicit in ontologies is used to describe the content of documents and allows reasoning about it.

For example, the availability of ontologically annotated documents is crucial in enabling the shift from keyword-based queries and navigation of predefined links provided by the HTML protocol on the Web, to semantic-driven search and navigation that can be effectively handled by automatic agents in Semantic Web applications (Maedche et al., 2003). In this sense, ontologies can be understood as the scaffolding of the Semantic Web.

During the last 20 years, many efforts have gone into the development of methodologies and tools to support users in the creation of ontologies. The need for developing ontologies in a faster and efficient way encouraged researchers on Ontological Engineering to define common and structured guidelines that could help ontology engineers in the ontology development process. The main purpose of these methodological works was to identify (a) the activities that needed to be carried out in any ontology development process from scratch, (b) the order in which these activities had to be performed, and (c) the human and technological support available for each of them. Classical methodologies for the development of ontologies are METHONTOLOGY (Fernández-López et al., 1999), DILIGENT (Pinto et al., 2004) and On-To-Knowledge (Staab et al., 2001). A detailed description of them can be found in Gómez-Pérez et al. (2003: 107-196).

Recently, a new paradigm for the development of ontologies has been proposed in the so-called NeOn Methodology (see (M. C. Suárez-Figueroa, 2010)). This new methodological approach comes to palliate the lack of guidelines in classical methodologies for building large ontologies embedded in ontology networks. It also aims at providing guides for the development of ontologies starting from available ontological and non-ontological resources, instead of assuming the development from scratch. A collaborative development by distributed teams of developers is also accounted for. Finally, this new paradigm also caters for the dynamic evolution of ontology networks. More details on the NeOn Methodology will be given in chapter 4. Indeed, the method we propose for knowledge acquisition and

ontology modeling has been developed in the framework of this methodology.

Regarding the development of tools for supporting users in the ontology development process (also known as ontology editors), some of the most popular ones nowadays are Protégé[7], TopBraid Composer[8] or NeOn Toolkit[9]. Ontology editors were initially conceived to avoid users having to implement ontologies directly in a formal language. They provide interfaces that help users carry out some of the main activities of the ontology development process. In this work we will present some ontologies developed with Protégé and the NeOn Toolkit (see 12). This latter editor is the one that currently supports more activities of the ontology development process by means of specific components (or plug-ins) for each of the activities. One of these plug-ins, the LabelTranslator NeOn Toolkit plug-in (Espinoza et al., 2008b), will be explained in chapter 11.

Apart from the methodological and technological efforts just mentioned, a lot of work has been devoted to the definition of languages to represent semantic content. Particularly important are the RDF (Resource Description Framework) (Klyne and Carroll, 2004) and OWL (Web Ontology Language) (Dean and Schreiber, 2004) languages. In fact, OWL reuses and extends RDF and its successor, the RDF Schema (also known as RDF(S)). These ontology languages have been developed in order to describe ontologies on the Web and represent information about web resources, so that information can be exchanged between applications minimizing loss of meaning. Depending on the expressiveness required and the reasoning possibilities expected, a different ontology language will be chosen. See Gómez-Pérez et al. (2003: 202) for more details on this. Here we will briefly refer to OWL and one of its variants, OWL DL, in which DL stands for Description Logics[10].

OWL is one of the most popular syntaxes in the current Semantic Web, and it is the one used in this work. Some of its main characteristics are: a) It allows the organization of classes in hierarchies and allows for subsumption between classes; 2) It permits to express unions and intersections of classes, as well as disjointness and exhaustiveness (e.g., by modeling that *cartoon dog* and *cartoon mouse* are disjoint, we are saying that the instances of cartoon dog can never belong to cartoon mouse); 3) It enables restrictions to be applied to some classes of the ontology (e.g., to say that *cartoon mouse* **only** *eats cartoon cheese*); 4) It admits cardinality restrictions (e.g., to say that *cartoon mouse* has two *cartoon mouse parents*); 5) It permits to define other characteristics of properties such as transitivity or inverse property. These characteristics of the OWL syntax have turned it into one of

[7] http://protege.stanford.edu/

[8] http://www.topquadrant.com/products/TB_Composer.html

[9] http://neon-toolkit.org/wiki/Main_Page

[10] Description Logics (DL) is the name for a family of knowledge representation formalisms that represent the knowledge of a domain by first defining the relevant concepts of the domain, and then using these concepts to specify properties of objects and individuals occurring in the domain. DL relies on first order logic (with some extensions) to define specify properties of classes and individuals, and allows for implicit knowledge to be automatically inferred (Baader and Nutt, 2002: 47).

the most expressive web languages, and all the ontology editors mentioned above support it.

Thanks to their flexibility and reasoning possibilities, ontologies are considered the most appropriate knowledge representation systems to bring semantics to the Web. Moreover, as we have briefly summarized, the methodological and technological support needed to build ontologies is already available and mature to make the vision of the Semantic Web a reality. The next step is to see how ontologies can face some of the challenges related to *heterogeneity* in the Web, in particular concerning multilingualism.

1.1.2 Multilingualism in the Semantic Web

The Web and its extension, the Semantic Web, are by nature distributed and heterogeneous (D'Aquin et al., 2008). Constantly, thousands of users all over the world are creating and updating knowledge. This dynamic and diverse scenario favors the creation of knowledge resources on the same domain represented by different formats and expressed in multiple natural languages (NLs).

This reality seems to be in contradiction with one of the features that ontologies should ideally have according to Studer's definition: ontologies should capture *consensual* knowledge. It is difficult to imagine that only a few number of agreed ontologies could exist, let alone if we take into account that ontologies are not created for the sake of it, but for their interaction with texts or for their use in multiple applications. In this sense, we will agree with Aussenac-Gilles et al. (2008), when they claim that

> (...) concept definitions can be the result of negotiations among domain experts or a compromise between various text sources. Thus the ontology is supposed to be the most consensual knowledge among the user community, and, at the same time, the most relevant one for the application.

Therefore, we assume that the knowledge represented in the ontology will be the result of agreement among a certain community of users, but another community may agree on a different representation.

By assuming this diversity in the Semantic Web, we need to search for strategies and provide solutions that make the Web feasible despite heterogeneity. In this thesis we are mainly concerned with the linguistic and cultural diversity in the Web, or, what is the same, with *multilingualism*.

Let us consider some statistics to start with. Although the highest number of Internet users is represented by English native-speakers with nearly half a billion users, this only represents around 30% of the total amount of internet users[11]. English users are closely followed by Chinese speakers, representing 23%, and followed at a long distance by Spanish speaking users with nearly 8%.

[11] Data obtained form the Internet World Stats web page at http://www.internetworldstats.com/stats7.htm [Accessed in July 2010].

These data reveal that multilingualism is a crucial issue that needs to be tackled for the definitive launching of the Semantic Web. On the one hand, this means that communities of "heterogeneous" users (users coming from different linguistic backgrounds) will create ontologies for their applications to benefit from Semantic Web technologies. This fact also involves that there will be more demand from methods and tools intended for heterogeneous users, not only due to their linguistic and cultural backgrounds, but also to different levels of expertise in ontology modeling. On the other hand, heterogeneity in the Web also means that ontologies will need to interact and interoperate with other ontologies in the Semantic Web expressed in different natural languages.

The impact of multilingualism in the Semantic Web, specifically in ontologies and Ontological Engineering, can be understood in terms of the following requirements, which are related to the open research problems we will address later on in this work:

- Users coming from different linguistic backgrounds, as well as expertise levels, require support in ontology modeling.
- Ontologies have to interact with information in several natural languages, so there is a need for strategies and models to deal with multilingual information in ontologies.

In this thesis we argue that the issue of multilingualism in ontologies can be examined from the perspectives afforded by a range of theories within the disciplines of linguistics, terminology and translation. As a matter of fact, *semantics* is at the core of all these disciplines, and each of them can provide valuable insight into the interaction between natural language expressions and the domain semantics represented in the ontology.

In chapter 2 we outline the broad *linguistic theoretical framework* in which our contributions to multilingualism in ontologies are supported. There we also describe the linguistic models we have chosen to systematically analyze the semantics of the linguistic constructs that are to be transformed into ontological structures in the research we perform on knowledge acquisition and ontology model. Here we are referring to the Lexical Constructional Model (Ruiz de Mendoza Ibáñez and Mairal Usón, 2006b, 2008), the Role and Reference Grammar (Van Valin and LaPolla, 1997), and the Generative Lexicon (Pustejovsky, 1995).

Then, we will also refer to terminology and translation theories in different chapters of this thesis in order to explain the interaction between multilingualism-multiculturalism and domain semantics. We are mainly referring to Cabré's Communicative Theory of Terminology (Cabré, 1999) and Temmerman's Sociocognitive Terminology Theory (Temmerman, 2000), as well as to Functionalist Theories to translation (Reiss and Vermeer (1984) and Nord (1997)).

After having introduced the fundamentals of ontologies and the need for dealing with multilingualism in the heterogeneous context of the Web, we provide an overview of the main goals and contributions of this work.

1.2 Goals and Contributions

Our main goal in this thesis is to advance the current state of the art in the interaction between natural languages and ontological representations in a multilingual scenario, and at two different stages: knowledge acquisition and ontology modeling from natural languages, and ontology localization to different natural languages. Regarding the first research topic, we face some of the issues that arise when newcomers to ontology engineering perform ontology modeling. In this context, we focus on providing **support in the knowledge acquisition activity and the subsequent ontology modeling** relying on natural language expressions. We also consider the fact that users may come from different linguistic communities, and that assistance has to be provided in multiple languages. As the second research topic is concerned, we deal with **ontology localization** issues and the problem of **representing multilingual information in ontologies**.

For tackling these issues we make several contributions related to the two main research topics.

1. Firstly, we provide a repository of multilingual patterns and a method for the reuse of ontological constructs that allow performing knowledge acquisition from expressions in natural language, and support the task of ontology modeling.

   The repository contains linguistic patterns associated to ontological constructs that model a certain parcel of knowledge in an ontology. On the one hand, the set of linguistic patterns included in the repository corresponds to linguistic structures based on verbal predicates that convey the conceptual relations captured in certain ontological structures. These linguistic structures have been termed **Lexico-Syntactic Patterns** based on previous literature, as will be explained in chapter 3. In order to provide support to different linguistic communities of users, Lexico-Syntactic Patterns have been identified for Spanish and English.

   On the other hand, the ontological structures included in the repository can be understood as small ontologies or building blocks that model a specific knowledge aspect and that can be reused in multiple ontologies during the ontology development process. These ontological compounds are known as **Ontology Design Patterns**, follow well recognized principles in Ontology Engineering, and are also considered to be good modeling solutions for design problems.

   The linking or **correspondence between Lexico-Syntactic Patterns and Ontology Design Patterns** is sustained by a manual analysis of the semantic representation of verbal predicates on the light of the Lexical Constructional Model and the Generative Lexicon.

   The main purpose for the creation of this repository of multilingual Lexico-Syntactic Patterns associated to Ontology Design Patterns is to serve as the

core of a system that will permit English- and Spanish-speaking users to express in natural language their modeling needs. Then, if a matching can be established between the natural language specification of the user and one of the Lexico-Syntactic Patterns in the repository, a modeling solution will be offered to the user in the form of one or several Ontology Design Patterns. The technological side of this approach has been performed and evaluated to a certain extent in this work, as described in chapter 7.

Finally, we propose a **method for guiding novice users in the reuse of Ontology Design Patterns** for ontology modeling, specifying the tasks they have to carry out when relying on the patterns repository and the system that implements them.

The particular contributions of this first research topic can be summarized in:

- An **analysis** of the deep semantic structure of some verbal predicates in English that convey the knowledge captured in Ontology Design Patterns according to the Lexical Constructional Model and the Generative Lexicon
- A **multilingual repository** (English and Spanish) of Lexico-Syntactic Patterns associated to the Ontology Design Patterns that model the semantics conveyed in the natural language expressions
- A **method** for supporting the acquisition of knowledge and the modeling of ontologies intended for novice users
- The **implementation** of a set of English Lexico-Syntactic Patterns in JAPE[12] rules for their reuse in Natural Language Processing (NLP) Applications
- A preliminary **evaluation** of the method and the implemented English Lexico-Syntactic Patterns with novice users in an academic setting

As can be observed, the multilingual repository contains Lexico-Syntactic Patterns in English and Spanish, but the analysis, implementation and evaluation phases have only been performed for the English language. The main reason for this is that most of the results and conclusions of these phases can be considered language-independent and can be easily extrapolated to other languages.

2. Regarding the second main contribution of this work, we develop a **model**, the Linguistic Information Repository (LIR), to enrich the linguistic layer of ontologies with multilingual information. Since the final aim of the model is to support the **localization** of ontologies to different natural languages, an

[12]JAPE stands for Java Annotation Pattern Language and is the formal language we have used to implement our patterns so that they can be processed by an NLP application.

analysis of the localization problem in ontologies is performed. This analysis is based on extant translation theories, such as Functionalist Theories, which provide an adequate framework along which ontology localization projects can be characterized.

The design of the model takes into account several requirements related with knowledge representation issues (dealing with the representation of linguistic and terminological information within and across languages, as well as cultural specificities), and technical aspects (regarding representation of linguistic information in ontologies and interoperability of the model with standards and models for the representation of lexical and terminological information in the Web).

Finally, an extension of the Ontology Metadata Vocabulary (OMV) is proposed to allow reporting about the linguistic information contained in ontologies at a metadata level. This extension called LexOMV would allow users to search for ontologies that have associated linguistic information in one or several languages.

The particular contributions in this sense are:

- An overview of the dimensions involved in the localization of ontologies
- An analysis of the **strategies to be followed in the localization of ontologies** drawing on Functionalist Theories to translation and Software Localization parallelisms
- An **overview of different modeling modalities** for representing multilingual information at the knowledge representation level in ontologies
- The **LIR model** for associating multilingual information with ontologies with the purpose of contributing to their localization to various natural languages
- A comparison of the LIR model to other models for associating multilingual information to ontologies
- LexOMV, an extension of OMV to report about multilingualism in ontologies at the metadata level

After an analysis of the state of the art in the topics dealt in this work, we will formulate the **assumptions** that we put forward in each case. In the first research subject regarding *knowledge acquisition and ontology modeling from natural languages*, the state of the art will be reviewed for the following topics:

1. knowledge acquisition from text focusing on verbal patterns, and knowledge acquisition from experts relying on controlled languages (chapter 3)

2. reuse of Ontology Design Patterns for ontology modeling, focusing on reuse methods (chapter 4)

Regarding the second research topic dealing with *ontology localization* and the *representation multilingual information in ontologies*, we will describe the state of the art on

1. translation strategies for the localization of ontologies (chapter 8)

2. models for representing multilingual information in ontologies (chapter 9)

3. requirements for an ontology localization model (chapter 10)

1.3 Methodology

The methodology applied in both research topics can be considered requirement-driven and empirically validated.

The approach that allows knowledge acquisition and ontology modeling by mapping natural language specifications to corresponding Ontology Design Patterns can be said to be design-focused and requirement-centered, in the sense that the main goal is to design a repository and a method, and implement a system that can meet the specified goals and requirements. These requirements are mainly related with providing users a transparent way of modeling an ontology starting from user specifications in natural language. The specific requirements, as well as the assumptions, will be detailed in sections 3.1.3, 3.2.3, and 4.5.

We also argue that the methodology regarding this contribution is empirical in the sense that the approach is evaluated with a number of test subjects to demonstrate its feasibility and analyze its performance.

Regarding the second contribution of this work, namely, the model to associate multilingual information with ontologies, the methodology applied here would also be considered requirement-driven. Here again the main goal is to develop a model which fulfills the requirements defined, being the most relevant ones, the representation of linguistic information according to current standards, and the definition of relations between lexicalizations within and across languages.

The practical feasibility of the design is then discussed by considering real needs derived from a large organization (FAO) as well as by a qualitative discussion of the advantages of the introduced model compared to extant models. As in the first contribution, particular requirements to be fulfilled by the model, as well as assumptions, will be defined in sections 9.4 and 10.4.

From now on, the two principal contributions of this thesis will be dealt separately in the document, with the exception of the theoretical framework that applies to both. This means that this document will be divided in two main blocks or parts, each of which containing chapters regarding the state of the art, open research problems, work assumptions, contributions, and evaluation. This will be explained in more detail in section 1.4.

1.4 Structure of the Document

The structure of this thesis is as follows. It comprises thirteen chapters, including this one, and is divided in two parts devoted to the two main contributions of this work.

After this introductory chapter, in **chapter 2** we present the theoretical framework in which the principal contributions of this thesis have been devised. We review the most relevant tenets of functional approaches to linguistics, particularly, Cognitive Linguistics, Role and Reference Grammar, the Generative Lexicon and the Lexical Constructional Model.

The following five chapters (chapter 3, 4, 5, 6, and 7) offer an insight into the method for supporting the activities of knowledge acquisition and ontology modeling intended for novice users, and based on the reuse of Ontology Design Patterns. These five chapters make up the **first part** of the thesis.

Chapter 3 provides an overview of the several approaches followed for the acquisition of knowledge from text and from domain experts, and points out their main benefits and drawbacks. This allows us to identify open research problems and work assumptions.

In **chapter 4** we introduce the philosophy embraced by the new paradigms on ontology modeling that favor the reuse of available resources. In particular, we focus on Ontology Design Patterns (ODPs) as reusable ontological resources. There we define ODPs, and identify the templates that describe them, the repositories that contain them, and the extant methods and tools that support their reuse. At the end of this section we also identify some open issues regarding the reuse of ODPs.

Chapter 5 is devoted to the description of the repository of Lexico-Syntactic Patterns (LSPs) associated to ODPs. For this aim, we provide in the first place a description of the ODPs studied in this research work. Then, we describe the strategies followed for the identification of candidate verbal patterns in English that convey the meaning represented in the selected ODPs. A deeper analysis of the semantics of those candidate verbal patterns that display a polysemous behavior is provided on the light of the Lexical Constructional Model and the Generative Lexicon. Finally, the repository of LSPs in English and Spanish associated to ODPs (*multilingual LSPs-ODPs pattern repository*) is presented.

After having described the benefits of reusing ODPs and having presented the repository of LSPs asssociated to ODPs, we propose a method for ontology modeling intended for novice users that involves the formulation in NL of modeling issues for a semi-automatic reuse of ODPs (**chapter 6**). This method is characterized by formulating modeling issues in several NLs (English and Spanish) relying on the multilingual LSPs-ODPs pattern repository.

The implementation of the English LSPs included in the repository, and the development of an application for the semi-automatic identification of modeling solutions in the form of ODPs from user statements is described in **chapter 7**. This is followed by a description of the Semantic Web portal that provides them on-line. Finally, we describe an experiment carried out with students to validate the method

proposed in chapter 4 and the application presented in chapter 7.

Chapter 8 opens the **second part** of this thesis devoted to the model for storing multilingual information in ontologies. It starts by defining the concept of Ontology Localization and by identifying the dimensions and problems involved in the localization of ontologies.

Chapter 9 deals with representational issues, and identifies the main advantages and disadvantages of current options to model multilingualism in ontologies. This allows us to define some open research problems and make some assumptions.

Then, in **chapter 10**, we spell out the requirements of a model for associating multilingual information to ontologies, namely, resource interoperability, localization, and accessibility.

In **chapter 11** we provide a detailed description of the model we propose for localizing ontologies, the LIR. At the end of this chapter we also include a brief description of the technological support that has been provided to the LIR model, since it is used by the LabelTranslator system, a plug-in of the ontology editor NeOn Toolkit. In the last section of this chapter we propose ontology metadata to report about multilingualism in ontologies.

The functionalities provided by the model are described in **chapter 12** on the basis of some examples. Then, the LIR model is compared against the modeling modality offered by the ontology languages RDF(S) and OWL for the association of linguistic information to ontologies. This comparison has been carried out with a real example of an ontology of the hydrographical domain.

Finally, **chapter 13** concludes this work and points to some future lines of research.

Chapter 2

Theoretical Framework

The work presented in this PhD thesis is considered within the broad theoretical framework of ***functional*** approaches to linguistics. Specifically, we identify our understanding of the interaction between semantics and language with that of the Cognitive Linguistics theories[1]. By adhering to functional linguistics we discard what has been considered the ***formal*** paradigm, which gives priority to grammatical description, pushing semantic analysis into the background[2].

Cognitive Linguistics embraces several approaches concerned with the relationship between human language, the mind and socio-physical experience (Evans et al., 2006). Broadly speaking, this theory states that language forms an integral part of human cognition (Peña Cervel and Samaniego Fernández, 2006). It emerged in the 1970s as a reaction to the dominance of *formal* approaches to language, and was influenced by several cognitive disciplines, particularly cognitive psychology. Cognitive linguistic practice can be divided into two main areas: cognitive semantics and cognitive (approaches to) grammar, being *construction grammar* one of the most relevant approaches in this context.

The reason for selecting this theory to explain how we understand the relation between language and ontological knowledge is because cognitive linguistics, and more specifically cognitive semantics, have investigated **how the semantic structure of concepts is encoded in language**. This view stays in accordance with our belief that there is a strict relation between *knowledge* as represented in an ontology, i.e., its conceptual structure, and the *structuring of knowledge* in language, i.e., the construction of meaning in language, since language is one of the most important means we have to convey and communicate knowledge.

Apart from these general considerations, there are some specific assumptions made by cognitive linguists that we also assume in our work (from Evans (2010)):

[1] Main representatives of this theory are Fillmore (1975), Lakoff (1987), Fauconnier (1985), M. Johnson (1987), Langacker (1987), Goldberg (1995), Talmy (2000), Croft and Cruse (2004), and Evans (2006), amongst others.

[2] The formal paradigm is identified here with the initial proposals by Chomsky in the definition of its Generative Grammar (see Chomsky (1957, 1965)).

1. Conceptual representation is the outcome of the nature of the bodies humans have and how they interact with the socio-physical world (the thesis of *embodied cognition* expressed by Lakoff (1987)).

2. Meaning, as it emerges from language use, is a function of the activation of conceptual knowledge structures as guided by context; hence there is no principled distinction between core meaning (semantics) and non-core meaning (pragmatics, social or cultural meaning). The merging of these two types of meanings is denominated encyclopedic meaning[3].

3. Encyclopedic meaning emerges in context. Encyclopedic meaning arises in context(s) of use, so that the selection of encyclopedic meaning is informed by contextual factors. The encyclopedic meaning view claims that word meanings do not exist, but are selected and formed from encyclopedic knowledge.

4. Lexical items are points of access to encyclopedic knowledge. Words are not containers that present "neat pre-packaged bundles of information". Instead, they provide access to particular parts of the network of encyclopedic knowledge.

This understanding of meaning is the one we propound in this work, and the one that has inspired us in the research we have conducted. Categorization, as performed in ontologies, is the product of human understanding of reality, thus we agree with the first assumption listed above. We also share the view that meaning is fundamentally guided by context, and that the pragmatic, social and cultural view of concepts cannot be separated from the so-called "core meaning". By assuming this we can explain linguistic phenomena such as polysemy, synonymy or figurative language, since the same word can have multiple meanings that are activated in context. Finally, the understanding of words as pointers to encyclopedic knowledge allows us to equate *words* to *concepts* that evoke vast repositories of networked knowledge relating to a particular domain. This is not to deny, however, that words have conventional meanings associated with them that can be captured in a specific context and for certain purposes. This is an issue we will return to later.

These principles support the so-called *experientialist* account propounded by cognitivist researchers that can be summarized as "language and the world have meaning only because human beings make them meaningful by interacting with objects" (Lakoff and Johnson, 1980: 159). By adopting this theoretical approach we reject some of the basic hypothesis previously formulated by *formal* linguistic theories such as Generative Grammar that hold that grammar could be studied

[3]Specific theories in Cognitive Semantics which adopt the encyclopedic approach include Frame Semantics (Fillmore, 1982) (Fillmore and Atkins, 1992), the approach to domains in Cognitive Grammar (Langacker, 1987), the approach to Dynamic Construal (Croft and Cruse, 2004), and the Theory of Lexical Concepts and Cognitive Models LCCM Theory (Evans, 2006).

independently of meaning, and that "core" meaning could be distinguished from "pragmatic social and cultural" meaning.

This *formal* approach adopted by generative linguists was denominated the *objectivist paradigm* (Lakoff, 1987). Amongst others, the main principles of *objectivism* are the following (Lakoff and Johnson, 1980: 198-209):

- Thought consists in the mechanical manipulation of abstract symbols.
- Symbols are meaningful inasmuch as they correspond to things in the external world.
- No matter how human bodies act and function in their environment. Concepts and reason exist independent of their own presence.
- Any machine that mechanically manipulates symbols that correspond to external reality is able to think and reason in a meaningful way.

According to these statements, objectivists believed that linguistic expressions corresponded to the world directly without the mediation of human understanding. This explanation of the relation between words and the external world had been traditionally represented by Ogden and Richards' *semiotic triangle* (see figure 2.1).

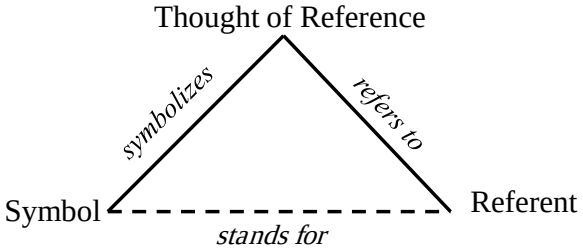


Figure 2.1: Ogden and Richards' semiotic triangle

What this triangular model basically explored was the relationship between words (symbols), the external world (referents), and the representation of referents in our minds (thoughts) in as much the same way as objectivists did. This view is still defended in some approaches to ontologies, as we will try to illustrate in the next section. Even when objectivist principles present sensible assumptions that can be logically applied to ontologies, we will support that the experientialist view provides a more flexible and comprehensive framework to account for ontological phenomena, particularly in a multilingual scenario.

Then, we will support our claims by briefly describing the evolution from an *objectivist* to an *experientialist* perspective in the neighboring field of Terminology. We will finish this overview of the theoretical approach adopted in this work by referring to the definition of concepts or categories, also from a cognitivist perspective.

Experientialism vs. Objectivism in Ontologies

If we extrapolate *objectivists* ideas to the Ontology Engineering field, or to the research in Artificial Intelligence in general, we may find many coincidences. In a way, ontologies try to structure the world by means of "abstract symbols" that refer to explicitly defined concepts, so that reasoning can be automated and information can be inferred. In the end, the purpose of ontologies is to imitate human reasoning.

The discussion between *objectivism* and *experientialism* reminds us of the much older philosophical account of *universals*, which goes back as far as Plato and Aristotle. Basically, this theory holds that universals are distinct from the particulars that instantiate them, and exist over and above of what we experience as real[4]. This theory, which has been termed *realism* assumes that universals "exist independently of the human capacity for thought and language" (Cocchiarella, 1996).

A realism-based approach to the construction of ontologies in the Biomedical domain has been assumed by modern philosophers such as Smith (2004) and Ceusters (Ceusters and Smith, 2006). According to Smith (2004), good ontologies are devoted "precisely to the representation of entities as they exist in reality". Only those terms that correspond to *universals* in reality, and thereby also to instances, can be considered *concepts* in ontologies. These authors avoid the use of the term *concept* in favor of *universal*, because they understand concepts as *subjective entities* used by a community of language users. Hence "the conceptualist interpretation is at base a report that simply describes the *use of language*" (Merrill, 2010), but which fails as an adequate basis for the semantics of biomedical terms.

While it may be convenient to adopt this approach in empirical sciences, other domains of knowledge may not have such a clear correspondence with entities in the real world, and notwithstanding, we should not deny their existence. In fact, Smith (2004) admits that the influence of the *concept view* in ontologies is due to much of the work on ontologies being concerned "with representations of domains, such as commerce, law or public administration, where we are dealing with the products of human convention and agreement - and thus with entities which are in some sense merely *conceptual*". So, in a way, he is making some concessions in favor of a conceptual approach for certain domains. The question then would be if such a conceptual approach could not be valid also for empirical sciences. Whereas for some *universals* there is a clear correspondence to instances in the world (for instance, *cell*, *DNA* or *lytic vacuole*), others are the product of human activity in its purpose of understanding and categorizing reality, as it is the case of *biotechnology* or *molecular genetics*. Can we still consider this latter type of terms *universals*?

At the other end of the spectrum we find the so-called *conceptual* philosophers like Abelard[5], who although admitting that universals "provide the semantic grounds for the correct use of predicate expressions", defended that these univer-

[4]The Medieval Problems of Universals, Standford Encyclopaedia of Philosophy http://plato.stanford.edu/entries/universals-medieval [Accessed in December 2008]

[5]Peter Abelard, Standford Encyclopaedia of Philosophy http://plato.stanford.edu/entries/abelard/ [Accessed in December 2008]

sals called concepts could not exist out of our intellect and "independently of the socio-biologically based capacity humans have for thought and language". In his view, the only reality that exists is in our cognition.

Modern philosophers like (Cocchiarella, 1996, 2001) propose a conciliatory solution in what has been called *conceptual realism*, which provides the basis of a "general conceptual-ontological framework", in which "beginning with thought and language, a comprehensive formal ontology can be developed". According to this theory "not only does conceptual realism explain how predication in thought and language is possible, but, in addition, it provides a theory of the nature of predication in reality through an analogical theory of properties and relations". According to this conciliatory view, language allows us to create an imitation of reality through the use of properties and relations between concepts.

It is in this context in which *experientialist* or *cognitivist* tenets are sustained. Cognitivism does not deny the existence of an objective reality or the possibility of accessing it. This theory simply maintains that there cannot be an objective and correct description of reality, but multiple ways of describing it. And in any case, these multiple descriptions result from a mental construction that humans make from their own experience.

The cognitivist approach to language and our understanding of the world could be schematically represented as a semiotic triangle following Ogden and Richards tradition, but the nature of the nodes would need to be redefined. In this sense, we propose a redefinition of the semiotic triangle as represented in figure 2.2. Ogden and Richards' *referents* or objects of the external world would become an *interpretation of the extralinguistic world* made by humans, or what is the same, an *ontology*. This interpretation of the world is in the end a classification or categorization that we humans make of reality with the aim of apprehend it or understand it. This allows us to represent not only objects that have an objective correspondence in reality, but also those artifactual objects created by humans.

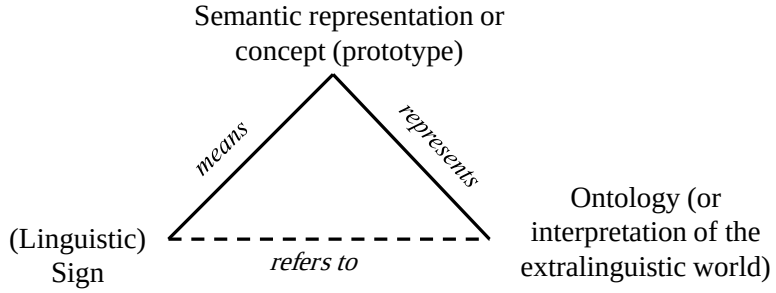


Figure 2.2: Redefinition of the semiotic triangle

Now, *linguistic signs* are seen as pointers to the interpretation of the external world (or the ontology) that we use to communicate and interact with others. Finally, *thoughts of reference* in Ogden and Richards's triangle would correspond here to the capacity of cognition or of organizing knowledge into classes that hu-

mans have, what we have called here the *semantic representation or concept*. In this context, classes are not understood as pre-packaged bundles of properties but as fuzzy categories definable according to contextual conditions. More attention will be devoted to the definition of categories below.

Then, assuming the views propounded by Cognitivism, we would support the idea of ontologies as products of language, or what is the same, of how a community of users[6] understands a certain parcel of the world. Taking this as baseline, we can accept the objective existence of some concepts that exhibit a set of "necessary and sufficient conditions", although bearing in mind that in that enumeration of necessary and sufficient conditions human intervention is present. At the same time, we claim that for most of the concepts those conditions cannot be so clearly delineated because their boundaries are not fixed. Therefore, when creating ontologies for computational purposes, human intervention will be required for determining how to define concepts. Decisive in this respect will be the context of the ontology restricted by the final purpose of the application in which the ontology is to be used.

In this sense, Guarino (1998) proposes an interesting distinction between *conceptualizations* and *ontologies* that we will adopt in this thesis. Whereas a *conceptualization* is understood as a "particular system of categories accounting for a certain vision of the world" in the conceptualist philosophical sense, an *ontology* is defined as "a *logical theory* accounting for the *intended meaning* of a formal vocabulary". This means that an ontology is an artificial artifact created for the purposes of a certain application that uses a specific vocabulary (concepts and relations) to describe certain aspects of a wider reality, which is represented by a conceptualization of the world. According to the author, "two ontologies can be different in the vocabulary used (using English or Italian words, for instance) while sharing the same conceptualization" (*ibidem*).

From this we can conclude that ontologies formalize a certain vision of the world as understood by a community of users for certain purposes, and that this can be achieved through language, because language reflects how a community of users understands reality. This claim is reinforced in current practice, because most of the knowledge structured in ontologies is obtained from descriptions in natural language, whether it be in text or speech input form. This also explains the fact that "the same parcel of the world" can be differently categorized depending on the community of users that is performing the knowledge representation. Therefore, we can argue that the world is out there, and may be the same for different speaker communities. But the way those communities interact with the objects of the world may vary from community to community, and this is reflected in the multiple categorizations that can be made of reality.

[6]Community is understood here not only as a community of people speaking the same language and coming from the same cultural environment, but also a community of experts in a domain against the general public, though having the same cultural and linguistic background.

This claim that ontologies are a product of language in the sense that they are developed by analyzing the meaning expressed in words is also reinforced by current terminology practice, as propounded in modern theories of Terminology such as the Communicative Theory of Terminology (Cabré, 1999) or the Sociocognitive Terminology Theory (Temmerman, 2000) (Temmerman and Kerremans, 2003). This semasiological approach (from terms to concepts) to Terminology was born in opposition to the traditional onomasiological approach (from concepts to terms) defended by the Vienna school represented by Wüster (1991). Traditional Terminology was based on a few premises which were considered to be unquestionable, as summarized in the following quote from Temmerman (2000: 1):

> (...) that concepts are clear-cut and can be defined on the basis of necessary and sufficient conditions, that univocity of terms is essential for unambiguous and therefore effective and efficient communication, and that figurative language and change of meaning are linguistic subjects which are of no concert to Terminology as Terminology restricts itself to the onomasiological perspective.

These principles were strongly rooted in objectivism. Followers of this tradition claimed that concepts should be studied and defined before terms, and only then, terms were assigned to them. In this way, the external world was assumed to exist independent of human observation and experience. Taking into account that the final aim of terminology work was to prepare terminology standards that would allow an unambiguous communication among experts, Traditional Terminology assumed that it was possible to achieve a unique correspondence between a term and a concept.

Without denying the validity of this approach for standardized communication, Traditional Terminology ignored many aspects of real communication between specialists (Cabré, 1999: 129). This provoked that terminology scholars, researchers and practitioners started questioning these principles and defining alternative ones that could account for the use of terms by domain experts in their communication. They realized that language played a determinant role in the conception and communication of categories. For instance, when giving names to specialized concepts, experts relied on their cognitive structures, previous experiences, and on their linguistic knowledge. Thus, terminology should not be studied independent of language.

Traditional terminologists also overlooked the fact that many concepts could not be clearly delineated. On the contrary, recent theories claim that the content of a term is never absolute, but relative to context and domain (Cabré, 1999: 132). This means that the same term can be differently defined or interpreted according to the professional discourse of the communicative situation. It is also a fact that concepts evolve in time as do their designations, demanding a study of language evolution.

Moreover, by assuming univocity between terms and concepts, Traditional Terminology would eliminate polysemy and synonymy. Both linguistic aspects occur very often in technical languages. According to Temmerman (2000: 133) polysemy is the consequence of changes in conception over a period of time, and synonymy reflects different perspectives in specialized discourse.

In the same sense, Traditional Terminology rejected figurative language, without taking into account that metaphorical models are as well present in specialized discourse, since they facilitate the understanding of new concepts by extrapolating similarities from well known ones.

In this way, current theories of Terminology place the study of terminology within Linguistics and propound the study of specialized languages in context. They also rely on cognitivist approaches in that they assume that concepts cannot be objectively defined by a number of discrete features, but that concepts have fuzzy boundaries and are often represented in terms of prototypes.

The definitions of categories or concepts is the last point we want to discuss before introducing the specific theoretical assumptions made for each contribution of this work.

Theories of Categorization

The classical theory of categorization has its roots in Aristotle's work and argues that categories are defined in terms of necessary and sufficient conditions, what makes them have clear boundaries (Peña Cervel and Samaniego Fernández, 2006: 247). And although this may be true for some categories, it cannot give account for others. Illustrative in this scenario were the experiments conducted by Rosch (1973, 1975) asking subjects to judge to what extent an entity could be regarded as a good example of a category. For example, when investigating the category of "furniture" most of the subjects would agree on "chairs" as very good representatives of the class furniture, whereas "vase" or "telephone" were considered peripheral examples. This new approach to categories was called the *prototye theory* (Labov (1973), Rosch (1973)), in which a prototype was considered a category member central to the category in question. As a consequence of this, members sharing many properties with the prototype would be indisputably considered members of the category, and others showing a lower degree of coincidence could still be considered members of that category. Because of this, prototype theorists admitted that categories have fuzzy boundaries.

Later on, cognitive linguists have seen a number of problems in the prototype theory. According to Croft and Cruse (2004: 87), in spite of relaxing the requirement that category features had to be necessary and sufficient, assuming a list of features that are central to the category and others that are more peripheric is still far too simplistic. Some of the main criticisms to this respect is that this model fails to "capture the full range of properties linked in complex chains of association and causation involved in a concept" (*ibidem*). It also fails to handle context sensitivity, since in a certain context some properties of a category may be considered "more

central" than others in a different context. Also, the fact that some features may be relevant for certain subjects, but not for others. Thus, the question that arises is where the boundaries to categories are. Croft and Cruse (2004: 90) illustrate this problematic issue with an example of a category in a multilingual scenario, and say to this respect that

> The location of the boundary of a category is independent of its prototype, that is to say, two categories may have the same prototype but different boundaries; likewise, two categories may have the same boundaries but different prototypes. Take the French word *corde* and its default English translation *rope*. A questioning of native speakers of the two languages suggests that the prototypes of the two categories are very close: both put forward the same sort of things as best examples. However, their boundaries differ. *Le Petit Larousse* defines *ficelle* (string) as "un corde mince"; a parallel definition of string as "a thin rope", would seem very odd. That is to say, *ficelle* falls within the (default) boundary of the category CORDE, but string falls outside the boundary of the category ROPE (emphasis in the original).

This can be due to the contextual factor or to idiosyncratic characteristics, which are related to the way we learn new concepts according to our previous knowledge, or as said in Murphy (2002: 63)

> People do not rely on simple observation or feature learning in order to learn new concepts. They pay attention to the features that their prior knowledge says are the important ones. They may make inferences and add information that is not actually observed in the item itself. Their knowledge is used in an active way to shape what is learned and how that information is used after learning.

Studies in Cognitive Linguistics go a step further in which they admit that it is possible to determine boundaries for a category. It is the case of Croft and Cruse (2004: 87) who are in favor of a **dynamic approach** in which boundaries are created "at the moment of use". This means that in a certain context or for a certain application boundaries will be determined so that we are able to assert what is "inside" the category and what has to be "left out". So, basically, they agree with the fact that boundaries are fuzzy in the sense that "different subjects make different judgments as to the location of boundaries, and the same subject will make different judgments under different contextual conditions".

Stepping now to the research in ontologies, we argue that this is a very suitable approach for understanding concepts or classes in ontologies. Ontology engineers and domain experts will have to reach an agreement on which are the categories they want to include in a certain ontology, and which the boundaries they want to assign to them. This decision will be based on their knowledge of the domain, their interests, and the needs of the final application, among many other contextual conditions.

Having defined the framework in which the two contributions of this work have been devised, we briefly summarize the theoretical assumptions made for each contribution.

2.1 Theoretical Assumptions

Multilingual Lexico-Syntactic Patterns for Ontology Modeling

Regarding the research work dealt in chapters 3 to 6, we rely on *functional* approaches to linguistics such as the Role and Reference Grammar (Van Valin and LaPolla, 1997), and on Cognitive Linguistic assumptions, particularly those formulated by Construction Grammar (Goldberg, 1995). In fact, we will adopt one linguistic theory which has seen the benefits of bringing together ideas from functionalists, cognitivists and constructionists theories, the Lexical Constructional Model (henceforth LCM).

Basically, functionalists believe that function and meaning are factors that condition form. This means that in a sentence, the semantics or meaning of the verb is what determines the syntactical structure, and not the other way round (Mairal Usón and Cortés-Rodríguez, 2006: 104). These approaches to the semantic representation of sentences have been termed "projectionist" approaches (Van Valin, 2004).

Constructionists, on the contrary, claim that "no strict division is assumed between the lexicon and syntax" (Goldberg, 1995: 7). It is assumed that lexicon and grammar form a continuum, and when trying to discover the semantics of a sentence one cannot analyze meaning and form as if they were two independent things, but meaning and form influence and restrict each other. To put it simple, in a sentence, the meaning of the verb restricts the type of arguments that can appear in the sentence, but, at the same time, those arguments can also restrict the meaning of the verb. In fact, constructionists have shown that in some sentences, the meaning of the verb proves insufficient to explain the occurrence of some arguments.

The LCM provides an alternative to these two approaches by offering a framework that considers a set of constraints imposed by the lexicon on the grammar, but accounting at the same time for syntactic elements that can influence or modify the original meaning of the lexicon (Ruiz de Mendoza Ibáñez and Mairal Usón, 2006b). According to Mairal Usón and Ruiz de Mendoza Ibáñez (2006), the reconciliation between functional, cognitivist and constructional paradigms is really necessary "if we (the LCM) want to account for the vast range of phenomena involved in meaning".

For this aim, the LCM model provides the necessary machinery for the representation of predicates in the form of **lexical templates** that mainly rely on the logical structures postulated in the Role and Reference Grammar, and combine elements from Wierzbicka's Natural Semantic Metalanguage (Wierzbicka, 1996), and Pustejovsky's Generative Lexicon (Pustejovsky, 1995), as will be explained

section 2.2.

Moreover, the templates in the LCM have been designed in such a systematic way so that a subsequent formalization and exploitation by NLP tools is possible. In fact, one of the main objectives of the LCM is to explicitly define lexical items so that it can contribute to provide semantics to the Web. We would like to further explore this in future work.

In this PhD, thus, we are particularly interested in the use of the lexical templates to represent the grammatical, semantic and pragmatic features of the verbal phrases we have identified as candidate verbal patterns to be included in the *multilingual LSPs-ODPs pattern repository*. By choosing this model we have discarded other models whose scope is limited to capture only those aspects of a word that are grammatically relevant (such as Generative Grammar), or others that include encyclopedic information but do not allow a systematic representation of semantics (such as Frame Semantics).

Regarding Frame Semantics (C. Johnson and Fillmore, 2000), its main purpose is also the provision of a complete account of the semantic and syntactic combinatorial properties of verbs. Those verbs that belong to the same semantic domain are included in the same category or *frame*, and their main combinatorial possibilities are accounted for by means of annotated sentences. This theory has led to the development of the FrameNet project[7]. Currently, FrameNet contains more than 960 semantic frames, exemplified in more than 150,000 annotated sentences. Some of these frames approach domains as different as *categorization, communication, cognition, emotion, ingestion, natural-features, becoming-aware* or *attack*. Let us take as example the categorization frame[8]. It is defined in the following manner:

> A *Cognizer* construes an *Item* as belonging to a certain *Category*. In this process, the *Cognizer* may either passively perceive the *Item* and note that it fits the *Criteria* for a *Category*, or, alternatively, actively examine the *Item* for certain *Criteria* that define a *Category* (...).

The so-called Core Frame Elements of this frame are Category, Cognizer, Criteria and Item, and are distinguished from the Non-core Frame Elements that are Circumstances, Manner or Means, among others. The sentences that exemplify the use of these elements are directly obtained from corpora.

It should be admitted that FrameNet is an extremely valuable source for the analysis of the behavior of verbal patterns. However, the construction of its frames follows a rather *ad hoc* procedure. The sentences taken as representatives of the different frames, as well as the core and non-core elements chosen for each frame seem to be quite arbitrary. Some of the criticisms that have been made to FrameNet is that "the authors do not explain the criteria for domain membership or the internal structure of the domain" (Faber and Mairal Usón, 1999: 74). This derives in a

[7] http://framenet.icsi.berkeley.edu/

[8] http://framenet.icsi.berkeley.edu/index.php?option=com_wrapper&Itemid=118&frame=Categorization& [Accessed in June 2010]

lack of systematization that makes it difficult to apply this model to new domains or lexical items.

Having justified the selection of the LCM for the analysis of candidate verbal patterns before they are included in the *multilingual LSPs-ODPs pattern repository*, in section 2.2 we offer a brief account of the origins and development of the LCM and the theories that make up its current configuration. This will help us understand the mechanisms that allow us to represent the semantic, syntactic and pragmatic features of verbal predicates into a single representation called **lexical template**.

Linguistic Information Repository for Ontology Localization

Cognitivist approaches to categorization and categories have also been the starting point in our proposal for a linguistic model to associate linguistic and multilingual information to domain ontologies (chapters 8 to 12). This model that we have termed *Linguistic Information Repository* or *LIR*, as mentioned before, is the second main contribution of this work (see chapter 11). In our approach, an ontology already available for a certain domain is reused for its localization into a different linguistic and cultural context. The lexical and semantic interpretation of ontology classes as well as potential categorization mismatches are captured in the LIR, which is implemented as an external linguistic model associated to the ontology.

We assume that the categorization represented in the ontology reflects a vision of the world according to certain contextual conditions, represented by the linguistic descriptions associated to the ontology. Our assumption in this sense is that some domains of knowledge admit categorizations more prone to be accepted or shared by multiple linguistic communities, because the knowledge they represent is the result of a standardization or normalization process. These domains of knowledge have been termed *internationalized* or *standardized* domains of knowledge.

On the contrary, some categorizations represented in ontologies reflect a certain vision of the world that may not be shared by other groups of people. This means that categories do not have the same boundaries across cultures, as shown in the example of Croft and Cruse (2004: 90) quoted above. These discrepancies tend to appear in multilingual and multicultural scenarios in which one and the same categorization is supposed to be employed in different linguistic environments. For practical reasons, an agreement could be reached on which boundaries categories should have and which cultural specificities should be left out. However, if the ontology cannot be modified, we will have to look for other ways of representing cultural discrepancies. These domains of knowledge have been called *culturally-influenced* domains in this work.

With the aim of reusing such categorizations in a multilingual environment, several solutions have been envisioned. The one analyzed in more detail in this thesis propounds to capture categorization mismatches in the external linguistic model. The chapters that will be dealing with the localization of ontologies and the

linguistic model we propose for localizing ontologies are chapters 7, 8, 9, and 10.

2.2 The Lexical Constructional Model

The LCM is a 'model for the investigation of meaning construction at all levels of linguistic description, including pragmatics and discourse" (Mairal Usón and Ruiz de Mendoza Ibáñez, 2009: 153). In this section we are interested in describing the principal tenets of the LCM, providing a historical account of the model's evolution to understand its current configuration. We argue that this model provides the necessary machinery to systematically explain the semantics of the verbal predicates we consider in this work, by combining ideas from a variety of approaches, specifically, Van Valin's Role and Reference Grammar and Pustejovsky's Generative Lexicon. Two sub-sections will also be devoted to these two approaches, section 2.2.1 and section 2.2.2, respectively. At the end of this section, in section 2.2.3, we provide a description of the LCM lexical template that serves the purpose of representing the semantic and argument structure of verbs. Finally, we also present our own adaptation of the LCM lexical template in which some elements of the meaning description have been made explicit for the sake of clarity.

The LCM was primarily influenced by the tenets of **Dik's Functional Grammar**, which put the lexicon in a prominent position. In this theory, terms are described by predicate frames that "project" their underlying structures. The lexicon is also hierarchically organized from an onomasiological viewpoint, i.e., according to the meaning of terms.

The next model influencing the LCM was the **Functional Lexematic Model** (Martín Mingorance, 1990, 1998). This model shared the onomasiological approach to the lexicon, but found that Dik's lexicon structuring did not follow a coherent methodology. Therefore, it proposed a structuring of the lexicon in terms of lexical fields or domains. Existence, Change, Possession, Speech, Emotion, Action, Cognition, Movement, Physical Perception and Manipulation are the lexical domains identified in the Functional Lexematic Model (Mairal Usón and Faber, 2007).

The primary task of the Functional Lexematic Model was to investigate the paradigmatic structure of the lexicon, i.e., the hierarchical structure of lexemes, and only afterwards the syntagmatic potential could be investigated, namely, the combinatorial properties of lexical items (Butler, 2009: 121). The Functional Lexematic Model had a strong influence in the configuration of the LCM model because it contributed to the integration of the "semantic aspects of the lexical structure with syntactic aspects, in terms of the linkage between semantically-base hierarchies and syntactic complementation patterns" (Butler, 2009: 123-124). However, this model failed to provide a systematic account of the mapping between semantics and syntax.

With the aim of filling this gap, the LCM turned to another functional theory, the **Role and Reference Grammar** (RRG), which attempts to provide a clear link-

age of semantic representation to syntactic structures. The complex machinery that this theory proposes to account for the relation between semantics and syntax will be explained in more detail in section 2.2.1, since it is currently employed by the LCM with some modifications. The RRG model relies on the lexical decomposition of verbs in order to establish relations among semantically related verbs and their arguments. An example of verb decomposition is illustrated by the verb *to kill*, which is decomposed into an activity (*to do*) carried out by an actor or doer (x) that *causes* that someone (y, the undergoer) is *dead*.

kill [**do**' (x, Ø)] CAUSE [BECOME **dead**' (y)][9]

The principles of this decomposition are also adopted by the LCM in the form of *lexical templates*, as will be spelled out in section 2.2.1. However, there was not a clear methodology on how this decomposition had to be carried out, and which elements had to be considered "un-decomposable". For instance, in the previous example, the activity represented by the verb *kill* is further decomposed in *to do*, but *dead* is not further decomposed.

This lack of systematicity made the authors of the LCM search for other theories that could give account of a systematic decomposition of predicates to arrive at undefinable elements. The LCM introduced then a set of primitives that to a greater extent correspond to Wierzbicka's *semantic primitives or primes*, and which have been proposed in the framework of the **Natural Semantic Metalanguage** (NSM) theory. Semantic primitives are defined as "elements which can be used to define the meaning of words and cannot be defined themselves; rather, they must be accepted as *indefinibilia*" (Wierzbicka, 1996: 10).

The whole list of primes can be seen in table 2.1, and is also available on-line[10]. The use of Wierzbicka primitives or indefinables is one of the main innovations with respect to the original RRG proposal, which is in favor of the systematization of the templates creation.

Then, with the purpose of further enriching the semantic description of lexical units, the LCM incorporated Mel'cuk's *lexical functions* to capture "those pragmatic and semantic parameters that are idiosyncratic to the meaning of a word and that allow to distinguish one word off from others within the same lexical hierarchy" (Mairal Usón and Ruiz de Mendoza Ibáñez, 2008). These lexical functions have been collected by Mel'cuk and his team in the so called *Explanatory Combinatorial Dictionary* (Mel'Cuk, 1988) in the context of the **Meaning-Text Theory**[11].

Lexical functions are defined as dependencies (f) that are associated with lexemes (L), and that produce further lexemes (L') which play a specific syntactic role (Mel'Cuk and Polguère, 1987):

[9] Example extracted from Van Valin (2005).

[10] http://www.une.edu.au/bcss/linguistics/nsm/semantics-in-brief.php#model [Accessed in June 2010]

[11] http://meaningtext.net

Gramatical category	NSM Semantic Prime
Substantives	I, YOU, SOMEONE, PEOPLE, SOMETHING/THING, BODY
Relational substantives	KIND, PART
Determiners	THIS, THE SAME, OTHER/ELSE
Quantifiers	ONE, TWO, SOME, ALL, MUCH/MANY
Evaluators	GOOD, BAD
Descriptors	BIG, SMALL
Mental predicates	THINK, KNOW, WANT, FEEL, SEE, HEAR
Speech	SAY, WORDS, TRUE
Actions, events, movement, contact	DO, HAPPEN,MOVE, TOUCH
Location, existence, possession, specification	BE (SOMEWHERE),THERE IS, HAVE, BE (SOMEONE/SOMETHING)
Life and death	LIVE, DIE
Time	WHEN/TIME, NOW, BEFORE, AFTER, A LONG TIME, A SHORT TIME, FOR SOME TIME, MOMENT
Space	WHERE/PLACE, HERE, ABOVE, BELOW, FAR, NEAR, SIDE, INSIDE
"Logical" concepts	NOT, MAYBE, CAN, BECAUSE, IF
Intensifier, augmentor	VERY, MORE
Similarity	LIKE

Table 2.1: NSM semantic primes

f (L) = L'

For example, the lexical function OPER_1 specifies a verb for a noun denoting an action that takes as its grammatical subject the name of the agent of the action, and as its direct object, the noun itself (see examples below).

OPER_1 (QUESTION) = ASK
OPER_1 (PREGUNTA) = HACER

The result of applying the function OPER_1 to the nouns *question* in English and *pregunta* in Spanish are the verb forms *ask* and *hacer* respectively. Lexical functions are largely used to account for syntagmatic relations between lexemes. However, in the LCM, lexical functions are used to establish semantic distinctions between lexemes in a given domain, i.e., they are applied from a paradigmatic viewpoint to distinguish the more general lexical items from the more specific ones within a lexical class. See some further examples of lexical functions in table 2.2.

Lexical Fuctions	Definition
ANTI	Antonym
CAUS	Cause
CULM	The highest point of
DEGRAD	To get worse
FACT	Be realized
INCEP	The beginning of
INSTR	Instrument
LOC	Spatial location
MAGN	Intensely, very, to a very high degree
OBSTR	To function with difficulty

Table 2.2: Lexical Functions and their meaning

Let us take for example the verbs *understand* and *grasp* to see how lexical functions specify the semantic properties that differentiate one lexeme from another within a given domain. Both verbs belong to the COGNITION domain, and are defined by the semantic primitive *know*, which describes mental predicates in Wierzbicka's list of indefinables semantic primitives. To define the semantics of the verb *understand*, the lexical function CULM is used to describe that "the end-point of knowing something" has been reached (Ruiz de Mendoza Ibáñez and Mairal Usón, 2006a).

understand: [$\text{CULM}_{12[All]}$] know' (x, y)

In the case of the verb *grasp*, the process of understanding involves a larger degree of difficulty, and therefore the lexical functions MAGN, meaning "intense", and OBST, meaning "to function with difficulty", would be added to the semantic representation of the verb specifying "the larger degree of difficulty involved in carrying out the action" (*ibidem*). The use of such lexical functions specifying the semantic and pragmatic properties of the lexical items allows to differentiate them within the lexicon of a language.

grasp: [MAGNOBSTR & $\text{CULM}_{12[All]}$] know' (x, y)

The next step in the configuration of the LCM was the inclusion of **Construction theories**, particularly that of Goldberg (1995, 2006). These theories posit that lexicon and grammar form a continuum, and that if there is a clash between the semantics of a lexical entry and a construction, the former adapts itself to the construction. In this respect, the authors of the LCM assume different levels of meaning construction, in which lexical templates represent the most basic level. Then, higher cognitive processes (such as metaphor and metonymy) will modify those lexical templates to adapt them to more complex constructions.

Apart from the semantic representations of lexical items accounted for in Level 1, the model consists of three additional levels that build on the basic one, and that are *subsumed* into a higher level or act as a linguistic *cue* for the activation of the next level. The four levels recognized by the LCM are illustrated in figure 2.3 (Mairal Usón and Ruiz de Mendoza Ibáñez, 2009), and explained in the following[12]:

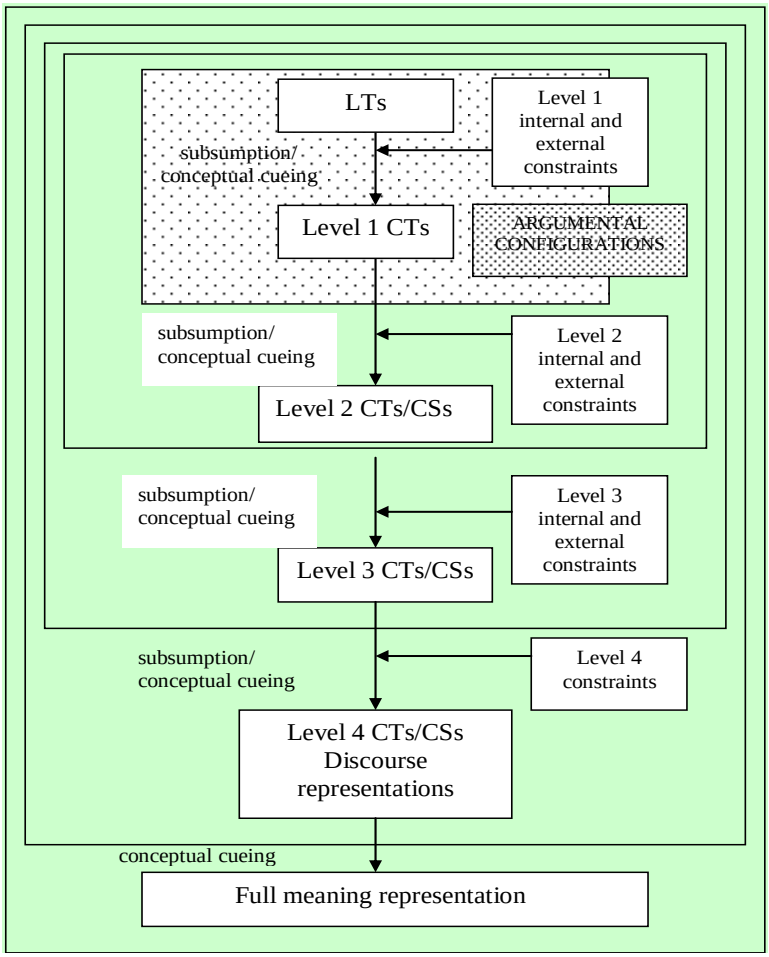


Figure 2.3: Levels of the Lexical Constructional Model

- Level 1 - argumental layer - accounts for the semantic representation of syntactic elements
- Level 2 - implicational layer - is concerned with meaning captured in constructions (pragmatics)

[12]LT stands for lexical template, CT stands for constructional template, and CS for conceptual structure.

- Level 3 - illocutionary layer - deals with traditional illocutionary force or intention of the speaker

- Level 4 - discourse layer - addresses discourse aspects, specifically cohesion and coherence

Lexical templates represent the classical argumental layer in Level 1. This is the central module and consists of "elements of syntactically relevant semantic interpretation based on the principled interaction between *lexical and constructional templates*" (Mairal Usón and Ruiz de Mendoza Ibáñez, 2008). Lexical templates are defined as low-level semantic representations of the syntactically relevant content of a predicate, whereas constructional templates represent high-level semantic representations derived from multiple lower-level representations. This is the level we are interested in in this work, because our purpose is to capture the semantics of the syntactically relevant aspects of predicates.

Then, a set of internal and external constraints explain the shift from one level to the next one. Internal constraints are those specified in the lexical template modeling process, whereas external constraints are concerned with the cognitive processes that allow the fusion of one construction into a different one (Butler, 2009: 136). However, we will not further elaborate on Levels 2 to 4, since these will not apply to our patterns[13].

Regarding the LCM templates, we still need to refer to a further theory that has influenced the most recent work on this lexical and constructional templates, and which has improved their format and theoretical significance: the **Generative Lexicon**, a theory developed by Pustejovsky (1995). The aim of this incorporation is to better establish the interrelations that hold between the two components of the lexical templates, namely, the semantic component and the syntactic component. Basically, what the mechanisms of the Generative Lexicon allow to do is to relate the **semantic primes** and the **lexical functions** with the **event and argument structures** provided in the RRG's logical structures. The enrichment of the lexical templates in such a way may help predicting when a lexical structure takes part in a construction at a higher level. The details of the improvement of LCM lexical templates with some mechanisms of the Generative Lexicon also deserve a separate section (see section 2.2.2).

All in all, this brief historical account of the LCM primarily aimed at showing that this model represents a reconciliation between often far apart linguistic paradigms. Basically, it brings together two opposite perspectives: the **projectionist** approach, represented by the Role and Reference Grammar or by Dik's Functional Grammar; and the **constructional** approach, as propounded by Goldberg (1995, 2006) or Pustejovsky (1995), amongst others.

The main aspect of the LCM we are interested in in this PhD work, is the use of **lexical templates** for the representation of predicates. After devoting some time

[13]For a detailed account of the model the reader can refer to http://www.lexicom.es

to spelling out the theories of RRG and the Generative Lexicon, we will describe the LCM lexical templates, as we will employ them later in chapter 5.

2.2.1 Role and Reference Grammar

The RRG is a monostratal theory of syntax which posits a single syntactic representation for each sentence, which is linked to a semantic representation by means of a set of linking rules called linking algorithm (see figure 2.4). By *linking* it is understood the relations between the meaning of a predicate and the different morphosyntactic patterns that a predicate can subcategorize. The RRG linking system is bidirectional, in that it maps both from syntax to semantics and from semantics to syntax. The two directions represent what speaker and hearer do in a communicative situation.

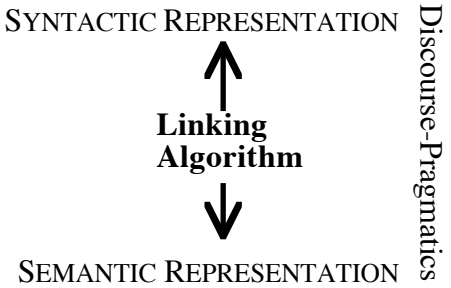


Figure 2.4: RRG linking algorithm

The RRG uses a decompositional system to represent both the semantic structure and the argument structure of verbs and other predicates (Ruiz de Mendoza Ibáñez and Mairal Usón, 2008). This representation is built around the notion of *Aktionsart* categories (Vendler, 1967) that divides verb classes into states, activities, achievements, semelfactives, and accomplishments together with their corresponding causatives.

- **States** denote static situations that are atelic, i.e., that do not have a temporal boundary (e.g., know, have, love).
- **Activities** are verbs that encode dynamic states of affairs that do not have a temporal boundary (atelic) (e.g., walk, think, drive).
- **Achievements** express changes of states that are telic (have a temporal point), and that do not take place over an extended period of time, i.e., they express momentaneous changes of states (e.g., shatter, pop, explose).
- **Accomplishments** denote changes of state that are telic and that have duration in time (e.g., freeze, dry, learn).

The complete list of categories and its corresponding logical structures can be seen in table 2.4. States and activities are primitives, while accomplishments

and achievements consist of either a state or activity that involve a result or consequence introduced by the operators BECOME or INGR[14], respectively. **Active accomplishments** represent telic uses of activity verbs (e.g., walk home, drink a beer, march to the field). **Semelfactives** are also a specific type of activities that encode punctual events that do not result in a state, for example, flash, glimpse, or sneeze. Finally, **causatives** can participate in any logical structure, in which the cause of a state, activity, achievement or accomplishment is explicitly mentioned. The logical structure of these sort of derived categories will also be included in table 2.4.

Verb Class	Logical Structure (LSs)
STATE	predicate' (x) or (x,y)
ACTIVITY	do' (x, [predicate' (x) or (x, y)])
ACHIEVEMENT	INGR predicate' (x) or (x,y), or INGR do' (x, [predicate' (x) or (x, y)])
SEMELFACTIVE	SEML predicate' (x) or (x,y), or SEML do' (x, [predicate' (x) or (x, y)])
ACCOMPLISHMENT	BECOME predicate' (x) or (x,y), or BECOME do' (x, [predicate' (x) or (x, y)])
ACTIVE ACCOMPLISHMENT	do' (x, [predicate' (x, (y))]) & INGR predicate' (z, x) or (y)
CAUSATIVE	α CAUSE β, where α, β are LSs of any type

Table 2.3: RRG logical structures

The variables x and y correspond to the actor and the undergoer of an action. The logical structures of the *Aktionsart* categories permit the identification of the arguments of the predicate that will be projected from the semantic meaning, i.e., the semantic representation will determine to a large extent the syntactic representation of the clause. Examples of English verbs with the different logical structures are given in table 2.4 for the sake of clarity[15].

Coming back to the linking algorithm, we pointed out the existence of two directions. One direction can be said to go from the syntactic representation of the predicate to its semantics. This direction describes the workflow followed by the hearer of an utterance. Conversely, in the opposite direction, the semantic representation of the predicate is *projected* from the lexical representation of the verb which determines to a large extent its syntactic structure. This workflow would be adopted by the speaker in a communication act, and is also the workflow that has received more attention in the RRG theory. As already explained in section 2.1, in this approach the syntactic structure of the sentence can largely be determined on the basis of the meaning or lexical representation of the arguments.

[14]INGR means ingressive, i.e., that denotes the onset of an action.

[15]From Van Valin (2005).

Verb Class	Examples	Logical Structures
STATE	Sam is a lawyer	be' (Sam, [lawyer'])
ACTIVITY	Lee drank beer	do' (Lee, [drink' (Lee, beer)])
ACHIEVEMENT	The vase shattered	INGR shattered' (vase)
SEMELFACTIVE	Lee sneezed	SEML do' (Lee, [sneeze' (Lee)])
ACCOMPLISHMENT	The water froze	BECOME frozen' (water)
ACTIVE	Paul ran to the store	do' (Paul, [run' (Paul)])
ACCOMPLISHMENT		& INGR be-at' (store, Paul)
CAUSATIVE	The dog scared the boy	[do' (dog, Ø] CAUSE [feel' (boy, [afraid'])]

Table 2.4: Examples and instances of RRG logical structures

The problem arises when a verb participates in a construction that cannot be inferred from its primitive lexical representation. Or, what is the same, when the argument structure of a predicate is insufficient to explain the occurrence of one constituent. One way out is to posit an additional verb sense to account for the participation of the verb in a different alternation. However, at this stage, the same authors of the RRG propose to incorporate some mechanisms of *constructionist* approaches, specifically the notion of co-composition from the Generative Lexicon, so that the "new" semantics of the verb can be derived from a combination of the semantics of the verb and the semantics of the new constituent.

Let us exemplify this. As we have seen in the examples in table 2.4, *Lee drank beer* is an **activity** because it encodes a dynamic and atelic state of affairs. However, *Lee drank **a** beer* would become an **active accomplishment** because the end of the activity happens when the beer is drank, so that it has a temporal boundary. A projectionist theory would have to account for all the structures in which a certain verb can occur in, whereas a constructionist approach would define an underspecified sense, and would maintain that further senses can be derived or generated by combining the semantics of the new constituents. In this example, the new constituent would be represented by the direct object of the sentence being a referential argument (**a** beer), as opposed to the primary object of the verb drink (beer), which does not refer to a specific participant, but characterizes the action.

This generative mechanism, as well as the main tenets of the Generative Lexicon, will be further explained in the next section. Finally, in section 2.2.3, we will see how the different approaches presented so far are combined in the lexical templates proposed by the LCM.

2.2.2 The Generative Lexicon

The Generative Lexicon (Pustejovsky, 1995) is "both a linguistic theory of semantic interpretation and a theory of lexical semantic knowledge representation" (Buitelaar, 1998: 29). This means that it provides some guidelines on how to interpret the semantics of lexical items, but it also provides the mechanisms to represent

that semantics. The main idea of the Generative Lexicon is that the meaning of lexical items should not be decomposed in a set of senses, but that the different senses of a word are composed or activated each time depending on the context. In this way, different word senses are conflated in a single "meta-entry" (Buitelaar, 1998: 62).

One of the main objectives of the theory of the Generative Lexicon is to account for polysemy in natural language. Pustejovsky believes that lexical items have a semantic representation conventionally assumed, but that it can be modified due to certain constraints (one sense is activated against other potential senses).

In order to explain the property of multiple subcategorizations being associated with a common underlying meaning, the author defines a complex machinery of lexical semantic descriptions that allow him to explain how the basic meaning of a lexical item can allow or generate different readings according to the types of complements that accompany it. This "allowing of different readings" also called "underspecification" is explained through a complex machinery of lexical semantic knowledge representation.

A generative lexicon is a system that involves four levels of semantic representations (Pustejovsky, 1995: 58):

1. **Argument structure**, which specifies the number and type of arguments that a lexical item carries. The number of arguments has a correspondence with syntactic constituents. Arguments can be divided into *necessary* and *optional*. Among the optional arguments we find *default arguments*, which are not necessarily expressed syntactically, and *shadow arguments*, which can only be expressed by discourse specification.

2. **Event structure**, which characterizes not only the basic event type of a lexical item, but also internal, subeventual structures. Events can be classified into three types: states, processes and transitions.

3. **Qualia structure**, which represents the different modes of predication possible with a lexical item. The *qualia* structure determines the meaning of a lexical entry, and determines its systematic polysemy by representing how its different semantic aspects are related to each other (Buitelaar, 1998: 32).

4. **Lexical Inheritance Structure**, which identifies how a lexical structure is related to other structures in the dictionary.

Consequently, the semantics of a lexical item can be defined by the following tuple:

α = <A, E, Q, I>

> (...) where A is the argument structure, E is the especification of the event structure, Q provides the binding of these two parameters in the *qualia* structure, and I is an embedding transformation, placing α within a type lattice, determining what information is inheritable from the global lexical structure (Buitelaar, 1998: 62).

The semantics of the verb are seen as centrally defined by the *qualia*, but constrained by the arguments and the events.

In the following we will explain the four levels of semantic representation with more detail. In the analysis of the verbal structures that we have identified as LSPs in this research work, we will make use of the argument, event and *qualia* structures as proposed by Mairal Usón and Ruiz de Mendoza Ibáñez (2008) in the most recent version of the LCM model.

Argument structure. According to Pustejovsky (1995: 63) "the argument structure for a word can be seen as a minimal specification of its lexical semantics. By itself, it is certainly inadequate for capturing the semantic characterization of a lexical item, but it is a necessary component". This view coincides with the consideration of arguments in RRG's logical structures.

Pustejovsky distinguishes three types of arguments (Pustejovsky, 1995: 63-64):

- True arguments (ARG): syntactically realized parameters of the lexical item (e.g., **John** arrived late)
- Default arguments (D-ARG): parameters which participate in the logical expressions in the *qualia*, but which are not necessarily expressed syntactically (e.g., John built the house **out of bricks**)
- Shadow arguments (S-ARG): parameters which are semantically incorporated into the lexical item. They can be expressed only by operations of subtyping or discourse specification (e.g., Mary buttered her toast **with an expensive butter**)

Event structure. Regarding events, the author says (Pustejovsky, 1995: 68)

> ...finer-grained distinctions are necessary for event descriptions in order to capture some phenomena associated with aspect and *Aktionsarten*. Assuming this is the case, we need a means for both representing the subeventual structure associated with lexical items while expressing the necessary relation between events and the arguments of a verb.

Pustejovsky argues that in a verb several events take place which are intertwined, and that it is necessary to identify those events (or subevents) because each of them may be given prominence in different contexts. Pustejovsky refers to this as *event headedness*, i.e., the property of events of being ordered not only temporally, but also according to their prominence.

For example, the verb *build* contains two subevents e_1 and e_2. e_1 represents the *activity* or *process* of building a house, whereas e_2 represents the *state* or *result* of building a house. This justifies the construction of sentences like *He is building a house*, in which the process is headed, against constructions like *He built a house*, in which the final result is given prominence.

This ordering restriction (RESTR) that takes place on events can be of three types:

- *exhaustive ordered part of* (e$<\infty$), one subevent preceding the other. E.g.: build, arrive (when building a house, the action of building precedes the result or state of a house being built)
- *exhaustive overlap part of* (e$o\infty$), two subevents occurring simultaneously. E.g.: accompany (the action of going with someone somewhere and being with that person occur at the same time)
- *exhaustive ordered overlap* (e$<o\infty$), two simultaneous subevents, where one starts before the other. E.g.: walk, begin (the action of starting walking and the walking action are simultaneous, but the one has happened before the other)

If the subeventual structure is left *underspecified*, this means that none of the subevents has been *headed* (in Pustejovsky's terminology). As a result of this, we would be dealing with a polysemic verb in which several interpretations are possible. However, if we focus on or head one of the subevents, we will be able to account for the participation of the verb in a specific alternation. Here we could also refer to the previous example of the verb *build* in order to justify its participation in different verbal alternations, depending on which subevent, the activity or the state, is headed.

Finally, we should refer to the generative mechanism of *co-composition*[16]. Co-composition is defined as an operation in which verbal complements carry information that acts on the verb, taking the verb as argument and shifting its event type. This is a further mechanism to capture polysemy and obviate the need for listing multiple senses. To put it in simple words, co-compositionality is a form of coercion or selection mechanism that picks out knowledge from the verbal complement (as stored in its *qualia* structure) and influences the verbal semantics by also selecting one of its senses.

Let us illustrate this operation by means of the verb *float*, whose reading can change from an activity to an active accomplishment. According to Van Valin (2004) *float* would be defined as a process verb in which something is moving as in *The bottle is floating in the river*. If the verb is combined or composed with a prepositional phrase of the sort *into the cave*, as in *The bottle floated into the cave*, the result is an event change in which the verb has become an active accomplishment, because a final result has been achieved. The new meaning that has been generated or derived by co-composing the verb with a complement specifies the lexical meaning of the original verb and exists in the phrase *float into*. We will see further examples of this generative mechanism in section 5.2.

[16]The other two generative devices or operations that generate polysemy are type coercion and selective binding, but they will not be dealt in this PhD work. For more information on this we refer the reader to Pustejovsky (1995), Chapter 7 on *Generative Mechanisms in Semantics*.

***Qualia* structure**. According to the Generative Lexicon, lexical items encode semantic information in the *qualia*, i.e., by stating the *qualia* structure of a certain lexical item we are able to define the essential aspects of a word's meaning. In the following we define the four essential aspects of the *Qualia* structure, which permit to capture the meaning of a lexical item (Pustejovsky, 1995: 76):

- FORMAL: that which distinguishes a lexical item within a larger domain (hyponymy)
- CONSTITUTIVE: the relation between an object and its constituent parts (meronymy)
- TELIC: its purpose and function (function)
- AGENTIVE: factors involved in its origin or "bringing about"(causality)

These four roles are considered necessary for the semantic description of a lexical item, although not all of them are always present in one and the same lexical item. Let us illustrate the *qualia* structure of the noun *novel* (adapted from Pustejovsky (1995: 78).

novel	
QUALIA	FORMAL = book (x)
	TELIC = read (y, x)
	AGENT = write (z, x)

Table 2.5: *Qualia* structure of the noun *novel*

This representation specifies that the lexical item *novel* is defined as a book, whose function is to be read, and which is the result of a writing process by a writer. This representation enables two actions: on the one hand, to encode information about properties and activities related with the noun novel, and, on the other, to interpret sentences like *Mary began the novel*, in which we understand that she began to *read* the novel, because we know that Mary is not a writer.

Since we are interested in the semantic description of verbs, in the following we present the representation of the event (EVENTSTR), argument (ARGSTR) and *qualia* structure (QUALIASTR) of the verb *build* in table 2.6 (adapted from Pustejovsky (1995: 103)).

To *build* is an accomplishment verb that involves a process and a resulting state ordered by the *exhaustive ordered part of* ($<\infty$) restriction. The sub-event headed is the process one, which means that the activity of building a house is *headed* against the final result. Then, the argument structure is represented by three arguments, two syntactically realized arguments (the animate individual that builds the house and the resulting artifact, i.e., the house), and one default argument (the material of which the house is made of), which is not syntactically realized. The

semantic properties of the arguments involved in the event are specified by means of the *qualia* structure. Three *qualia* structures are present in this verb. The formal *quale* indicates the final result of the building activity, and involves ARG_2. The agentive *quale* describes the process and involves ARG_1, the person carrying out the activity, and the default argument (*D-ARG*), which describes the material used in the building of the artifact.

build	
EVENTSTR:	E_1 = e_1: [process] E_2 = e_2: [state] *Restr* = [$<\infty$] *HEAD* = [e_1]
ARGSTR:	ARG_1 = [x: animate individual] ARG_2 = [y: artifact] *D-ARG* = [z: material]
QUALIASTR:	Q_F = [exist (e_2, y)] Q_A = [build_act (e_1, x, z)]

Table 2.6: Event, argument and *qualia* structures of the verb *build*

By allowing a verb's membership in a particular semantic class to emerge from the composition of the sentence it appears in, we obviate the need to enumerate separate senses for the distinct semantic classes associate with that verb (Pustejovsky, 1995: 197). Such a representation of verbs in the Generative Lexicon provides a more granular semantic description of lexical items, while also connecting directly with the syntactic expressiveness of the semantic types.

Mairal Usón and Ruiz de Mendoza Ibáñez (2008) saw in this formalism some similarities with the lexical templates proposed in the LCM model, mainly, that both representations included an *event structure* description, which to a large extent coincides with the *Aktionsart* module of the RRG, and a semantic module or *qualia structure* that permits to account for the semantic properties of arguments and events.

In the next section, we explain the current configuration of the LCM Lexical Templates, and also present the template that we will use in our analysis of LSPs in chapter 5 of this PhD work.

2.2.3 LCM Lexical Templates

In an attempt to provide RRG's logical structures with a richer semantic decomposition, the LCM proposed the notion of lexical template. Originally, lexical templates consisted of two modules: the semantic module, and the logical representation or *Aktionsart* module. The basic representation of a lexical template looked as follows:

predicate: [SEMANTIC MODULE <lexical functions>] [AKTIONSART MODULE <semantic primes>]

The Aktionsart module relied on the inventory of logical structures as developed in RRG. However, every predicate was semantically decomposed in an attempt to provide "typologically valid representations" (Mairal Usón and Ruiz de Mendoza Ibáñez, 2009: 164). For this aim, Wierzbicka's semantic primes were employed in the definition of predicates. The Semantic module, on the other hand, attempted to capture the semantic and pragmatic properties of the predicate, and it made use of Mel'cuk lexical functions.

Let us analyze the example of the verb **realize**, defined as *bring vividly before the mind*, according to The New Shorter Oxford English Dictionary[17]. The lexical template that would correspond to this verb is represented below (extracted from Ruiz de Mendoza Ibáñez and Mairal Usón (2006a)):

realize: [INSTR (see)$_{12}$ LOCin(body_part: mind) & CULM$_{12[All]}$] [**know'** (x, y)]
where x=1 and y=2

The rightmost hand part of the template includes the RRG logical structure representation of a *state* predicate with two arguments, x and y. The primitive predicate describing realize is *know*, a mental predicate according to Wierzbicka's primitives (see section 2.1). The semantic and pragmatic properties are shown in the leftmost hand part of the template formalized by the lexical functions, INSTR (see)$_{12}$ LOCin(body_part: mind) expressing that the cognizer (x) *sees* a mental percept in his or her mind, conceptualized as a location and represented as an abstract body part that is in a partitive relation to *body*. Then, CULM expresses the culmination of the process, as in the verbs of understanding introduced in section 2.1.

In the new version of the LCM lexical templates proposed by Mairal Usón and Ruiz de Mendoza Ibáñez (2008), the two modules are maintained, but differently structured and incorporating features from Pustejovsky's Generative Lexicon, particularly, the event and *qualia* structures. For the sake of clarity, we will reproduce the same example presented above with the verb *realize*, but now according to the new representation.

In this new representation, we clearly distinguish two subevents: the perception event and the understanding event. The first event is encoded as an agentive *quale*, as it is the kind of action performed to obtain knowledge, and involves the two arguments (x, y). The understanding event is encoded in the formal *quale*, and also involves the two arguments.

As pointed out by the authors, this new representation formalism has the ad-

[17] Oxford University Press 1973, 1993.

realize	
EVENTSTR:	**know'** (x, y)
QUALIASTR:	Q_A: LOCin(body_part: mind, see' (x, y))
	Q_T: CULM know' (x, y $<_{All}>$)

Table 2.7: LCM lexical template for the verb *realize*

vantage of explicitly assigning semantic descriptions to the event and argument structures of the predicate. By way of illustration, in the *realize* example the lexical function ALL is integrated as a restriction over the second argument (y) in the logical structure. Furthermore, if the same verb participates in different subcategorization frames, the mechanisim of foregrounding or *heading* one of the *quales* in the *qualia* structure acts as constrain and determines the specific syntactic realization of the predicate.

This is clearly exemplified in chage-of-state verbs like *break* (Mairal Usón and Ruiz de Mendoza Ibáñez, 2008). Break is decomposed in a formal *quale* and an agentive *quale*. The formal *quale* encodes the result of the break activity, that is, the second argument (y) being broken. The agentive *quale* refers to the actual act of breaking something and involves the first argument, the actor (x) of the breaking act. If the agentive *quale* is headed, the verb can be constructed in a transitive (causative) structure as in *Peter broke the window*. If, on the other hand, the formal *quale* is headed, the verb will subcategorize an intransitive structure of the form *The window broke*.

As the same authors put it (Mairal Usón and Ruiz de Mendoza Ibáñez, 2008):

> (The fact that the semantic and eventive modules are closely intertwined) has interesting consequences in the semantics-to-syntax mapping possibilities of a predicate since, as pointed out in Pustejovsky (1995: 101-104), individual *qualia* compete for projection, and there are mechanisms such as foregrounding or 'focalizing' a single *quale* of the verbal semantic representation.

For the purposes of this work **we combine the structure of the LCM templates as proposed in** Mairal Usón and Ruiz de Mendoza Ibáñez (2008) **with the proposal made by Pustejovsky, in which event and argument structure are separately specified**. We believe that by keeping event, argument and *qualia* structures independent from each other, we are able to account for each of these structures in more detail. However, we adopt Mairal and Ruiz de Mendoza's event structure based on the *Aktionsart* module and Wierzbicka's semantic primitives, because this formalism allows us to encode those meaning elements that differentiate one predicate from others in the same domain according to the lexicon philosophy. In this way, we make a modest contribution to the research on models that aim at describing and explaining meaning construction, as is the case of the LCM.

The template designed for this PhD work can be seen in table 2.8. LCM EVENTSTR stands for the *Aktionsart* module. GT stands for Generative Lexicon and indicates that event (EVENTSTR), argument (ARGSTR) and *qualia* (QUALIASTR)

Lexical Template	
verbal pattern	infinitive form
LCM EVENTSTR	*Aktionsart* module
GT EVENTSTR:	$E_1 = e_1$: [state, activity, achievement, etc.] $E_2 = e_2$: [state, activity, achievement, etc.] *Restr* = [$<\infty$, $o\infty$, $<o\infty$] *HEAD* = [e_1 e_2]
GT ARGSTR:	ARG_1 = [human, artifact, class, etc.] ARG_2 = [human, artifact, class, etc.] *D-ARG* = [human, artifact, class, etc.] *S-ARG* = [human, artifact, class, etc.]
GT QUALIASTR:	Q_F = [hypernymy-hyponymy] Q_C = [meronymy] Q_T = [function] Q_A = [origin, cause]

Table 2.8: Lexical template proposed for the analysis of *candidate verbal patterns*

structures are represented according to the formalisms defined by Pustejovsky (1995: 105).

The analysis of the verbal predicates which are investigated in this PhD thesis will be conducted in chapter 5, section 5.3 *LSPs on the light of the Lexical Constructional Model*, once we have described the strategies for identifying *candidate verbal patterns* that convey the knowledge captured in a subset of selected ODPs.

2.3 Summary

This chapter had the objective of describing the theoretical underpinnings in which the present work is supported. We understand the interaction between semantics and natural languages in the broad framework of functional-cognitive theories, which propound the analysis of function and meaning of language in context, over form. In particular, we rely on Cognitive Linguistics' *experientialists* account to describe how we understand knowledge as represented in ontologies and its relation to the language used to convey and organize that knowledge. We support the idea that ontologies can be understood as products of language, or what is the same, of how a certain community of users understands a parcel of the world under certain conditions.

By committing to these theoretical assumptions, we explain the approaches taken in the two principal contributions of this thesis, namely, the *multilingual LSPs-ODPs pattern repository* for knowledge acquisition and ontology modeling, and the LIR for ontology localization.

For the first research topic, we rely on the LCM, a model that brings function-

alists, cognitivists and constructionists theories together to describe the meaning of predicates. By applying this model to the analysis of candidate verbal patterns that convey the knowledge captured in ODPs, we systematically describe those patterns that present interesting uses (such as polysemy) to establish a more reliable correspondence to its ontological structure.

The LCM model is then devoted a whole section (section 2.2). The LCM relies in its turn on two further models that it adapts for its own purposes, and which have also received some attention in sections 2.2.1 and 2.2.2, respectively. These models are the RRG and the Generative Lexicon. By detailing the main principles of these models, our purpose was to facilitate the understanding of the lexical template that we propose for the analysis of some of the most interesting candidate verbal patterns, as will be shown in section 5.3.

Regarding the second research topic, we rely on cognitivist approaches to categorization to explain how we understand the different types of ontologies that exist regarding the domain of knowledge they represent. Here we distinguish between *internationalized* or *standardize* domains of knowledge vs. *culturally-influenced* ones. In the case of culturally-influenced domains of knowledge, boundaries of categories may not be shared among the linguistic communities involved in a localization project. Disagreements in this sense can be accounted for at the lexical layer, as we propose in chapter 9.

Part I

Multilingual Lexico-Syntactic Patterns for Ontology Modeling

Chapter 3

Knowledge Acquisition for Ontology Modeling

The importance of language for the extraction of knowledge and information has led to the use of texts in the construction of several types of resources, such as dictionaries, terminologies, or ontologies, to mention but a few. This process is assumed to reduce the time spent on knowledge acquisition directly from domain experts. The need for automating the process of knowledge acquisition has constituted a field of research for more than twenty five years. In Ontology Engineering, the knowledge acquisition process has been mainly applied to learning the terms or concepts relevant for a given domain as well as the relations holding between them. The activity of relying on (semi-)automatic methods to transform unstructured (e.g. plain text), semi-structured (e.g. folksonomies, HTML pages), and structured data (e.g. data bases) in conceptual structures is known as Ontology Learning (M. C. Suárez-Figueroa, 2010). The task of learning the extensions or instances for concepts and relations is commonly known as Ontology Population (Cimiano, 2006: 26). In the context of this thesis, we will use the more general term *knowledge acquisition* because we will be analyzing approaches in the Terminology field, and also because the level of automation is different in each approach.

In this dissertation work, we are mainly interested in the acquisition of concepts and the relations holding between them from texts in different languages with the purpose of building an ontology or enriching it with new concepts and relations. We have investigated two types of approaches

1. Approaches relying on **linguistic patterns**, specifically verb-centered patterns, to automatically extract information from texts

2. Approaches involving domain experts in both the elicitation process and the ontology development process relying on the so-called **controlled languages**

As we will explain in the following sections, both approaches have drawbacks that could be overcome by combining some of the strategies that they explore.

Approaches based on linguistic patterns have to deal with the problems imposed by the processing of unstructured data in NL. These problems are mainly related with language ambiguities and invalid retrieved contexts. Apart from that, they do not offer any support in the modeling task, i.e., once knowledge has been obtained, users have to decide how to model it in the ontology, which demands expertise in ontology modeling.

Methods based on controlled languages are intended at domain experts without knowledge engineering background, who are assumed to learn the controlled language to formulate sentences that are directly translatable into formal representations. Even so, these approaches demand that users not only learn the controlled grammar, but also understand what they can model with it, so that they can make the right modeling choice.

Our solution suggests that we should **rely on NL**, specifically on verbs, to obtain the knowledge to be included in the ontology, because verbs are the major conveyors of conceptual relations (as shown by most of the studies that have dealt with this research issue and which are described in section 3.1.1). Instead of directly relying on corpora, we propose to **involve domain experts** in the ontology modeling task. In this way, we aim to avoid some of the problems posed by language ambiguities and invalid retrieved contexts. On top of that, we should also support users in the task of selecting the formal representation that better meets their modeling needs. For this purpose, we build on the repository of LSPs associated to ODPs to propose a method and a system to perform knowledge acquisition and ontology modeling in a semi-automatic way.

The present chapter will then be structured around two types of knowledge acquisition: knowledge acquisition from text and knowledge acquisition from experts. Our objective is to give an outline of the main trends in these areas and point out some of the drawbacks that could be overcome by a **hybrid approach** involving strategies from the two approaches. This intermediate method, which is one of the main contributions of this PhD, will be described in chapter 4.

3.1 Knowledge Acquisition from Text

One of the main objectives of knowledge acquisition from text is to reduce the time and efforts necessary in the development of resources such as ontologies or terminologies. This has been regarded as a critical bottleneck in the consolidation of ontologies as the basis for the Semantic Web. Several approaches have been investigated in the acquisition of knowledge from text using different methods and techniques, from which we distinguish three main trends:

- approaches relying on statistical measures about co-occurrence of terms
- approaches that apply machine learning algorithms
- approaches relying on regular expressions that usually convey a relation of interest, the so-called pattern-based approaches

Statistical and machine learning approaches are often used in combination with linguistic-based methods, what makes it difficult to draw a sharp line between the different methods. Broadly speaking, statistical methods can be said to be mainly based on calculating statistical metrics about the frequency with which some words appear related to others in the same context. Examples of methods that use statistical measures to enrich ontologies with further concepts can be found in Agirre et al. (2000) or Faatz and Steinmetz (2002). Other approaches such as Xu et al. (2002) or Schutz and Buitelaar (2005) combine linguistic and statistical processing for relation extraction.

Ontology Learning methods based on machine learning algorithms make use of regularities within example data to make inferences about unknown data (Cimiano, 2006: 62). A good exponent of the use of machine learning algorithms in extracting conceptual relations is the work by Maedche and Staab (2000, 2001). In this approach, the authors firstly rely on the analysis of syntactic dependences of terms, and then determine the confidence of the discovered relation by means of an algorithm that uses concepts and relations in a domain taxonomy, and the learned concept pairs.

One advantage of **pattern-based approaches** over statistical methods is the possibility of identifying the type of relation existing between two elements with a high degree of confidence. Within the pattern-based approaches, which are the ones that interest us in the present research, we can make a basic distinction between approaches based on verbal expressions and approaches that exploit the internal relations of noun phrases. A further distinction can be made between research studies focusing on taxonomic or meronymic relationships (Marshman et al., 2002), (Cimiano et al., 2005), and others that put the emphasis on the identification of non-taxonomic relations, also called *ah-hoc* relations, which are specific of certain domains (Marshman and L'Homme, 2006), (Sánchez and Moreno, 2008).

In the following we will give a brief account of the **state of the art on pattern-based approaches for learning conceptual relations with the objective of building ontologies and terminologies**. We will focus on the type of patterns employed (verb-oriented vs. noun-phrase based) the type of relations discovered (taxonomic, meronymic, ad-hoc relations), and the language of the patterns.

Pattern-based approaches

The idea of applying linguistic patterns to the discovery of semantic relations was introduced in Computation by Marti Hearst in the early 1990s (Hearst, 1992). The goal of her research was the automatic acquisition of hypernym-hyponym relations from corpora to enrich lexical resources such as WordNet. Hearst defined lexico-syntactic patterns as linguistic structures that are "easily recognizable, that occur frequently and across text genre boundaries, and that indisputably indicate the lexical relation of interest".

An example of a pattern used by Hearst is

NP_0 such as {NP_1, NP_2 ... (and|or)} NP_n

where NP stands for Noun Phrase followed by the conjunction *such as*, and then a list of Noun Phrases linked by the conjunctions *and* or *or*. An example of a sentence in English containing that kind of construction from Hearst' work is *The bow lute, such as the Bambara ndang, is plucked and has an individual curved neck for each string*. The rest of patterns defined by Hearst with the aim of automatically identifying hypernym-hyponym relations are reproduced in figure 3.1 for convenience.

such NP as {NP,}* {(and|or)} NP *…works by such authors as Herrick, Goldsmith, and Shakespeare…*
NP {, NP}* {,} or other NP *Bruises, wounds, broken bones or other injuries…*
NP {, NP}* {,} and other NP *…temples, treasuries, and other important civic buildings…*
NP {,} including {NP,}* {or|and} NP *All common-law countries, including Canada and England…*
NP {,} especially {NP,}* {or|and} NP *…most European countries, especially France, England and Spain…*

Figure 3.1: Hearst's patterns

Since then, several authors have applied lexico-syntactic patterns for the automatic discovery of semantically related lexical items in English with different purposes. In particular, some approaches have taken the set of patterns described by Hearst and have complemented them with patterns based on verbal constructs or noun phrases to discover hypernym-hyponym relations, see for instance Finkelstein-Landau and Morin (1999), Snow et al. (2004), or Etzioni et al. (2004). The main reason for extending the original set of Hearst patterns is that these patterns have proven to appear rarely in texts, a drawback that Cimiano and Staab (2004) try to overcome by using the Web as corpus.

Others have applied very much the same methodology as Hearst for the identification of meronymy relations. This is the case of Berland and Charniak (1999). Examples of prepositional phrases conveying the part-whole relation are *...the basement **of** a building...* or *...basement **in** a building....* However, these authors include a verbal-centered pattern (*...is in...*) without explicitly referring to it, and only focusing on the preposition. It is the case of the construction ... *the basement **is in** a four-story apartment building...* (Berland and Charniak, 1999). Neither Hearst nor Berland and Charniak do explicitly mention the patterns *is a* or *is in*, probably because its high degree of ambiguity requires to rely on further linguistic elements, such as definite or indefinite articles, prepositions, etc. Finally, it is worth mentioning that we also find translations of Hearst patterns to other languages, such as

German (*NP und andere NP; NP bzw. NP; NP sowie NP, NP wie NP*) in Xu et al. (2002).

Equally, some effort has gone to the discovery of patterns based on head nouns or noun phrases. Here we find the work of Hahn and Schnattinger (1998) in which constructions such as the genitive case in English *NP's NP* or appositions *NP NP* are investigated. In the same line we find the work of Iwanska et al. (2000) for taxonomy relations or the research by Vela and Declerck (2009) for taxonomy and meronymy relations in the German language. The main drawback of these approaches is related with the difficulties to carry out the task without supervision, since a lot of spurious and ambiguous results are obtained. For instance, genitives (*the girl's mouth*) can also express possession (*Mary's toy*), kinship (*Mary's brother*), and many other relations (Girju and Moldovan, 2003).

Next, we will refer to the so-called *sentence-level patterns* or *verb-centred patterns*, i.e., those patterns in which verbs carry the semantics of the relation. Reliability is the main feature of these patterns, although they have proven scarce in recall. Needless to say that most of the research on linguistic patterns has been widely done for the English language, specifically in the Ontology Engineering domain, although we also find some works in French and Spanish in the Terminology area. In our case, we aim at identifying the same set of linguistic patterns for English and Spanish, in an attempt for promoting the construction of a multilingual repository.

3.1.1 Verb-centred Patterns for Knowledge Acquisition

In this section our purpose is to give an overview of knowledge acquisition approaches that rely on **verb-centered patterns** to extract conceptual relations from texts. As stated in Schutz and Buitelaar (2005), the role of verb as a central connecting element between concepts is undeniable.

Conceptual relations are defined in Feliu Cortés (2004: 27) as elements "that link two or more specialized knowledge units in a particular subject field", and they are formally represented as **R (a b, n)**, where **R** represents the relation, **a** and **b** are knowledge units, and **n** foresees the case when a relation links more than the two elements **a** and **b**.

From an Ontology Engineering viewpoint, it can be argued that the knowledge units correspond to concepts or classes that specify the domain and range in an ontological relation, and that the relation between them can be represented by an object- or data type property. From now on, we will refer to these relations as semantic or conceptual relations.

Our main objective in this section is to give an account of the different types of semantic relations and languages that have been investigated, and how these approaches have influenced our work. Most of the approaches dealt in this section have their roots in the Terminology domain. Some of them have been applied later on to the discovery of conceptual or ontological relations. Others follow Hearst's legacy of automatically enriching on-line lexical resources and have been thought

for the enrichment of ontologies with new relations, concepts and instances.

In the first group we find the work of the French TIA[1] special interest group. In this context, Aussenac-Gilles et al. (2000) propose TERMINAE and CAMÉLÉON (Séguéla, 2001). TERMINAE is a method to guide ontology modeling from texts by using NLP tools. It was created to support the building of terminologies and ontologies. In this framework, CAMÉLÉON was developed as a tool for identifying semantic relations from texts by applying a pattern-based approach (Aussenac-Gilles, 2005).

The TERMINAE method consists of four steps : (i) *corpus definition*, (ii) *text analysis* with NLP tools (from which CAMÉLÉON is used to extract terms and relations), (iii) *normalization* (concept identification and description with the help of semantic relations) and (iv) *formal representation* (in a language close to Description Logic). The CAMÉLÉON pattern repository includes more than 100 patterns. Some of them developed by the authors themselves, others adapted from other works (Rebeyrolle and Tanguy, 2000) and (Marshman et al., 2002), in which terminologists and linguists have identified patterns manually or with the help of corpora processing tools. Some of the French patterns included in CAMÉLÉON can be seen in figure 3.2. Most of the patterns convey the relations of *hyperonymy-hyponymy* and *meronymy*, whereas a smaller group of patterns suggests *function* relations.

Authors	Verbs and verbal phrases	Pattern types
Aussenac-Gilles and Jacques, 2006	définir, être-un, et Adv, sorte de, inclure, partie de, situé dans, c-à-dire	hyperonymy-hyponymy, meronymy, synonymy
Rebeyrolle and Tanguy, 2000	désigner, appeler, signifier, être-un	definitional patterns
Marshman et al., 2002	est (Adv/un/le), appelé* + terme, il s'agit (là) d*, nommé, terme + constitu*, être considéré* comme, terme + design, y compris*, il y a type* d*, catégorie*, sorte*, distinger clase* d*, constitue*, regroup* sont les suivant*, conten*/contien*, compos* d*, comport*, posséd*/possèd*, constitu* de, compren*, équipé d*, disposer d*, partie*, être dote* d*, présen*, compt*, adjoindre, assemblage d*, déten*/détien*, divisé* en, formé* d*/par, inclu*, incorporé*, intégré* à, muni* d*, pourvu d*, rassembl*, retrouvé* dans, se subdivisier en ; faire (Adv) partie* d*, intégré à/dans, à base d*, formé* d*, héberge*, incorporé*, introduit* dans, rich* en; transmettre, effectuer, permettre d*/permet*, s'utiliser (pour/dans le sense de/comme), capable* d*, pour (+le/la) + verb, ser* (à/de), destiné* à, néccessaire* à, par + term, rôle*, fonction*, grâce à, par l'intermédiaire d*, conçu*, au moyen d*, consist* à, à l'aide d* + term, à travers, applicable* à, …	hyperonymy-hyponymy, meronymy, function

Figure 3.2: Verbal patterns in French included in CAMÉLÉON

[1] http://tia.loria.fr

An evaluation performed on the CAMÉLÉON system is described in (Aussenac-Gilles and Jacques, 2008). The purpose of this experiment was to evaluate the precision and recall of the patterns included in the repository depending on the domain of knowledge of the different corpora. One of the main conclusions reached in this experiment was that some of the patterns considered "generic" or highly productive showed performance variability depending on the domain of knowledge. They were relevant for certain domains but yielded very few contexts in others. Also depending on the domain, some polysemous patterns identified one relation or another. Therefore, a supervised approach was finally recommended, in which experts were expected to evaluate the contexts before enriching the ontology.

Marshman et al. (2002), also in the field of terminology, investigate ways to help terminologists extract conceptual relations automatically from corpora, and for this endeavor, they propose to rely on inventories of linguistic patterns. Their work focuses on the identification of patterns for French, and is based on the notion of *knowledge-rich contexts* (Meyer, 2001). Knowledge-rich contexts designate "a context indicating at least one item of domain knowledge that could be useful for conceptual analysis". In Marshman et al. (2002), the authors manually analyze texts in two domains -computers and genetics- with the aim of identifying verbal patterns that activate knowledge-rich contexts. Then, they propose the use of patterns to speed up the extraction of new knowledge-rich contexts by automating the process. They focus on patterns indicating hyperonymy, meronymy and function (see also figure 3.2). In subsequent studies, Marshman (2007) investigates so-called *ad hoc* relations in the medical domain, specifically relations of *association* and *cause-effect*, this time in English and French. With the aim of facilitating the identification of verbal patterns they use the TermoStat (Drouin, 2003) term extractor tool.

A similar approach is followed by Feliu and Cabré (2002) and Feliu Cortés (2004), also with the aim of supporting the process of terminology extraction. The authors focus on specialized corpora and identify patterns with the help of the Mercedes (Vivaldi, 2003) term identifier for the relations of similarity, hyperonymy-hyponymy, sequentiality, causality, meronymy, and association. See figure 3.3 for some examples of these patterns. In order to validate the set of identified patterns in Feliu Cortés (2004), the author extracts contexts that contain the patterns from a specialized corpora (Corpus Tècnic de l'IULA) by means of the BwanaNet corpus search engine [2] and obtains precision and recall data.

Next, we will refer to two further studies on patterns in Spanish for the automatic extraction of terms and contexts for terminology work. In the first one by Alarcón Martínez and Sierra Martínez (2003) and Sierra et al. (2008), the authors focus on the analysis of what they refer to as *definitional verbal patterns*. In a first stage, they work on the manual extraction of patterns from both machine readable dictionaries, and scientific and technical corpora, because they argue these are good containers of definitional contexts. Then, they establish a typology of defini-

[2] http://brangaene.upf.es/bwananet/indexes.htm

Authors	Verbs and verbal phrases	Pattern types
Feliu and Cabré, 2002	és a dir, assemblar-se a, diferenciar-se de, ser el contrari a, ser, considerar-se, ser com, :, iniciar-se en, produir-se en, tenir lloc a nivell de, quedar encarat amb, realitzar-se, situar sobre, registrar-se en/a/des de, evidenciar-se a, originar a, veure's en, ocórrer, aparèixer fins que, propagar-se, continuar per/fins, arribar a, apropar-se, allunyar-se ; dependre de, fer que, ser la causa de, deure's a, implicar, aparèixer, contribuir a, dependre de, donar lloc a, reforçar, provocar, augmentar, produir ; servir com a, realitzar-se amb ; definir-se X grups, constar de ; tenir, mostrar , incluoure, caracteritzar-se per, presentar ; correlacionar-se, correspondre a, intervenir, ...	similarity, hyperonymy-hyponymy, sequenciality, causality, meronymy, association

Figure 3.3: Feliu and Cabré's verbal patterns in Catalan

tions that consists of four types of definitions: analytical definition, synonymical definition, functional definition, and meronymical definition. See examples of each type of definition in figure 3.4. The authors also develop an automatic extractor of definitional contexts (ECODE, (Alarcón Martínez et al., 2008)) based on the previously identified patterns and decision trees, and evaluate it with a subset of patterns. They obtain a precision of over 50% of corrected classified elements, and find out that some verbs (such as *denominar*, *definir*, *entender* or *significar*) are more reliable than others, and that the decision tree inferences need to be improved. For further details on the evaluation see Sierra et al. (2008).

Authors	Verbs and verbal phrases	Pattern types
Sierra et al., 2008	referir, representar, ser, significar, caracterizar, comprender, concebir, conocer, considerar, definir, describir, entender, identificar, visualizar, emplearse, encargar, funcionar, ocupar, permitir, servir, usar, utilizar, componer, consistir, constar, contar, constituir, contener, incluir, integrar, equivaler, llamar, nombrar, ser igual, ser similar	definitional patterns (analytical, functional, meronymy, synonymy)

Figure 3.4: Sierra et al.'s definitional patterns in Spanish

Also for Spanish, but this time for the language of classification, we find the work of (Aguado de Cea and Álvarez de Mon, 2006) and (Alvarez de Mon and Aguado de Cea, 2006). The authors extract a set of verbs of classification from an *ad hoc* corpus of textbook documents related to the following subjects: histology, biology and zoology. Then, they use the concordance functionality of the corpus of current Spanish of the *Real Academia de la Lengua* (CREA) to manually analyze the previously extracted set of verbs. After a careful analysis of the concordances the authors conclude that it is necessary to talk about the phraseology of classification, and not only about verbs, because "it is really the combined presence of several lexical items which is really significant, as well as some other paralinguistic information". See figure 3.5 for some examples of verbs of classification.

Finally, we will refer to several studies that have focused on the meronymy re-

Authors	Verbs and verbal phrases	Pattern types
Aguado de Cea and Álvarez de Mon, 2006	clasificar en, clasificar según/de acuerdo con, clasificar como, (entre otros/el/la/los/las que) figurar, figurar (tipos/clases), distinguirse (por/atendiendo a), distinguir, dividirse en, (entre) comprender	classification

Figure 3.5: Aguado de Cea and Álvarez de Mon's classification patterns in Spanish

lation in Spanish. In this sense, we find Climent Roca's PhD Thesis (Climent Roca, 2000), in which an in-depth analysis of *partitive noun phrases* is performed with the aim of contributing to the representation of such relations in computational lexicons. Although focusing on noun phrases rather than verbal patterns, we find the analysis and classification of meronymy very useful.

Soler and Alcina (2008), more in line with the previous studies for the semi-automatic extraction of terms and relations, rely on a corpus of texts about ceramics to identify linguistic structures that relate *wholes* and their *parts*. For their purpose, they rely on the Concord application of the WordSmith tools (Scott, 1999) that helps them identifying knowledge rich contexts containing partitive relations. They take Winston et al.'s classification (Winston et al., 1987) of six subtypes of partitive relations (*component-object*, *member-collection*, *mass-count*, *material-object*, *characteristic-activity*, and *place-area*) and find out that most of the relations identified in their corpus relate to the types *component-object* and *material-object*. The first subtype is considered to be the most representative one in the partitive relations (Winston et al., 1987), whereas the justification for the abundance of the second subtype has to do with the domain dealt in the documents, in which many descriptions of chemical compounds in ceramics are present. The list of patterns is included in figure 3.6. A further classification of patterns according to the subtype of partitive relation they convey would have been of great value for this research. In any case, they carry out an evaluation for precision and recall of the identified patterns obtaining results of over 80% precision for 17 patterns out of 52.

Authors	Verbs and verbal phrases	Pattern types
Soler and Alcina, 2008	constituid* por, provist* de, confeccionad* con, ric* en, contien*, compone* de, compuest* por, consta* de*, a base de*, contempla* en, añad*, incorpora*, dotad* de, formad* por, conten*, forma* parte, obenid* a partir de, adiciona*, elaborad* con, parte de*, hech* con, inclu*, existencia de* en, proporción* de, basad* en, cunet* con, integrad*, presen*, es el elemento, emplea*, *divid*, introdu*, utili*, mineralogía es, dispon*, instala*, corr* a cargo de*, *fabrica*, interv*, usad* para, comprend*, conform*, prepar*, tiene* montad*, distingu*, realizad*, situad*, admit*, distribu* en, composici*	meronymy

Figure 3.6: Soler and Alcina's meronymy verbal patterns in Spanish

It should be observed that in most of the pattern repositories analyzed so far, the authors include some paralinguistic symbols, such as colon, to indicate that

they also appear in combination with linguistic patterns and are an invaluable help to convey the conceptual relations in question.

Now we turn to two research works that investigate the use of verb-centered patterns in English for the task of ontology learning.

The first of them is the work by Cimiano and Wenderoth (2005, 2007) in with they investigate the impact of some verbs that convey Pustejovsky's *qualia* structures[3] (Pustejovsky, 1995) in the acquisition of semantic relations from the Web. In this context, the authors propose a set of lexico-syntactic patterns for learning *formal*, *constitutive*, *telic* and *agentive* relations. They also reuse Hearst's patterns for the *formal* role, which they interpret as conveying the hyperonym-hyponym relation. The *constitutive* relation is understood by the authors as the relation between objects and its parts. The *agentive* role is identified with verbs denoting actions that bring the object into existence, and the *telic* role describes the function of the object. The set of patterns identified in this work can be seen in figure 3.7. The authors perform an experiment with users in which they want to find out which are the more "prototypical" verbal phrases for each *qualia* structure.

Authors	Verbs and verbal phrases	Pattern types
Cimiano and Wenderoth, 2007	a (x) is a kind of, (x) and other, (x) or other, such as (x), (x) and other, (x) or other, especially (x), including (x), (x) is made (up) of, (x) comprises, (x) consists of, purpose of a (x) is, a (x) is used to, to * a (x) new, to * a (x) complete, a (x) new has been, a (x) complete has been *	formal role (hyperonym-hyponym), constitutive role (meronymy), telic role (function), agentive role (originator)

Figure 3.7: Cimiano and Wenderoth's verbal patterns for *qualia* in English

The second approach in this context aims at automatically learning *ad-hoc* or *non-taxonomic* relationships also using the Web as corpus. This work is described

Authors	Verbs and verbal phrases	Pattern types
Sánchez and Moreno, 2008	suffer from, is associated with, is treated with, is caused by, accelerates, is associated with, is inherited, affects, causes, reduces, increases, develops	*ad-hoc* or non-taxonomic

Figure 3.8: Sánchez and Moreno's *ad-hoc* verbal patterns in English

in Sánchez Ruenes (2007) and Sánchez and Moreno (2008). The starting point of this approach are domain relevant concepts and taxonomical patterns. This step allows authors to retrieve a first set of related words by means of taxonomical or hyperonymy-hyponymy relations that will become the set of seed words for the subsequent unsupervised domain relation extraction. The taxonomical patterns used in the first stage are those defined by Hearst (1992), and the ones by Grefenstette (1992) based on noun phrases (combination of adjectives and nouns:

[3]For a detailed description of Pustejovsky's theory see section 2.2.2.

breast cancer). Some of the *ad-hoc* relations they discover for the medical domain can be seen in figure 3.8. Finally, they evaluate some or the learned relations manually, and the resulting ontologies against lexicons or ontologies in the same domain (WordNet[4], MESH[5]).

Authors	Main goal	Type of learned relations	Sources used for learning	Language of patterns	Tool support	Evaluation
Aussenac-Gilles and Jacques, 2006	Learning concepts and relations for terminology and ontology building	hyperonymy-hyponymy, meronymy, synonymy	Domain texts	French	CAMÉLÉON (Séguéla, 2001)	Expert
Marshman et al,. 2002	Learning terms and relations for terminology building	hyperonymy - hyponymy, meronymy, function	Domain texts	French	Not mentioned	Expert
Marshman, 2007	Learning terms and relations for terminology building	association, cause-effect	Domain texts (medical domain)	French and English	TermoStat, Term Extractor (Drouin, 2003)	Expert
Feliu and Cabré, 2002 (Feliu, 2004)	Learning terms and relations for terminology and ontology building	similarity, hyperonymy-hyponymy, sequenciality, causality, meronymy, association	Domain texts (IULA's Technical Corpus)	Catalan	BwanaNet corpus search engine, and Mercedes term identifier (Vivaldi, 2003)	Expert
Sierra et al., 2008	Learning terms and relations for terminology building	definitional patterns (analytical, functional, meronymy , synonymy)	Dictionaries and domain texts (IULA's Technical Corpus)	Spanish	ECODE (Alarcón Martínez et al., 2008) and BwanaNet corpus search engine	Expert
Aguado de Cea and Álvarez de Mon, 2006	Learning concepts and relations for ontology building	classification	Textbook documents (histology, biology, zoology)	Spanish	CREA corpus concordancer	Expert
Soler and Alcina, 2008	Learning terms and relations for terminology building	meronmymy	Domain texts (ceramics)	Spanish	Concord – WordSmith (Scott, 1999)	Expert
Cimiano and Wenderoth, 2005, 2007	Learning concepts and relations for ontology building	hypernym-hyponym, meronymy, function, origin	The Web	English	Commercial Web search engines	User

Figure 3.9: Summary of knowledge acquisition approaches

To finish, in table 3.9 we include a summary of the approaches analyzed for learning concepts or terms and relations with the objective of building terminologies and ontologies. We only include those approaches that focus on verb-centered

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

[5]http://www.ncbi.nlm.nih.gov/mesh

patterns and perform a manual identification of linguistic patterns previous to the learning process.

3.1.2 Main Limitations of Pattern Approaches for Knowledge Acquisition from Text

Despite the quantity and quality of patterns identified in the research works introduced above, most of the authors are in favor of a supervised approach in which the user (terminologist or ontology engineer) validates the knowledge-rich contexts identified by the patterns before including terms and relations in the final resource (terminology or ontology). One of the principal reasons for this is the *noise* or *invalid knowledge-rich contexts* that some of these patterns generate, which sometimes exceed the number of good matches, a disadvantage that we try to avoid in the approach proposed in this work (see chapter 6).

In the following, we detail some of the features of candidate knowledge-rich contexts which invalidate them for their direct reuse in the final resource. In Marshman (2008), the author offers a comprehensive analysis of those features of candidate knowledge-rich contexts that make them useless for a semi-automatic approach to knowledge acquisition from text. As claimed by this author, the **reliability or certainty** of the information needs to be previously assessed. A candidate knowledge-rich context would be considered *unreliable* or *uncertain* if certain lexical elements are present in the context. These lexical elements are:

1. indicators of quantification (E.g., *some* X are classified into...)
2. hedging (E.g., X is *basically* classified into...)
3. use of modal verbs (E.g., X groups of Y *may* be distinguished...)
4. use of negation (E.g., one *cannot* distinguish X from Y...)

This author also argues that the simple pattern structure consisting of two items appearing on either side of the verb or marker is in fact rarer than one might wish (Marshman, 2007). Usually, a lot of words and modifiers appear in between, which make the identification of terms and relations even harder for an automatic system.

Soler and Alcina (2008) also refer to the problems of **polysemy**, **anaphora**, **morpho-syntactic variety**, and domain **dependence**. **Polysemy** has to do with the fact that one pattern can be indicative of more than one relation within the same corpus. **Anaphora** refers to the use of a grammatical substitute (pronoun or pro-verb) in the patterns context instead of its antecedent. The **morpho-syntactic variety** in some languages poses some obstacles in the identification of patterns because of the different forms that a pattern can take. Finally, a further problem with patterns is that they may need to be adapted or tuned for each new **domain**. So, unless we rely on a tool that supports an automatic identification of patterns in every new corpus of texts, the user is expected to define a specific set of domain

relations with valid patterns for each new domain, as also stated in (Aussenac-Gilles and Jacques, 2006).

It should also be highlighted that most of the authors that have dealt with building ontologies or terminologies from text emphasize the importance of making a careful and wise choice of the corpus from which terms and relations are going to be extracted (Condamines and Rebeyrolle, 2000). The importance of the **corpus selection** has to do with the fact that texts can be of very different nature, content and genre (Condamines, 2002). For example, it is well known that we will most probably find definitions of terms in pedagogical handbooks or "documents that popularize technical (and scientific) information" (Aussenac-Gilles and Jacques, 2008). This poses a further obstacle in the activity of ontology learning, which may also demand the intervention of domain experts.

Apart from the disadvantages mentioned, in the case of approaches for building ontologies, domain experts are also left with the task of modeling the acquired concepts and relations in the ontology, or at least supervising it. From all the approaches reviewed, only Sánchez Ruenes (2007) proposes an automatic evaluation of the relations learned against available lexicons and ontologies in the domain. However, for some domains in which comparable resources are scarce, the author also suggests expert's intervention (Sánchez Ruenes, 2007: 108).

Taking into account all the limitations identified so far, we summarize those aspects of the knowledge acquisition approaches for which human intervention or supervision is assumed

1. selection of the corpus from which knowledge is to be acquired
2. validation of obtained knowledge-rich contexts (because of uncertainty aspects)
3. occurrence of polysemy, anaphora, and morphosyntactic variety
4. tuning of patterns for specific domains
5. formalization or modeling of concepts and relations in the ontology

Having analyzed the state of the art in knowledge acquisition and ontology learning approaches based on verbal patterns, as well as the main limitations they expose, in the next section we propose several assumptions that will be taken into account in our specific contribution to this topic.

3.1.3 Open Research Problems and Work Assumptions

As can be seen from the previous research on knowledge acquisition and ontology learning based on linguistic patterns, the process is far from being automatic, and human intervention is still required at different stages. The stage that has received less attention is the fact that domain experts are the ones who have to eventually decide how to formalize or model the learned knowledge. From our viewpoint,

this is still an open research problem that needs to be tackled. In the following, we point out the main differences between our proposal and the state of the art presented, and detail the assumptions made for our work.

The approach we define in this PhD work is very close to the described studies in what regards the identification of patterns. We do also believe that *verb-centered patterns have the advantage of reliably conveying a semantic relation between concepts*. However, there are two central aspects that differentiate our proposal from the ones analyzed above:

1. We **do not pursue the automatic extraction of concepts and relations directly from domain texts or corpora**, but from utterances made by domain experts working in the development of an ontology.

2. We lay particular **emphasis on modeling** or, what is the same, on the formal representation of the obtained knowledge in the ontology, **focusing on newcomers to ontology engineering** as our target users.

Additionally, it should also be mentioned that we deal with some other conceptual relations that have not been considered in previous approaches, and which are motivated by the starting point of our research: a subset of ODPs, as will be further explained in chapter 4. We also believe that it is necessary to **rely on linguistic models to analyze the deep semantics** of the identified linguistic constructs so that definitive statements can be made about their semantic values. For this objective, we draw on the **lexical template** we have proposed, which in its turn is based on the one provided by the LCM. For more details on this see chapter 5).

We expect that this new perspective in the use of linguistic patterns will avoid some of the problems mentioned above, and thus, we formulate the following assumptions:

- We assume that the problem of making a **the right choice of a corpus** would be solved if domain experts previously agree on the extent of the ontology and the information they want to obtain when using the ontology for a certain application.

- If domain experts are involved in the development of an ontology, they may be sure about the information they want to represent in the ontology, which would **avoid the use of uncertainty** aspects.

- Regarding **polysemy**, we think that it is important to identify which patterns are ambiguous and enable various modeling possibilities in the ontology, and make users aware of that.

- The use of **anaphora** could be avoided by giving domain experts some indications on the kind of utterances expected from their side.

- The problem of the **morpho-syntactic variety** of patterns could be approached by relying on sound NLP tools.

- The issue of **domain dependence** could be solved if a distinction is made between patterns that can be considered general and that may appear across several domains of knowledge, and patterns that are more recurrent in certain domains of knowledge.

Next, we will review the state of the art in knowledge acquisition approaches that count on domain experts and support them in the activity of ontology modeling by means of controlled languages.

3.2 Knowledge Acquisition from Experts

The idea of involving domain experts in the development of ontologies is thought to overcome some of the obstacles posed by traditional knowledge acquisition approaches from text. As reported in section 3.1, learning concepts and relations from texts is not a trivial task, which most of the time still requires supervision from domain experts to discard noisy contexts. If domain experts are to build their own ontologies, the complex task of knowledge acquisition would be avoided, on the one hand, and, on the other, more and more organizations would be encouraged to introduce Semantic Web technologies in their information systems. However, **the cornerstone of these approaches is to allow domain experts to input the necessary knowledge without understanding the formal or computational properties of the underlying knowledge representation language** (Pulman, 1996).

In this scenario, some researchers started to look at formulae to bring ontology languages closer to the average user. Some practical experiences in teaching ontology representation paradigms to novice users revealed that they had difficulties in understanding the logical formalisms used to encode ontology models (Rector et al., 2004). As introduced in chapter 1, DL (Baader and Nutt, 2002) is one of the most followed paradigms for the creation of ontologies, on which OWL relies. This representation language is based on first-order predicate logics and demands good background in Logics. Since most ontology editors support DL, they are considered quite inaccessible to all but ontology modeling experts (Cregan et al., 2007; Dolbear et al., 2007; Horridge et al., 2006; Kaljurand and Fuchs, 2007).

Thus, one of the main difficulties to face when using DL lies in making facts explicit that are otherwise implicit in NL expressions. For example, in case we want to state that herbivores are animals that eat plants, it has to be made explicit that it is only plants that herbivores eat, i.e., that the relation *eat* in regard to herbivores can only be established to *plants*. This would be formulated in DL in a compacted way by means of a mathematical symbol as in

$$Herbivore \forall eatPlant$$

This symbol is translated as *allValuesFrom*, which means that the object of the predicate *eat* in this specific relation can **only** be *plant*.

A further example described in Rector et al. (2004) refers to the property of disjointness among classes that do not share instances or individuals. Let us imagine

that in an ontology about pizzas we want to define `Meat`, `Fish` and `Vegetables` as different types or classes of pizza toppings. Unless we define a relation of disjointness among the three classes, any instance of one of the classes could be considered an instance of the other classes. This means that in DL classes are overlapping until disjointness is entered. Again, statements that are usually implicit in NL have to be made explicit when modelling ontologies in DL. Otherwise, they could lead to inconsistencies in reasoning.

These examples illustrate two of the most common problems faced by newcomers to DL. With the aim of overcoming such difficulties, efforts were directed to the creation of simplified syntaxes including elements of NLs that tried to disguise Logics. In this sense, research was devoted to the creation of Controlled Languages (from now on CLs) to make ontology languages more readable and understandable to non ontology experts.

In the next section, we restrict the state of the art on CLs to those that have been designed to facilitate the development of ontologies in the OWL-DL syntax to non ontology experts. In this regard, we will consider the Manchester OWL Syntax (Horridge et al., 2006), Attempto Controlled English (ACE) (Kaljurand and Fuchs, 2006), (Fuchs et al., 2006), (Kaljurand and Fuchs, 2007), the Rabbit syntax (Dolbear et al., 2007), the Sydney OWL Syntax (Cregan et al., 2007), and CLOnE (Controlled Language for Ontology Editing) (Funk, Davis, et al., 2007; Funk, Tablan, et al., 2007).

3.2.1 Controlled Languages in Ontology Engineering

As in other domains in which CLs have been widely applied (machine translation, generation of technical documents, etc.), these are understood as "subsets of natural languages whose grammars and dictionaries have been restricted in order to reduce or eliminate both ambiguity and complexity" (Schwitter, 2007).

The concept of controlled language was proposed in the 1930s by linguists such as Charles Kay Ogden[6], who sought to establish a "minimal" variety of English, in order to make it accessible and usable internationally for as many people as possible (especially non-native speakers) (Schwitter, 2007). As stated in Schwitter (2004), grammatical restrictions result in less complex and less ambiguous sentences, and lexical restrictions reduce the size of the vocabulary and the meaning of the lexical entries for a particular application domain. In general, three types of CLs can be distinguished that fulfill different purposes:

1. CLs designed to help in the writing of technical manuals so that they are clear and unambiguous for readers

2. CLs designed to be used in the translation of documents

[6]In 1930, Charles Kay Ogden publishes the book *Basic English: A General Introduction with Rules and Grammar*, London: Treber, which can be fully accessed at http://ogden.basic-english.org/booksum1.html

3. CLs developed for making it easier to authors to acquire knowledge and build computable models

Regarding the first two groups of CLs, some examples of early CLs include Caterpillar Fundamental English[7] or Simplified English[8]. International companies such as IBM, Ericsson or Boeing saw great benefits in the employment of such CLs for the production of user documentation in several NLs to aid the translation process, whether performed by humans or by machine translation systems (Adriaens and Schreurs, 1992).

Here we are especially interested in the third group of CLs developed for being easily processed by computers. In this context, examples of early CLs are Atemptto Controlled English (Fuchs et al., 1998), or PENG Processable English (Schwitter and Ljungberg, 2002), both examples of syntaxes translatable to first-order predicate logic developed with the objective of stating the properties or constraints of software systems.

These initial approaches showed the main benefits of CLs in computation, which can be summarized as follows:

- CLs can be accurately and efficiently processed by a computer.
- CLs are close enough to natural language, so that users can easily understand and use them.
- CLs avoid ambiguity because the same construct always produces the same result.

Turning now to those CLs that came into existence to make the OWL-DL syntax more readable to non-logicians, one of the first approaches was the one introduced by the Manchester Syntax (Horridge et al., 2006). In this syntax, logical expressions in DL are substituted by NL keywords in English as can be seen in table 3.1.

Logical expression	Meaning	Equivalent English keyword
$\cap$	intersectionOf	and
$\cup$	unionOf	or
$\forall$	allValuesFrom	only
$\exists$	someValuesFrom	some

Table 3.1: Keywords for symbols in Manchester Syntax

In this way, the sentence introduced in the previous section about herbivores would become *Herbivore eat **only** Plant*. The main drawback of such syntax is the

[7]Caterpillar Corporation: Dictionary for Caterpillar Fundamental English. Caterpillar Corporation (1974).

[8]http://www.simplifiedenglish-aecma.org/Simplified%20English.htm

artificiality of the formulations that just manage to somehow disguise the underlying DL syntax. In addition to that, users have to be conscious of the importance of explicitly declaring that "it is only plants that herbivores eat, and nothing else", i.e., users have to understand one of the basic principles of DLs and apply it by means of the appropriate formulation and use of the keyword.

Shortly after the appearance of the Manchester Syntax, other CLs were created adopting the philosophy behind the Manchester Syntax of making the OWL syntax accessible to the average user. These CLs were OWL ACE (or a subset of ACE for OWL (Kaljurand and Fuchs, 2006)), Rabbit (Dolbear et al., 2007), and the Sydney OWL Syntax (Cregan et al., 2007). The motivation behind their creation was the unnaturalness still present in the Manchester Syntax caused, amongst others, by the lack of determiners, the use of singular forms to refer to classes, or the heavy use of parentheses (Kaljurand and Fuchs, 2007).

OWL ACE, Rabbit and the Sydney OWL Syntax are based on well-defined subsets of the English language that translate directly into OWL. OWL ACE and the Sydney OWL Syntax make use of an intermediate syntax between the controlled language and OWL (Discourse Representation Structure in the case of OWL ACE, and OWL Functional-Style Syntax for the Sydney Syntax) (Schwitter et al., 2008). Rabbit, however, utilizes the GATE[9] NLP architecture to convert the controlled language into OWL.

Some examples of sentences produced by the use of these CLs can be seen in table 3.2. Two examples of renderings of OWL axioms into OWL ACE, Rabbit and Sydney OWL Syntax have been included. The first axiom expresses the relation of *subclass of* between two ontology classes (bournes and streams). The second example expresses the relation between a class (river stretch) and its parts (confluences), and additionally states that those parts can be two at the most.

In any case, users are required to become familiar with the languages before editing ontologies. Whereas ACE and the Sydney Syntax are intended for people with no training in formal logics, Rabbit identifies as end users domain experts aided by knowledge engineers. The fact that Rabbit's creators consider that domain users have to be helped by ontology engineers when using the CL somehow hints at the difficulties that CLs may still impose to end users. In fact, sentences resulting from the use of the three CLs sound unnatural. Examples of sentences or even tool support are foreseen to help users familiarize with the languages. Regarding these three initiatives, a task force was formed in 2007 to work towards a common Controlled Natural Language Syntax for OWL 1.1 (Schwitter et al., 2008), because approaches were found to be similar in form and purpose.

Finally, we will refer to the CLOnE approach and its software implementation CLIE. CLOnE is also a CL, based on the English grammar, that relies on the GATE architecture for matching the sentence in controlled language to a syntactic

[9]GATE stands for General Architecture for Text Engineering, and refers to an open source tool for the development of NLP applications. This tool will be explained in more detail in chapter 7, since it has been employed in the development of the application we propose in this work for reusing ODPs in ontology modeling.

Syntax	Renderings
OWL	SubClassOf (OWLClass(bourne), OWLClass(stream)) SubClassOf (OWLClass(river-strech), ObjectMax-Cardinality(2), ObjectProperty(has-part), OWL-Class(confluence))
OWL ACE	Every bourne is a stream. Every river-stretch has-part at most 2 confluences.
Rabbit	Every Bourne is a kind of Stream. Every River Stretch has part at most two confluences.
Sydney OWL Syntax	Every bourne is a stream. Every river stretch has at most 2 confluences as a part.

Table 3.2: Examples of OWL ACE, Rabbit and Sydney OWL Syntax

rule that determines the nature of the ontology element to be modeled. Users are supposed to easily learn the language by following examples and guiding rules. The language and the editor are intended for users without expertise in ontology modeling, although resulting sentences may also remind of the syntax underlying the ontology, as in the previous examples (see table 3.3).

Syntax	Rendering
OWL	SubClassOf (OWLClass(bourne), OWLClass(stream)) SubClassOf (OWLClass(river), DataTypeProperty(has), OWLClass(name))
CLOnE	Bournes are types of streams. Rivers have string names.

Table 3.3: Examples of CLOnE

Some of these languages have been implemented in well-known ontology editors such as Protégé, or in other recently created editors that aim at including more NL components to make them user-friendly to non-experts as GINO (Bernstein and Kaufmann, 2006). GINO is a controlled language ontology editor that uses a NL interface for guiding the user during the edition and querying of the ontology. GINO incrementally parses the input and makes suggestions to the user considering the structure and vocabulary of the ontology being developed or already available.

Similarly to GINO, other ontology question-answering systems such as PowerAqua (López et al., 2006) or Querix (Kaufmann et al., 2006), also make use of powerful NL interfaces and NLP tools for bridging the gap between formal logics and the general public to make users feel more comfortable when communicating with machines in NL. A description of these tools is, however, out of the scope of this research work. Nevertheless, no matter if we are dealing with ontology edit-

ing or ontology querying, the major concern is related to the barrier between the ontology architecture and the average user eager to benefit from the Semantic Web.

To sum up, in figure 3.10 we compare the analyzed CLs to edit ontologies taking into account the following criteria: tools in which they have been implemented, language on which they are based, type of end users considered in each approach, and requirements imposed to users.

	Manchester Syntax	**OWL ACE**	**Rabbit**	**Sydney OWL Syntax**	**CLOnE**
Supporting Tools	Protégé 4.0 plug-in	Protégé 4.0 plug-in GINO	Protégé 4.0 plug-in GATE	Not specified	CLIE GATE
Subset of NL	English	English	English	English	English
End users	Domain experts	Domain experts	Domain experts together with knowledge engineers	Domain experts	Domain experts
User requirements	Not specified	Training and practice	Not specified	Not specified	Training by following examples and guiding rules

Figure 3.10: Summary of CLs for building ontologies

3.2.2 Main limitations of CLs in Knowledge Acquisition for Ontology Modeling

Up to now, the reviewed approaches have as starting point the OWL DL syntax and they create a layer above it, supposedly closer to the syntax of a NL than to formal logics. However, CLs are still quite accurate reflections of the underlying ontological structure.

Apart from the unnaturalness of sentences, we see some drawbacks in the analyzed CLs that should be overcome when aiming at making OWL ontologies accessible to people with no training in formal logics. These problems can be summarized as follows:

1. CLs do not provide users help in solving modeling difficulties
2. CLs require some efforts on the side of the user to learn, read and write statements
3. CLs have been developed as subsets of the English language

Regarding **problem 1.**, we claim that users may have more difficulties in finding out which ontology structure or element allows them to represent certain content in the ontology, than in selecting the rule that the CL offers to model that ontology structure. In order to understand this position, let us consider the following example. Imagine the user wants to model the relation between a "river stretch" and its "confluences", as in some of the examples in table 3.2. The user will have to be previously aware of the fact that a part-whole relation (part-of) is holding between the two concepts. Then, as a second step, (s)he will search for the corresponding CL formulation (e.g., "has-part" in OWL ACE, or "has...as part" in the Sydney Syntax) to model that relation in the ontology.

Selecting the ontological relation that the untrained user needs is not a trivial task, as some experiments have revealed (Aguado de Cea et al., 2008). In the mentioned experiments, Computer Science students with some background in modeling had to identify the most appropriate modeling solution for modeling a problem expressed in NL such as: *A research plan is composed by a theoretical plan and an experimental plan*. Results showed that nearly half of the solutions (41%) were erroneous according to the golden standard. It is worth mentioning that the part-whole relation (part-of) was mainly confused with the subclass-of relation, among other erroneous solutions. For more details about these experiments we refer the interested reader to Aguado de Cea et al. (2008).

This gives just some hints of the difficulties untrained users face when having to choose the most appropriate modeling solution. One could argue that this is to a lesser extent related to the essence of CLs themselves. However, we consider that most of the problems domain experts have when developing ontologies are rather related to modeling decisions, than to choosing the CL syntax to express them. We believe that this is a more demanding and complex issue not really considered by the approaches to CLs analyzed in section 3.2.1, and which should be handled together. In fact, the analyzed approaches on CLs do not provide the user with any guidelines for making that kind of modeling decisions.

Regarding **problem 2.**, it must also be noted that learning to use a CL is by no means trivial, let alone if it is fairly close to logics. The implications are not only limited to learning some new grammar structures or rules, but to understanding what they represent and imply when modeling. And this brings us to the previous point (**problem 1.**), since the difficulties in learning new rules is tightly connected with the content they allow to model in the ontology.

Additionally, some experiments have revealed that users prefer the use of full NL when interacting with machines because "they can communicate their information need in a familiar and natural way without having to think of appropriate keywords in order to find what they are looking for" (Kaufmann and Berstein, 2007). This result has been obtained in recent usability studies conducted to investigate how useful NL Interfaces are to find data in the Semantic Web. From the four interfaces tested by the 48 users involved in the experiment, the one that required full English questions was judged to be the most useful and "best-liked query interface".

As far as **problem 3.** is concerned, to the best of our knowledge, CLs aimed at helping users to semantically represent domain content in OWL are only available for the English language. From our point of view, this represents an obstacle to the development of ontologies for applications that need to interact with languages different from English.

3.2.3 Open Research Problems and Work Assumptions

A key debate that takes place once and again is the dichotomy between *naturalists* vs. *formalist* approaches to CLs (Clark et al., 2009). The set of approaches presented in section 3.2.1 can be said to follow a *formalist* paradigm, since they comply with the conditions of being "well-defined, predictable, and deterministically translatable into a formal representation". On the other hand, *naturalist* CLs are closer to the user, but suffer the inconvenience of having to deal with language ambiguities.

It is undeniable that language ambiguities demand sound NLP tools to discern the correct interpretation of a sentence in a certain context. However, it is unquestionable as well that formalist approaches require a great effort on the side of the user in two aspects:

- the time and effort the user has to put in learning the language
- the idea that the more "controlled" the language is, the more the user needs to understand the underlying representation language, or in our case, the logic formalisms underlying ontology modeling

We consider these two aspects as being open research problems in the CLs research, mainly if such approaches are intended for untrained users in ontology modeling. For these reasons, we opt for a ***naturalist approach*** that has the following advantages:

1. Domain experts are allowed to express ontology specifications in full NL and do not need to learn a CL.
2. By expressing what is to be modeled in the ontology in NL, **domain experts move away from ontology modeling paradigms and underlying representation languages, and concentrate on their modeling needs**.

In this context, we formulate the following assumptions:

- We believe that if users are allowed to express ontology specifications in full NL, the ontology modeling task will be perceived as a simple task.
- Additionally, if users can express their modeling needs in their own language, more and more users will adopt ontologies for their applications.

- By providing some guidelines or recommendations to users on the kind of input that is expected from them, we can avoid some of the problems attributed to *naturalist* approaches vs. *formalist* ones, namely, use of uncertainly or anaphora (see also section 3.1.2 on pattern approaches for knowledge acquisition from text).

Apart from relying on a naturalist approach, the most innovative aspect of our proposal is that we establish a correspondence between NL assertions and ontology components considered good practices in Ontological Engineering, and not simply formalizations in an ontology language, as was the case of CLs. The ontology components we are referring to in this work are the so-called Ontology Design Patterns. More details about this type of patterns is given in chapter 4.

3.3 Summary

In this chapter we provide a description of the state of the art in knowledge acquisition approaches both from text and from experts. Each type of knowledge acquisition process is devoted a separate section. The last part of each section is dedicated to an analysis of the main limitations of both approaches, open research problems, and the assumptions that we take as starting point in our work.

From the different approaches on knowledge acquisition from text, we have focused on those that exploit the idea of applying linguistic patterns to the discovery of semantic or conceptual relations. Even when these linguistic patterns have showed to appear rarely in texts, they have proven to reliably convey a relation of interest. We present a total of ten research works on the identification of verbal patterns for several NLs. Most of them rely on a manual analysis of textual resources for the identification of verbal patterns that will be subsequently used for the semi-automatic identification of terms/concepts and the relations that hold between them. However, none of these approaches provides guidelines for the modeling task.

Regarding those approaches on knowledge acquisition from experts, we were interested in those that rely on CLs to help novice users in the ontology modeling task. We have described five approaches on CLs that require from users to learn and write statements following a certain syntax, and transform their statements into formal representations. Users need to be aware of the modeling possibilities offered by the CL and also of their modeling needs to write the appropriate statements. All CLs surveyed in this section are a subset of the English language.

Chapter 4

Ontology Design Patterns

The previous chapter presented a number of approaches to acquire knowledge with the aim of speeding up the construction of terminologies and/or ontologies. As has been reported, the investigation on knowledge acquisition was traditionally centered on textual resources that ontology engineers accessed to create ontologies. The main obstacles in this sense were represented by the complexities in the direct processing of NL. Researchers started then to look at domain experts and get them involved in the development of ontologies, but this involvement also showed severe limitations due to domain experts having to understand the representation paradigms underlying the encoding of ontology models. In order to address this bottleneck, current trends suggest the emergence of hybrid approaches that try to combine strategies from both approximations. These approaches can be divided into those that try to combine the automatic extraction of information from text with the participation of domain experts, and those that propose the **assistance of ontology engineers and domain experts with semi-automatic means**.

It is in this latter framework in which our approach for knowledge acquisition and ontology modeling is in line with. Basically, **we achieve the acquisition of domain knowledge by processing linguistic structures formulated by users that describe the knowledge they want to represent in the ontology**. After that, we offer users the most appropriate ontological structure to model the knowledge expressed in the linguistic construct. These ontological structures or ontology modeling components correspond to ODPs in this research work. As we will explain in the following sections, the aim on focusing on ODPs as starting point in our research is motivated by the fact that ODPs (a) follow well recognized principles in Ontological Engineering, (b) represent modeling solutions agreed by ontology engineers, and (c) guarantee the design adequacy of the final ontology.

After selecting the subset of ODPs we want to focus on in our work, we identify the linguistic structures that are recursively used to express the knowledge captured in those ODPs. These linguistic structures have been called **Lexico-Syntactic Patterns** or **LSPs** (inspired by Hearst (1992)), and have been defined as

> *(...) formalized linguistic schemes or constructions derived from regular expressions in NL that consist of certain linguistic and paralinguistic elements, following a specific syntactic order, and that permit to extract some conclusions about the meaning they express* (Aguado de Cea et al., 2008).

A correspondence between LSPs and ODPs is established after a manual analysis of the semantics captured in the linguistic expressions (see chapter 5). The set of correspondences between linguistic expressions and ontological representations is stored in a repository that will be the core of the method for knowledge acquisition and ontology modeling that we suggest in this work.

This modeling method based on patterns is to be understood within the wider framework of a new ontology modeling paradigm that emphasizes the reuse of available knowledge resources, from which ODPs are a principal exponent. The new paradigm we are referring to here is the NeOn Methodology (M. C. Suárez-Figueroa, 2010). Additionally, users are to be assisted during the whole process by means of a supporting tool, whose development has also been initiated in the research conducted in this PhD work.

Thus, Chapter 4 of this dissertation is devoted to the definition of Design Patterns in general, and its application in Ontological Engineering. Then, we present the NeOn Methodology, in which the method we propose is to be understood. Finally, we discuss some open issues and formulate our work assumptions.

4.1 Design Patterns

The term design pattern was introduced in the seventies by Christopher Alexander in the Architecture domain for designating those modeling solutions that after being recurrently used for solving similar design problems, could be identified as generalized design solutions to be applied whenever a similar problem appeared (Buschmann et al., 1996). In Alexander's own words, patterns described solutions "in such a way that you can use this solution a million times over, without ever doing it the same way twice".

In the mid 1980s, W. Cunningham and Beck (1987) adapted Alexander's ideas to software development, but it was not until the publication of the book *Design Patterns - Elements of Reusable Object-Oriented Software* (Gamma et al., 1995) that design patterns became broadly used in object-oriented software design. Since then, design patterns have been applied in a great variety of areas within Computer Science.

The benefits of design patterns in Software Engineering are well known, and can be summarized in three points, as in Prechelt (1997):

- design patterns allow less experienced users to produce a better design
- design patterns encourage recording and reusing best practices even for experienced designers

- design patterns can improve communication by defining a common design terminology

Nowadays, design pattern reuse in object-oriented software design is an extended practice, supported by design pattern repositories and manuals as the one by Gamma et al. (1995), or the one by Buschmann et al. (1996). Templates describing design patterns typically contain the following information (Svátek, 2004):

- name of the pattern
- problem description
- suggested solution
- implementation guidelines
- discussion on consequences of using the pattern

The *name* is considered the identifier for the pattern. The *problem description* and *discussion on consequences* sections are expressed in natural language, whereas the *suggested solution* often has the form of a UML[1] diagram with abstract classes to be filled in with specific concepts. And similarly, the *implementation guidelines*usually contain source code with abstract roles that are to be replaced with specific content.

Pattern repositories are usually integrated in software tools in order to allow a quicker access and integration of the patterns. Nevertheless, most of the existent manuals or repositories presuppose prior design knowledge and expertise, and do not provide any guidance to users. It is assumed that expert users select the most adequate design pattern for their modeling needs relying on prior experiences and the descriptions included in the templates.

The assumption of users prior knowledge and other limitations of the reuse of patterns are being recently discussed in public forums by experts in the Software Engineering domain (Fayad and Srikanth, 2007). The main limitations are related to the lack of general methodologies or standards for the reuse of patterns, since some efforts in that sense are limited to steps or recommendations for local use developed by the authors of the manuals themselves. Likewise, templates follow different styles depending on the manual, so that some of the steps or approaches given by certain authors cannot be extrapolated or reused in searching other design pattern repositories. Finally, an additional limitation reported by practitioners is related to the efforts that the search activity requires, which, apart from being time consuming, demands a careful analysis of the templates on the part of the user.

In the next section, our objective is to give a similar overview on design patterns but now in the Ontology Engineering domain.

[1] Unified Modeling Language, see http://www.omg.org/uml

4.2 Design Patterns in Ontology Engineering

It is not until the beginning of the 21st century that design patterns are fully introduced in this domain by ontology researchers such as Gangemi (2005), Rector and Rogers (2004), Svátek (2004) or the W3C Consortium[2]. In this scenario, ODPs have been defined as "archetypal solutions to design problems" (Gangemi, 2005), and are assumed to produce the same benefits in the modeling of ontologies as in object-oriented software design, namely: faster and better design, reuse of best practices, and fluent communication among designers.

One of the first approaches addressing the reuse of knowledge components by means of patterns had been that of (Clark and Porter, 1997; Clark et al., 2000). Clark et al. (2000) defined *knowledge patterns* as "general templates denoting recurring schemata, and their transformation (through symbol renaming) to create specific theories". Reusability of knowledge patterns or "mini-theories" was seen as a way of improving efficiency in knowledge-based systems. Likewise, the interest in proposing design patterns that could be reused by unexperienced users grew also in the Bioinformatics field (Reich, 2000), where the modeling of biological knowledge posed great challenges to experts in the domain.

Soon afterwards, Svátek (2004) and Gangemi (2005) explicitly referred to **Ontology Design Patterns for the Semantic Web**. Svatek defined them as "building blocks" that could be highly beneficial for ontology developers, explicitly referring to the business domain. Gangemi, after extensive experience in ontology design projects in several domains such as fishery techniques or legal norms, presented patterns for solving *content* design problems in OWL or other logical languages.

By *content* design problems, Gangemi (2005) referred to design problems of classes and properties *specific of certain domains*. For instance, the *participation content design pattern* for modeling the participation of objects in events. Though general enough and reusable in ontologies of several domains, these patterns already contain certain domain information, as opposed to patterns that deal with structural problems. Indeed, these patterns can be considered small ontologies that address specific modeling issues, and that can be directly reused by importing them in the ontology being built (Presutti et al., 2009).

Design patterns were also seen as appropriate solutions to the difficulties that ontology languages imposed to users. In Rector et al. (2004) and Egaña et al. (2008), the authors provide a summary of their experiences in teaching OWL DL and state that "for most people it is very difficult to understand the logical meaning and potential inferred statements of any DL formalism". As a result of that, design patterns emerged as a way for helping ontology practitioners to model OWL ontologies, since they could simply reuse the pattern in their ontologies without having to understand the logic behind. To this end, the W3C Semantic Web Best Practices and Deployment Working Group[3] proposed patterns for solving design

[2]World Wide Web Consortium, see http://www.w3.org

[3]http://www.w3.org/2001/sw/BestPractices/

problems for OWL, independently of a particular conceptualization.

In (Presutti et al., 2008) and (M. C. Suárez-Figueroa et al., 2007) ODPs have been classified into different groups, namely:

- *Logical ODPs*. These patterns are content-independent and typically solve problems of *expressivity* in a certain ontology language. For instance, the need for expressing relations among three or more concepts (also known as *n-ary relations*) in OWL, which can be quite complex for users without a strong background in OWL (Presutti et al., 2009).

- *Content ODPs*. These are patterns that solve design problems of specific domains. For instance the relation between *collections* and the entities that are *members* of that collection.

- *Architectural ODPs*. These patterns describe the overall structure of the ontology (either internal or external) that is convenient with respect to a specific ontology-based application (Presutti et al., 2009).

- *Mapping ODPs*. The purpose of this type of patterns is to define semantic associations between two existing ontologies (Presutti et al., 2008).

- *Reasoning ODPs*. These patterns are applications of Logical ODPs oriented to obtain certain reasoning results. Examples of Reasoning ODPs include: subsumption, inheritance or materialization patterns (Presutti et al., 2008).

- *Presentation ODPs*. These patterns are designed to deal with usability and readability of ontologies. For example, the naming pattern defines naming conventions of ontology concepts and properties.

- *Reengineering ODPs*. These patterns define methods for transforming non-ontological resources to ontological resources (García-Silva et al., 2008). For example, there is a pattern for transforming a classification schema into an ontology.

In the Ontology Engineering domain, researchers have tried to apply lessons learned from the reuse of design patterns in Software Engineering to overcome the limitations reported by practitioners in that domain. Several experiments with design patterns have empirically proven that design patterns can benefit ontology development, as reported in Blomqvist et al. (2009). As a consequence of this, a lot of effort has gone into the following actions:

1. creation of **templates** to systematically describe ODPs
2. creation of on-line **repositories** to enable an easy access and reuse of ODPs
3. creation of guidelines or **methods** for the reuse of ODPs
4. development of **tools** for supporting the reuse of ODPs

In the following, we will devote four sub-sections to each of the actions taken so far in the state of the art to support the reuse of ODPs in the Ontology Engineering Domain.

4.2.1 Templates for ODPs

In M. C. Suárez-Figueroa et al. (2007), a template has been proposed to systematically describe ODPs inspired by the work in Software Engineering. The purpose of this template is to lay the foundations for a standard description of ODPs. The proposed template contains the following information:

- **General Information**, which includes name, and identifier (an acronym that consists of: component type + component + number) and the type of ontology pattern (Logical ODP, Content ODP, etc.)
- **Use Case**, a description in natural language of the problem to be addressed with a real example
- **ODP**, which includes the proposed solution in different formats (UML graphical representation and OWL code), accompanied by a description in natural language
- **Comments**, which refers to remarks for clarifying the use of the pattern, and relations to other patterns

The template describing the Logical Pattern for Modeling Disjoint Classes extracted from (M. C. Suárez-Figueroa et al., 2007) is included below by way of example (see Figure 4.1).

4.2.2 ODPs Repositories

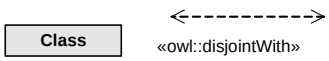
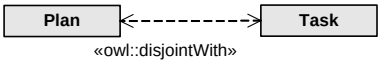
At the present stage of the research on ODPs, we find several repositories that contain design patterns. All of them are public and available on-line, which contributes to the idea of reusability of design solutions.

One of the earliest repositories to appear was the **LoaWiki:CPRepository** maintained by the Laboratory for Applied Ontology of the Italian National Research Council[4]. This repository is devoted to Content ODPs, and does not contain any other type of design patterns.

Pioneer in the biological domain was the **Ontology Design Patterns (ODPS) Public Catalogue**[5] of ODPs for bio-ontologies, created by researchers at the University of Manchester working on the GENE ONTOLOGY project.

Finally, we will refer to the **Ontology Design Patterns Portal**[6], a Semantic Web portal developed by researchers participating in the NeOn project. The main

[4] http://wiki.loa-cnr.it/index.php/LoaWiki:CPRepository
[5] http://www.gong.manchester.ac.uk/odp/html/index.html
[6] www.ontologydesignpatterns.org

Slot	Value
General Information	
Name	Disjoint Classes
Identifier	LP-Di-01
Type of Component	Logical Pattern (LP)
Use Case	
General	Express that an element, belonging to a certain group or set, cannot belong to another group or set. In other words, express that two different sets are disjoint.
Examples	Suppose that someone wants to express that 'plans' are disjoint with 'tasks'.
Ontology Design Pattern	
Informal	
General	Instantiate the class `Class` and the object property `disjointWith`.
Examples	Create the classes 'Plan' and 'Task', and assert that 'Plan' is 'disjointWith' 'Task'.
Graphical	
(UML) Diagram for the General Solution	
(UML) Diagram for Examples	
Formalization	

General	`Class(Class partial OntologyElement)`
	`Class(Property partial OntologyElement)`
	`Class(ObjectProperty partial Property)`
	`ObjectProperty(disjointWith domain(Class) range(Class))`

Examples	`DisjointClasses(Plan Task)`
Relationships	
Relations to other modelling components	Possible use of this LP in the following LPs: LP-NR-01,LP-NR-02 and LP-SV-02. Possible use of this LP in APs and CPs.

Figure 4.1: Template describing the *logical pattern for disjoint classes*

objective of this portal is to promote collaboration among ontology practitioners in sharing, updating and improving patterns created in the most different domains. A capture of the main page of the www.ontologydesignpatterns.org portal is to be seen in figure 4.2.

Figure 4.2: Ontology Design Patterns Portal screenshot

A part from a description of the pattern in NL containing the fields described above in the template, these repositories offer a UML diagram illustrating the pattern in question. See captures of the Object-Role Content ODP from the Ontology Design Patterns Portal in figure 4.3, and the Defined Class Logical ODP from the GENE ONTOLOGY project in figure 4.4.

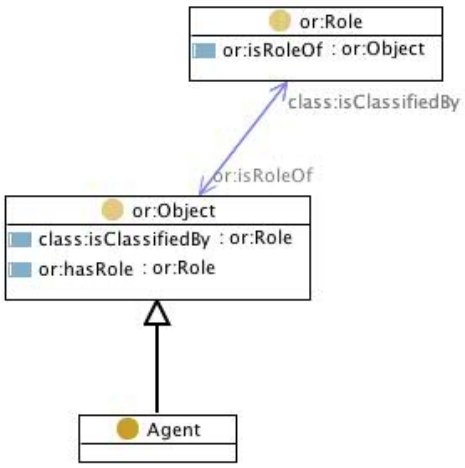


Figure 4.3: *Agent role pattern* from the Ontology Design Patterns Portal

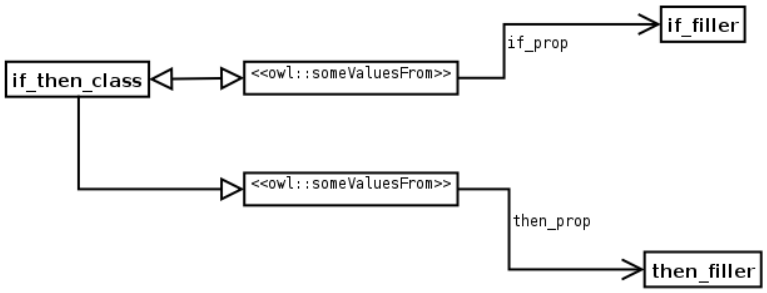


Figure 4.4: *Defined class pattern* from GENE ONTOLOY project

4.2.3 ODPs Reuse Methods

In terms of methods for guiding the selection and reuse of ODPs, research was nearly inexistent at the time of initiating the investigation presented in this work. The creation of methods was deemed crucial for an effective and efficient reuse of ODPs in ontology modeling. Experiences in Software Engineering had already revealed the need for such methods to actually palliate the main limitations of the reuse practice: the time that had to be spent in the analysis of pattern templates, and the difficulties faced by users without modeling experience, as reported in section 4.2.

In Ontology Engineering, some experiments on the reuse of ODPs also confirmed this supposition (see Aguado de Cea et al. (2008) for more details). These experiments involved Computer Science students and Ontology Engineering PhD students, and showed that the selection of the ODP or ODPs that better match a modeling aspect expressed in NL is not a flawless task.

In these experiments, students were provided with several modeling problems in NL and were asked to give a modeling solution. Examples of these sentences

are given below:

1. *Tasks are management tasks, financial tasks, marketing tasks, and control tasks*.

2. *A research plan is composed of a theoretical plan and an experimental plan*.

In a first stage, students did not get any external support, whereas in a second stage, they were provided with a catalogue of ODPs (particularly M. C. Suárez-Figueroa et al. (2007)), containing a selection of Logical ODPs and Content ODPs. Results showed that around 50% of the participants had difficulties in selecting the most appropriate pattern according to the experts. Representative examples of some common mistakes were:

- **subclass-of relation**, mainly mistaken with exhaustive classes or disjoint classes;

- **exhaustive classes**, mainly mistaken with subclass-of relation or disjoint classes; and

- **part-whole relation**, mainly mistaken with subclass-of relation.

While there had been some initiatives for helping users in the process of adapting or implementing ODPs by means of wizards, as the ones provided by the CO-ODE project[7] for the Protégé ontology editor (Egaña Aranguren et al., 2007), the search and selection tasks remained untreated. Users were assumed to access available repositories, carefully analyze the templates, and select the most appropriate pattern for their modeling needs. This process, apart from being highly time-consuming, proved to be by no means trivial.

In this context, M. Suárez-Figueroa et al. (2009) propose a method for developers with prior modeling experience: the *XD method* (*eXtreme Design method*). This method has to be understood within the wider framework of the NeOn Methodology that will be explained in section 4.3. This method focuses on the reuse of *Content ODPs*, and consists of the following tasks:

- Task 1.: Identify the set of requirements to be addressed from the Ontology Requirement Specification Document (ORSD) (obtained from the Ontology Specification Activity proposed in the NeOn Methodology, as will be explained in section 4.3).

- Task 2.: Identify patterns repositories.

- Task 3.: Divide ontology requirements into smaller parts or partial problems.

[7] http://www.co-ode.org/downloads/wizard/

- Task 4.: Match partial problems to identified patterns. This task consists in identifying candidate patterns to solve partial modeling needs. This is considered one of the hardest task of the process, but at the time of writing this document, it has to be carried out manually by the user, although the authors admit that tool support may be needed (M. Suárez-Figueroa et al., 2009).

- Task 5.: Select patterns to be reused. If manual matching was performed this is a decision-making process, where the usefulness of the pattern is weighted against the overhead of reusing it.

- Task 6.: Apply selected patterns and compose them to solve the problem addressed in the initial requirement. This task may involve different actions, such as specialization of Content ODPs or combination of several Logical ODPs.

- Task 7.: Evaluate solutions by querying the ontology.

- Task 8.: Integrate partial solutions in the complete solution.

In Presutti et al. (2009), the authors propose some subtasks for the main Tasks identified in the XD method. We would like to devote some attention to the subtasks proposed for Task 1, because they clearly illustrate the difficulties existing in matching modeling problems expressed in NL to ODPs.

Once ontology designers and domain experts get an idea of each other's tasks in the development process and domain experts are taught about the method and tools to be applied during the project, domain experts are asked to write *requirement stories*. Requirement stories are descriptions of real scenarios that sample the typical facts that should be stored in the ontology. Consider the following example of a requirement story of the domain of *Tourist information about cities*[8].

> Rome is the capital of Italy, it is located in the Lazio region. Rome has two airports. Fiumincino airport is served by Alitalia flights, while Ciampino is served by Ryanair and Wizzair. Rome has several train stations, the main station is Termini located in the center of Rome, but there are also the Trastevere station in the west part of Rome (the Trastevere district), and Tiburtina in the south east.

Then, requirement stories have to be transformed into Competency Questions (CQs). CQs (Grüninger and Fox, 1994) are understood as questions that the ontology is assumed to be able to answer once its development has been completed. This is suggested to be done by the ontology engineers participating in the project, with the help of domain experts. The strategy to follow consists in splitting the story

[8]This story has been obtained form one of the training tutorials about XD held at the K-CAP conference in Redondo Beach California, in September 2009. See http://ontologydesignpatterns.org/wiki/Training:Extreme_Design_%28XD%29:_Pattern-based_Ontology_Design/Hands-on_session_K-CAP_tutorial

into simple shorter sentences, and derive "abstractions" from those sentences, i.e., derive sentences that refer to general classes or facts instead of specific facts. Only afterwards, are CQs formulated.

In the example above, *Rome is the capital of Italy, it is located in the Lazio region*, the sentence would be split into two simpler sentences:

1. *Rome is capital of Italy.*
2. *Italy is located in the Lazio region.*

After that, we would generalize and say that

1. *Some cities are capital cities of countries*, or *Countries have capital cities*
2. *Cities are located in regions*

Finally, we would formulate the following CQs and their corresponding answers:

- CQ_1: Which are the capital cities of (European) countries? Italy is capital city of Rome, Paris is capital city of France...
- CQ_2: In which regions are cities located? Rome is located in the Lazio region; Venice is located in the Veneto region; Florece is located in...

The CQs that result from this process are used to identify candidate Content ODPs. According to the authors, if ontology engineers have a good knowledge of available Content ODPs, this task should not involve further difficulties. Otherwise, they propose to carry out keyword search in pattern repositories.

After the performance of these tasks, ontology engineers would continue with Task 5 to 8, to complete the process. As will be explained in the next section, the authors of the XD method provide tool support only for Tasks 5 and 6.

4.2.4 Tools for supporting ODPs Reuse

Apart from the mentioned wizard created within the CO-ODE project for supporting the implementation of ODPs in ontologies, to the best of our knowledge the only tool support available nowadays is the one developed for the XD method described above.

The XD tool has been designed as a plug-in of the ontology editor NeOn Toolkit[9]. Its current prototype focuses on the reuse of Content ODPs. It mainly supports Task 5 and Task 6 of the method. Task 5 regards the selection of the pattern to be reused from the ones that match the modeling problem. Task 6 deals with the specialization and integration of the pattern in the final ontology. However, the "matching" task (Task 4) is still to be performed manually and some supporting component is expected to be included in the tool in the future.

[9]The plug-in can be downloaded from the following URL http://neon-toolkit.org/wiki/XDTools

4.3 NeOn Methodology as Framework for the Reuse of ODPs

In M. Suárez-Figueroa et al. (2009), it is stated that very few methodologies for the development of ontologies explicitly mention the use of patterns, and if mentioned "they are usually proposed as a kind of *additional support* that may guide developers within any methodology". In the framework of the NeOn Methodology, however, the reuse of ODPs is understood as a an "ontology development method" *per se*.

The NeOn Methodology owes its name to the project in which it has been developed, the NeOn project [10]. This new methodological paradigm is grounded on traditional methodologies for the development of single ontologies such as METHONTOLOGY (Fernández-López et al., 1999), On-To-Knowledge (Staab et al., 2001), or DILIGENT (Pinto et al., 2004). These methodologies identify a set of activities to build ontologies from scratch and provide some guidelines for the execution of those activities. However, they neglect some aspects of the development process that have proven decisive in the current age of the Semantic Web, namely, (a) the reuse of existing ontological and non ontological resources, (b) the dynamic evolution of ontologies, or (c) the fact that some ontology projects may require a network of ontologies[11], as opposed to a single ontology, built by distributed teams that work collaboratively.

With the aim of palliating those shortcomings, the **NeOn Methodology** (M. C. Suárez-Figueroa, 2010) identifies a set of flexible scenarios in the ontology development process that can be combined according to the ontology requirements and the existing resources in the domain. The 9 scenarios identified so far are represented in 4.5 by directed arrows and numbered circles. Each scenario covers a specific process or activity that has to be followed to develop an ontology whenever certain requirements or premises are given. For most of the scenarios prescriptive methodological guidelines are given. These guidelines define precisely the set of activities or tasks that are to be performed in each scenario, and the state inputs and outputs of each task, the actors involved, and the existence of techniques and tools to be used.

Any combination of scenarios should include Scenario 1, because this scenario is made up of the core activities that are to be performed in any ontology development process. In fact, most scenarios will be included in the development process once some of the activities in Scenario 1 have been carried out.

Following Scenario 1 of the NeOn Methodology, any ontology development should start with the knowledge acquisition activity. Simultaneously, ontology developers should perform the Ontology Specification activity, whose objective is to

[10] http://www.neon-project.org

[11] An ontology network is defined as *a collection of ontologies related together via a variety of different meta-relationships such as mapping, modularization, version, and dependency relationships* (Gómez-Pérez and Suárez-Figueroa, 2009).

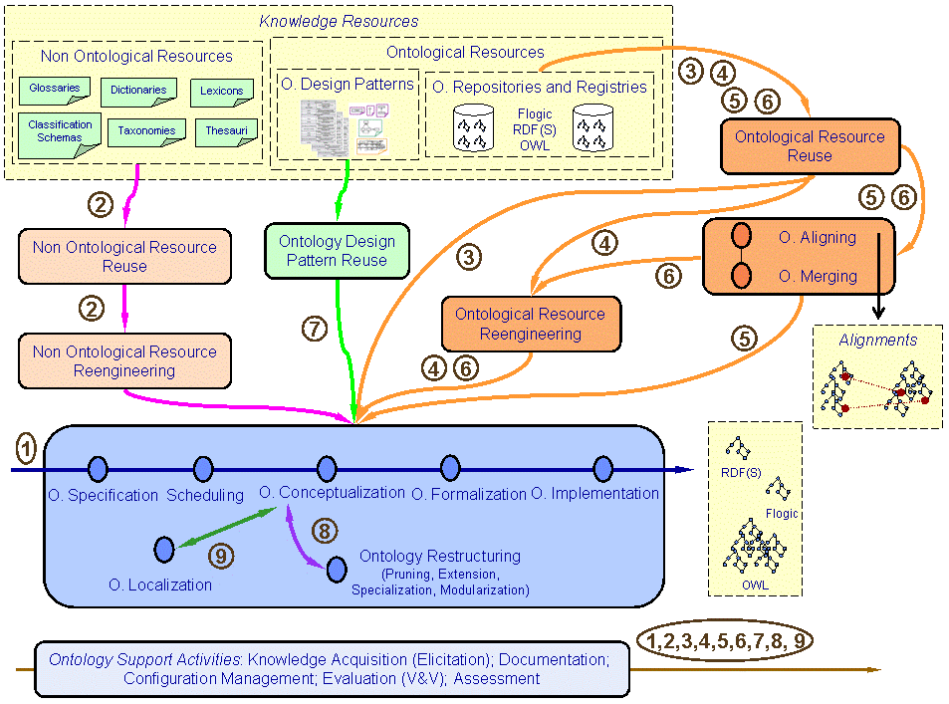


Figure 4.5: NeOn Methodology scenarios for building ontology networks

obtain the so-called *Ontology Requirements Specification Document* (ORSD) as output. In this document, the following aspects of the ontology are specified:

- purpose
- scope
- implementation language
- target group
- intended uses
- requirements

The set of requirements that the ontology network should fulfill is mainly expressed in the form of competency questions (CQs) (Gómez-Pérez and Suárez-Figueroa, 2009). CQs also serve as benchmarks or evaluation frameworks for the ontology, because for the ontology to be complete and correct, it has to represent the knowledge specified in the CQs as well as their solutions.

The formulation of CQs is carried out by the so-called Ontology Development Team, which consists of domain experts and ontology engineers. It can be considered an initial knowledge acquisition strategy that will allow knowledge engineers

to delimit the extent and coverage of the ontology in the first place. It is also a very convenient strategy to obtain the main concepts, relations and instances that are to be represented in the ontology. An example of the set of CQs formulated for the development of ontologies in the e-employment domain within the SEEMP project can be seen in Figure 4.6 (obtained from M. C. Suárez-Figueroa et al. (2009)).

In most of the ontology development projects, the CQs strategy to acquire knowledge is complemented by semi-automatic methods for obtaining additional concepts, relations and instances that are to be included in the ontology. Some of these approaches for the semi-automatic acquisition of knowledge have been introduced in chapter 3. In the approach we envision for the modeling of ontologies intended at novel users, **CQs will be taken as the starting point for the formulation in NL of the domain knowledge to be represented in the ontology**, as was the case of the XD method presented in section 4.2.3. See also section 4.5.

After the Ontology Specification activity, the NeOn Methodology guidelines advise to carry out a quick search for existing knowledge resources using the terms in the ORSD. The main objective of this task is to obtain an overview of candidate resources that could be eventually reused in the ontology development process. Experiences in several projects have shown that making use of existing knowledge resources considerably reduces the time and efforts involved in the ontology development process (Gómez-Pérez and Suárez-Figueroa, 2009). In this sense, the NeOn Methodology considers two types of knowledge resources: ontological resources (ontologies, ontology modules, ontology statements, or ontology design patterns), and non-ontological resources (thesauri, lexicons, classification schemas, and databases).

Next, it is advisory to perform the Scheduling activity, since, by then, users must be in the position of estimating the time that each of the remaining activities will approximately take. Afterwards, ontology developers carry out the rest of the activities (Conceptualization, Formalization, and Implementation) following the guidelines provided in METHONTOLOGY (Fernández-López et al., 1999) or On-To-Knowledge (Staab et al., 2001). According to METHONTOLOGY, the Ontology Conceptualization activity includes tasks such as, (1) identification of the main concepts to be included in the ontology; (2) building of initial concept taxonomies; (3) building of *ah-hoc* binary and n-ary relation diagrams between concepts of the ontology or with concepts of other ontologies; and (4) description of attributes, instances and axioms. Finally, concepts and relations are implemented in a formal language by means of any ontology editor tool.

The scenario that interests us in this research is **Scenario 7: Ontology Design Pattern Reuse**. This scenario would normally come into scene if the terms collected in the ORSD, and more specifically the set of CQs, bring ontology developers to conclude that some ODPs could be selected and reused to model the knowledge expressed in the CQs. In this context, the reuse of ODPs is considered a strategy for the development of ontologies *per se*, which can still be combined with other scenarios or can be employed on its own.

In Scenario 7, ODPs are the basis for ontology design, taking as input the on-

	Ontology Requirements
	b. Functional Requirements: Groups of Competency Questions
	CQG1. Job Seeker (14 CQ)
	CQ1. What is the Job Seeker's name? Lewis Hamilton
	CQ2. What is the Job Seeker's nationality? British; Spanish; Italian; French;
	CQ3. What is the Job Seeker's birth date? '13/09/1984; 30/03/1970; 15/04/1978
	CQ4. What is the Job Seeker's contact information? Tel: 34600654231. Email: jsanz@fi2.upm2.es
	CQ5. What is the Job Seeker's current job? Programmer; Computer Engineer; Computer Assistant
	CQ6. What is the Job Seeker's desired job? Radio Engineer; Hardware designer; Software Engineer
	CQ7. What are the Job Seeker's desired working conditions? Autonomous; Seasonal Job; Traineeship; Consultant
	CQ8. What kind of contract does the Job Seeker want? Full time; Partial time; Autonomous; Seasonal Job
	CQ9. How much salary does the Job Seeker want to earn? 3000 Euros per month, 40000 Euros per year
	CQ10. What is the Job Seeker's education level? Basic education; Higher education/University
	CQ11. What is the Job Seeker's work experience? 6 months, 1 year, 2 years
	CQ12. What is the Job Seeker's knowledge? Java Programming; C Programming, Database Administration
	CQ13. What is the Job Seeker's expertise? Software Engineering
	CQ14. What are the Job Seeker' skills? SQL programming, network administration
	CQG2. Job Offer (11 CQ)
	CQ15. What is the employer's information? CEFRIEL Research Company, Milano, Italy; ATOS, Madrid, Spain
	CQ16. What kind of job does the employer's offer? Java Programmer; C Programmer, Database administration
	CQ17. What kind of contract does the employer's offer? Seasonal Job; Autonomous
	CQ18. How much salary does the employer's offer? 3500 Euros, 3000 USD
	CQ19. What is the economic activity of the employer? Research; Financial; Education; Industrial
	CQ20. What is the description of the job offer? Sun Certified Java Programmer
	CQ21. What are the working conditions of the job offer? Full time; Partial time; Autonomous; Seasonal Job
	CQ22. What is the required education level for the job offer? Basic education; Higher education/University
	CQ23. What is the required work experience for the job offer? 1 year, 2 years, 3 years, 4 years, 5 or more years
	CQ24. What is the required knowledge for the job offer? Java, Haskell, Windows
	CQ25. What are the required skills for the job offer? ASP Programmer, Data warehouse, Hardware programming

Figure 4.6: Examples of CQs in the e-employment domain (SEEMP project)

tology requirements identified in the Ontology Specification activity, and resulting in ODPs integrated in the ontology network. As introduced in section 4.2, some

efforts have been devoted to the development of templates, repositories, methods and tools to support the reuse of ODPs. It is in this context that we have to understand the **XD method** described in section 4.2.3, which comes to support Scenario 7. The method that we have investigated in this PhD thesis is also intended to assist Scenario 7. It will be described in chapter 7.

4.4 Open Research Problems and Work Assumptions

To the best of our knowledge, at the moment of designing our contribution on knowledge acquisition and ontology modeling basing on ODPs, the XD method presented in section 4.2.3 was also ongoing work. In fact, both works have been devised in the framework of the NeOn Methodology and are inspired in this new paradigm.

As already reported, the XD method is intended for ontology engineers in general, and so is the tool provided for supporting this method, i.e., the XD NeOn Toolkit plug-in. Despite the major benefits provided by the XD method and tool in the tasks of searching, specialization, and integration of patterns in the final ontology, the XD tool still leaves users with the arduous task of selecting the pattern or patterns that better match their modeling problems. It is worth mentioning that the same method designers already showed interest in supporting the "matching" task (M. Suárez-Figueroa et al., 2009). However, this functionality was not available at the time of writing this document.

In the case of the method and tool proposed in this research work, **our objective is exactly to contribute to supporting users in the matching task**, and therefore, filling this gap. This is even more justified if we take into account that the target users of our method and tool are untrained users in ontology modeling.

The assumptions underlying our proposal of a method and tool for the ODPs Reuse activity based on a repository of LSPs associated to ODPs are listed below:

- We believe that it is feasible to intuitively identify linguistic structures that convey the meaning captured in ODPs.

- In establishing a correspondence between linguistic structures and ODPs, linguistic theories that analyze meaning construction in language can help confirming correspondences based on intuition, or explaining the behavior of ambiguous uses.

- Users without expertise in ontology modeling will encounter difficulties in the ODPs matching task.

- We believe that methods or guidelines designed for experts may not be helpful for novice users, so that specific guidelines should be defined for each type of target user.

- Prescriptive methodological guidelines that define precisely the tasks that are to be performed in each step, as well as the actors involved, can contribute to a better fulfillment of the activity.

4.5 Summary

Along this chapter we have introduced *design patterns* as they are understood in Computer Science in general, and, in particular, in the Ontological Engineering field. The importance of ODPs lies in providing consensual modeling solutions to less experienced users. ODPs are at the center of the repository we want to build for supporting a method and a tool that will enable pattern reuse to novice users.

After reviewing the state of the art on templates, repositories, methods and tools for supporting the reuse of ODPs in the ontology development process, we have presented the NeOn Methodology. Contrary to traditional methodologies, the NeOn methodology identifies a set of possible scenarios in any ontology development process according to requirements and available resources in the domain. Much emphasis is put in the reuse of ontological and non-ontological resources instead of promoting ontology development from scratch. This is why ODPs are seen as attractive resources that are worth constituting a scenario in the methodology by themselves.

Open research problems and work assumptions are presented regarding the lack of guidelines and tool support in one of the most complex tasks for novice users, namely, the matching task between a modeling problem and the ODPs that solve it.

Finally, we summarize the main contributions in this research topic, which will be presented in the next three chapters (chapter 5, chapter 6, and chapter 7)

1. a repository of LSPs associated to ODPs
2. a method to guide novice users in the ODPs reuse activity
3. a tool that relies on the repository of LSPs associated to ODPs to semi-automate the pattern matching task
4. a preliminary evaluation of the method and a subset of patterns in an academic setting

The *multilingual LSPs-ODPs pattern repository* is therefore the core of the proposed method and tool, since support will only be provided for the modeling components included in it. This repository is what demands the most effort on the side of the repository designer, but releases end users from having to understand ontology representation formalisms. A description of the patterns included in the repository is provided in chapter 5. Besides, the templates that have been designed to publish and share our LSPs in the Ontology Design Patterns Portal are also presented in chapter 7.

Then, the method we propose has the objective of guiding users in the formulation in natural language of the ontology specifications, as well as in the refinement of the input, in case no matches are obtained in the repository. The method is presented in chapter 6. As in the rest of methods identified in the framework of the NeOn Methodology, a template has been created to describe the ODPs Reuse activity, as well as a workflow diagram to indicate the execution order of each of the tasks that make up the activity.

Chapter 5

Multilingual LSPs-ODPs Pattern Repository

Chapter 5 will be devoted to the core of the proposed approach: the multilingual repository of LSPs associated to ODPs (henceforth *multilingual LSPs-ODPs pattern repository*).

Both the method and the tool we want to propose for supporting the reuse of ODPs have to rely on a repository that contains LSPs associated to ODPs. Thanks to this repository, whenever a user introduces in the system a sentence in NL that finds a match or correspondence with one LSP in the repository, a further correspondence with be established with one or several ODPs. Therefore, **this repository can be viewed as the means to achieve knowledge acquisition and ontology modeling** from NL statements.

Our strategy to identify LSPs is based on the assumption that any language has a number of lexical and compositional or generative mechanisms that reliably convey certain semantics (see section 2.2). Our research focuses on verb-centered LSPs, which are mainly composed by tuples of subject-verb-object. We claim that for expressing the relations holding among the concepts we want to model in ontologies, we rely on verbal expressions in affirmative or declarative sentences. In this kind of patterns, verbs are the ones that carry the semantics of the relation.

Thus, in this chapter, our purpose is to explain the different stages and strategies followed with the objective of:

1. Selecting a subset of representative ODPs
2. Obtaining and identifying *candidate verbal patterns* that convey the conceptual structures captured in a subset of ODPs
3. Analyzing the semantics of candidate verbal patterns that display polysemic or ambiguous uses by means of LCM lexical templates (see section 2.2.3), to establish a definitive correlation to one or several ODPs
4. Describing the multilingual LSPs-ODPs pattern repository

The different tasks or steps followed for the development of the multilingual LSPs-ODPs pattern repository are illustrated in Figure 5.1. Each one of these stages will be explained in more detail in the different sections of this chapter.

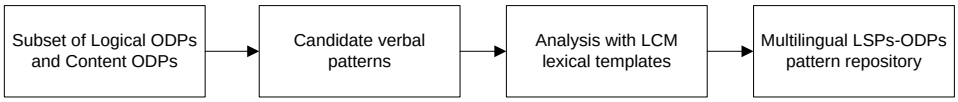


Figure 5.1: Steps in the development of the multilingual LSPs-ODPs pattern repository

5.1 Selection of Logical and Content ODPs

The starting point in the identification of LSPs corresponding to ODPs were the conceptual relations captured in a subset of ODPs, spelled out in the templates designed for their description in M. C. Suárez-Figueroa et al. (2007), Presutti et al. (2008), and the on-line repository Ontology Design Patterns Portal[1]. See figure 4.1 in section 4.2.1 for an example of an ODP template.

From the whole set of ODPs described in those resources, we selected a subset of basic or fundamental ODPs that are general enough so as to be used across domains of knowledge. This subset includes some Logical ODPs and Content ODPs that will be described in the following.

Logical ODPs

Logical ODPs were of great interest for our research for two main reasons:

- Logical patterns are content independent, which means that they can be used across domains of knowledge
- Logical patterns help to solve design problems where the primitives of the ontology representation language do not (always) directly support a certain logical construct

We will argue that the fact that Logical ODPs are intended to solve those design problems raised by the most basic representation options of ontology languages (OWL DL specifically), is an overriding reason for the purposes of this research. We should bear in mind that the method we propose for ontology modeling is intended at newcomers to ontology engineering, and they may need support even with the most elemental ontological constructions, such as the *subclass of* relation.

From the Logical ODPs described in M. C. Suárez-Figueroa et al. (2007), we chose the ones included in Table 5.1, because of their outstanding employment in general and domain ontologies.

[1] www.ontologydesingpatterns.org

Table 5.1: Subset of Logical ODPs selected for the LSPs-ODPs pattern repository

Logical ODPs for LSPs-ODPs pattern repository	
1	Logical ODP for Modeling a Defined Class
2	Logical ODP for Modeling a SubClassOf Relation
3	Logical ODP for Modeling Multiple Inheritance between Classes
4	Logical ODP for Modeling an Equivalence Relation between Classes
5	Logical ODP for Modeling an Object Property
6	Logical ODP for Modeling a Datatype Property
7	Logical ODP for Modeling a Universal Restriction
8	Logical ODP for Modeling Disjoint Classes
9	Logical ODP for Modeling Exhaustive Classes
10	Logical ODP for Modeling Specified Values

In the following, we provide a brief description of each of the patterns mentioned above, including the identifier they have received in the pattern repositories available in M. C. Suárez-Figueroa et al. (2007), and a brief description of their use. Descriptions and examples have been obtained and/or adapted from the ones included in the templates defined in M. C. Suárez-Figueroa et al. (2007).

1. *Logical ODP for Modeling a Defined Class* (LP-DC-01). This pattern expresses that elements which satisfy a given set of conditions belong to a certain group or set. To put it in other words, this means that if a class of elements satisfies a set of "necessary and sufficient conditions", then, it must be a member of a group or set. E.g., let us suppose that someone wants to express that a workflow which includes one or more business tasks is a business plan.

2. *Logical ODP for Modeling a SubClassOf Relation* (LP-SC-01). This pattern expresses that the elements that belong to a certain group or set, also belong to a more general set. E.g., suppose that someone wants to model that any business task is a task.

3. *Logical ODP for Modeling Multiple Inheritance between Classes*(LP-MI-01). It models elements that belonging to a certain group or set, also belong to several more general sets. E.g., someone wants to model that any beginning of selling process is a beginning task and also a business task.

4. *Logical ODP for Modeling an Equivalence Relation between Classes* (LP-EQ-01). This pattern allows to model that two groups have precisely the same set of elements. E.g., let us assume that someone wants to express that business tasks are the same as commercial tasks.

5. *Logical ODP for Modeling an Object Property* (LP-OP-01). It models that elements that belong to a certain group or set have a relationship or link with elements that belong to a different group or set. E.g., someone wants to model that business plans have business tasks.

6. *Logical ODP for Modeling a Datatype Property* (LP-DP-01). It models elements that belong to a certain group or set and that have a relationship or link with elements of a group or set of the type of literals, values, etc. E.g., someone wants to express that tasks have a name and a description.

7. *Logical ODP for Modeling a Universal Restriction* (LP-UR-01). This pattern expresses that a set of elements only have relationships to elements belonging to another group or set. E.g., suppose that someone wants to express the set of individuals that only have the relationship "has business task" with the individuals in the class "business task".

8. *Logical ODP for Modeling Disjoint Classes* (LP-Di-01). It models that elements that belong to a certain group or set cannot belong to another group or set, i.e., the two sets are disjoint and do not share instances. E.g., suppose that someone wants to model that plans and tasks are disjoint.

9. *Logical ODP for Modeling Exhaustive Classes* (LP-EC-01). It expresses that a general group or set is the union of several more specific groups or sets, which in its turn are mutually disjoint. E.g., someone wants to express that types of control task can only be a begin task, and end class or a sequential task.

10. *Logical ODP for Modeling Specified Values* (LP-SV -01). This pattern allows to represent that a class has a set of descriptive values for features, and that those values are different between or among them. E.g., someone wants to model that business plans can only have three values regarding their acceptance status, that is, they can only be accepted, non-accepted, or in process of revision.

Content ODPs

A selection of Content ODPs was also made on the basis of relevance. Content ODPs solve design issues that may rise in ontologies dealing with certain domain content. Contrary to Logical ODPs, Content ODPs represent modeling solutions that are not directly supported by logical constructs of ontology languages. The number of Content ODPs is quite large, and is increasing constantly as new solutions to ontology modeling problems are found by ontology engineers working in new domains.

In this research, we restrict our selection of Content ODPs to a sample of them that, despite being common to certain domains, are general enough to appear across

most domains of knowledge. Good representatives in this sense are the Content ODPs for modeling participation, location or the part-whole relation. The complete list of Content ODPs dealt in this work can be seen in Table 5.2.

Table 5.2: Subset of Content ODPs selected for the LSPs-ODPs pattern repository

Content ODPs for LSPs-ODPs pattern repository	
1	Content ODP for Modeling Participation
2	Content ODP for Modeling Co-participation
3	Content ODP for Modeling Location
4	Content ODP for Modeling Object-Role
5	Content ODP for Modeling Simple Part-Whole Relation
6	Content ODP for Modeling Constituency
7	Content ODP for Modeling Componency
8	Content ODP for Modeling Collection-Entity

The Content ODPs we have chosen to be included in the LSPs-ODPs pattern repository are defined in Presutti et al. (2008) or in the Ontology Design Patterns Portal, with the exception of the Simple Part-Whole Relation pattern that is defined in M. C. Suárez-Figueroa et al. (2007). See descriptions below:

1. *Content ODP for Modeling Participation* (CP-PA-01). This pattern models the participation of sets of elements in events. E.g., let us suppose that someone wants to express that an actor participates in a film premiere.

2. *Content ODP for Modeling Co-participation* (CP-CPA-01). This pattern represents the participation of two elements in a same event. E.g., someone wants to model that two actors participate in the same film premiere.

3. *Content ODP for Modeling Location* (CP-LO-01). It models a generic, relative localization holding between any groups or elements. E.g., someone wants to model that a city is located in a certain county or region.

4. *Content ODP for Modeling Object-Role* (CP-OR-01). This pattern allows to model sets of elements and the role they play. E.g., let us imagine that someone wants to express that an old glass is used as a flower pot.

5. *Content ODP for Modeling Simple Part-Whole Relation* (CP-PW-01). This pattern models general relations between wholes and their parts in a transitive manner, i.e., the part of an object is also part of the whole that contains the object. E.g., someone wants to model that the brain is part of the human body, and that the *substantia nigra* is part of the brain.

6. *Content ODP for Modeling Constituency* (CP-CONS-01). This pattern represents the constituents of a layered structure. E.g., someone wants to model that a table is constituted of different types of wood.

7. *Content ODP for Modeling Componency* (CP-COM-01). This pattern represents that objects are either proper parts of other objects, or have proper parts. This relation is understood in a non-transitive way, i.e., the part of an object is not part of the whole that contains the object. E.g., suppose that someone wants to model that a turbine is a part of the engine. This does not mean that the parts of the turbine are also parts of the engine.

8. *Content ODP for Modeling Collection-Entity* (CP-CE-01). It models the relation between collections or groups and its members. E.g., someone wants to model that the Louvre has a collection of Aegyptian objects.

Once we had selected the subset of ODPs we wanted to investigate, the next step was to look in the languages object of this research for forms of realizing the conceptual knowledge captured in those ODPs.

5.2 Strategies for the Identification of *candidate verbal patterns*

With the aim of identifying *candidate verbal patterns* we explored three strategies that will be explained in the following:

1. Manual analysis of handbooks, encyclopedic documents, and ORDS documents for the discovery of a tentative list of "seed" verbs

2. Use of pairs of conceptually related terms in a search engine to retrieve additional seed verbs

3. Use of seed verbs and state of the art verbal patterns to look for concordances in corpus and analyze knowledge-rich contexts

Since the selected subset of ODPs represents some of the most basic relations of any domain of knowledge, we decided to look at descriptive documents or encyclopaedic sources. These documents contain the lexical, grammatical and rhetorical features employed in the description and organization of any domain of knowledge. We prepared an *ad hoc* corpus, i.e., a "special purpose corpus, a corpus whose composition is determined by the precise purpose for which it is to be used", as proposed by Pearson (1998: 48). In this way, one of the problems of corpus based research for extracting linguistic knowledge was avoided, namely, the rare presence of the patterns targeted.

The *ad hoc* corpus was mainly composed of two types of documents. On the one hand, we collected Web documents mainly concerned with Natural Science disciplines, i.e., biology, chemistry, astronomy, etc., in which classifications

and partonomy descriptions are usual discourse subjects. On the other hand, we selected several Ontology Requirement Specification Documents (ORSD) used in European and national projects for the development of ontologies (SEEMP[2], NeOn[3], SensorGrid4Env[4], MiO[5]). The sections that interested us from the different ORSD were mainly: (a) the CQs section, and (b) those sections of the ORSD in which descriptions of the domain were included. This initial corpus served to establish a tentative list of "seed" verbs and to manually discover its main patterns of use.

The second strategy for finding out seed verbs expressing the relations captured in ODPs was inspired by Hearst (1992, 1998) and Finkelstein-Landau and Morin (1999). These authors propose to collect pairs of terms linked by the selected relation in lexicons (for instance WordNet[6]) or thesauri, and then look for sentences in which the conceptually related terms occur. For this purpose we used the Google search engine, introduced the pair of terms, and looked for sentences in which both were related. The results obtained were not so fruitful because the system returned documents in which the two terms appeared, but not necessarily in the same sentence or in the nearby context. See Figure 5.2 for an example of the retrieved sentences in which the pair of terms "book" and "chapter" were selected to look for verbs indicating the *part whole* relation. This research has been reported in Sabou et al. (2009).

*by Andrew Gelv, this **book** includes a **chapter** on juggling and unicycling. Most people*
*Francombe. This **book** includes an interesting **chapter** on the history of juggling and covers*
*the **book** is particularly strong. Abelson's **chapter** (with a third attack on BASIC) provides*
*than a graduation. The **book** finishes with a **chapter** on buying and chartering a boat. No*
*The Brothers Karamazov [f] Part II, **Book** V, **chapter** 4. [p] Eknath Easwaran, [f] The*
*40 years that one more **book** or even one more **chapter** could not add much that is new.*
*Flower [f] is a unique **book**. Beginning at **Chapter** XV (the Original Gate), it propounds*
*are the core concerns throughout this **book**. In **Chapter** 1 Nigel Parton focuses on the assu*

Figure 5.2: Examples of conceptually related terms in nearby context

Finally, the intial set of verbs was complemented by some of the verbal patterns identified in previous research on knowledge acquisition from text introduced in chapter 3. From those works we considered the verbal patterns identified for English in Cimiano and Wenderoth (2005, 2007) (see table 3.7 in section 3.1.1), with the exception of the ones identified for the relation of *function* and *originator*. Regarding the verbal patterns for Spanish, we included the ones identified in Alvarez de Mon and Aguado de Cea (2006) and some from Soler and Alcina (2008) that were less domain specific (see tables 3.5 and 3.6, respectively, in section 3.1.1).

[2]http://ec.europa.eu/information_society/activities/egovernment/research/projects/seemp/index_en.htm

[3]http://www.neon-project.org/nw/Welcome_to_the_NeOn_Project

[4]http://www.semsorgrid4env.eu/home.jsp

[5]http://www.cenitmio.es/

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

The whole set of seed verbs was then used to search for sample sentences in the Web and in on-line corpora to retrieve knowledge-rich contexts. The advantages of using on-line corpora is that these corpora permit to look for concordances, and it made it very easy to identify knowledge-rich contexts (see an example of the search for "classified" in Figure 5.3). The corpora chosen for our purposes were:

- British National Corpus (BNC)[7]
- Cobuild Concordance and Collocation Sampler[8]
- Corpus of current Spanish of *Real Academia Española* (CREA)[9]

The BNC contains a 100.000 million word collection of samples of written and spoken English from a wide range of sources, and represents a wide cross-section of current British English, both spoken and written. The Cobuild Concordance and Collocation Sampler is composed of 56 million words of contemporary written and spoken text. And, finally, the CREA Corpus for the Spanish language contains more than 150 million words and nine sub-corpus, from which we focused only on the Science and Technology sub-corpus with 10% of all the documents of the CREA.

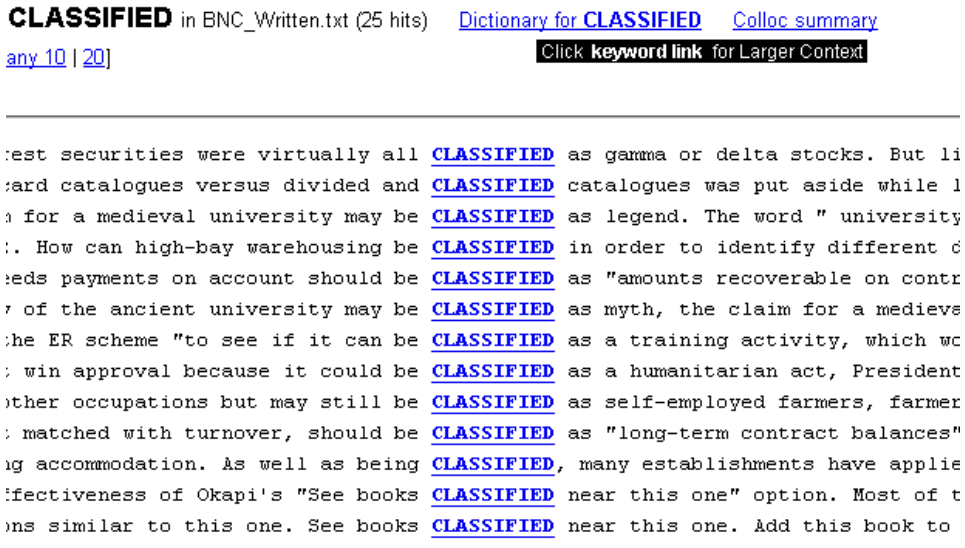


Figure 5.3: Search for "classified" concordances in the BNC

The result of this initial step was a set of candidate verbal patterns (such as, be, classify as, include) and sentence structures (be a(n), be something that, be known as) that we related with the set of Logical and Content ODPs previously selected, as shown in table 5.4. Only the English verbal patterns have been included in these table.

[7] http://www.natcorp.ox.ac.uk/
[8] http://www.collins.co.uk/Corpus/CorpusSearch.aspx
[9] http://corpus.rae.es/creanet.html

	ODPs	candidate verbal patterns	LCM lexical template	LSP-ODP pattern repository
Logical	**Defined Classes**	be + prep.	n.a.	Table 5. 29
	Subclass of	be a(n), be type of, be either…or	Table 5. 4 Table 5. 5	Table 5. 18 Table 5. 29 Table 5. 30
		classify, classify as, fall into	Table 5. 6 Table 5. 8	Table 5. 18 Table 5. 27 Table 5. 28
		classify in/into, categorize in/into, group in/into	Table 5.7	Table 5. 18 Table 5. 30
		divide in/into, separate in/into	Table 5. 7	Table 5. 32 Table 5. 30
		include	Table 5. 10	Table 5. 32
		belong to	Table 5. 12	Table 5. 18
		there is/are	n.a.	Table 5. 18 Table 5. 30
	Multiple Inheritance	be (an), be type of	Table 5. 5	Table 5. 19
		classify as	Table 5. 8	Table 5. 19
	Equivalent Relation	know as, call, refer to as	n.a.	Table 5. 20
	Object Property	have	Table 5. 13 Table 5. 14	Table 5. 21 Table 5. 31 Table 5. 33
	Datatype Property	be	n.a.	Table 5. 22 Table 5. 33
		have	Table 5. 14	Table 5. 22 Table 5. 33
	Universal Restriction	be only/just/exclusively	n.a.	Table 5. 31
	Disjoint Classes	differ, be different from, be either…or	Table 5. 5	Table 5. 23 Table 5. 30
	Exhaustive Classes	see Subclass of	n.a.	Table 5. 30
	Specified Values	can/may be…(either)…or	n.a.	Table 5. 24
Content	**Participation**	participate, take part in, involve in	n.a.	Table 5. 25
	Co-Participation	see Participation	n.a.	Table 5. 26
	Location	locate, find, set, situate, place	n.a.	Table 5. 27
	Object-Role	use as, work as, act as, serve as	n.a.	Table 5. 28
	Part-Whole Relation	divide in/into, separate in/into	Table 5. 9	Table 5. 32
		include	Table 5. 15	Table 5. 32
		belong to	Table 5. 11	Table 5. 34
		have	Table 5. 13	Table 5. 33
		contain, form part of, consist of, comprise, compose, make up, constitute	Table 5. 15	Table 5. 34
	Constituency	see Part-Whole Relation	Table 5. 15	Table 5. 34
	Componency	see Part-Whole Relation	Table 5. 15	Table 5. 34
	Collection-Entity	see Part-Whole Relation	Table 5. 15	Table 5. 34

Figure 5.4: Summarizing table: from ODPs to the English LSPs-ODPs pattern repository

As a result of the identification of candidate verbal patterns for the specific subset of ODPs, we could draw some previous conclusions:

- Some verbs showed **polysemic behavior** and were found in sentences that conveyed the semantics of several ODPs. This is the case of verbs such as:
  - be
  - classify
  - divide
  - include
  - belong
  - have
  - contain
- The behavior of some verbs changed according to their syntagmatic relations, or what is the same, to the elements that accompany them. These elements are normally prepositions, nouns or adverbs that modify or restrict the meaning of the verb. Some examples have been listed below:
  - classify *as* vs. classify *into*
  - be *type* of vs. be *part* of
  - be *only*
- Some verbs needed to be analyzed taking into account the whole sentence structure they were inserted in (*be either...or, be something that, may/can be*)

For those verbs that exhibited a polysemic behavior, we relied on the lexical template proposed in section 2.2.3, which resulted after combining the decompositional system of the LCM lexical templates for representing the semantic and argument structure of verbs, with Pustejovsky's event and *qualia* structures. By relying on these templates, the correspondence between the linguistic structure and the conceptual structure representing that meaning in an ontology is more straightforward and reliable. The rest of verbs and verbal phrases were directly formalized and included in the *multilingual LSPs-ODPs pattern repository*.

The correspondence between the initial set of ODPs, their related candidate verbal patterns, the LCM lexical templates for describing polysemous uses, and the tables that make up the final LSPs-ODPs pattern repository are illustrated in figure 5.4. Note that this analysis has only been performed for the English language. In case no LCM lexical template is available for some of the candidate verbal patterns, it has been marked with the *n.a.* abbreviation for not applicable.

5.3 LSPs on the light of the Lexical-Constructional Model

As introduced in section 2.2.3, the template we propose in this work for analyzing candidate verbal patterns is based on the lexical template proposed by Mairal Usón and Ruiz de Mendoza Ibáñez (2008) in the LCM, in combination with the formalisms defined by Pustejovsky (1995: 104 ff.) in the framework of the Generative Lexicon theory. This template, which we will refer to as LCM lexical template, allows us to account for those properties of a lexical item which go beyond those aspects of meaning that are grammatically relevant.

LCM lexical templates provide the mechanisms to

1. define rich semantic representations of verbal predicates
2. account for the semantics-to-syntax mapping in a systematic way
3. establish paradigmatic relations between verbal predicates that belong to the same domain
4. analyze and represent verbal polysemy

Again we include the template designed for analyzing the "deep semantics" of the linguistic structures that we have previously identified in domain documents (see table 5.3). In this way we can reliably predict the meaning of linguistic structures and establish a link or correspondence to the ontological constructs that better represent the meaning conveyed by those structures. We argue that this decomposition of verbal templates in their semantic and pragmatic properties is, if possible, more urgent, if our purposes involve the use of these patterns in a system for the automatic transformation of natural language input into ontological constructs.

LCM EVENTSTR stands for the *Aktionsart* module as understood by Mairal Usón and Ruiz de Mendoza Ibáñez (2008) in the LCM. GT stands for Generative Lexicon and indicates that event (EVENTSTR), argument (ARGSTR) and *qualia* (QUALIASTR) structures are represented according to the formalisms defined by Pustejovsky (1995: 105 ff.).

For the description of the lexical templates in this section we will proceed in the following way: First, we will describe GT EVENTSTR, GT ARGSTR and GT QUALIASTR, and then we will look into the LCM EVENTSTR. This order reproduces the steps taken for the semantic description of the verbal predicates. First, we identify the type of event or (sub)events involved in the verbal structure. Then, we identify the number and the type of arguments that participate in the structure. In the third step, we define the *qualia* structure, whose purpose is to specify the semantic properties of each of the arguments and (sub)events. As a corollary, the LCM EVENTSTR brings together the descriptions provided by the rest of structures.

From all the verbal predicates and phrases identified in our corpora we only include here i) some verbs whose sense specification was relevant for the subsequent correspondence to ODPs, and ii) some verbs that present polysemous uses,

Lexical Template	
verbal pattern	infinitive form
LCM EVENTSTR	*Aktionsart* module
GT EVENTSTR:	E_1 = e_1: [state, activity, achievement, etc.] E_2 = e_2: [state, activity, achievement, etc.] *Restr* = [$<\infty$, $o\infty$, $<o\infty$] *HEAD* = [e_1 e_2]
GT ARGSTR:	ARG_1 = [human, artifact, class, etc.] ARG_2 = [human, artifact, class, etc.] *D-ARG* = [human, artifact, class, etc.] *S-ARG* = [human, artifact, class, etc.]
GT QUALIASTR:	Q_F = [hypernymy-hyponymy] Q_C = [meronymy] Q_T = [function] Q_A = [origin, cause]

Table 5.3: LCM lexical template

and whose disambiguation was needed for an appropriate mapping to ODPs. We have also restricted this analysis to verbal phrases in English, although most of the conclusions are also valid for the equivalent verbal phrases in Spanish. The subset of verbs and verbal phrases for which we formulate lexical templates is listed in the following:

1. be a(n)
2. be either... or...
3. classify
4. classify into
5. classify as
6. divide into
7. include
8. belong to
9. have
10. contain

1. be a(n)

The lexical template for the verbal phrase *be a(n)* is shown in table 5.4. This verbal pattern participates in sentences like *A cat is an animal*. Here we are dealing with a state verb that involves only one event (e_1). The argument structure consists of two *true arguments* (i.e., those syntactically realized) and one default argument (i.e., relevant for the *qualia* but usually not syntactically realized). The true arguments are specified as generic names for classes or categories. The first argument (x) refers to the subclass, and the second argument (y) to the superclass of the relation. The default argument refers to the criterion or criteria underlying any classification act.

In the *qualia* structure, the formal *quale* specifies the nature of the event, which belongs to the "existence" lexical domain (Faber and Mairal Usón, 1999: 279). Finally, the LCM EVENTSTR relies on the primitive predicate BE, defined as a "specification" primitive, and on the "relational substantive" for taxonomy, KIND (see Wierzbicka's semantic primes included in table 2.1, section 2.1). It indicates that the relation existing between the two arguments of the verbal predicate is one of a subclass to its superclass.

Note that this verbal phrase would encode a different meaning without the complement introduced by the indefinite article *a*. Thus, here, the complement and the type of arguments that participate in the construction constrain the meaning and syntactic realization of the verb *be*. This is a clear **example of what Pustejovsky defined as co-composition, a generative mechanism in which complements carry information which modifies or specializes the original meaning of the verb**. To this respect, *to be* is a highly polysemous verb, and the sense represented here corresponds to one of its multiple senses.

Lexical Template	
verbal pattern	be a(n)
LCM EVENTSTR	be kind of (x, y)e_1
GT EVENTSTR:	E_1 = e_1: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [subclass] ARG_2 = y [superclass] *D-ARG* = v [criteria]
GT QUALIASTR:	Q_F = [exist in a relation of kind of to (e_1, x, y)]

Table 5.4: Lexical template for *be a(n)*

The same lexical template would apply for the *be a type of* construction and its variants (e.g., be a kind of, be a class of, etc.). In the same sense, this lexical template would also be valid for sentences like *Cats are animals*, in which the indefinite article has disappeared, but the arguments (x, y) are compulsorily in plural. However, should the second argument (y) be represented by an adjective, instead

of a class name representing the superclass, we would be describing one of the properties of the first argument (x), as in *Cats are flexible*.

2. be either... or...

The pattern analyzed in table 5.5 is also a state verb involving one event and two arguments, which are generic names for classes or categories (x, y). The difference between this pattern and the previous one is that the order of the arguments has been inverted. In the present case, the superclass is introduced by the first argument (x) and the subclasses are represented in the second argument (y). An additional difference is that the second argument (y) is represented by the union of two sub-arguments that are disjoint between them, as syntactically marked by the conjunction *or*.

Lexical Template	
verbal pattern	be either... or...
LCM EVENTSTR	have types (x, y)e_1
GT EVENTSTR:	E_1 = e_1: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [superclass] ARG_2 = y [subclass $\cup$ subclass] [subclass $\neq$ subclass] *D-ARG* = v [criteria]
GT QUALIASTR:	Q_F = [exist in a relation of kind of to (e_1, x, y) have types (e_1, x, y)] Q_C = [be only types (e_1, y), be disjoint (e_1, y)]

Table 5.5: Lexical template for *be either... or...*

The formal *quale* provides two types of information. First, that the event belongs to the "existence" lexical domain, as in the previous pattern, and, second, that the sub-arguments included in *y* are types of the superclass.

The nature of the second argument (y) is restricted by the constitutive *quale* and also exposes two characteristics. The first argument (x) is equal to the sum of the two subclasses that make up the second argument (y). This is represented in the argument structure by means of the union symbol ($\cup$) between the two subclasses. Furthermore, the two subclasses that make up the superclass are incompatible or disjoint between them, which means that they do not share instances of the real world. This is expressed by means of the negative symbol ($\neq$) in the argument structure of the second argument (y).

These particular features or restrictions of the second argument (y) could have been expressed by means of Mel'Cuk lexical functions (see table 2.2 in section 2.1). The function ALL applied to argument *y* would mean that the two subclasses

are the only subtypes of the superclass, and the function ANTI for antonym, could have pointed out to the fact that the subclasses are disjoint. Yet, we preferred the use of logical symbols because they are closer to the formalisms used in ontology construction.

The LCM EVENTSTR tells us that the structure *be either... or...* is defined by the primitive predicate HAVE and the relational substantive TYPE, which define the type of relation between the superclass and its subclasses.

3. classify

In table 5.6, we represent the lexical template for the verb *classify*. According to Faber and Mairal Usón (1999: 284), *classify* belongs to the lexical domain of position verbs, and it is defined as "to put something in a particular position/order". However, we argue that the new position or order into which something is put, is the result of a movement act. This assumption will also allow us to decompose the semantic meaning of the verb *classify* into the *movement* semantic prime MOVE, and the verbal form of the *space* semantic prime PLACE (see table 2.1 in section 2.1).

Classify represents an activity as in *I am classifying the books*. However, it can also become a telic predicate when the result is headed: *I have classified the books*. In this respect, we have decided to treat this verb as an active accomplishment, taking into account that the contexts we retrieved conveyed this meaning. Its semantic representation can be seen in table 5.6.

Lexical Template	
verbal pattern	classify
LCM EVENTSTR	[do' (x, move to (other) place' (x, y)]e_1 & [BECOME be in (new) place (y)]e_2 $E_1 <\infty E_2$
GT EVENTSTR:	E_1 = e_1: [process] E_2 = e_2: [state] *Restr* = [$<\infty$] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [human] ARG_2 = y [class] *D-ARG* = w [criteria]
GT QUALIASTR:	Q_F = [classified (e_2, y)] Q_A = [classify_act (e_1, x, y, w)]

Table 5.6: Lexical template for *classify*

The event structure encodes two subevents: an activity or process and a final resulting state. The relation between the activity and the result involves an *exhaustive ordered part of* restriction, as signalized by the symbol $<\infty$ (see section

2.2.2). Each subevent maps onto one *quale*: the agent *quale* specifies the nature of the activity subevent, and the formal *quale* maps onto the result of the activity.

Depending on which event is headed, we would have a non-telic interpretation (e_1) or a telic interpretation (e_2), in which the final result would be foregrounded. In the present template, we head the activity, as indicated by the HEAD property (*HEAD* = [e_1]). The telic interpretation, however, would ask for *into which classes are books classified?* The explicit mention of the classes into which something is classified is additional information that complements or composes the semantics of the verb *classify*. This could be analyzed in the same template, but we have decided to create a new one for the *classify into* phrase for the sake of clarity (see table 5.7).

Regarding the argument structure of the non-telic interpretation of *classify*, there are two true arguments and one default argument. In the LCM EVENTSTR we try to combine event, argument and *qualia* structure by making use of the semantic primitives MOVE and PLACE. This structure should be rendered as a movement action that results in the object being in a new place or position. The movement activity precedes the state represented by the new position and which is modified by the telic operator (BECOME).

4. classify into

When the subclasses into which a class is classified are mentioned explicitly, the lexical template is extended in order to account for the new position of the classes, which was unknown in the previous structure. This additional meaning is introduced by the preposition *into*, as represented in table 5.7. The new derived sense is generated because of what Pustejovsky calls co-composition method (section 2.2.2), in which the meaning of the verb and the meaning of the preposition are combined.

The event structure in this case is quite complex. It consists of four subevents: two activities or processes and two states. The first process and state (e_1 and e_2) coincide with the previous template of the verb classify, and correspond to the first part of the formal and agentive *qualia*, respectively. The agent *quale* corresponds to the classify act according to certain criteria, and the formal *quale* to the result of the classification (*I am classifying -have classified- the books according to the topic*).

Now, if we want to specify the classes into which something has been classified, we understand that a set of disjoint subclasses is created, and that the items that form part of the superclass are distributed in more specific subclasses (e.g., *I have classified the books into Natural Science books and Literature books*). The set of subclasses is assumed to make up the superclass, which is represented by the constitutive *quale*, and they are considered disjoint among them. The second part of the agentive *quale* describes the change of position, in which the superclass is now distributed into the different subclasses. The second part of the formal *quale*

Lexical Template	
verbal pattern	classify into
LCM EVENTSTR	[do' (x, move to (other) place' (x, y) & place in (x, y, v)]e_1 e_3 & [BECOME be in (new) place (y) & have types (y, v)]e_2 e_4 $<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4)
GT EVENTSTR:	E_1 = e_1: [process] E_2 = e_2: [state] E_3 = e_3: [process] E_4 = e_4: [state] *Restr* = [$<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4)] *HEAD* = [e_4]
GT ARGSTR:	ARG_1 = x [human] ARG_2 = y [superclass] ARG_3 = v [subclasses (subclass $\neq$ subclass $\neq$ subclass...)] *D-ARG* = w [criteria]
GT QUALIASTR:	Q_F = [classified (e_2 y), exist in a relation of kind of to (e_4 y, v)] Q_C = [be only types (e_4, y), be disjoint (e_4, y)] Q_A = [classify_act (e_1, x, y, w), move into new class (e_3 y, v)]

Table 5.7: Lexical template for *classify into*

indicates the existence of a set of subclasses that are in a relation of "kind of" with respect to the superclass.

The final state (e_4) is headed, which corresponds to the set of subclasses that now exist and into which the items of the superclass have been divided. This representation matches sentences like *Membrane proteins are classified into integral proteins and peripheral proteins*, in which the result of the classification is foregrounded. This is the type of sentences we have come across in our domain corpora.

The argument structure accounts for four arguments: three true arguments and one default argument. ARG_3 is represented by the subclasses, which have the property of being disjoint among them, so that the items of the superclass can be indisputably placed under the right subclass.

The temporal sequence between subevents is encoded by means of the relation $<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4), which means that the first pair process-state (E_1, E_2) precedes the second pair process-state (E_3, E_4).

The LCM EVENTSTR represents an active accomplishment involving an activity and a final resulting state modified by the telic operator BECOME. The activity event maps to the agentive *quale* and describes the movement and placing

actions. The final result indicates that the subclasses have been classified or are in a new place, and that the result of the classification consists of the superclass staying in a relation of "have types" to the subclasses.

It is also important to note that sometimes a further argument is included with the aim of determining the number and type of classes into which a class is divided. Let us refer again to the *proteins* example but slightly modified: *Membrane proteins are classified into **two categories**: integral proteins and peripheral proteins*. In this sentence, the generic class name "category" has been included, together with the number of categories into which membrane proteins are classified. It could be stated that this argument has a referential (cataphoric) nature, because it introduces the number of categories that are going to be specified after the colon.

When dealing with the verb *classify*, this information could be said to be redundant, but we will see that in other verbal patterns it is needed for disambiguation. A further verbal pattern follows the same pattern as *classify* and *classify into* is *group*. Consider some examples below:

(1) *I am grouping the papers based on their length.*

(2) *Papers are grouped into three categories: short papers, long papers, and position papers.*

5. classify as

Next, we will deal with a further co-compositionally derived sense of the verb *classify*, in which its original meaning is constrained by the use of the prepositional phrase introduced by *as*. Once again, the event structure depicts four subevents and four arguments (see table 5.8).

What interests us here is that the order of the second and third argument has been inverted, so that subclasses come first in the order, and then the superclass to which they belong is mentioned. This is also reflected in the formal *quale* related to the fourth subevent (e_4) that establishes a relation of *subclass of* or *kind of* between arguments. Note also that the constitutive *quale* present in the previous template for *classify into* has disappeared, and, therefore, **exhaustiveness cannot be assured for the second argument (y) in this construction**.

Finally, we will refer to a verb that can participate in the three senses and alternations analyzed here for the verb classify: *categorize*. This is illustrated by the examples below:

(3) *Cigars are categorized by the country where they were made.*

Lexical Template	
verbal pattern	classify as
LCM EVENTSTR	[do' (x, move to (other) place' (x, y) & place in (x, y, v)]e_1 e_3 & [BECOME be in (new) place (y) & be kind of (y, v)]e_2 e_4 $<\infty$ (E_3, E_4), $<\infty$ (E_1, E_2)
GT EVENTSTR:	E_1 = e_1: [process] E_2 = e_2: [state] E_3 = e_3: [process] E_4 = e_4: [state] *Restr* = [$<\infty$ (E_3, E_4), $<\infty$ (E_1, E_2)] *HEAD* = [e_4]
GT ARGSTR:	ARG_1 = x [human] ARG_2 = y [subclasses (subclass $\neq$ subclass $\neq$ subclass...)] ARG_3 = v [superclass] *D-ARG* = w [criteria]
GT QUALIASTR:	Q_F = [classified (e_2 y), exist in a relation of kind of to (e_4 y, v)] Q_A = [classify_act (e_1, x, y, w), move into (super)class (e_3 y, v)]

Table 5.8: Lexical template for *classify as*

(4) *These novels are categorized into beginner, intermediate, and advanced levels.*

(5) *Historical research is categorized as a qualitative research method.*

A further example in which the co-composition generative mechanism is clearly manifested is represented by the verb *fall* and its phrasal structure *fall into*. Originally, to fall denotes a change of position, and would typically involve an activity followed by a resulting state. However, when combined with the preposition *into*, and specifically within a construction of the type *fall into the group/class/category...*, it acquires the meaning of the *classify into* verbal phrase. See an example below:

(6) *The psalms fall into different categories, such as hymns, thanksgivings, laments, royal psalms, pilgrimage songs, etc.*

6. divide into

The next verbal predicate we would like to analyze with respect to its semantic, syntactic and pragmatic properties is *divide*. Faber and Mairal Usón (1999:

282) regard it as belonging to the *movement* lexical domain, in which something is "moved apart" from something else. This basic meaning is found in sentences such as:

(7) *Most bog garden and waterside plants should be divided during the dormant season.*

(8) *The disparities of opinion divided the party.*

(9) *Profits have been divided.*

This division in its "physical" sense refers to parts, areas or differentiated groups that result from a moving, separation or even cutting process. *Divide* is more often than not used in combination with the preposition *into*, which introduces the number and type of parts that result from the dividing process, as in *The field is divided into three panels or compartments*. To illustrate this, consider the lexical template for the verbal phrase *divide into* in table 5.9.

Lexical Template	
verbal pattern	divide into
LCM EVENTSTR	[do' (x, move apart' (x, y) & create parts (x, v)]e_1 e_3 & [BECOME be moved apart (y) & have parts (y, v)]e_2 e_4 $<\infty$ (E_3, E_4), $<\infty$ (E_1, E_2)
GT EVENTSTR:	E_1 = e_1: [process] E_2 = e_2: [state] E_3 = e_3: [process] E_4 = e_4: [state] *Restr* = [$<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4)] *HEAD* = [e_4]
GT ARGSTR:	ARG_1 = x [human] ARG_2 = y [whole] ARG_3 = v [parts] *D-ARG* = w [criteria]
GT QUALIASTR:	Q_F = [divided (e_2 y), have parts (e_4 y, v)] Q_C = [made up of (e_4 y, v)] Q_A = [divide_act (e_1, x, y, w), move apart from something else (e_3 y, v)]

Table 5.9: Lexical template for *divide into*

The event structure indicates that *divide into* is an active accomplishment verb that involves two activities or processes and two result states ordered by the relation *exhaustive ordered part of* $<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4), in which each process subevent precedes each resulting state subevent. The final event is headed, which means that the final division result is foregrounded. With regard to the argument

structure, there are three true arguments and one default argument that represents the criteria behind the division process.

The nature of ARG_2 and ARG_3 is crucial for the meaning of this structure. ARG_2 (y) represents the object or artifact that is to be divided into parts, and ARG_3 consists of the parts (v) that make up the object. The relation between the object and its parts is expressed in the constitutive *quale*. Then, the agentive and formal *qualia* specify the activities and results of the subevents involved in this structure.

The first part of the agentive *quale* expresses that the effector (x) carries out a divide act according to certain criteria (w) in order to make the resulting state come about (y). This *quale* maps to e_1 subevent, and the first part of the formal *quale* maps to the (intermediate) resulting state e_2.

The second part of the agentive *quale* expresses the divide act *per se* in which the parts (y) are created or actually "moved apart from the whole (v)" (e_3), which results in the definitive identification of the parts (e_4) in which the whole is divided. The kind of sentences that would match this lexical template, in which the parts are foregrounded, are:

(10) *This chapter is divided into land use studies, landscape studies, and landscape evaluation.*

The LCM EVENTSTR represents an active accomplishment involving an activity and a final resulting state modified by the telic operator BECOME. The semantic primes employed here are MOVE and PART, the relational substantive for partonomy (table 2.1). The activity event maps to the agentive *quale* and describes the movement action and the creation or bring into being of the parts. The final result maps to the formal *quale* and indicates that the parts are created and made explicit.

Consider now sentences (11) and (12) extracted from the Web:

(11) *Electrochemical cells can be divided into two categories: galvanic cells and electrolytic cells.*

(12) *The following suggested readings are divided into primary resources (i.e., original literature) and secondary sources (i.e., scholarly writings).*

In these sentences, the relation between ARG_2 and ARG_3 is not the one of a whole to its parts, but that of a superclass to its subclasses. We can state that since the nature of the arguments has changed, there has also been a shift of meaning from a *part-whole* relation to a *subclass of* relation. The eventual structure would remain the same, and the argument and *qualia* structures would now correspond to the ones of the *classify into* lexical template dealt with in table 5.7.

Divide into, thus, is considered an **ambiguous verb that can convey two different types of relations, namely, between an object and its parts, and between**

a class and its subclasses. This could be regarded as a metaphorical use of the physical sense of *divide*, which projects this separation and identification of the parts of an object to the subclasses that make up a class.

In fact, cognitivists working on metaphors like Lakoff (1987) propose some basic frames from the most direct experience of humans that are then extrapolated to other spheres. The ones that interest us here are the *part-whole frame* and the *container-content frame*. The first one is related to the perception and experimentation we have of our bodies as wholes from which parts can be distinguished. The second also considers the body as a container in which activities such as breathing, eating, etc., take place. According to Lakoff, this perception would be extrapolated to other real or abstract concepts such as classes, in this case, which can be divided into subclasses in order to organize and understand domains of knowledge.

For all these reasons, we will contend that **the nature of the arguments plays a decisive role in verbs such as *divide into* or *separate into*, since they will restrict the sense of the relation**. This also justifies the inclusion of a generic name describing the argument type, which normally appears introducing a further (default) argument, as in the sentences below:

(13) *The book is divided into **chapters** (...).*

(14) *The Congo is divided into six **provinces**: Leopoldville, Kasai, Kivu, Katanga, Equator and Eastern.*

(15) *Materials can be divided into two basic **categories**: structural and functional.*

7. include

In this subsection we are going to analyze the semantics of the verb *include*. This verb exhibits a curious behavior. Faber and Mairal Usón (1999: 291) consider it a possession verb and define it as "to have something within as a part". The same definition applies also to *contain*. Sentences that comply with this definition are listed below:

(16) *Members of the committee include Mrs Milton Bernet, Mrs J. Clinton Bowman, Mrs Rollie W. Bradford, etc.*

(17) *Miscellaneous soils include sticky substances and colorless liquids.*

(18) *Some new cars include iPod cables in the dash.*

(19) *Industrial constructions include warehouse and factory units.*

In sentence (16) the verb *include* is used with the sense of people belonging to a group. In sentence (17), we find a relation of an object to the materials or substances it is made up of. And finally, in sentences (18) and (19), we identify a

relation between a whole and its parts. However, in the corpora analyzed in this work, we also come across sentences in which the relation existing between the two arguments of the sentence is from a superclass to its subclasses. Consider the sentences below:

(20) *Single-seeded dry fruits used for flavoring include cumin, dill, fennel, and angelica.*

(21) *Products of crude oil refineries include gasoline, diesel fuel, heating oil, kerosene, jet fuel, bunker fuel oil, and liquified petroleum.*

According to sentences (20) and (21), we would either regard the verb *include* as an activity belonging to the general lexical domain of "movement", in which someone places something in a class or group, as in the case of the *classify* verb. Actually, these two senses of the verb *include* are to be found in the Oxford Shorter Dictionary, which reads:

> **include**
> 2. Contain as part of a whole or as a subordinate element (...)
> 3. Place in a class or category; treat or regard as part of a whole (...)

Definition number 3. would be more in line with the definition of the verb *classify into*. As in the case of this latter verb, the result of the movement and placement activities is what is foregrounded in the sentences. For this reason, we would propose the same template for the verb *include* (see table 5.10).

Nevertheless, and contrary to the *classify into* lexical template, the human actor would become a default argument, since it is not syntactically realized in the sentences. In addition to that, the second argument (y) cannot be said to be composed of all the subclasses that make up the superclass, i.e., exhaustiveness is not a feature of the *y* argument defined by the constitutive *quale*. Despite all these differences, this sense of the verb *include* would allow the so-called container subject alternation (Levin, 1993: 82), in which the whole or superclass is expressed in a prepositional phrase, as was the case of *classify into*. This means that the sentences above introduced could be also formulated as:

(22) *Cumin, dill, fennel, and angelica are included into single-seeded dry fruits used for flavoring.*

(23) *Gasoline, diesel fuel, heating oil, kerosene, jet fuel, bunker fuel oil, and liquified petroleum are included into (the class of) products of crude oil refineries.*

However, this alternation would not be possible in the case of sentences conveying the part-whole relation, because the parts, members or substances are part of the wholes, and the latter cannot exist without them. Consider the sentences below[10]:

[10]The asterisk symbol indicates that the sentence is grammatically incorrect.

Lexical Template	
verbal pattern	include
LCM EVENTSTR	[do' (x, move to (other) place' (x, y) & place in (x, y, v)]e_1 e_3 & [BECOME be in (new) place (y) & have types (y, v)]e_2 e_4 $<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4)
GT EVENTSTR:	E_1 = e_1: [process] E_2 = e_2: [state] E_3 = e_3: [process] E_4 = e_4: [state] *Restr* = [$<\infty$ (E_1, E_2), $<\infty$ (E_3, E_4)] *HEAD* = [e_4]
GT ARGSTR:	*D-ARG* = x [human] ARG_2 = y [superclass] ARG_3 = v [subclasses (subclass $\neq$ subclass $\neq$ subclass...)] *D-ARG* = w [criteria]
GT QUALIASTR:	Q_F = [classified (e_2 y), exist in a relation of kind of to (e_4 y, v)] Q_C = be disjoint (e, y) Q_A = [classify_act (e_1, x, y, w), move into new class (e_3 y, v)]

Table 5.10: Lexical template for *include*

(24) **Mrs Milton Bernet, Mrs J. Clinton Bowman, Mrs Rollie W. Bradford are included into members of the committee*.

(25) **Sticky substances and colorless liquids are included into miscellaneous soils*.

(26) **Warehouse and factory units are included into industrial constructions*.

In this respect, we regard the *subclass of* meaning of *include* as being composed of the primary sense of *include*, plus the meaning of the prepositional phrase introduced by *into*, although this is not lexicalized usually. On the contrary, the *part-whole* relation expressed by the verb *include* would rely on the primary sense of *include* as described in table 5.15 for the verb *contain*. In this sense, it can be said that we have moved from a co-compositionally formed sense to its primary sense. Accordingly, *include* in the sense of part-whole relation would only participate in the latter alternation. A complete description of the lexical template for *contain* and related verbs can be seen in table 5.15.

8. belong to

A similar situation as the one described above for the verbs *divide* and *include* occurs in the case of the verbal phrase *belong to*. In its basic meaning, it is defined as "to be the property of a person or thing"[11]. We contend it is a "possession" verb that establishes a relation between a thing and its possessor. However, this relation can be also understood as the relation between a part and the whole, a member and a group, or even a subclass and its superclass. These could be regarded as metaphorical uses of the container-content frame, extrapolated to organisms, groups, classes, etc.

Consider the following examples:

(27) Cape Verde Islands belong to Portugal.

(28) The children belong to the football team.

In those sentences, the meaning of *belong to* has been shifted to "be part or member of a group, organization, etc.". The template describing this meaning is illustrated in table 5.11. The lexical template describes an event structure that involves one state (e_1) and two arguments (x, y). The formal *quale* specifies the nature of the state event stating that there is a relation of part-whole between the arguments. The arguments can be rendered as parts or members (x) staying in a relation of *part of* with the whole or group (y). The LCM EVENTSTR relies on the specification primitive BE and the relational substantive PART.

Lexical Template	
verbal pattern	belong to
LCM EVENTSTR	be part of (x, y)
GT EVENTSTR:	$E_1 = e_1$: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [part / member] ARG_2 = y [whole / group]
GT QUALIASTR:	Q_F = [exist in a relation of part of to (e_1, x, y)]

Table 5.11: Lexical template for *belong to*

A shift in the arguments from generic names of groups or objects to classes or categories would derive in a shift in meaning. However, in the great majority of the examples identified in our corpora, the verb *belong to* conveying a relation of *subclass of* between categories is accompanied by a generic word for classes. See, for instance, the sentences below:

(29) *Thyroid medicines belong to the **general group of** hormone medicines.*

[11] http://www.merriam-webster.com/dictionary/belong

(30) *Detergent actives belong to the chemical **class**.*

This use of a generic word may be due to the *subclass of* relation being a metaphorical use of the primary meaning of the verb *belong to*. The template for this additional sense of *belong to* is illustrated in table 5.12. This template reminds us of the template for the *be a(n)* phrase in table 5.4.

Lexical Template	
verbal pattern	belong to
LCM EVENTSTR	be kind of (x, y)
GT EVENTSTR:	E_1 = e_1: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [subclass] ARG_2 = y [superclass] *D-ARG* = v [criteria]
GT QUALIASTR:	Q_F = [exist in a relation of kind of to (e_1, x, y)]

Table 5.12: Lexical template for *belong to the class of...*

9. have

As in the previous case of *belong to*, *have* is also a possession verb that establishes a relation between a person or object and its parts, properties, entitlements, etc. *Have* is a highly polysemous verb, and as in the case of previous verbs analyzed here, the nature of the arguments will restrict the meaning of the verbal predicate.

Lexical Template	
verbal pattern	have
LCM EVENTSTR	have part(s) (x, y)
GT EVENTSTR:	E_1 = e_1: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [whole] ARG_2 = y [part]
GT QUALIASTR:	Q_F = [exist in a relation of part of to (e_1, x, y)]

Table 5.13: Lexical template for *have (as part)*

In sentences such as *Cars have height-adjustable steering columns*, the relation established between subject and object is a relation of a whole to its parts (see table 5.13). However, in the sentence *Some cars have basic warranties*, we are describing an extrinsic feature of a car contract, and not one of its parts (see table 5.14). Both uses of the verb *have* are very common in the context of descriptive and encyclopedic documents, which are the ones we have taken into account in this research work.

Lexical Template	
verbal pattern	have
LCM EVENTSTR	have property(s) (x, y)
GT EVENTSTR:	E_1 = e_1: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [human, object, animate-individual, artifact, etc.] ARG_2 = y [properties]
GT QUALIASTR:	Q_F = [exist in a relation of kind of to (e_1, x, y)]

Table 5.14: Lexical template for *have (as property)*

10. contain

Finally, we will analyze the verb *contain*. As happened in the case of *include*, Faber and Mairal Usón (1999: 291) consider it a "possession" verb defined as "to have something within as a part". The Oxford Shorter Dictionary provides the following definition:

> **contain** 1. Include as a part or the whole of its substance or content; comprise.

Apart from the recursiveness of the definitions provided for *include* and *contain* in the Oxford Shorter Dictionary, this definition indicates that the verb does not share the sense of *subclass of* conveyed by the active accomplishment in the case of *include*. In table 5.15 we define its event, argument and *qualia* structures.

Lexical Template	
verbal pattern	contain
LCM EVENTSTR	have parts (x, y)
GT EVENTSTR:	E_1 = e_1: [state] *HEAD* = [e_1]
GT ARGSTR:	ARG_1 = x [human, object, animate-individual, artifact, etc.] ARG_2 = y [part, constituent, member, etc.]
GT QUALIASTR:	Q_F = [exist in a relation of part of to (e_1, x, y)]

Table 5.15: Lexical template for *contain*

The event structure involves a state, which is mapped onto the formal *quale*. The formal *quale* describes that there is a meronymy relation between the arguments. Depending on the nature of the arguments, the relation would be further specified as a relation between a whole and its parts, a group and its members, or an object and its constituents (material or substance). The LCM EVENTSTR

makes use of the primitives HAVE and PART to define a relation of possession between the first and the second argument.

Although the relation of meronymy is not a simple one, and could be said to include a family of relations, we will restrict here to three types of relation: part-whole, group-member, and object-constituent. We consider that these three relations embrace the great majority of meronymical relations, which could be considered more specific. Some examples of them are: mass-count, characteristic-activity, step-process, or place-area (see chapter 3). According to Climent Roca (2000)[12], these are the three basic mereologic schemas. It could also be argued that we focus on these three types because they have been adopted in lexical resources such as WordNet.

Other verbs that follow the same template as the one provided for *contain* are: *comprise*, *make up*, *compose*, *constitute*, *be part of*, and *form part of*. Contain, comprise and consist are normally used in the active voice, whereas make up, compose and constitute are more usual in the passive voice. Interestingly enough, in the corpora analyzed we find all these verbs conveying the different types of meronymical relations mentioned above. The examples below show how the same verbal form can convey the different types of meronymy relations (part-whole, group-member, and object-constituent).

(31) Foam machines that produce such stock consist of two or more pumping units. (part-whole)

(32) This board consists of two of the trustees of the college, the director, and two members of the board of freeholders. (group-member)

(33) Crude oil consists of hydrocarbons. (object-constituent)

(34) All organisms are made up of one or more cells. (part-whole)

(35) The Working Group is made up of Administrative Office staff, Federal Defender Organization attorneys, a private criminal defense attorney representative, and Department of Justice representatives. (group-member)

(36) Bones are made up of calcium, phosphorous, sodium, and other minerals, as well as the protein collagen. (object-constituent)

These polysemous uses will have to be disambiguated to be appropriately modeled in the ontology. In the next section we include the repository of English and Spanish LSPs associated to their corresponding ODPs. The ambiguity just mentioned in the case of verbs that convey meronymical relations will be also reproduced in the repository by establishing a relation of *1 LSP to N disjoint ODPs*. This means that since every verb can convey different types of meronymy relations, we will have to find out which relation applies depending on the arguments (are they

[12] Section 4.2 of the on-line version of his PhD work [Accessed in April 2010].

parts of a whole?, members that belong to a group?, or constituents of an object?). Although disambiguation strategies are out of the scope of this work, we suggest some solutions that would need to be further developed (see section 6.4 in chapter 6).

5.4 Multilingual LSPs-ODPs Pattern Repository

In this section, we present the complete *multilingual LSPs-ODPs pattern repository* that we have designed as the core of our approach for knowledge acquisition and ontology modeling intended for novice users. It contains a total amount of 34 LSPs in English and 35 LSPs in Spanish, hence its multilingual nature. There is no doubt that the current repository could be enriched with additional LSPs, and it would also be desirable. With this aim, we have published the English LSPs in the Ontology Design Pattern Portal[13], so that users and researchers in the domain can contribute to enlarge the repository, as will be explained in section 7.2. However, we were not so much interested in the quantity of patterns but in its quality, which justifies the detailed analysis of some verbs on the light of the LCM (section 5.2).

As already introduced in this work, the repository has to be understood as a bridge, as the means to connect NL expressions and ODPs. Whenever an input statement in English or Spanish produced by the user of our method matches an LSP, the ontological structure needed for modeling the semantics of the input will be output in the form of an ODP or a combination of them. However, as has been outlined in section 5.1, the correspondence between LSPs and ODPs is not always direct or 1 to 1 correspondence.

More often than not, an LSP corresponds to a combination of several ODPs, because the same linguistic structure conveys information that has been encoded separately in various ODPs. And there is still a further possibility, namely, the correspondence of one LSP to pairwise disjoint ODPs. This means that the information conveyed by the same linguistic structure can have various modeling possibilities which are not compatible. The complexities in the correspondences between LSPs and ODPs are mainly due to the polysemic uses of some LSPs, as has been already pointed out in section 5.2. Several strategies to solve the ambiguities caused by polysemic LSPs have been devised (see section 6.4).

The different type of correspondences between LSPs and ODPs can be summarized as follows:

1. 1 to 1 correspondence: 1 LSP corresponds to 1 ODP
2. 1 to N correspondence: 1 LSP corresponds to a combination of 2 or more ODPs
3. 1 to *N disjoint* correspondence: 1 LSP corresponds to a set of disjoint ODPs

[13] http://ontologydesignpatterns.org

ODPs Type	Type of correspondence	EN	Table nº	ES	Table nº
1 LSP corresponds to 1 ODP					
Logical ODPs	LSPs for *subclass of* relation ODP	5	T 5.18	7	T 5.35
	LSPs for *multiple inheritance* ODP	1	T 5.19	2	T 5.36
	LSPs for *equivalence relation* ODP	1	T 5.20	1	T 5.37
	LSPs for *object property* ODP	1	T 5.21	1	T 5.38
	LSPs for *datatype property* ODP	2	T 5.22	3	T 5.39
	LSPs for *disjoint classes* ODP	1	T 5.23	1	T 5.40
	LSPs for *specified values* ODP	1	T 5.24	1	T 5.41
Content ODPs	LSPs for *participation* ODP	1	T 5.25	1	T 5.42
	LSPs for *co-participation* ODP	2	T 5.26	2	T 5.43
	LSPs for *location* ODP	1	T 5.27	1	T 5.44
	LSPs for *object-role* ODP	1	T 5.28	1	T 5.45
1 LSP corresponds to a combination of 2 or more ODPs					
Logical ODPs	LSPs for *defined classes* and *subclass of* relation ODPs	4	T 5.29	4	T 5.46
	LSPs for *subclass of* relation, *disjoint classes* and *exhaustive classes* ODPs	4	T 5.30	3	T 5.47
	LSPs for *object property* and *universal restriction* ODPs	1	T 5.31	1	T 5.48
1 LSP corresponds to a set of disjoint ODPs					
Logical and Content ODPs	LSPs for *subclass of* relation and *part whole* relation ODPs	2	T 5.32	1	T 5.49
	LSPs for *object property*, *data type property* or *part whole* relation ODPs	1	T 5.33	1	T 5.50
Content ODPs	LSPs for *part whole* relation, *constituency*, *componency* or *collection-entity* ODPs	5	T 5.34	4	T 5.51
Total number of LSPs		34		35	

Figure 5.5: Summarizing table of LSPs-ODPs correspondences

The *multilingual LSPs-ODPs pattern repository* will be organized in these three groups according to the correspondence type. In the table included in Figure 5.5, we summarize the type and number of LSPs according to the ODPs they correspond to. We have also included the number of LSPs that we have identified for each relation in English (EN) and in Spanish (ES), as well as the number of the table that contains them in the repository. The order presented in this table is the order in which the LSPs will be presented in both the English and the Spanish LSPs-ODPs pattern repository.

We will start by introducing the LSPs for English, and then we will present the Spanish LSPs-ODPs pattern repository. LSPs are considered to be language dependent and, despite some overlapping, they have to be discovered for each new language. In order to describe LSPs in a systematic way, we have designed a template that consists of four slots, as shown in Table 5.16. The information contained in each table refers to:

- **LSPs Identifier.** This mandatory slot contains an acronym composed of:

LSP, plus the acronym of the relation captured by the ODP, plus the ISO-639 code for representing the name of the language for which the LSP is valid.

- **NeOn ODPs Identifier.** This mandatory slot inherits the ODP identifier used in the ODPs repository in M. C. Suárez-Figueroa et al. (2007) and Presutti et al. (2008). If an identifier is not contained in those repositories, the acronym is created according to the rules defined in the following. Identifiers are composed of the component type (e.g. LP standing for Logical Pattern, or CP for Content Pattern), component (e.g. SC standing for SubClassOf), and number of the pattern (01).

- **Formalization.** This mandatory slot includes the various LSPs that express the relation contained in the corresponding ODP or ODPs. LSPs have been formalized according to an extension of the BNF[14] notation (see Table 5.17 for Symbols and Abbreviations created for this purpose).

- **Examples.** This optional slot shows some examples of sentences in NL that match the LSPs in question.

Table 5.16: LSPs-ODPs pattern repository template

LSP Identifier	An acronym composed of LSP + ODP component + ISO code for language
NeOn ODPs Identifier	An acronym composed of component type + component + number
Formalization	LSPs formalized according to BNF extension
Examples	Sentences in NL that exemplify the corresponding LSPs

Following the above described template, we present a total of **34 LSPs for English**, and **35 for Spanish**. The elements represented in the formalized patterns are considered to be necessary for identifying the relation of interest expressed by the pattern. A summary of the main elements used for the notation of the formalized patterns is given below. All elements have been described in table 5.17.

A *Noun Phrase* (NP) is a phrase whose main word is a noun or a pronoun, and that is optionally accompanied by a set of modifiers, as for example, determiners, adjectives, etc. NPs represent the arguments of a predicate, which in ontologies usually correspond to classes, properties or individuals. The semantic role attached to each NP in the patterns has been made explicit in angle brackets $\langle ... \rangle$. It is important to note that in the LSPs included in our repository, we only distinguish between classes and properties, but not between classes and individuals. Indeed, sometimes the same pattern can convey a relation between classes, and also be-

[14] BNF stands for Backus-Naur Form and is a programming language that relies on well-defined symbols and unambiguous syntactic rules. See (*ISO/IEC 14977:1996 - Information technology - Syntactic metalanguage - Extended BNF*, 1996).

tween classes and instances. In the latter case, we are in favor of relying on NL processing tools to find out if the arguments of the sentence are referring to classes or to instances. However, this is out of the scope of this work.

Verbs expressing the conceptual relation in question are represented by its lemma or base form. The elements represented in the pattern are the ones considered to be necessary for the pattern to express a certain relation. Optional elements, i.e. the ones that may appear or not without modifying the basic meaning of the pattern, have been indicated by the use of [...].

In table 5.17 we include the complete set of symbols and abbreviations created *ad hoc* for the formalization of the LSPs presented in this work. Any additional element appearing in the sentence but not captured in the LSP should be in principle ignored, because it does neither provide any information nor affect or modify the semantics of the sentence in NL.

Table 5.17: LSPs Symbols and Abbreviations

SYMBOLS & ABBREVIATIONS	DESCRIPTION
AP$\langle ... \rangle$	Adjectival Phrase. It is defined as a phrase whose head is an adjective accompanied optionally by adverbs or other complements as prepositional phrases. AP is followed by the semantic role played by the concept it represents in the conceptual relation (for instance, property) in angle brackets.
CATV	Verbs of Classification. Set of verbs of classification plus the preposition that normally follows them. Some of the most representative verbs in this group are: classify in/into, categorize in/into, group in/into, or fall into in English; and *clasificar en* or *agrupar en* in Spanish.
CD	Cardinal Number
CN	Class Name. Generic names for semantic roles usually accompanied by preposition. Two main groups have been identified: CN conveying classification (CN-CATV) (class, group, type, subtype, subclass, category, species, family, order, example in English; or *clase, tipo, grupo, subtipo, subclase, categoría, especie, familia, orden* in Spanish) and CN conveying mereological relations (CN-PART) (part, set, member, constituent, component, element, piece, item, layer in English; or *parte, miembro, componente, elemento, pieza, segmento, porción, pedazo, trozo, fragmento* in Spanish). If not otherwise specified, CN can include generic names such as period, area or phase.

Table 5.17: LSPs Symbols and Abbreviations (continued)

PART	Verbs of Mereology. Set of verbs conveying the relation existing between a whole and its parts. Some of the most representative ones in English are: contain, form part of, consist of, comprise, be composed of, be made up of, be formed of, be part of, be constituted of, belong to. In Spanish we have identified: *formar*, *integrar*, *constituir*, *ser parte de*, *formar parte de*, *comprender*, *componerse de*, or *descomponerse en*.
NP⟨...⟩	Noun Phrase. It is defined as a phrase whose head is a noun or a pronoun, optionally accompanied by a set of modifiers, and that functions as the subject or object of a verb. NP is followed by the semantic role played by the concept it represents in the conceptual relation in question in ⟨...⟩, e.g., class, subclass, part, property, value, object, etc.
PARA	Paralinguistic symbols like colon, or more complex structures such as, as follows, *como por ejemplo*, *a saber*, *entre otros*, etc., that introduce a list.
PREP	Prepositions
QUAN	Quantifiers such as all, some, most, many, several, every, etc., in English, or *algún*, *varios*, etc., in Spanish.
REPRO	Relative pronouns such as that, which, whose, etc., in English, or *que*, *cuyo*, etc., in Spanish.
(...)	Parentheses group two or more elements.
(*)	Asterisk indicates repetition.
[...]	Elements in brackets are meant to be optional, which means that they can be present either at that stage of the sentence or not. By default of appearance, the semantics of the pattern remains unmodified.
NEG	Negative. Elements preceded by this abbreviation should not appear in the pattern.

5.4.1 English LSPs-ODPs Pattern Repository

With no claim of being exhaustive, in the following we present a set of English patterns that correspond to ODPs and that have been formalized according to the templates introduced in the previous section. The final aim of this task is to facilitate their subsequent implementation in the NL processing tool GATE, as will be explained in section 7.1.

The English and Spanish repositories have been organized according to the types of correspondences between LSPs and ODPs. As already explained, the symbols used in the formalization follow an extension of the BNF notation, in which additional *ad hoc* symbols have been created with the aim of systematizing and simplifying the notation. The whole set of English LSPs-ODPs templates and their corresponding code in JAPE rules has been made available for their reuse in NLP tools through the Ontology Design Patterns Portal (for more on this see section 6.2).

1 to 1 correspondence: 1 LSP corresponds to 1 ODP

In this section we present those LSPs in English that have a direct correspondence with one ODP. As will be explained in the following, the patterns that belong to this type are

(1) LSPs corresponding to *subclass-of relation* ODP (table 5.18)

(2) LSPs corresponding to *multiple inheritance* ODP (table 5.19)

(3) LSPs corresponding to *equivalence relation between classes* ODP (table 5.20)

(4) LSPs corresponding to *object property* ODP (table 5.21)

(5) LSPs corresponding to *datatype property* ODP (table 5.22)

(6) LSPs corresponding to *disjoint classes* ODP (table 5.23)

(7) LSPs corresponding to *specified values* ODP (table 5.24)

(8) LSPs corresponding to *participation* ODP (table 5.25)

(9) LSPs corresponding to *co-participation* ODP (table 5.26)

(10) LSPs corresponding to *location* ODP (table 5.27)

(11) LSPs corresponding to *object-role* ODP (table 5.28)

(1) The LSPs represented in table 5.18 correspond to the *subclass-of relation* ODP. These patterns can be said to straightforwardly convey a relation of subclass-of between the arguments of the predicates. Neither disjointness nor exhaustiveness can be assured when constructions like these are expressed by the user. However, such characteristics of the subclass-of relation should be specified in the ontology. This means that additional strategies will have to be drawn up to obtain that information from the user when following the method (for more on this see section 6.4 in chapter 6).

Table 5.18: LSPs corresponding to *subclass-of relation* ODP

LSP Identifier: LSP-SC-EN	
NeOn ODPs Identifier: LP-SC-01	
Formalization	
1	[(NP⟨$subclass$⟩,)* and] NP⟨$subclass$⟩ be [CN-CATV] NP⟨$superclass$⟩
2	[(NP⟨$subclass$⟩,)* and] NP⟨$subclass$⟩ classify as NP⟨$superclass$⟩
3	[(NP⟨$subclass$⟩,)* and] NP⟨$subclass$⟩ (belong to) | (fall into) CN-CATV NP⟨$superclass$⟩
4	There are QUAN CN-CATV NP ⟨$superclass$⟩ PARA [(NP ⟨$subclass$⟩,)* and] NP ⟨$subclass$⟩
5	[A(n) | QUAN] example of | CN-CATV NP⟨$superclass$⟩ be | include [PARA] [(NP⟨$subclass$⟩,)* and] NP⟨$subclass$⟩
Examples	
1	*An orphan drug is a type of drug.* *Odometry, speedometry and GPS are types of sensors.*
2	*Prefixes and suffixes are classified as affixes.*
3	*Thyroid medicines belong to the general group of hormone medicines.* *Starfish fall into the class Asteroidea.*
4	*There are several kinds of memory: fast, expensive, short term memory, and long-term memory.*
5	*Some examples of peripherals are keyboards, mice, monitors, printers, scanners, disk and tape drives, microphones, speakers, joysticks, plotters and cameras.* *Types of criteria for assessing applications are: quality, safety and efficacy.*

(2) The pattern formalized in table 5.19 for the *multiple inheritance* ODP reminds us of some of the LSPs introduced in table 5.18 for the *subclass-of relation*. In fact, the only difference is that the elements of a class correspond now to two different super-classes, what is termed in ontology engineering as *multiple inheritance*.

Table 5.19: LSPs corresponding to *multiple inheritance* ODP

LSP Identifier: LSP-MI-EN	
NeOn ODPs Identifier: LP-MI-01	
Formalization	
1	NP$\langle subclass \rangle$ be | be classify as NP$\langle superclass \rangle$ and NP$\langle superclass \rangle$
Examples	
1	*Amphibians are water-living and land-living animals.*

The LSP in table 5.20 conveys the relation existing between two sets or groups that have different names.

(3) Let us imagine two ontologies modeling the same domain of knowledge that are to be merged. In one of them the class defining the group of frogs from the *Dendrobatidae* family that is native to Central and South America has the name of *poison dart frogs*, whereas in the other ontology the same class has been termed *poison-arrow frogs*. Then, ontology engineers may want to establish a relation of *equivalence* between both classes. We believe that the way we have to express this equivalence in language is captured by verbal phrases such as *know as*, *call*, or *refer to as*.

Table 5.20: LSPs corresponding to *equivalence relation between classes* ODP

LSP Identifier: LSP-EQ-EN	
NeOn ODPs Identifier: LP-EQ-01	
Formalization	
1	NP $\langle class \rangle$ be (also | likewise) | know as | call | (refer to as) NP $\langle class \rangle$
Examples	
1	*Poison dart frogs are also known as poison-arrow frogs.*

(4) In table 5.21, we represent a relation between objects that belong to different classes, i.e., and *object property* relation. This relation could be directly established if no other relation from the ones identified in the set of Logical and Content ODPs has been previously identified. This is represented by the abbreviation NEG before the parenthesis including the verbs *be*, *have*, verbs of classification and partonomy (The rest of verbs identified in the Content ODPs for *location*, *participation* and *co-participation* should also be included in the parenthesis. Although this has been taken into account in the implementation, it is not represented here for the sake of conciseness).

Table 5.21: LSPs corresponding to *object property* ODP

LSP Identifier: LSP-OP-EN	
NeOn ODPs Identifier: LP-OP-01	
Formalization	
1	NP$\langle class \rangle$ VB NEG (be | have | CATV | PART) NP$\langle class \rangle$
Examples	
1	*Birds build nests.*

(5) When the relation is to be established between the elements that belong to a class and the properties or attributes that define this class, we make use of the *datatype property relation*, see table 5.22.

Properties are normally defined by elements of the type literals or values (boolean values) in an ontology. Let us take example number 2 of this LSP, *Metals are lustrous, malleable and good conductors of heat and electricity*. The characteristics of metals expressed by means of adjectives tell us that for the properties *lustrousness*, *malleability* and *conductivity*, the boolean value would be set to true for metals. This means that the adjectives listed in the LSP-DP-EN are not the properties *per se* but the values. This has to be taken into account when processing this type of sentences.

Table 5.22: LSPs corresponding to *datatype property* ODP

LSP Identifier:LSP-DP-EN	
NeOn ODPs Identifier:LP-DP-01	
Formalization	
1	Property | characteristic | attribute of NP$\langle class \rangle$ be [PARA] [(NP$\langle property \rangle$,)* and] NP$\langle property \rangle$
2	NP$\langle class \rangle$ be [(AP$\langle property \rangle$,)*] and AP$\langle property \rangle$
Examples	
1	*Properties of mammals are hair, sweat glands, milk, and giving live birth.*
2	*Metals are lustrous, malleable and good conductors of heat and electricity.*

(6) In table 5.23 we identify one LSP that allows us to directly identify that two classes cannot share instances between them, i.e., a relation between *disjoint classes*. Disjointness is a relation that has to be made explicit when modeling classifications in ontologies. This avoids errors in the ontology reasoning process. As we will see in table 5.30, some LSPs expressing subclass-of relation can also be asserted to determine disjointness.

Table 5.23: LSPs corresponding to *disjoint classes* ODP

LSP Identifier: LSP-Di-EN	
NeOn ODPs Identifier: LP-Di-01	
Formalization	
1	NP$\langle class \rangle$ differ | be different | be differentiate from NP$\langle class \rangle$
Examples	
1	*Non-opioid agents differ from opioid agents.* *Universal aspects of language are differentiated from language-specific ones.*

(7) The next pattern included in this section is the LSP corresponding to *specified values* ODP (see table 5.24). According to the definition of specified values given in section 5.1, such a formulation conveys the relation between a class and a set of descriptive values that are different among them. This optionality is given in the language by modal verbs followed by the verb *be* and a set of adjective phrases separated by the conjunction *or*, as can be seen in both examples.

Table 5.24: LSPs corresponding to *specified values* ODP

LSP Identifier: LSP-SV-EN	
NeOn ODPs Identifier: LP-SV-01	
Formalization	
1	NP$\langle feature \rangle$ can | may be [(AP$\langle value \rangle$,)*] or AP$\langle value \rangle$
Examples	
1	*Size may be small, medium, or big.* *Business plans can be accepted, non-accepted, or in process of revision.*

The LSPs represented in tables 5.25 and 5.26 could be considered specifications of the basic *object property* relation that one could expect in certain domains of knowledge.

(8) The *participation* ODP represents the participation of an object in an event.

Table 5.25: LSPs corresponding to *participation* ODP

LSP Identifier: LSP-PA-EN	
NeOn ODPs Identifier: CP-PA-01	
Formalization	
1	NP$\langle object \rangle$ participate | take part in | be involved in (NP$\langle event \rangle$,)* and] NP$\langle event \rangle$

Table 5.25: LSPs corresponding to *participation* ODP (follow-up)

	Examples
1	*Engineering project managers participate in writing specifications, researching, and selecting suppliers and materials.* *Players are involved in competitions.*

(9) The *co-participation* ODP represents that two objects participate in the same event.

Table 5.26: LSPs corresponding to *co-participation* ODP

LSP Identifier: LSP-CPA-EN	
NeOn ODPs Identifier: CP-CPA-01	
Formalization	
1	(NP$\langle object \rangle$,)* and NP$\langle object \rangle$ participate | (take part) | (be involved) in [(NP$\langle event \rangle$,)* and] NP$\langle event \rangle$
2	NP$\langle object \rangle$ participate | (take part) | (be involved) with NP$\langle object \rangle$ in [(NP$\langle event \rangle$,)* and] NP$\langle event \rangle$
Examples	
1	*Aldo Gangemi and Valentina Presutti participate in the ISWC 2007 conference.*
2	*Action Engine participates with Microsoft at 3GSM World Congress.*

(10) The next LSP corresponds to the *location* ODP and has been included in table 5.27. As in the case of the LSPs for *participation* and *co-participation*, the LSP for location can be considered a specification of the basic *object property* relation that one could expect in certain domains of knowledge. This pattern refers to a generic, relative localization, holding between any entities.

Table 5.27: LSPs corresponding to *location* ODP

LSP Identifier: LSP-LO-EN	
NeOn ODPs Identifier: CP-LO-01	
Formalization	
1	NP$\langle place \rangle$ be | has (locate | find | set | situate | place | (a site)) in [(NP$\langle location \rangle$,)* and] NP$\langle location \rangle$
Examples	
1	*T-cadherin is located in the nucleus and in the centrosomes.*

(11) The *object-role* ODP is introduced in table 5.28. This relation allows us to

model that a class of elements is used with a certain function o plays a certain role. In the Ontology Design Patterns portal, we find two further specifications of the *object-role* pattern, namely, *participant-role* and *agent-role*. These have not been considered in our work.

Table 5.28: LSPs corresponding to *object-role* ODP

LSP Identifier: LSP-OR-EN	
NeOn ODPs Identifier: CP-OR-01	
Formalization	
1	NP$\langle object \rangle$ (be used) | work | act | serve as [(NP$\langle role \rangle$,)* and] NP$\langle role \rangle$
Examples	
1	*Gold is used as the reflective layer on some high-end CDs* *Induced bronchus-associated lymphoid tissue serves as a general priming site for T cells.*

1 to N correspondence: 1 LSP corresponds to a combination of 2 or more ODPs

Next, we include the templates for those linguistic structures that do not find a one to one correspondence with an ODP, but which represent a more complex meaning structure that requires from several ODPs to be appropriately modeled in the ontology. As will be explained below, the patterns that belong to this type are the following:

(12) LSPs corresponding to *defined classes* and *subclass-of relation* ODPs (table 5.29)

(13) LSPs corresponding to *subclass-of relation*, *disjoint classes*, and *exhaustive classes* ODPs (table 5.30)

(14) LSPs corresponding to *object property* and *universal restriction* ODPs (table 5.31)

(12) Under this category, we find patterns like the one represented in table 5.29, the LSPs corresponding to *defined classes* and *subclass-of relation*. When defining a class by referring to its superclass and additionally mentioning the property or attribute that makes it different from the superclass (and its sibling classes), we are expressing a *subclass-of* relation, and are also making explicit those properties that make the class *defined*. This combination of Logical ODPs is represented in table 5.29. *Defined* classes stay in opposition to *primitive* classes. When defining

primitive classes, we assert the set of *necessary* conditions that a class must satisfy. However, when describing *defined* classes, both *necessary* and *sufficient* conditions have to be expressed.

Table 5.29: LSPs corresponding to *defined classes* and *subclass-of relation* ODPs

LSP Identifier : LSP-DC-SC-EN	
NeOn ODPs Identifier : LP-DC-01 + LP-SC-01	
Formalization	
1	[A | any] NP⟨$subclass$⟩ be [a | any] NP⟨$superclass$⟩ REPRO VB [(NP⟨$class$⟩,)* and] NP⟨$class$⟩
2	[A | any] NP⟨$subclass$⟩ be [a | any] NP⟨$superclass$⟩ PREP [(NP⟨$class$⟩,)* and | or NP⟨$class$⟩
3	[A | any] NP⟨$subclass$⟩ REPRO VB [(NP⟨$class$⟩,)* and | or NP⟨$class$⟩ be [a] NP⟨$superclass$⟩
4	[A | any] NP⟨$subclass$⟩ PREP [(NP⟨$class$⟩,)* and] NP⟨$class$⟩ be VB [a] NP⟨$superclass$⟩
Examples	
1	*A device is any machine or component that attaches to a computer. Non-narcotic analgesics are drugs that have principally analgesic, antipyretic, and anti-inflammatory actions.*
2	*A vegetarian pizza is a pizza without fish or meat.*
3	*A workflow that contains at least one business task is a business plan.*
4	*Animal with backbones are called vertebrates.*

(13) A further combination of Logical ODPs is represented by the set of LSPs contained in table 5.30 for *subclass-of relation*, *disjoint classes* and *exhaustive classes* ODPs. The identification of these linguistic patterns that require the combination of the *subclass-of* relation with the two further characteristics of this relation, namely, disjointness and exhaustiveness, can greatly benefit the modeling of ontologies by non-experts. Good practices in ontology engineering recommend the further specification of these two characteristics whenever a subclass-of relation is modeled in ontologies.

In the patterns captured below, we argue that these three aspects of knowledge are conveyed in the same sentence. Even so, due to the importance of these characteristics, we believe that users should be asked for confirmation so that they are also aware of the implications that this type of modeling can have in the ontology. In case they did not meant to make those strong statements about disjointness and

exhaustiveness, they are given the option to correct or complete the statement. This is further explained in section 6.4, chapter 6.

Table 5.30: LSPs corresponding to *subclass-of relation*, *disjoint classes* and *exhaustive classes* ODPs

LSP Identifier : LSP-SC-Di-EC-EN	
NeOn ODPs Identifier : LP-SC-01 + LP-Di-01 + LP-EC-01	
Formalization	
1	NP⟨$superclass$⟩ be | CATV [either] NP⟨$subclass$⟩ or | and NP⟨$subclass$⟩
2	NP⟨$superclass$⟩ CATV CD CN-CATV [PARA] [(NP⟨$subclass$⟩,)*and] NP⟨$subclass$⟩
3	There are CD CN-CATV NP⟨$superclass$⟩ [PARA] [(NP⟨$subclass$⟩,)* and] NP⟨$subclass$⟩
4	NP⟨$superclass$⟩ be divided | separate in|into CD CN-CATV [PARA] [(NP⟨$subclass$⟩,)* and] NP⟨$subclass$⟩
Examples	
1	*Animals are either vertebrates or invertebrates.*
2	*Membrane proteins are classified into two categories, integral proteins and peripheral proteins.* *Flat roofing materials fall into three categories: built-up felt roofing, mastic asphalt and single-ply membranes.* *Malignant mixed tumors are grouped into 3 categories: carcinoma ex pleomorphic adenoma, true malignant mixed tumor (carcinosarcoma), and metastasizing mixed tumor.*
3	*There are two types of narcotic analgesics: the opiates and the opioids.*
4	*Marine mammals are divided into three orders: Carnivora, Sirenia and Cetacea.*

(14) Finally, we will refer to the LSPs corresponding to *object property* and *universal restriction* (see table 5.31). These patterns model a relation that can only be established between members of two groups or classes. Let us illustrate this by means of the example provided in the template.

The combination of these two Logical ODPs means the set of elements that belong to the class of *Herbivores* can only be in a relation of *eat* to the class of *Plants*, i.e., the relation *eat* cannot be established to any other class. In this pattern, it is not exactly the verb that gives us the possibility of asserting the relations that exist between classes, but the adverb (just, only , exclusively) that modifies the verb.

Table 5.31: LSPs corresponding to *object property* and *universal restriction* ODPs

LSP Identifier : LSP-OP-UR-EN	
NeOn ODPs Identifier : LP-OP-01 + LP-UR-01	
Formalization	
1	NP$\langle class \rangle$ VB NEG(be | have | CATV | PART) just | only | exclusively NP$\langle class \rangle$
Examples	
1	*Herbivore eat only plants.*

1 to *N disjoint* correspondence: 1 LSP corresponds to a set of disjoint ODPs

The third and last type of LSPs collected in this repository embraces those linguistic patterns that are considered polysemous, i.e., that correspond to two or more ODPs that represent incompatible modeling options. Most of the verbs and verbal phrases contained in these LSPs have been analyzed in section 5.2 on the light of the Lexical Constructional Model, so that their semantics could be determined. Here we are referring to those LSPs that convey the following relations:

(15) LSPs corresponding to *subclass-of* or *part-whole* relations ODPs (table 5.32)

(16) LSPs corresponding to *object property*, *datatype property* or *simple part-whole relation* ODPs (table 5.33)

(17) LSPs corresponding to *simple part-whole relation*, *constituency*, *componency* or *collection-entity* ODPs (table 5.34)

(15) As already mentioned, the linguistic structures included in table 5.32 convey two different types of ontological relations *subclass-of* and *part-whole*. Only the nature of the arguments can help us disambiguate. Some of the strategies devised for solving this ambiguity problem in the framework of the method proposed in this PhD work have been included in chapter 6, section 6.4.

Table 5.32: LSPs corresponding to *subclass-of relation*, or *simple part-whole relation* ODPs

LSP Identifier : LSP-SC-PW-EN	
NeOn ODPs Identifier : LP-SC-01 CP-PW-01	
Formalization	
1	NP⟨$class$⟩ include [(NP⟨$class$⟩,)* and] NP⟨$class$⟩
2	NP⟨$class$⟩ be divided | separate in|into [CN] [(NP⟨$class$⟩,)* and] NP⟨$class$⟩
Examples	
1	*Arthropods include insects, crustaceans, spiders, scorpions, and centipedes. (LP-SC-01)* *Reproductive structures in female insects include ovaries, bursa copulatrix and uterus. (CP-PW-01)*
2	*Seed producing plants are divided into angiosperms and gymnosperms. (LP-SC-01)* *Cells are divided into distinct sub-cellular compartments. (CP-PW-01)*

(16) In the LSPs included in table 5.33, the ambiguity problem is, if possible, more complex, because the nature of the arguments can be very similar, but a modeling decision has to be taken, considering the rest of classes represented in the ontology. Let us take the example *Birds have feathers* to illustrate this issue.

- **1st case.** The user is creating an ontology about animals, and may be interested in identifying those *parts* of birds that make them different from other animals.
- **2nd case.** The user is creating an ontology about birds, and may want to classify them according to the color of their feathers. In that case, a more proper modeling solution may be represented by the *object property* relation, because further characteristics of feathers can be asserted.
- **3rd case.** The user may only want to define feathers as a *property* of birds, because no further information about feathers is required.

The ontological context will be decisive in such a modeling issue. We believe that the user has to be made aware of such a problem and take part in reaching a solution. Further strategies would need to be investigated for this specific pattern. In the example of *Water areas have names in natural language*, the discussion would be between modeling it as an object property or as a datatype property, depending on the information that the user would need to include about names.

Table 5.33: LSPs corresponding to *object property* or *datatype property* or *simple part-whole relation* ODPs

LSP Identifier : LSP-OP-DP-PW-EN	
NeOn ODPs Identifier : LP-OP-01 LP-DP-01 CP-PW-01	
Formalization	
1	NP$\langle class \rangle$ have NP$\langle class \rangle$
Examples	
1	*Birds have feathers.* *Water areas have names in natural language.*

(17) The last set of LSPs which corresponds to a set of disjoint ODPs is the one that captures meronymic relations: *part-whole relation*, *constituency*, *componency* or *collection-entity* ODPs, as shown in table 5.34. The semantic role attached to the NPs in the LSPs only refers to *wholes* and *parts* for being considered the most general arguments.

The way in which the various specializations of the part-whole relation are captured by the different types of Content ODPs has been described in section 5.1:

- *simple part-whole relation* is a transitive relation between objects and their parts
- *constituency* represents the constituents of a layered structure, including material or substance
- *componency* is a non-transitive relation between objects and their proper parts
- *collection-entity* models members and groups

By identifying these LSPs and careful analyzing their semantics in section 5.2, we will be able to make the user aware of the polysemous uses of these patterns and the options (s)he has for modeling that content. Depending on the type of arguments involved in the sentences, one ODP will be more appropriate than the other. We believe that for taking a final choice, interaction of the user with the system will be needed. For more on this see section 6.4 in chapter 6.

Table 5.34: LSPs corresponding to *simple part-whole relation* or *constituency* or *componency* or *collection-entity* ODPs

LSP Identifier : LSP-PW-CONS-COM-CE-EN	
NeOn ODPs Identifier : **CP-PW-01** **CP-CONS-01** **CP-COM-01** **CP-CE-01**	
Formalization	
1	[(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$ PART NP$\langle whole \rangle$
2	NP$\langle whole \rangle$ PART [(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$
3	NP$\langle whole \rangle$ be PART [CD] CN-PART [PARA] [(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$
4	CN-PART NP$\langle whole \rangle$ be [PARA] [(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$
5	NP$\langle whole \rangle$ include | (be divide in|into) | (be separate in|into) CD CN-PART [PARA] [(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$
Examples	
1	*Proteins form part of the cell membrane.*
2	*Lysosomes contain enzymes.* *Most clays consist of flat particles.* *Water is made up of hydrogen and oxygen.* *The United Arab Emirates is a country composed of seven emirates or sheikdoms.*
3	*A state machine workflow is made up of a set of states, transitions, and actions.*
4	*The parts of a tree are the root, trunk(s), branches, twigs and leaves.*
5	*The cerebrum is divided in two parts: the right cerebral hemisphere and left cerebral hemisphere.*

The same comments made for the LSPs in English can also be extrapolated for the LSPs in Spanish. For this reason, in section 5.4.2 we only include the Spanish repository without comments.

5.4.2 Spanish LSPs-ODPs Pattern Repository

1 to 1 correspondence: 1 LSP corresponds to 1 ODP

In this section we present those LSPs in Spanish that have a direct correspondence with one ODP. The patterns that belong to this type are the following:

(1) LSPs corresponding to *subclass-of relation* ODP (table 5.35)

(2) LSPs corresponding to *multiple inheritance* ODP (table 5.36)

(3) LSPs corresponding to *equivalence relation between classes* ODP (table 5.37)

(4) LSPs corresponding to *object property* ODP (table 5.38)

(5) LSPs corresponding to *datatype property* ODP (table 5.39)

(6) LSPs corresponding to *disjoint classes* ODP (table 5.40)

(7) LSPs corresponding to *specified values* ODP (table 5.41)

(8) LSPs corresponding to *participation* ODP (table 5.25)

(9) LSPs corresponding to *co-participation* ODP (table 5.43)

(10) LSPs corresponding to *location* ODP (table 5.44)

(11) LSPs corresponding to *object-role* ODP (table 5.45)

Table 5.35: LSPs corresponding to *subclassOf relation* ODP

LSP Identifier : LSP-SC-ES	
NeOn ODPs Identifier : LP-SC-01	
Formalization	
1	NP$\langle subclass \rangle$ ser un|una [CN-CATV] NP$\langle superclass \rangle$
2	NP$\langle subclass \rangle$ clasificarse como NP$\langle superclass \rangle$
3	NP$\langle subclass \rangle$ clasificarse dentro de [CN] NP$\langle superclass \rangle$
4	[(NP$\langle superclass \rangle$,)* and] NP$\langle superclass \rangle$ (pertenecer a) CN-CATV de NP$\langle superclass \rangle$

Table 5.35: LSPs corresponding to *subclassOf relation* ODP (follow-up)

5	Hay QUAN CN-CATV de NP $\langle superclass \rangle$ PARA | por ejemplo|entre otros/as [(NP $\langle subclass \rangle$,)* y] NP $\langle subclass \rangle$
6	[Un | QUAN] ejemplo[s] de | CN-CATV NP$\langle superclass \rangle$ ser [PARA] [(NP$\langle subclass \rangle$,)* y] NP$\langle subclass \rangle$
7	Entre los|las NP$\langle superclass \rangle$ figurar los|las [(NP$\langle subclass \rangle$,)* y] NP$\langle subclass \rangle$
Examples	
1	*El dos es un número par.*
2	*La pimienta común (Piper nigrum) se clasifica como perteneciente al género Piper.*
3	*Esta grave enfermedad neurodegenerativa se clasifica dentro del grupo de las enfermedades hereditarias recesivas.*
4	*Los primates y los cetáceos pertenecen a la clase de los mamíferos.*
5	*Hay varios tipos de oraciones predicativas como, por ejemplo, las transitivas y las intransitivas.*
6	*Ejemplos de artrópodos son los crustáceos, los insectos y los arácnidos.*
7	*Entre los animales en peligro de extinción figuran el cari cari, los zamuros, los caimanes y los garzones.*

Table 5.36: LSPs corresponding to *multiple inheritance* ODP

LSP Identifier: LSP-MI-ES	
NeOn ODPs Identifier: LP-MI-01	
Formalization	
1	NP$\langle subclass \rangle$ ser NP$\langle superclass \rangle$ y NP$\langle superclass \rangle$
2	NP$\langle subclass \rangle$ (pertenecer a) | (classificarse como) [CD-CN] NP$\langle superclass \rangle$ y NP$\langle superclass \rangle$
Examples	
1	*Los anfibios son acuáticos y terrestres.*
2	*Las bases de ADN pertenecen a dos familias llamadas purinas y pirimidinas.*

Table 5.37: LSPs corresponding to *equivalence relation between classes* ODP

LSP Identifier: LSP-EQ-ES	
NeOn ODPs Identifier: LP-EQ-01	
Formalization	
1	NP ⟨$class$⟩ [también] (ser conocidos/as como) (denominarse|llamarse|conocerse) [también] [como] NP ⟨$class$⟩
Examples	
1	*Los ordenadores portátiles también son conocidos como portátiles o notebooks.* *Los Óxidos Metálicos se denominan también Óxidos Básicos.*

Table 5.38: LSPs corresponding to *object property* OP

LSP Identifier: LSP-OP-ES	
NeOn ODPs Identifier: LP-OP-01	
Formalization	
1	NP⟨$class$⟩ VB NEG (ser | tener | CATV | PART) NP⟨$class$⟩
Examples	
1	*Los sensores mandan señales*

Table 5.39: LSPs corresponding to *datatype property* ODP

LSP Identifier:LSP-DP-ES	
NeOn ODPs Identifier:LP-DP-01	
Formalization	
1	Las propiedades | características de los/las NP⟨$class$⟩ ser [PARA] [(NP⟨$property$⟩,)* y] NP⟨$property$⟩
2	Los/las NP⟨$class$⟩ caracterizarse por [ser | su/s | el/la/los/las] [(AP⟨$property$⟩,)*] y AP⟨$property$⟩
3	NP⟨$class$⟩ ser [(AP⟨$property$⟩,)*] y AP⟨$property$⟩
Examples	
1	*Las propiedades de los minerales son el color, el brillo, la densidad y la dureza.*

Table 5.39: LSPs corresponding to *datatype property* ODP (follow-up)

2	*Las Aplicaciones se caracterizan por su especialización, flexibilidad, rapidez, exactitud y comodidad de trabajo.* *Estos bosques se caracterizan por ser heterogéneos y frágiles.*
3	*Las células son microscópicas.*

Table 5.40: LSPs corresponding to *disjoint classes* ODP

LSP Identifier: LSP-Di-ES	
NeOn ODPs Identifier: LP-Di-01	
Formalization	
1	NP⟨$class$⟩ diferir | diferenciarse | distinguirse de | NP⟨$class$⟩ [en|por NP⟨$property$⟩] | [en REPRO VP NP⟨$property$⟩]
Examples	
1	*Los Onychopliora difieren de los artrópodos en la ausencia de apéndices segmentados.* *Los ambientes de agua dulce difieren de los marinos por la menor salinidad y la mayor influencia del clima.*

Table 5.41: LSPs corresponding to *specified values* ODP

LSP Identifier: LSP-SV-ES	
NeOn ODPs Identifier: LP-SV-01	
Formalization	
1	NP⟨$feature$⟩ poder ser [(AP⟨$value$⟩,)*] or AP⟨$value$⟩
Examples	
1	*Las valoraciones pueden ser positivas, negativas o neutras.*

Table 5.42: LSPs corresponding to *participation* ODP

LSP Identifier: LSP-PA-ES	
NeOn ODPs Identifier: CP-PA-01	
Formalization	
1	NP⟨$object$⟩ participar en (NP⟨$event$⟩,)* y] NP⟨$event$⟩

Table 5.42: LSPs corresponding to *participation* ODP (follow-up)

	Examples
1	*Científicos españoles participan en proyectos europeos.* *Empresas del Grupo OHL participan en Fundaciones de Protección Ambiental.*

Table 5.43: LSPs corresponding to *co-participation* ODP

LSP Identifier: LSP-PCP-ES	
NeOn ODPs Identifier: CP-PCP-01	
	Formalization
1	(NP$\langle object \rangle$,)* y NP$\langle object \rangle$ participar en [(NP$\langle event \rangle$,)* and] NP$\langle event \rangle$
2	NP$\langle object \rangle$ participar [junto] con NP$\langle object \rangle$ en [(NP$\langle event \rangle$,)* y] NP$\langle event \rangle$
	Examples
1	*Actores y directores participan en un festival de cine.*
2	*La Cámara participa junto con ocho empresas de la piedra natural en la feria Xiamen Stone de China.*

Table 5.44: LSPs corresponding to *location* ODP

LSP Identifier: LSP-LO-ES	
NeOn ODPs Identifier: CP-LO-01	
	Formalization
1	NP$\langle place \rangle$ estar | (estar localizado/a) | (estar situado/a) | encontrarse en [(NP$\langle location \rangle$,)* y] NP$\langle location \rangle$
	Examples
1	*La Reserva Nacional Pacaya-Samiria esta localizada en la región Amaz óónica del Perú.* *La escuela se encuentra en Madrid.*

Table 5.45: LSPs corresponding to *object-role* ODP

LSP Identifier: LSP-OR-ES
NeOn ODPs Identifier: CP-OR-01
Formalization

Table 5.45: LSPs corresponding to *object-role* ODP (follow-up)

1	NP⟨$object$⟩ (utilizarse|usarse para|de) | (servir de|para) | (actuar de) | (valer para) [(NP⟨$role$⟩,)* y] NP⟨$role$⟩
Examples	
1	*Los rascacielos manglares se utilizan para desalinar el agua del mar.* *Los antibióticos sirven para tratar las bacterias.*

1 to N correspondence: 1 LSP corresponds to a combination of 2 or more ODPs

Next we include the templates for those linguistic structures in Spanish that do not find a one to one correspondence with an ODP, but which represent a more complex meaning structure that requires from several ODPs to be appropriately modeled in the ontology. The patterns that belong to this type are the following:

(12) LSPs corresponding to *defined classes* and *subclass-of relation* ODPs (table 5.46)

(13) LSPs corresponding to *subclass-of relation*, *disjoint classes*, and *exhaustive classes* ODPs (table 5.47)

(14) LSPs corresponding to *object property* and *universal restriction* ODPs (table 5.48)

Table 5.46: LSPs corresponding to *defined classes* and *subclass-of relation* ODPs

LSP Identifier : LSP-DC-SC-ES	
NeOn ODPs Identifier : LP-DC-01 + LP-SC-01	
Formalization	
1	[Un/a] NP⟨$subclass$⟩ ser [un/a] NP⟨$superclass$⟩ REPRO VB [(NP⟨$class$⟩,)* y] NP⟨$class$⟩ | VB NP⟨$class$⟩
2	[A | any] NP⟨$subclass$⟩ be [a | any] NP⟨$superclass$⟩ PREP [(NP⟨$class$⟩,)* and | or NP⟨$class$⟩
3	[Un/a] NP⟨$subclass$⟩ REPRO VB [(NP⟨$class$⟩,)* y NP⟨$class$⟩] ser | llamarse|denominarse | VB [un/a] NP⟨$superclass$⟩
4	[Un/a] NP⟨$subclass$⟩ PREP [(NP⟨$class$⟩,)* y] NP⟨$class$⟩ ser | llamarse|denominarse | VB [un/a] NP⟨$superclass$⟩

Table 5.46: LSPs corresponding to *defined classes* and *subclass-of relation* ODPs (follow-up)

Examples	
1	*Una impresora es un periférico de computadora que permite producir una copia permanente de textos o gráficos de documentos.* *Las proteínas son compuestos nitrogenados que forman los tejidos y líquidos orgánicos.*
2	*Los repetidores son dispositivos con un sólo puerto de entrada y un sólo puerto de salida.*
3	*Las proteínas que catalizan la transferencia de los fosfolípidos se llaman flipasas.*
4	*Los animales con esqueleto externo se llaman vertebrados.*

Table 5.47: LSPs corresponding to *subclassOf relation, disjoint classes and exhaustive classes* ODPs

LSP Identifier : LSP-SC-Di-EC-ES	
NeOn ODPs Identifier : LP-SC-01 + LP-Di-01 + LP-EC-01	
Formalization	
1	Los/las NP$\langle superclass \rangle$ clasificarse en | dividirse en [CN] [los|las siguientes] [CD-CN] [PARA] [(NP$\langle subclass \rangle$,)* y] NP$\langle subclass \rangle$
2	Se distinguen [los|las siguientes] [CD-CN] de NP$\langle superclass \rangle$: [(NP$\langle subclass \rangle$,)* y] NP$\langle subclass \rangle$
3	Hay CD CN-CATV de NP $\langle superclass \rangle$ PARA [(NP $\langle superclass \rangle$,)* y] NP $\langle superclass \rangle$
Examples	
1	*Los hongos se clasifican en cuatro grandes grupos: Ficomicetos, Ascomicetos, Basidiomicetos y Deuteromicetos.*
2	*Se distinguen dos tipos de tilacoides: los tilacoides de las granas y los tilacoides del estroma.*
3	*Hay dos tipos de facturas: estándar y con confirmación de recepción.*

Table 5.48: LSPs corresponding to *object property* and *universal restriction* ODPs

Table 5.48: LSPs corresponding to *object property* and *universal restriction* ODPs (follow-up)

LSP Identifier : LSP-OP-UR-ES	
NeOn ODPs Identifier : LP-OP-01 + LP-UR-01	
Formalization	
1	NP⟨$class$⟩ sólo|únicamente|exclusivamente VB NEG (ser | tener | CATV | PART) NP⟨$class$⟩
Examples	
1	*Los herbívoros sólo se alimentan de plantas.*

1 to *N disjoint* correspondence: 1 LSP corresponds to a set of disjoint ODPs

Finally, the last type of Spanish LSPs collected in this repository embraces those linguistic patterns that are considered polysemous, i.e., that correspond to two or more ODPs that represent incompatible modeling options. Here we are referring to those LSPs that convey the following relations:

(15) LSPs corresponding to *subclass-of* or *part-whole* relations ODPs (table 5.49)

(16) LSPs corresponding to *object property*, *datatype property* or *simple part-whole relation* ODPs (table 5.50)

(17) LSPs corresponding to *simple part-whole relation*, *constituency*, *componency* or *collection-entity* ODPs (table 5.51)

Table 5.49: LSPs corresponding to *subclass-of relation*, or *simple part-whole relation* ODPs

LSP Identifier : LSP-SC-PW-ES	
NeOn ODPs Identifier : LP-SC-01 CP-PW-01	
Formalization	
1	NP⟨$class$⟩ dividirse en [CN] [(NP⟨$class$⟩,)* y] NP⟨$class$⟩
Examples	
1	*Las grasas se dividen en saturadas e insaturadas. (LP-SC-01)* *Las provincias se dividen en comunas. (CP-PW-01)*

Table 5.50: LSPs corresponding to *object property* or *datatype property* or *simple part-whole relation* ODPs

LSP Identifier : LSP-OP-DP-PW-ES	
NeOn ODPs Identifier : LP-OP-01 LP-DP-01 CP-PW-01	
Formalization	
1	NP$\langle class \rangle$ tener NP$\langle class \rangle$
Examples	
1	*Los mamíferos tienen vértebras.* *Los clientes tienen identificadores.*

Table 5.51: LSPs corresponding to *simple part-whole relation* or *constituency* or *componency* or *collection-entity* ODPs

LSP Identifier : LSP-PW-CONS-COM-CE-ES	
NeOn ODPs Identifier : CP-PW-01 CP-CONS-01 CP-COM-01 CP-CE-01	
Formalization	
1	[(NP$\langle part \rangle$,)* y] NP$\langle part \rangle$ PART NP$\langle whole \rangle$
2	NP$\langle whole \rangle$ PART [CD] [CN-PART] [PARA] [(NP$\langle part \rangle$,)* y] NP$\langle part \rangle$
3	CN-PART [en|de los/las que PART] NP$\langle whole \rangle$ be [PARA] [(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$
4	NP$\langle whole \rangle$ dividirse en CD CN-PART [PARA] [(NP$\langle part \rangle$,)* and] NP$\langle part \rangle$
Examples	
1	*Las proteínas forman parte de la estructura de todas las células y tejidos del cuerpo.* *bla.*
2	*Todas las grasas contienen grasa saturada e insaturada.* *La cámara de senadores se compone de dos miembros por cada estado.* *El manto superior de la tierra se compone de hierro y silicatos de magnesio.* *The United Arab Emirates is a country composed of seven emirates or sheikdoms.*

Table 5.51: LSPs corresponding to *simple part-whole relation* or *constituency* or *componency* or *componency* ODPs (follow-up)

3	*Las partes de una flor son el cĺiz, la corola, el androceo y el gineceo. Las partes en las que se divide el intestino son el duodeno, el yeyuno, el íleon y el intestino grueso.*
4	*El cerebro se divide en dos partes: un hemisferio izquierdo y un hemisferio derecho.*

5.5 Summary

In this chapter we have described the steps followed for the construction of the **multilingual LSPs-ODPs pattern repository**, which is the core component of the method and the tool we propose for knowledge acquisition and ontology modeling based on ODPs reuse.

In the first place we have presented the set of Logical and Content ODPs taken as starting point in our research due to their relevance in any ontology development process and their reuse across domains of knowledge.

In a second stage, we have described the strategies employed for the identification of *candidate verbal patterns* that convey the relations captured in the different ODPs. We have made use of an *ad hoc* corpus and the Web to perform an initial search for verbs and verbal phrases that express the ontological structures we are looking for. This search has allowed us to create informal intuitive links between the identified linguistic structures and the ODPs. Some links respond to common sense, and their corresponding linguistic structures have been directly formalized and introduced in the *LSPs-ODPs pattern repository*. Other *candidate verbal patterns* have required a closer analysis mainly due to their polysemic uses.

A systematic and thorough analysis of those *candidate verbal patterns* that exhibited a polysemic behavior has been conducted in a third stage. For this aim we have relied on the lexical templates provided by the Lexical Constructional Model, in combination with the mechanisms defined in the Generative Lexicon theory. This dissection activity has allowed us to discover the deep semantics of some verbs and establish more reliable links to ODPs. Moreover, the need for refinement and/or disambiguation strategies has already been pointed out when the repository is to be used in a semi-automatic fashion.

The fourth step has consisted in the development of a typology of correspondences between LSPs and ODPs. The purpose of such a typology is to account for the differences between the way in which semantics are expressed in language, and how it is captured in ODPs. Then, we present the templates that describe LSPs, and the formalization followed for the creation of LSPs according to a BNF extension. Finally, we include the set of tables that make up the *multilingual LSPs-ODPs pattern repository*. In total, tables contain 34 LSPs in English and 35 in Spanish.

Chapter 6

ODPs Reuse Method for Novice Users

Researchers in Ontology Engineering have seen the implication of domain experts in the development of ontologies as one crucial aspect for the definitive launching of the Semantic Web. The traditional knowledge acquisition process from text has proven to be a time-consuming process that could be accelerated if domain experts got a more prominent role in ontology development. On the contrary, as reported in chapter 3, section 3.2, experiments on controlled languages have also evidenced that domain experts require the support of ontology engineers at different stages of the ontology development process. This arises the question of whether domain experts could create ontologies on their own, or if they will always require some support from ontology engineers.

Our believe is that domain experts should get some background in knowledge representation in ontologies so that they understand in which way do ontologies structure knowledge, and how much knowledge can be represented in ontologies. For these reasons, the method that we propose for knowledge acquisition and ontology modeling based on Ontology Design Patterns reuse assumes that a team of domain experts and ontology engineers carries out the Ontology Specification activity. Let us recall that this activity is the first one in Scenario 1 of the NeOn Methodology, as explained in section 4.3. The importance of carrying out this activity in a team is twofold:

1. the Ontology Specification activity helps users to precisely determine what they want to model in the ontology

2. the formulation of Competency Questions (CQs) with the support of ontology experts helps users parcel their domain of knowledge in small and manageable bits of knowledge

Once the ontology requirements have been formulated mainly in the form of CQs, the reformulation in statements that can be converted into ontological structures should be quite immediate. This is the second assumption in the method we

propose. Once CQs and their corresponding answers have been formulated, resulting in a parceling or sectioning of the domain of knowledge to be modeled (see Figure 4.6 in section 4.3), domain experts should be in the position of making affirmative statements of what is to be definitely included in the ontology. Examples of this will be presented below.

In this way, and contrary to what was proposed by the approaches relying on Controlled Languages (CLs), our method adopts a more "naturalist" approach (Clark et al., 2009) by allowing domain experts to formulate in full Natural Language (NL) their modeling needs. The main advantage of such an approach is that it does not require the user spending time and efforts learning a CL. And most importantly, the user does not need to understand the ontology structure which underlies the syntax defined by the CL. Nonetheless, there is one main drawback in naturalist approaches, namely, the ambiguities present in NL that have to be dealt with in order to find out what the user means to model. Such ambiguities do not happen in "formalist" approaches to CLs. In spite of that, we believe that the advantages of naturalist approaches outweigh any disadvantages, if our final aim is to bring ontology technologies closer to the general user.

In this sense, the approach we support regarding the use of CQs is quite close to the one proposed in the XD method (Presutti et al., 2009) for ontology engineers in general, as presented in section 4.2.3. However, we will use the ORSD, including the CQs, as a background document that domain experts use to finally formulate in declarative sentences what they want to include in the ontology. At this stage, some recommendations will be given to domain experts to help them in the formulation task, so that they are aware of the type of sentences expected from them. This will allow us to automate the matching task (Task 4 in the XD method), because in our approach we cannot assume domain experts to have good knowledge of ODPs.

It is exactly in this task where the method we propose for reusing ODPs differs from the XD method. We believe that finding the most appropriate design pattern for a requirement or modeling need expressed in NL is by no means trivial, and that some support should be provided, if our aim is to actively involve domain experts in the development of ontologies. In this sense, the method we propose needs some supporting tool to help users in a "semi-automatic matching" of ODPs. Both method and tool rely on the *multilingual LSPs-ODPs pattern repository* described in chapter 5. This repository has to be implemented in a Natural Language Processing (NLP) tool that automates the reuse of ODPs.

In the next section, we provide a filling card (see Figure 6.1) defining the ODPs reuse activity intended for novice users, as suggested by the NeOn Methodology for all the activities taking part in the ontology development process. Then, we present the tasks into which this activity is decomposed, which should provide the user helpful guidance to carry out the ODPs reuse activity. To illustrate the method we include an example of use. Finally, we offer an overview of the interaction between the methodological guides and the technological support needed for this activity.

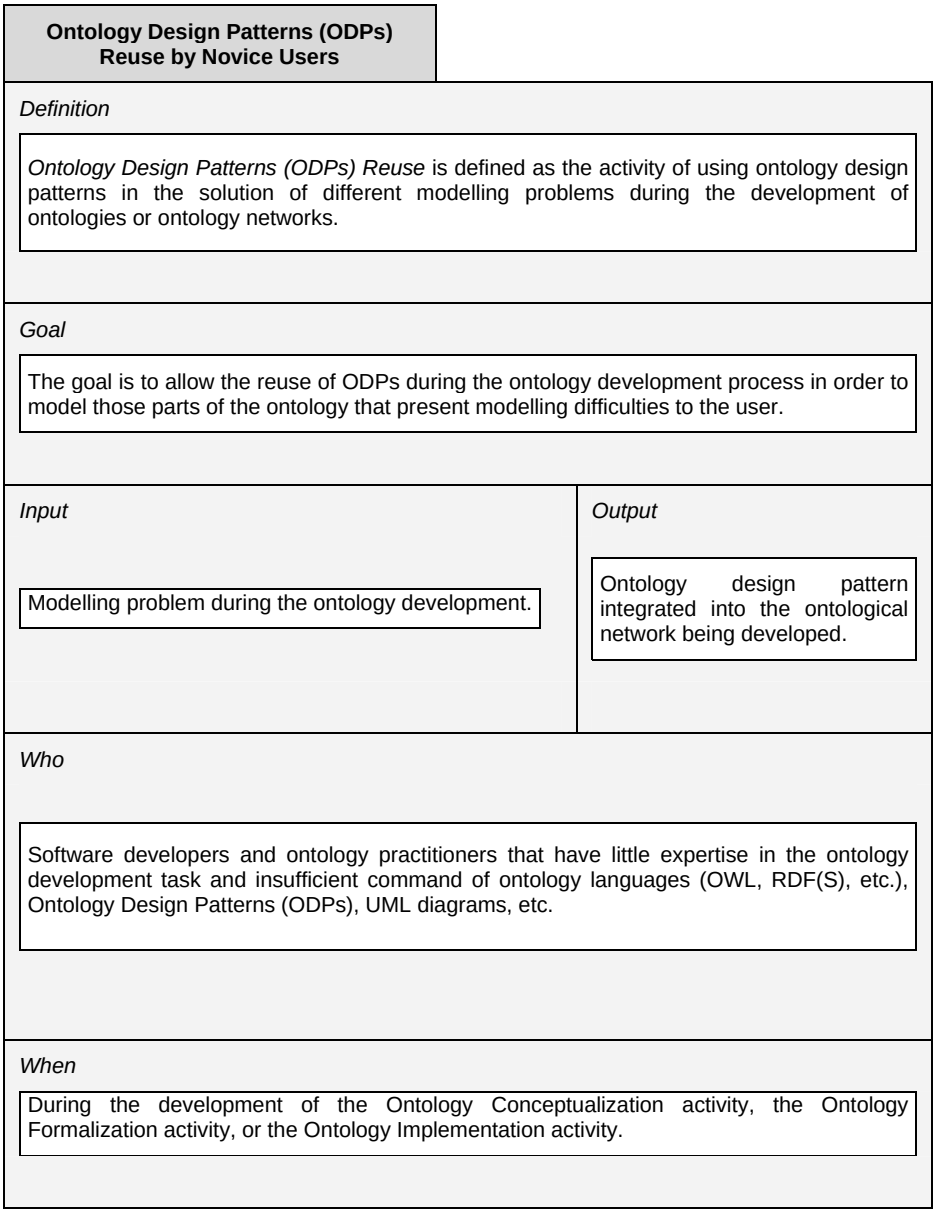


Figure 6.1: Filling card for the ODPs reuse activity aimed at novice users

6.1 Methodological Guides

The filling card provided for the ODPs reuse activity (see figure 6.1) offers information about *definition*, *goal*, *inputs*, *outputs*, *who* carries out the task, and *when* the task should be performed. This activity is decomposed in several tasks that should be carried out in the prescribed order to obtain the expected outputs. Figure

6.2 illustrates the workflow of this activity.

1. **Task 1. NL Formulation.** The goal of this task is to formulate in full NL the domain aspect to be modeled: the user has difficulties in modeling a certain domain parcel and expresses that knowledge in NL. We assume that users introduce correct information from the content viewpoint. Taking into account that the formulation has been derived from the ORSD, specifically from the set of CQs, the content should be reliable because it has been previously validated by the development team. Additional guidance is provided to the user by means of some *recommendations* accompanied by real examples in NL of the type of input that is expected. Recommendations have been included in Table 6.1.

Recommendations
1. Express one topic or idea per sentence. Avoid coordination of phrases, and use only when necessary. *Falls are types of incidents, which can happen in hospitals.* WRONG *Falls are types of incidents. Falls happen in hospitals.* RIGHT
2. Include in each sentence subject, verb and object (SVO) (Do not use pronouns instead of nouns!) *They receive assistance.* WRONG *Patients receive assistance.* RIGHT
3. Avoid using neither interrogative nor negative sentences. *Chairs are not considered mobility aids.* WRONG *Mobility aids are walking sticks, walking frames, crutches, wheelchairs, walking tripods, callipers, orthotics, and prosthetic devices.* RIGHT
4. Avoid including redundant or unnecessary information that does not add new content to the idea. *According to many people, medications can cause falls.* WRONG *Medications cause falls.* RIGHT
5. End up each sentence with full stop.
6. In enumerations, use comas to separate elements. *Examples of Fall Minimation Strategies are restraint, safety devices, protocols, intervention, and procedures.* RIGHT

Table 6.1: Recommendations for task 1.

2. **Task 2. Input Refinement.** The goal of this task is to refine the input from Task 1. This task is only carried out when there is no direct correspondence to one ODP. The reasons for refinement may be related with ontology enrichment needs or lexical ambiguities. Different strategies may be adopted

to solve this lack of correspondence, such as, user interaction with the system or search in external ontologies or lexicons. Examples of refinement strategies will be given in section 5.4. This task should be repeated until the exact matching is obtained.

3. **Task 3. Pattern Validation.** The goal of this task is to confirm that the resulting ODP or ODPs meet the user's expectations. The final result is expected to be returned to the user in the form of a UML diagram (instantiated with information from the NL formulation), and the corresponding OWL code. As we may recall, the corresponding OWL code was already contained in the templates that described ODPs (M. C. Suárez-Figueroa et al., 2007), (Presutti et al., 2008), and in the on-line Ontology Design Patterns Portal. Therefore, the task output is one or several ODPs ready to be integrated in the ontology network being developed. Technological support for this latter task has not been further elaborated in this work.

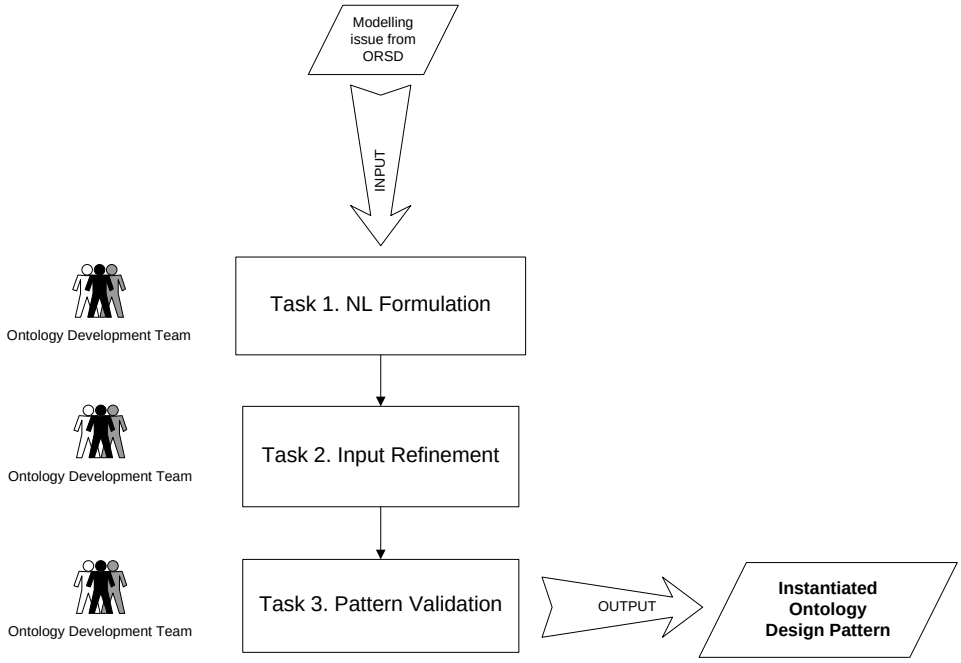


Figure 6.2: Method for the reuse of ODPs aimed at novice users

6.2 Example of Use

In the following we provide an example of the guidelines application with a simple example of an ontology development project in the Health Care domain.

Let us assume that a group of domain experts would like to create an ontology modeling the types of incidents that can happen in hospitals. The group of experts

formulates a set of competency questions (CQs) with the support of a team of ontology engineers. The set of CQs specifies the content requirements that the ontology should satisfy, i.e., the concepts, relations and restrictions that should be modeled in the ontology, so that the ontology is able to answer those questions once modeled. See table 6.2 for some examples of CQs.

ORSD	Competency Questions (CQs)
CQ1	What type of fall caused the incident? Trip, stumble, slip, collapse, loss of balance.
CQ2	Which elements were involved in the fall? Cot, bed, chair, stretcher, therapeutic equipment, steps, clothing, bed linen.
CQ3	Who observed the fall? Staff, visitor, family, another patient.
CQ3	...

Table 6.2: Example of CQs of the Health Care Domain

Taking those CQs as starting point, the user formulates statements of the knowledge represented in the CQs following the recommendations given in table 6.1, as suggested by Task 1 of the method: **NL Formulation**. In this sense, experts should take into account that they need to fragment one idea per sentence, as suggested by Recommendation number 1. They should also notice that each sentence has to be finished with full stop (Recommendation 5), and that comas have to be used to separate elements in an enumeration (Recommendation 6). Regarding CQ1 in table 6.2, this means that the question and its corresponding answer would be broken into two sentences expressed in the following way:

1. Falls cause incidents.

2. There are different types of falls: trip, stumble, slip, collapse, and loss of balance.

The formulations resulting from Task 1 would be introduced in a system that would provide a matching to an ODP or to a set of ODPs, thanks to the *LSPs-ODPs pattern repository*. If no direct matching is found, a refinement of the input would be needed, as proposed by **Task 2. Input Refinement**.

If a matching has been found, the third and last task will be performed. This task, **Task 3. Pattern Validation**, will consist in the confirmation on the part of the user of the validity of the resulting ODP or ODPs instantiated with the information from the initial formulation. As already pointed out, Task 3 has been left for future work.

6.3 Methodological and Technological Interaction

Once the method has been introduced, we would like to provide an overview of the interaction between the methodological and technological components of this approach, before presenting the implementation work that has been done in chapter 7. Figure 6.3 illustrates this interaction.

The process starts with the formulation in NL of the domain aspect to be modeled (**Task 1. NL Formulation**). Taking as input the set of CQs from the ORSD and the recommendations provided by the method in table 6.1, novice users are asked to formulate the domain aspect they aim at modeling in the ontology and introduce a sentence in the system (step 1 in Figure 6.3).

The next step (step 2) consists in processing the input sentences with NLP tools. We have opted for performing this analysis with GATE, the General Architecture for Text Engineering developed at the University of Sheffield (H. Cunningham et al., 2009). Details of the GATE Architecture and the annotations used for our purposes will be given in chapter 7.

Once the input sentence has been annotated with GATE, the result will be compared against the *multilingual LSPs-ODPs pattern repository* to look for correspondences (step 3). As already explained in section 5.4, correspondences between LSPs and ODPs are not always direct correspondences. Let us recall that they can be 1 to 1 correspondence, 1 to N correspondence, and 1 to N disjoint correspondence. According to the matching modality three different situations can arise:

1. When the matching between the annotated sentence and the multilingual LSPs-ODPs pattern repository is 1 to 1, the system will identify the appropriate ODP that solves the user modeling problem, and show the results to the user.

2. When the matching results in a 1 to N correspondence, as was the case of the *LSPs for subclass-of relation, disjoint classes and exhaustive classes ODPs*, a refinement will be needed to make sure that all modeling options apply.

3. If the third correspondence modality happens to match, 1 to N disjoint correspondence, as in the *LSPs for subclass-of relation or simple part-whole relation ODPs*, then a disambiguation strategy has to take place.

And there is still a fourth option, in which no matching at all is found. In that case, the user also needs to be asked to check and refine the input. Different refinement and disambiguation strategies have been investigated, as reported in Aguado de Cea et al. (2008) or Montiel-Ponsoda, Aguado de Cea, Gómez-Pérez, and Suárez-Figueroa (2008), and will be summarized in section 6.4.

Once the refinement or disambiguation tasks have been performed as proposed in **Task 2. Input Refinement**, results can be displayed in the form of a UML diagram instantiated with the information from the input sentence, as shown in figure 6.3, or by displaying the corresponding OWL code (step 5).

Finally, the user will be asked to validate the obtained result (**Task 3. Pattern Validation**). Since this approach is aimed at novice users, the returned pattern or OWL code should be accompanied by an explanation in NL of the modeling possibilities offered by the matched pattern.

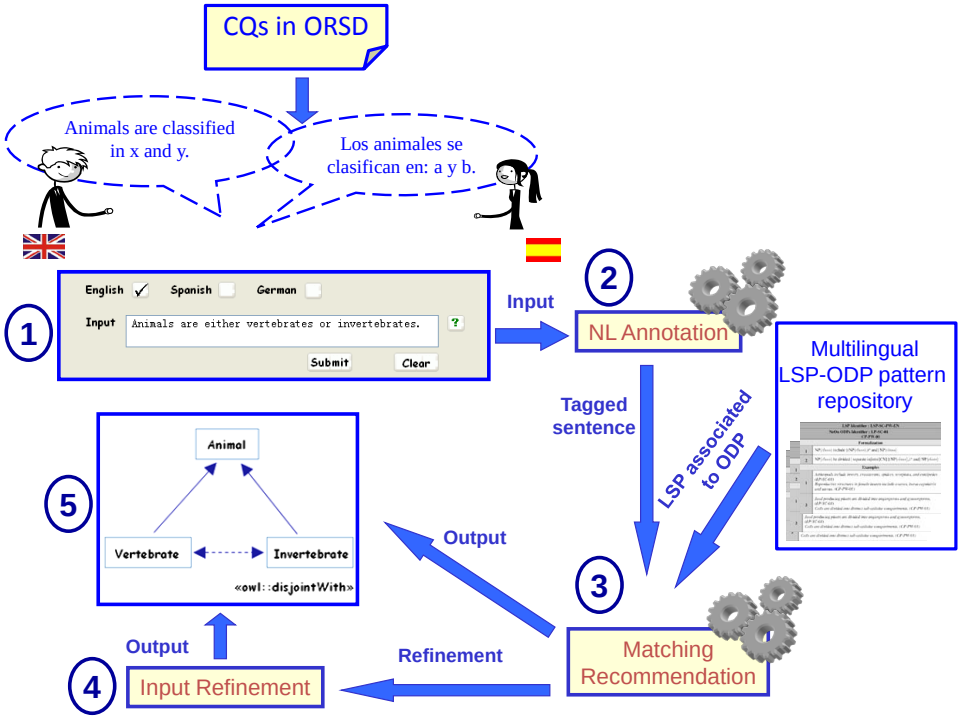


Figure 6.3: Overview of the proposed approach for the reuse of ODPs

Steps 4 and 5 of the proposed approach for the reuse of ODPs are out of the scope of the work presented in this thesis, and have not been implemented.

6.4 Strategies for solving NL Ambiguities in LSPs

In this section, our aim is to exemplify some of the strategies that we have devised to support users in the performance of the second task proposed in the method, **Task 2. Input Refinement**[1]. As already pointed out, this task should be carried out whenever an LSP matches N disjoint ODPs that need to be disambiguated, or N ODPs that need to be refined.

In the following we provide one example for each use.

[1]These strategies have been presented in several papers such as (Montiel-Ponsoda, Aguado de Cea, Gómez-Pérez, and Suárez-Figueroa, 2008), (Aguado de Cea et al., 2008) or (Aguado de Cea et al., 2009).

Example 1

Task 1. NL Formulation: Let us imagine that the user introduces the following sentence in English in the input window

Arthropods include insects, crustaceans, spiders, scorpions, and centipedes.

For exemplifying the method, we assume that the user wants to represent a subclass-of relation.

Task 2. Input Refinement: The system would identify that the resulting annotated sentence has a correspondence with the *LSP for modeling subclass-of relation or simple part-whole relation ODPs* (see table 5.32 in the *multilingual LSPs-ODPs pattern repository* presented in section 5.4).

Whenever the correspondence is 1 LSP to N disjoint ODPs, a disambiguation process is needed. As already introduced, this situation results from the ambiguity present in the polysemous verb *include*, since it can correspond to two ODPs, one modeling the subclass-of relation, and the other modeling the simple part-whole relation.

For these cases, an option would be to interact with the user by means of so-called *refining questions*. In this example, questions would be:

1. Are insects, crustaceans, spiders, scorpions, and centipedes, **types of** arthropods?
2. Are insects, crustaceans, spiders, scorpions, and centipedes, **parts of** an arthropod?

The answer to the first question should be yes, and to the second, no, if the input sentence wants to model a subclass-of relation, as we suppose in this example. In this way, the system would help users to come to the right decision.

Task 3. Pattern Validation: The system would return the user a UML diagram representing an ODP for the subclass-of relation and instantiated with the information from the NL formulation, plus a description. The user would only need to validate it.

Seemingly, a sentence like *Birds have feathers* corresponding to the *LSP for object property, datatype property and simple part-whole relation* needs to be disambiguated because three different modeling solutions are possible. Here, however, we are not only dealing with the multiple polysemous senses a verb can have, but also with the different modeling decisions the user can take according to his or her needs, that is,

1. feather as a class related to the class bird
2. feather as a property of the class bird

3. feather as parts of bird

Conscious of the intricacy, but at the same time the importance, of such a modeling problem, other strategies can be proposed to solve this kind of polysemy. One of them would be to search for ontologies in the Web already modeling that kind of knowledge. Thanks to Semantic Web Search Engines such as Watson[2] this can be easily done nowadays. Results of this search could be shown to the user, so that (s)he chooses how to model it. This strategy has not been further explored, but is left for future work.

Example 2

For this second example, let us assume that the user introduced the following sentence expressing a subclass-of relation.

Task 1. NL Formulation:

Vertebrates are classified into mammals, amphibians, reptiles, and birds.

Once the correspondence to the subclass-of relation ODP has been obtained, it would be recommendable from an ontological viewpoint to enrich this relation with knowledge about disjointness and exhaustiveness. This is a typical case of ODPs that are commonly used in combination. With the aim of making users aware of this fact, and support them in the reuse of best practices in ontology modeling, it would also be advisable to ask them to refine the input.

A similar strategy to the one presented in Example 1 has been also designed to find out if the classes in a subclass-of relation are additionally disjoint and/or exhaustive.

Task 2. Input Refinement: Regarding exhaustiveness, the question could be

1. Are there any other types of vertebrates?

If the answer is yes, the system would ask the user if (s)he would like to introduce the missing class(es) or group(s). The question could be

2. Would you like to add any other types of vertebrates for making this list complete or exhaustive?

If the answer is yes, the system would offer the user the possibility of introducing the missing subclasses in the input window. In this example, the type *fish* is missing to reach an exhaustive enumeration of vertebrates. The user would be

[2] http://watson.kmi.open.ac.uk/WatsonWUI/

made aware of this, and would introduce the new subclass. Then, the system would model those classes according to the Exhaustive Classes ODP, and would proceed to ask about disjointness. If, for any reason, the user decides not to introduce the missing class, the system would directly proceed to ask about disjointness.

Regarding disjointness, the question could be

3. Can a certain vertebrate belong to the group of mammals, amphibians, reptiles, birds, and fish at the same time?

The answer should be no, and the system would further model those subclasses as Disjoint Classes. This kind of dependencies between ODPs are illustrated by a state diagram in figure 6.4.

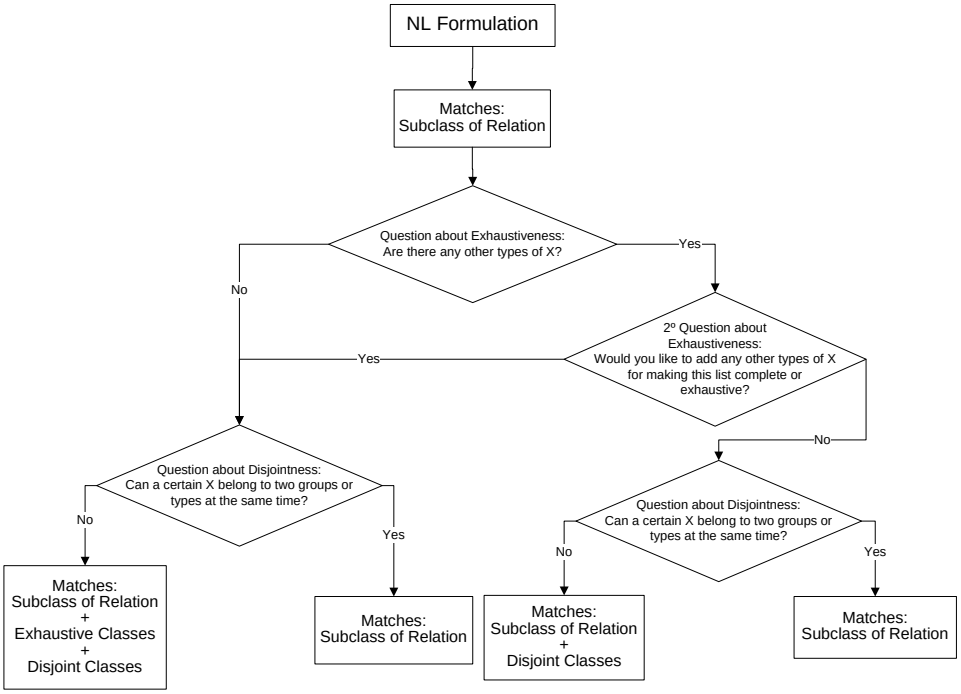


Figure 6.4: Dependencies between ODPs: subclass-of relation, disjoint classes and exhaustive classes

Regarding the third and last task, **Task 3. Pattern Validation**, the system we devise returns the user a UML diagram representing the ODPs with information from the NL sentence, as in figure 6.5 for the sentence in Example 2. This diagram should be accompanied by an explanation in NL of the model to instruct the user in the modeling of ontologies. In this way, the user has a new opportunity to check if the returned UML diagram complies with his or her expectations. If (s)he finally accepts the output, it is then integrated into the ontology being developed. As already pointed out, the technological components required to support this functionality are out of the scope of this work.

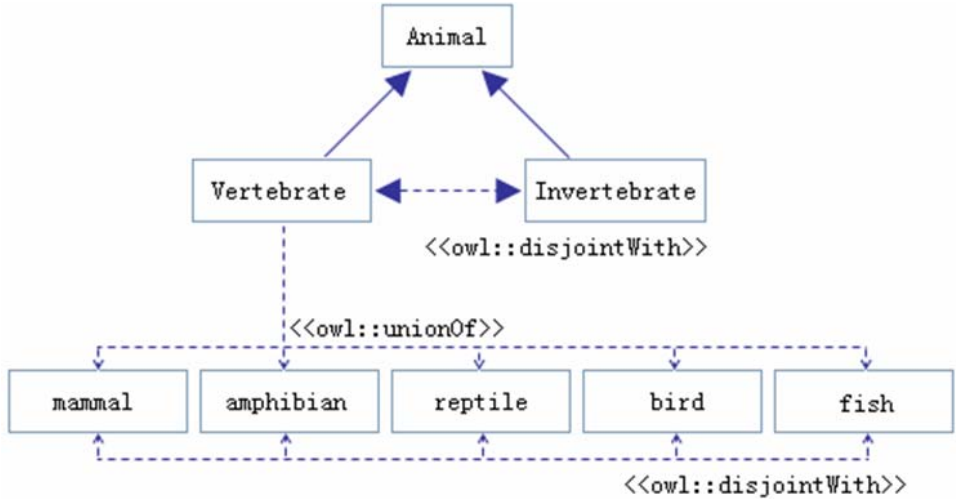


Figure 6.5: Example of an instantiated UML diagram

6.5 Concluding Remarks

From an Ontology Engineering viewpoint, the main benefits of this method are summarized in the following.

On the one hand, it allows the performance of the knowledge acquisition activity from domain experts by means of the formulations in NL of what is to be modeled in the ontology. We should recall that the formulations in NL are based on the set of CQs agreed on previously by the domain experts and ontology engineers that make up the Ontology Development Team.

Another interesting aspect of this approach is that it allows domain experts to use full NL when interacting with a system that supports a semi-automatic reuse of ODPs for ontology modeling.

On the other hand, this method ensures an appropriate modeling of the domain aspect expressed in the NL formulation because it reuses consensual verified solutions represented by ODPs. Thus, it prevents the use of bad practices in ontology modeling.

Chapter 7

LSPs Implementation and Evaluation

As already described in chapter 6, our approach for the reuse of ODPs involves the annotation of the input provided by the novice user in NL, and its comparison against the *multilingual LSPs-ODPs pattern repository*. If a matching is to be obtained between the annotations of the input and one of the LSPs, the ODP or ODPs associated to the LSP are returned to the user as the solution for the modeling problem expressed by him or her in NL.

With the aim of performing the annotation of the NL input, we decided to use GATE[1]. GATE (H. Cunningham et al., 2002, 2009), the General Architecture for Text Engineering, is a framework for the development and deployment of software components for NLP. Its basic component, ANNIE, is an information extraction system that relies on basic processing resources. Additionally, GATE has a large number of plug-ins that can be combined to create different NLP applications.

For the purposes of this research we relied on the annotations provided by ANNIE processing resources and by some additional processing resources that will be described below. One of the most important processing resources in our application is the JAPE transducer. JAPE stands for Java Annotation Patterns Engine and is a grammar that "provides a finite state transduction over annotations based on regular expressions", as documented in (H. Cunningham et al., 2000). To put it in simple words, JAPE allows users to identify certain structures in documents relying on available annotations (previously provided by other processing resources in GATE), and create new annotations. Basing on this, we created our own annotations that corresponded to the elements in the different ODPs included in our repository. Some examples of these new annotations are *subclass, superclass, object property, disjoint class, participant,* or *event*. In this sense, GATE provided us with the needed functionalities to perform the *annotation* and *matching recommendation* steps (step 2 and 3) in the method we propose for the semi-automatic reuse of ODPs (see section 6.1).

[1] http://gate.ac.uk/

In section 7.1, our aim is to present the application we created for the identification of ODPs from NL input[2]. Then, we describe one JAPE rule by way of example. The complete JAPE code has been made available at the Ontology Design Patterns Portal, as will be described in section 7.2, and can be freely accessed and downloaded. Note that we created an application only for the processing of English sentences. Although GATE also supports some processing resources for Spanish, not all the processing resources needed in our application were available for Spanish. Therefore, the implementation of the Spanish LSPs repository is left for future work.

Finally, in section 6.3 we describe an experiment we conducted to validate both, the method for the reuse of ODPs proposed in chapter 4, and the application developed in GATE for the annotation and matching of LSPs to ODPs.

7.1 LSPs Implementation in GATE

With the aim of supporting the automatic recognition of ODPs in the formulations provided by the users of our method, we created a GATE application called **LSPs application**. An application in GATE consists of a set of processing resources executed following a sequential order over a set of documents contained in a corpora. A capture of GATE's interface is included in figure 7.1. There, we see the LSPs application, the corpora over which the application is run, and the set of processing resources that make up the LSPs application. Processing resources can be added or removed according to the needs of the application. Similarly, new documents can be added to the corpora or removed from it.

A snapshot of the LSPs application pipeline can be seen in figure 7.2. Most of the processing resources we employed in our application belonged to the ANNIE plug-in, GATE's basic component, as depicted in figure 7.2. The rest of them were obtained from other GATE plug-ins.

ANNIE consists of several processing resources that perform annotations over documents in NL. Annotations are made up of features with a name and a value. For example, ANNIE's Tokeniser can identify the word *animal* in a document, and associate to that word the feature *Token*, with the name *kind*, and the value *word*, as in

```
Token.kind == word
```

If additional information from the Part-of-Speech (POS) tagger has been obtained, a further annotation of the same word will be

```
Token.pos == NN
```

[2]For the actual implementation task of the JAPE rules, we visited the Natural Language Processing Group at the University of Sheffield. We are greatly indebted to Diana Maynard for her support with the JAVA programming code.

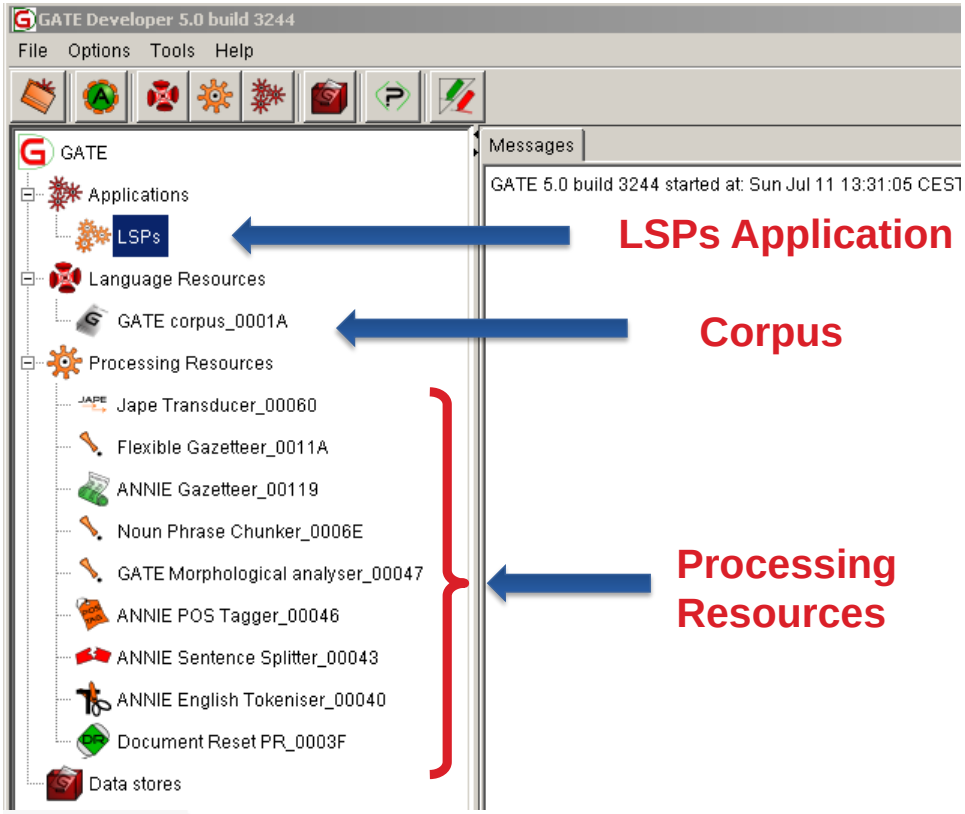


Figure 7.1: GATE's main interface

in which NN stands for "noun in singular" according to the Hepler Tagger (see (H. Cunningham et al., 2009)).

ANNIE contains the following processing resources:

- Document Reset: clears existing annotations in the document, so that no annotations are embedded in the document, and the document can be brought to its original status
- Tokeniser: divides the text into tokens, such as number, punctuation or word, and adds a Token annotation to it
- Sentence Splitter: divides the text into sentences
- POS-Tagger: adds part-of-speech information to Token annotations
- Gazetteer: contains lists of words grouped in categories to perform Named Entity (NE) recognition or key phrase lookups
- Orthomatcher: adds identity relations between NE found by the NE Transducer to perform co-reference

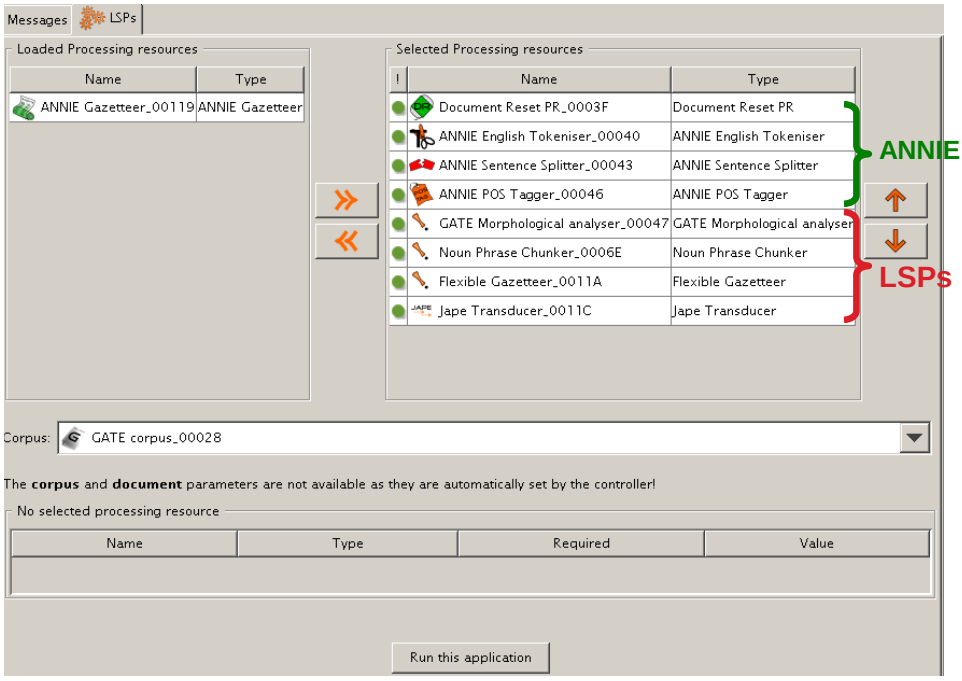


Figure 7.2: Sequential order in the execution of the processing resources in GATE

- JAPE transducers: executes JAPE rules to create complex annotations based on the results of the previous processing resources

The resources contained in ANNIE were complemented by further processing resources to obtain additional annotations that were needed by our LSPs application. These resources are

- Morphological Analyser: adds morphological information (lemma and affixes of words)
- Noun Phrase Chunker: identifies noun phrases
- Flexible Gazetteer: is a gazetteer that allows us to create our own lists of NEs or key words to perform lookups

It is worth mentioning the role of the Flexible Gazetteer. This processing resource permits the creation of wordlists that are to be identified by the *Lookup* annotation. For the LSPs application, we created our own lists of some key words and key verbs that allow us to unambiguously identify the linguistic structures we are interested in.

In figure 7.3, we have included a snapshot of one of the gazetteers we created for identifying those words that introduce noun phrases that are in a relation of subclasses to its superclass. That specific wordlist was named CN, for “class name”,

and generated an annotation of the feature type Lookup, as shown below:

```
Lookup.majorType == CN
```

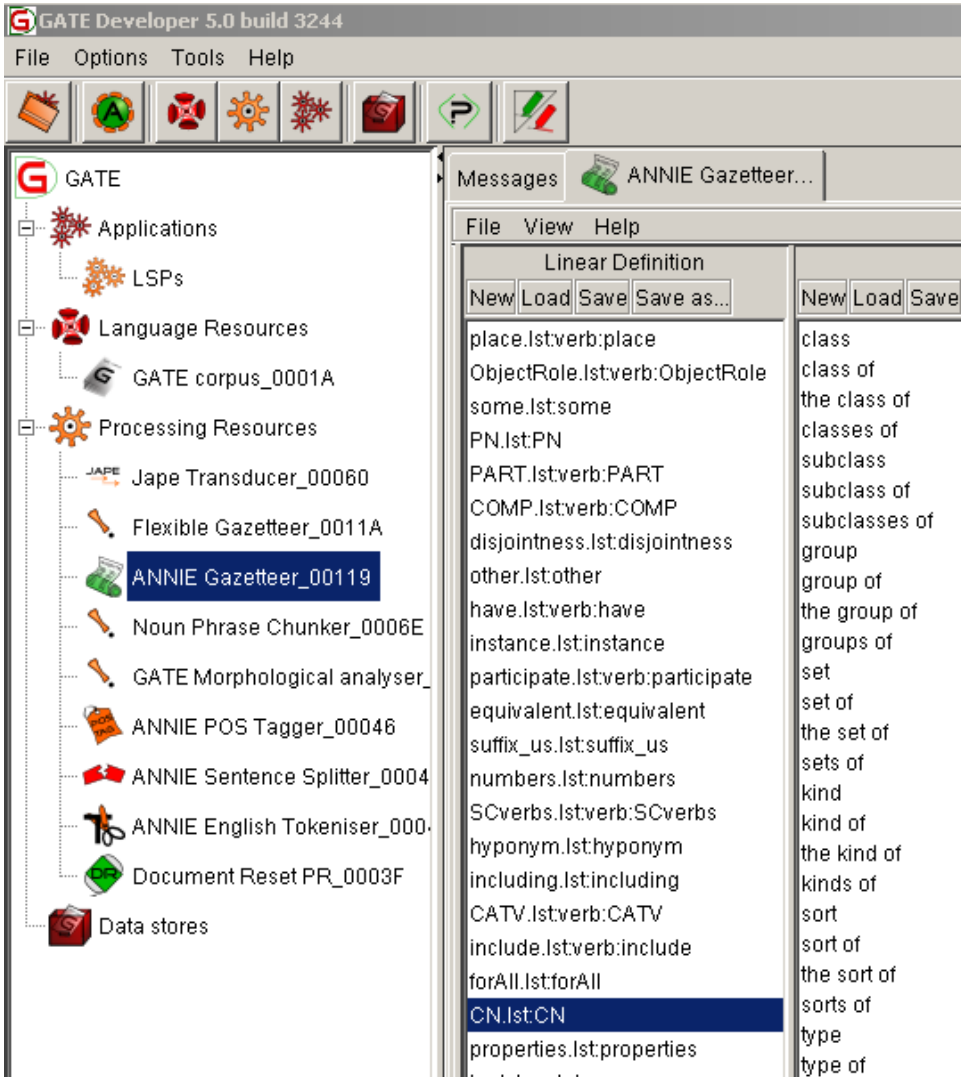


Figure 7.3: Snapshot of a gazetteer in GATE

As can be observed in the pipeline we defined (see figure 7.2), the last processing resource to be run is the JAPE transducer. The JAPE transducer needs to rely on the rest of annotations provided by the processing resources previously run, to create complex annotations as a result of the execution of the JAPE rules that we created.

As reported in the GATE User Guide (H. Cunningham et al., 2009), JAPE consists of a set of phases, each of which contains pattern/action rules. JAPE rules

are divided into two parts: the so-called left-hand-side (LHS) of the rule, which contains the annotation pattern, and the right-hand-side (RHS) of the rule, which contains the "annotation manipulation statements". This means that the LHS of the rule needs to match certain annotation patterns in the document, so that the RHS can perform a certain action. See figure 7.4 for an example of a JAPE rule to match e-mail addresses in texts.

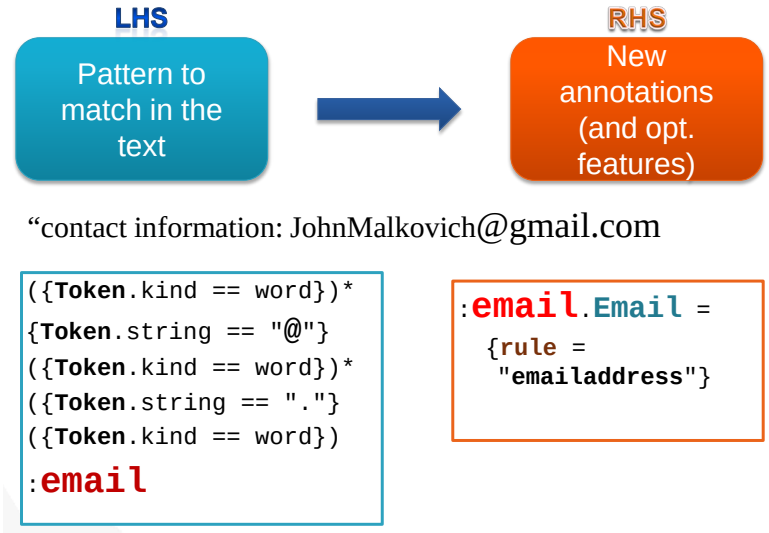


Figure 7.4: The two phases of a JAPE rule

This is the structure of a JAPE rule: The LHS of the JAPE rule is separated from the RHS by the symbol −>. The LHS relies on annotations, and optionally, on their features and values. Any annotation to be used must be included in the input header, and is to be enclosed in curly braces.

Besides the possibility of expressing annotations or strings of text, the presence or absence of annotations can also be indicated. For instance, if we want to match the Token annotation for the @ symbol as in figure 7.4, we would add that annotation in curly braces, as in

```
({Token.string == "@"})
```

However, if we want the LHS to match whenever the same annotation is not present, we will have to express it by means of the symbol `!`, as in

```
({!Token.string == "@"})
```

Finally, the optionality of an annotation will be indicated by means of the `?` symbol, as in

```
({Token.string == "@"})?
```

The LHS of the rule also permits the employment of the traditional Klene operators (H. Cunningham et al., 2002) to combine annotation patterns:

- | meaning "or"
- * meaning 0 or more occurrences
- + meaning 1 or more occurrences
- ? meaning 0 or 1 occurrences

Finally, each pattern to be matched in the LHS is enclosed in round brackets and can have a label attached to it, as it is the case of **email** in our example in figure 7.4. And the same happens with the RHS of the rule. The names given to the LHS and RHS of the rule will appear as the new annotations generated from the application of the JAPE rules.

Next, we include one of the JAPE rules created for the identification of LSPs 1 of the *LSPs corresponding to subclass of relation ODP* in table 5.18 by way of example. This JAPE rule has been given the name SC1_1 in the repository of JAPE rules.

[(NP$\langle superclass \rangle$**,)* and] NP**$\langle superclass \rangle$ **be [CN-CATV] NP**$\langle superclass \rangle$

The first step is to create a matching pattern for a list of noun phrases. Since a list of noun phrases is a recurrent pattern in the set of LSPs, it is advisable to create a Macro with the name LIST. In this pattern we specify that whenever a list of noun phrases followed by a comma appears in the text (from 0 to 10 times) followed optionally by a comma before a coordinating conjunction ("and" in this case), and a further noun phrase, then the pattern LIST is matched.

```
Macro:LIST
(
(NounChunkToken.string == ",")[0,10]
NounChunk
(Token.string == ",")?
Token.category == CC
NounChunk
)
```

The macro LIST will be the first element of the LHS in this SC1_1 JAPE rule. Each of the noun phrases identified here will correspond to the subclasses of the pattern, hence the label assigned to the LIST is **subclass**, as can be seen in the JAPE code included below.

Then, we specify a lookup annotation with the feature "minorType" and value "be" to match any verbal form of the verb to be. After this, a determiner can be optionally matched (note the symbol ?). Then, another lookup annotation, this time with feature "majorType" and value "CN" (Class Name) should be matched. To end up, a last noun phrase (NounChunk) has to be matched that will correspond to the **superclass**.

After this, the RHS is specified separated by ->. This symbol is followed by the label assigned in the LHS to the Noun Chunk, **superclass**, and the name of the new annotation (Superclass) separated by a dot. Then, JAVA code is used for the second part of the RHS, and for the specification of the second new annotation (Superclass).

```
   (
(LIST):subclass
Lookup.minorType == be
(Token.category == DT)?
Lookup.majorType == CN
(NounChunk):superclass
)

->
:superclass.Superclass = rule="SC1_1",
{
   // "subclass" matches LHS label
List annList = new ArrayList((AnnotationSet)bindings.get("subclass"));

   //sort the list by offset
Collections.sort(annList, new OffsetComparator());

   //iterate through the matched annotations
for(int i = 0; i < annList.size(); i++)
{ Annotation anAnn = (Annotation)annList.get(i);

   // check that the new annotation is a NounChunk
if ((anAnn.getType().equals("NounChunk")) )

   { FeatureMap features = Factory.newFeatureMap();

   // change this for a different rule name"
features.put("rule", "SC1_1");

   // change "Subclass" for a different annotation name
annotations.add(anAnn.getStartNode(),
anAnn.getEndNode(),"Subclass",features);}}}
```

If this JAPE rule is matched in any of the documents that form part of the corpora, the new annotations **subclass** and **superclass** will be created. Figure 7.5

illustrates the way in which annotations are assigned to text bits in documents. Each annotation is represented by a color. By selecting the annotations or markups from the right-hand side list, those annotations are visualized in the text. Then, by positioning the mouse on each of the colored bits of text, a small window pops up and shows the annotation (**Subclass** in the example in figure 7.5), and the name of the JAPE rule that generated it (SC1_1).

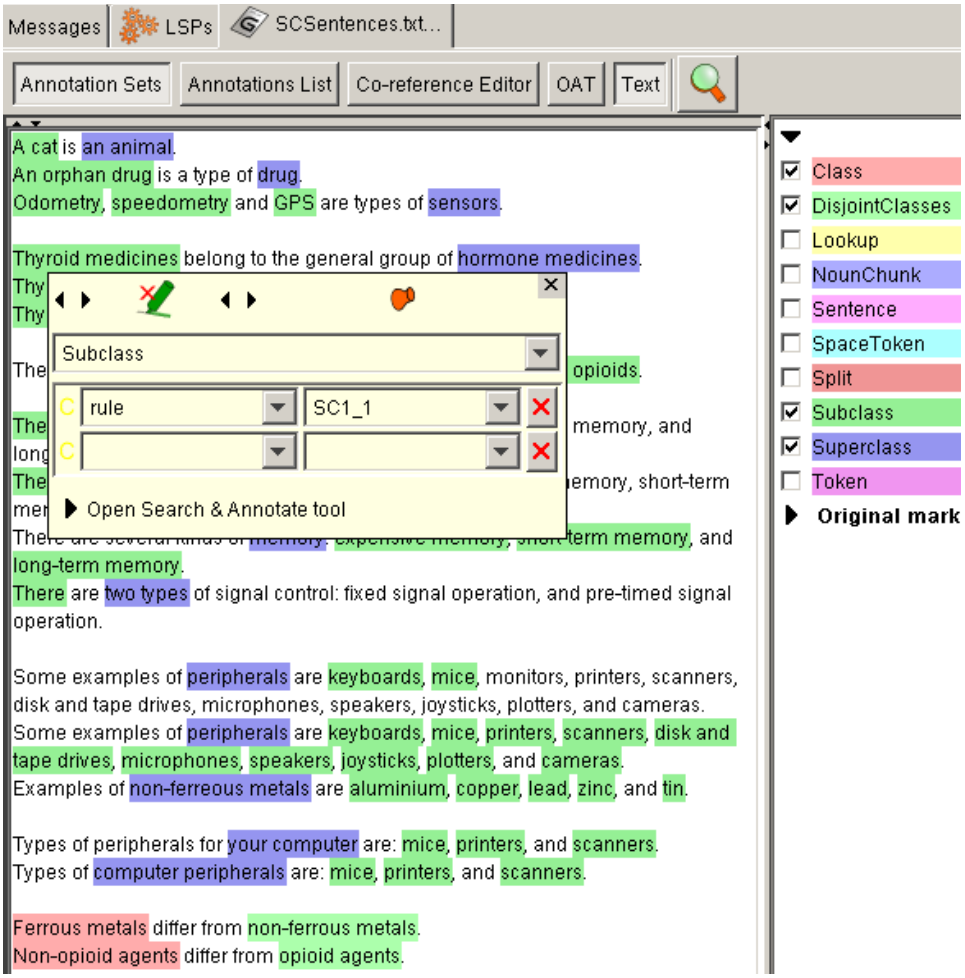


Figure 7.5: Annotations generated from the JAPE rule SC1_1

In figure 7.5, we can observe that the first three sentences of the document matched the JAPE rule presented above, which in its turn corresponds to LSP 1 of the *LSPs corresponding to subclass of relation ODP* in table 5.18. By positioning the mouse over the word *sensors*, we see the name of the annotation (**Subclass**) and JAPE rule.

The rest of JAPE rules created to match the English patterns defined in chapter 5, section 5.4.1 are available at the Ontology Design Patterns Portal, as will be

explained in section (section 7.2).

7.2 LSPs Publication in the ODPs Portal

In this section, we will refer to the publication of the English LSPs-ODPs pattern repository in the Ontology Design Patterns Portal, *www.ontologydesignspattern.org*. The aim of this contribution is to make LSPs, and their corresponding JAPE code, available to the Ontology Engineering Community, so that it can be reused or enhanced in further NLP applications.

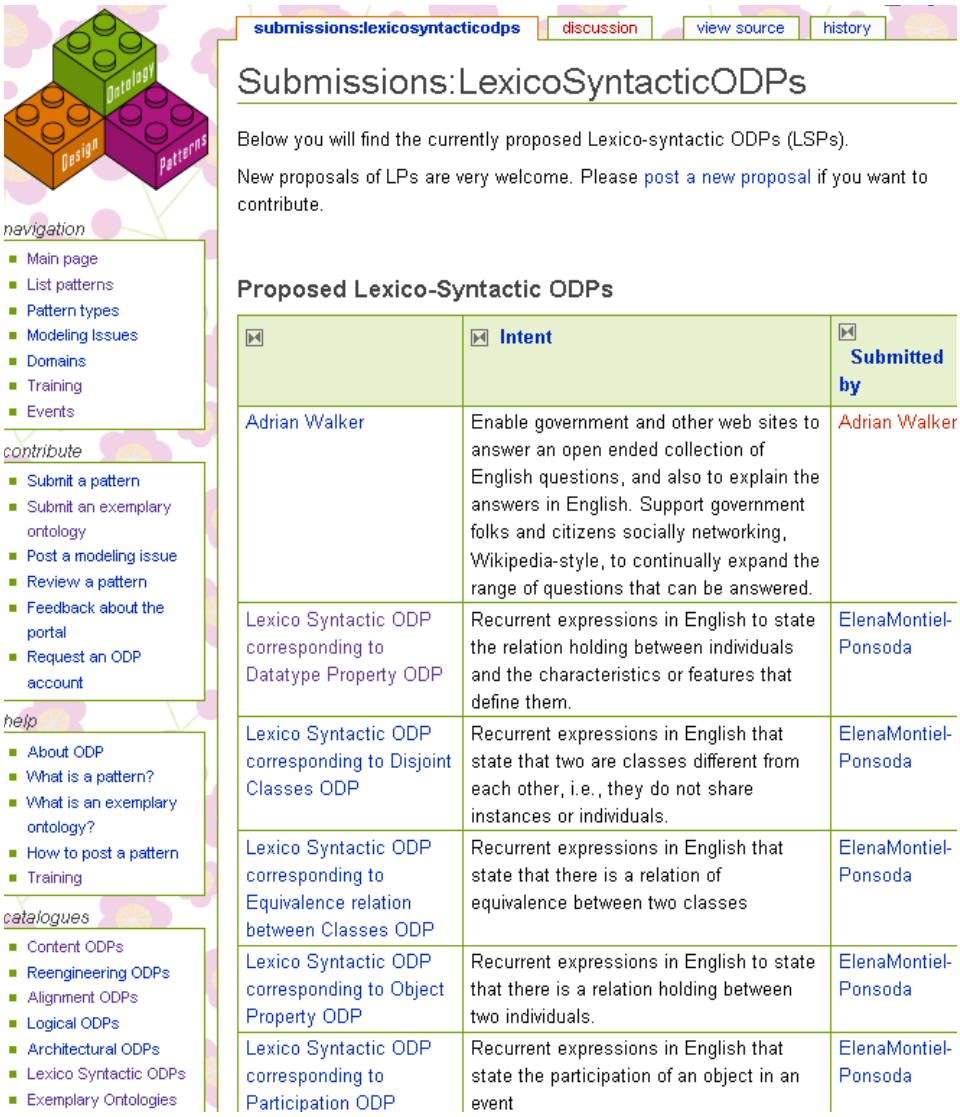


Figure 7.6: LexicoSyntacticODPs repository at Ontology Design Patterns Portal

A specific section has been created in the catalogue for the inclusion of Lexico-Syntactic Patterns[3] (**LexicoSyntacticODPs**), as can be seen in figure 7.6. According to the Ontology Design Patterns Portal philosophy, users are invited to submit proposals of patterns, which are assigned to at least two members of the ODP Quality Committee, who then provide a review. After the revision process, patterns can be certified and published in the official catalogue.

In order for LSPs to be included in the Ontology Design Pattern Portal, we provided some templates that contain the kind of information that should be filled in whenever a new pattern is proposed for the catalogue. The templates we provided consisted of two sections: *Description* section and *Cases* section, as can be seen in figures 7.7 and 7.8, respectively.

The **Description** section provides general information of LSPs. This section is based on the templates we created for the description of LSPs in our *multilingual LSPs-ODPs pattern repository* (see chapter 5). Other sections that were not originally included in those templates for describing LSPs were added for the sake of coherence with the rest of patterns in the Portal. The Description section includes the following subsections:

- **Name**: name of the pattern according to the original repository
- **Language**: language of the pattern. E.g., English, Spanish
- **Also known as**: alternative name of the pattern
- **Intent**: description of the semantics expressed by the pattern
- **Solution description**: description of the correspondence relation between the LSPs and the ODPs
- **Description of the correspondence relation between the LSPs and the ODPs**: 1:1, 1:N, 1:pairwise disjoint N
- **Related ODPs**: Link to the corresponding ODP or ODPs within the Ontology Design Patterns Portal, or to an external publication (paper, technical document, etc.) in its default
- **Web reference**: URL of the external publications
- **Authors**: names of the pattern creators
- **Submitted by**: name of the person who submitted the pattern

[3] On October, 8th 2010, the LexicoSyntacticODPs section of the Ontology Design Patterns Portal had been accessed 895 times.

In the **Cases** section we find three additional sections:

- **NL Formulation**: prototypical expressions in natural language (NL) exemplifying the pattern
- **LSPs Formalization**: formalization of the pattern using an extended version of the BNF notation
- **Reusable JAPE code**: link to a website containing JAPE code reusable for NLP applications that support JAPE language, such as GATE. The code is included accompanying each of the individual patterns, as can be seen in figure 7.8.

Description

Name	Lexico Syntactic ODPs corresponding to SubclassOf "or" Simple Part-Whole relation ODPs
Language	English
Also known as	LSP-SC-PW-EN
Intent	Ambiguous (or polisemic) expressions in English that can either state the relation holding between a class and its sublclasses, or a whole and its parts.
Solution description	The set of Lexico-Syntactic ODPs included here have a direct correspondence to both patterns: the Logical ODP for modelling "SubclassOf relation" or the Content ODP for modelling "Simple Part-Whole" relations, as described in the Technical report D5.1.1, NeOn project Deliverable (see Web Reference below).
Description of the correspondence relation between the LSPs and the ODPs	one LSP to pair-wise disjoint ODPs
Related ODP(s)	Submissions:http://ontologydesignpatterns.org/wiki/Submissions:PartOf
Web reference	http://www.neon-project.org/web-content/images/Publications/neon_2008_d2.5.1.pdf
Author(s)	Elena Montiel-Ponsoda, Guadalupe Aguado de Cea, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez
Submitted by	ElenaMontiel-Ponsoda

Figure 7.7: Description section at Ontology Design Patterns Portal

A link is provided to the table that contains the list of abbreviations and symbols used in the patterns formalization.

Figure 7.8: Cases section at Ontology Design Patterns Portal

7.3 Evaluation

This section is devoted to the description of an experiment that we performed with a twofold purpose. On the one hand, we aimed at demonstrating the viability of the proposed methodological guides for the reuse of ODPs by novice users (see chapter 6). On the other hand, we wanted to evaluate the performance of the **LSPs application** and the JAPE rules regarding the matching of LSPs in the NL input provided by users.

We will start by describing the experiment setting. Then, we will analyze the results obtained, and will draw some conclusions.

7.3.1 Experiment Setting

The experiment was performed in one session and involved one set of participants. Participants were students from several European countries visiting the Universidad Politécnica de Madrid to attend a course on "Ontologies and the Semantic Web". The course lasted a week and was part of the European exchange schema ATHENS[4], a program that organizes courses in major technological universities

[4]http://www.athensprogramme.com

twice a year. The course on "Ontologies and the Semantic Web" took place in November 2009 and involved 17 students.

Students were asked about their background and countries of origin. Most of them were computer science students with focus on different areas such as management engineering, software engineering, information systems in civil engineering or computer engineering. Seven European universities were represented: the Czech Technical University in Prague; three French universities, the Ecole Nationale Supérieure des Techniques Avancées, Mines ParisTech, and TELECOM ParisTech; the Katholieke Universiteit Leuven in Belgium; the Politecnico di Milano in Italy, and the Warsaw University of Technology in Poland. The language used in the course was English, although none of the students had English as its mother tongue.

The objective of the course was to provide students with a theoretical and practical understanding of ontologies. At the time of the experiment, students had received a broad introduction to the Semantic Web, and had been taught on theoretical and practical aspects of ontologies and ontology languages (RDF and RDF Schema), methodologies for the development of ontologies (specifically the NeOn Methodology) and some aspects of computational linguistics (terminology and multilingualism in ontologies). By the end of the course students had to apply the lessons learned in the development of a small ontology of the *Olympic Games* domain.

We believe that this set of participants could be considered a good representative of potential users of our method for the reuse of ODPs in ontology construction. Instead of having some background on other types of modeling, all of them were newcomers to ontology engineering. Besides, they had received an introductory course to ontologies and ontological engineering, and were interested in developing an ontology.

In this context, we organized a hands-on activity, in which students were asked to formulate in NL those modeling aspects they wanted to include in an ontology of the Olympic Games domain. They were given a short presentation on the Reuse of Ontology Design Patterns (ODPs) as one of the possible scenarios for building ontologies in the framework of the NeOn Methodology. Then, they were taught about ODPs (what they are; why users are encouraged to reuse them in the ontology development process; what the difficulties involved in the matching or selection tasks are) and were shown some examples.

After that short introduction, the method we propose in this thesis for the reuse of ODPs starting from formulations in NL was presented, and some examples were provided. Finally, students were asked to write sentences in English in which they expressed what they wanted to model in the ontology.

According to the lecturers of the course, students should have already performed the Ontology Requirements Specification activity (see section 4.3 in chapter 4). As a result of that, they should have obtained the Ontology Requirements Specification Document (ORSD), which would include the set of CQs that the ontology had to address, as explained in section 4.3. However, in order to control the

output of the experiment, and also to prevent students from abandoning the experiments for not having worked on the ORSD, we provided them with a set of 21 CQs they could use for the task. The table containing the CQs used in the experiment has been included in the Appendix (figure 13.1).

Students were also provided with the Recommendations table (see table 6.1 in section 6.1). Let us recall that the aim of the recommendations is to guide users in the kind of input that is expected from them in **Task 1. NL Formulation** of the method, as explained in chapter 6.

The time assigned to the task was 20 minutes. Once students had completed the exercise, they were asked to fill in a questionnaire about the hands-on activity. The questionnaire has also been included in the Appendix (figure 13.2). There, students were asked about the difficulty of the formulation task. We were also interested in their opinion about the Recommendations table. Besides, we asked them about the usefulness of the approach for newcomers to ontology engineering, and about the convenience of the CQs for the subsequent formulation of sentences. Answers to these questions will be commented in the next section.

7.3.2 Analysis of Results

A total of 15 students participated in the experiment, and 253 sentences were produced by them. A preprocessing of the sentences was needed in the first place. This was mainly due to the fact that students were not native speakers of English, and made mistakes that needed to be corrected for an adequate processing of the sentences. Consider the examples below:

(1) *Summer Olympic Games are made by different sports*.

(2) *Sports of Summer Olympic Games are Aquatics, Athletic, Gymnastic (...) and Volleyball*.

(3) *Winner of a discipline are People*.

Sentences (1), (2) and (3) are examples of grammatically incorrect sentences that we corrected before running the application. Misspellings were also corrected. Most of the students also forgot to separate the elements of an enumeration list by commas, despite being one of the recommendations. Therefore, commas had to be added when missing, since we had already checked the importance of them for a correct annotation process.

Some sentences contained symbols that had to be removed for a correct processing (such as sentence (4)), and others were formulated in an unnatural way (see sentence (5)).

(4) *Team XY has {members} as members*.

(5) *Olympic Game is organized in Beijin or Torino or Athens or Salt Lake or Sydney or Nagamo or Atlanta*.

We also had to discard some sentences that were unfinished or unreadable. Others were discarded because they missed the point of the activity, see sentences (6) and (7):

(6) *Competitors are swimming in the swimming-pool.*

(7) *The swimming-pool is full of water.*

In total, we discarded 15 sentences out of 253, which left us with **238 correct sentences from the point of view of content and grammar**. Then, we processed the corpora contained in the 15 documents, one per participant, and run the **LSPs application**. Numbers about the resulting matched ODPs have been summarized in table 7.1. The reason for only 9 patterns being matched in the corpora, out of the total of 17 patterns included in the repository (see summarizing table 5.5) is that the knowledge they convey was not expressed in the set of CQs.

Matched ODP or ODPs	Number of matches
Subclass-of relation ODP	117
Object property ODP	21
Simple part-whole relation, constituency, componency, or collection-entity ODPs	19
Participation ODP	19
Object property, datatype property, or simple part-whole relation ODPs	12
Defined classes and subclass-of relation ODPs	1
Subclass-of relation, disjoint classes, and exhaustive classes ODPs	5
Subclass-of relation or simple part-whole relation ODPs	10
Datatype property ODP	1
TOTAL	205

Table 7.1: Number of resulting annotations from the experiment with the LSPs application

A summarized in table 7.1, a total of **205 correct matchings** were produced, which amounts to 86.2% of correct matchings. This left us with 33 wrongly annotated sentences, or sentences that could not be annotated, out of the 238 sentences used in the experiment. First, we will comment on the correct matchings, and then on the wrongly annotated sentences.

As expected in any ontology, the great majority of patterns (117 matches) corresponded to the *subclass-of relation* ODP, and other patterns with which the subclass of relation is combined, namely,

- *Subclass-of relation*, *disjoint classes*, and *exhaustive classes* ODPs (5 matches)
- *Defined classes* and *subclass-of relation* ODPs (1 match)
- *Subclass-of relation* or *simple part-whole relation* ODPs (10 matches)

We should recall that the subclass-of relation also includes those sentences in which the instances of a class are specified. Examples will be given below.

For those linguistic structures that correspond to a set of disjoint ODPs, a further disambiguation process is needed, as explained in section 6.4, chapter 6. These patterns are

- *Subclass-of relation* or *simple part-whole relation* ODPs (10 matches)
- *Simple part-whole relation*, *constituency*, *componency*, or *collection-entity* ODPs (19 matches)
- *Object property*, *datatype property*, or *simple part-whole relation* ODPs (12 matches)

In this sense, the annotations provided by the **LSPs application** only pursue to make the user aware of this ambiguity problem, and invite him or her to perform the refinement task proposed in the method (see **Task 2. Input Refinement** in section 6.1). Refinement strategies have been outlined in section 6.4, but their implementation is out of the scope of this work.

The number of structures matching the "family" of *part-whole relations* is remarkable (19 matches). The strong presence of this relation, together with the subclass-of relation, confirms the argument that these relations are two of the most important ones in any organization and classification of knowledge.

Then, 21 sentences matched the *Object property* ODP. This pattern allows modeling *ad hoc* relations of any domain of knowledge, which are also very common in any ontology. In some cases, the relation can be additionally modeled by *datatype property*, or *simple part-whole relation* ODPs, as was the case in 12 matches. Here, refinement is also needed.

Finally, we will refer to the *Participation* ODP, which is also quite numerous in this corpora, 19 matches. This pattern models the relation between participants and events, which is very relevant in the Olympic Games domain.

Wrong annotations

As already mentioned, from the 238 sentences, 205 sentences were correctly annotated, and 33 could not be annotated by the application or were wrongly annotated because of two main reasons that we will try to illustrate in the following. This amounts to 13.8% of all annotated sentences. The principal reasons for wrongly annotated sentences are the following:

- Some of the sentences expressed modeling issues that have not been considered at this stage of the work. This amounted to 18 out of the 33 wrongly annotated or not annotated sentences.
- GATE did not provide the appropriate annotation from one of the basic processing resources. This left 15 sentences unannotated.

In the following, we provide some examples of these two cases.

For example, sentence (8) is expressing an axiom that could (and probably should) be included in the ontology, but which is not dealt in the present version of the repository.

(8) *If a swimmer breaks a rule is disqualified.*

Sentences (9) and (10) show another example of sentences that could not be annotated because their structure has not been considered in the current *LSPs-ODPs pattern repository*. This linguistic structure identifies the values (*25* or *559-1367243*) of data type properties (*age*, *ID*) of an instance (*Nadia*). This should also be considered in future work.

(9) *The age of Nadia is 25 years.*

(10) *The ID-number of Nadia is 559-1367243.*

Other type of sentences that could not be annotated are the so-called *n-ary relations*, i.e., relations in which more than two classes are involved, as in sentence (11). These are also very important relations to take into account in further versions of the *LSPs-ODPs pattern repository*.

(11) *People are participants in a discipline.*

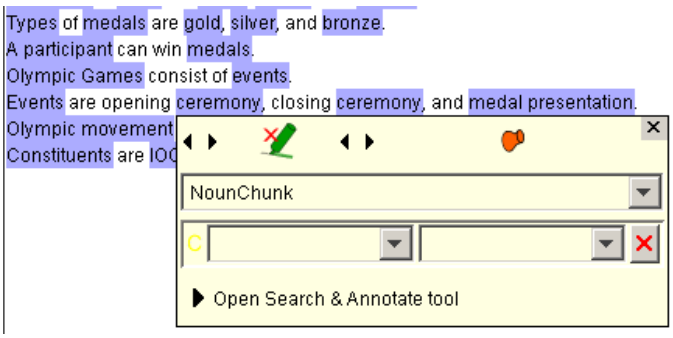


Figure 7.9: Example of a wrong annotation provided by GATE's noun chunker

Next, we will refer to those sentences that contained modal verbs, as sentence (12). From a modeling perspective, this formulation would not be valid, since the user, who is considered a domain expert, has to be certain about the fact that sports

consist of disciplines. This is an example of lack of *reliability or certainty* present in knowledge rich contexts, as explained in section 3.1.2. A recommendation in this sense should be included in the Recommendations table 6.1.

(12) *Sports may consist of disciplines*.

As already introduced, the second major reason for a sentence not being annotated is that GATE did not provide the appropriate annotation from one of the basic processing resources. Let us take for example sentence (12). As can be seen in figure 7.9, the Noun Phrase Chunker did not correctly identify *opening ceremony* and *closing ceremony* as a noun phrase. This caused for the rest of processing resources not to work properly. In fact, most of the wrong annotations provided by GATE resulted from the Noun Chunker.

(13) *Events are opening ceremony, closing ceremony, and medal presentation*.

A further problem, though not so common, was incorrect tagging. In sentence (14), *sports* is tagged as a verb instead of as a noun.

(14) *The Aquatics sports disciplines are diving, swimming, waterpolo, and synchronized swimming*.

Example of annotations

Let us now analyze in more detail the annotations obtained for the sentences provided by one of the participants, which are a representative subset of the results obtained for the whole corpora. Figure 7.10 shows the annotations generated by the **LSPs application** for the text provided by participant 7.

In the resulting annotations we can see examples of the LSP corresponding to the *subclass-of relation*, *disjoint classes* and *exhaustive classes* (see table 5.32 in chapter 5), identified by the annotations **Superclass** and **Subclass_Di_EC**. This is the case of sentences (15) and (16)

(15) *Olympic Games can be either Summer Olympic Games or Winter Olympic Games*.

(16) *The medals are either Gold, Silver, or Bronze*.

We also find examples of polysemous LSPs that need to be further disambiguated, because they are identified as corresponding to the *subclass-of relation* or *simple part-whole relation* (see tables 5.32 and 5.30). These are identified by the annotations **SuperclassWhole** and **Subclass_Di_ECPart**, as in sentences (17) and (18)

(17) *Summer Olympic Games include Aquatics, Athletics, Gymnastics, Judo, Archery, Taekwondo, Tennis, Handball, Football, and Cycling*.

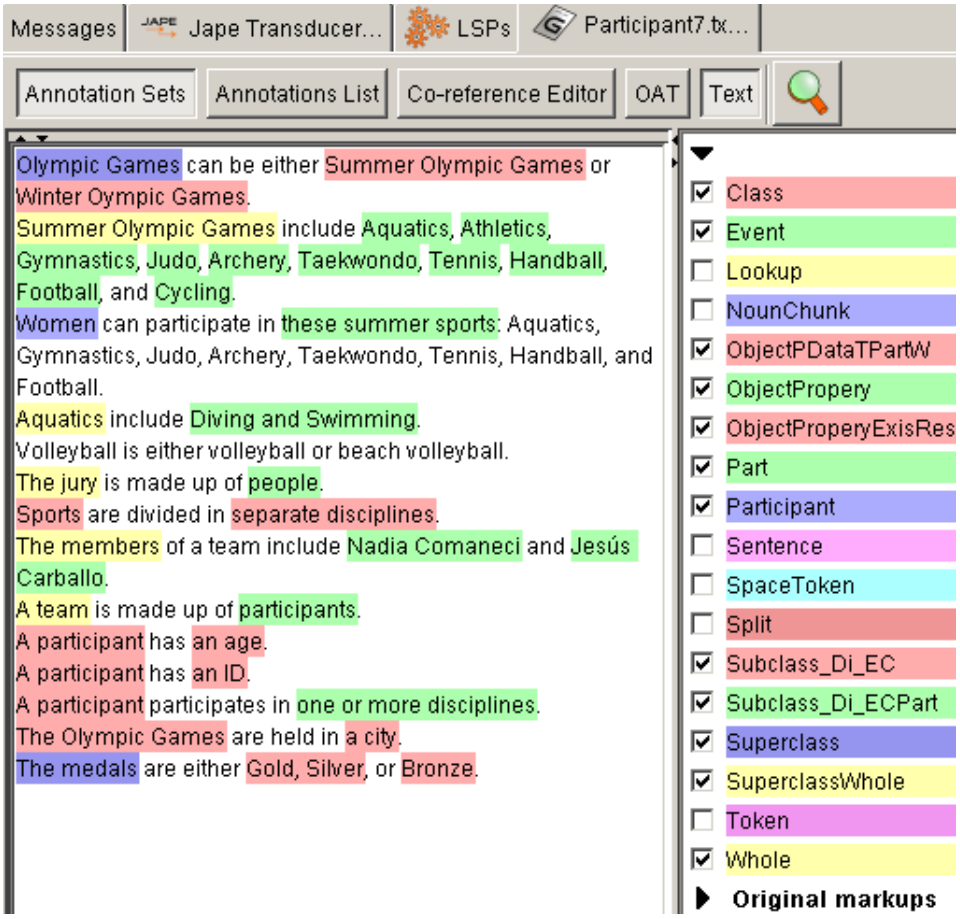


Figure 7.10: Annotations on the sentences provided by participant 7

(18) *Aquatics include Diving and Swimming*.

Regarding sentence (19), the relation expressed there is between a class and its instances. As we already mentioned in section 5.4, the identification of instances could be made by using processing resources that identify Named Entities, or by other additional strategies. However, at this stage of the work we consider this relation within the group of *subclass-of* relations.

(19) *The members of a team include Nadia Comaneci and Jesús Carballo*.

In sentences (20) and (21), the ambiguous LSP corresponding to *part-whole*, *constituency*, *componency* or *collection-entity* is identified by the annotations **Part** and **Whole**. As already described, this correspondence would also need to be further specified.

(20) *The jury is made up of people*.

(21) *A team is made up of participants*.

We also find an LSP expressing the relation between participants and events in sentence (22).

(22) *A participant participates in one or more disciplines*.

Finally, sentences (23) and (24) represent the ambiguous structure corresponding to *object property*, *datatype property*, or *part-whole relation* (see table 5.33). This is indicated by the annotations **Class** and **ObjectPDataTPartW**.

(23) *A participant has an age*.

(24) *A participant has an ID*.

Questionnaire

To conclude, we will refer to the questionnaire that students were asked to fill-in after the hands-on activity (see figure 13.2 in the Appendix). The following comments and suggestions were obtained from the questionnaire and will be taken into account in future work.

- All students agreed that CQ were very useful for the subsequent formulation of sentences. We also believe that a good set of CQs can help a lot in the rest of activities or tasks that are to be carried out in the development of the ontology. One of the students pointed out that CQs could be very useful "if carefully formulated". We already argued that CQs should be formulated by a team of domain experts and ontology engineers, and when formulated only by domain experts, they should always be checked by ontology engineers. The main reason for this is that ontology engineers can help domain experts carefully "parceling" and decomposing the domain of knowledge to the needed degree of granularity.

- All students also agreed on Recommendations being helpful, although some of them would have liked to have more examples, lists of verbs to use, or even lists of words that should not be used in sentences. We understand that people that do not have a good command of a language would feel more confident if getting lists of words or phrases that can be used. However, we did not want to restrict users too much in the kind of formulations they could produce to be consistent with the *naturalist* philosophy followed by our approach.

- Most of the students (12 out of 15) found the formulation task easy. One of them qualified if as too easy, and two students said they found it *a little bit difficult*. When asked about how the approach could be improved, some of them said that they would like to have more examples of correct sentences. One of them said that feedback on the correctness of sentences would be desired. In fact, feedback is something that the system, once implemented, should give to users.

- When asked about the appropriateness of the method for newcomers to Ontology Engineering, 13 students agreed that the approach can be very useful for novice users. Two of them were not sure about it. And one of the participants said that she was not an ontology expert and that our approach had helped her gaining an interesting insight into Ontological Engineering.

7.3.3 Concluding Remarks

The conclusions that we draw from this experiment are divided into those that are related with the results of the annotation process with the **LSPs application**, and those that refer to the approach we propose for the **reuse of ODPs by novice users**.

Regarding the results of the annotation process by the **LSPs application** we created in GATE, we think that results are very encouraging for further working in this direction. 86.2% of right annotations can be considered a satisfactory result. However, we were also made aware of some drawbacks:

1. On the design side, considerable effort is needed for the analysis of linguistic structures (LSPs) and the creation of rules (JAPE rules).

2. Sound processing resources are needed for complex applications to rely on them.

3. Grammar and content of the processed sentences have to be correct if an efficient processing is expected. In this sense, we argue that users should be allowed to carry out the formulation task in their own language. This requires for LPSs repositories to contain patterns in several natural languages.

As far as the guidelines are concerned, and considering the results of the questionnaire, we believe that users with a limited knowledge on ontology engineering can be encouraged to develop ontologies by themselves if guided by methods specifically aimed at them. The possibility of interacting with tools in full NL also contributes to bringing new technologies closer to the average user. The main benefits of the method proposed here are listed below.

1. Users are not required to have a deep knowledge on logics.

2. Users are allowed to formulate modeling issues in NL.

3. The modeling solutions provided as a result of the annotation and matching recommendation processes are based on consensual modeling solutions (ODPs), which guarantees the quality of the final ontology.

4. The method can also be regarded as a didactic approach that brings ontology engineering closer to the domain expert.

5. The implication of domain experts in the construction of ontologies contributes to their quality and subsequent adoption.

Apart from these benefits, we also need to further work on the following aspects:

1. Improvement of the Recommendations provided in **Task 1. NL Formulation**. For instance, provision of more examples of expected linguistic structures in NL.

2. Automation of refinement strategies in **Task 2. Input Refinement**, to refine LSPs and disambiguate polysemous LSPs.

3. Implementation of the system as a plug-in of an ontology editor.

7.4 Summary

In this chapter we have presented the GATE architecture for NLP, which is the framework we have used for creating our **LSPs application** to match NL formulations to the ODPs in the *English LSPs-ODPs pattern repository*. After a detailed description of the processing resources used in the LSPs application, we describe the JAPE rules that implement our LSPs for the English language.

Then, we present the templates that have been created for the publication of the LSPs-ODPs pattern repository in the Ontology Design Patterns Portal. In this way, not only the correspondence between linguistic descriptions (LSPs) and ODPs is made available for the community of ontology developers, but also the JAPE rules that can be reused in any NLP application relying on JAPE code.

After that, we describe the experiment we have conducted to evaluate both the performance of the LSPs application and the viability of the ODPs reuse method for novice users. Results are encouraging, but also suggest some enhancements. Regarding the LSPs application, the main disadvantages are related with the efforts that creating JAPE rules demands. This is not a simple task, and requires some programming practice. As for the method, positive feedback was given by the participants in the experiment, as well as some proposals for improvement.

Part II

Linguistic Information Repository for Ontology Localization

Chapter 8

Ontology Localization

This chapter introduces the second part of the PhD thesis. In the previous part we have been dealing with the interaction between natural languages and ontologies in the knowledge acquisition and ontology modeling activities. In that approach, our starting point were natural language expressions, and the pursued output an ontological representation of knowledge. In this second part of the research work, we take the ontology resulting from the modeling activity, and our aim is to provide a model, the Linguistic Information Repository or LIR, that allows for the knowledge represented in the ontology to be expressed in different natural languages. This means, we take as input the product of the modeling activity, namely, the ontology, and provide the modeling support that allows to associate multilingual information to the ontology. The activity of associating linguistic descriptions in multiple languages to an ontology for its reuse in other linguistic and cultural settings is known as ontology localization. The interaction between the different activities and components is illustrated in figure 8.1.

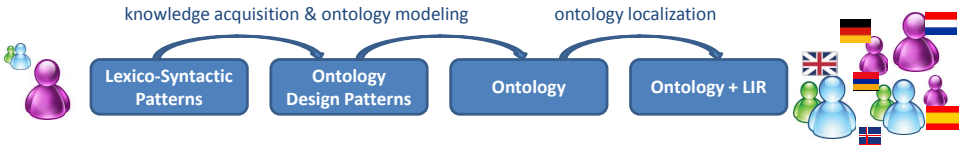


Figure 8.1: Interaction between the activities and components dealt in this thesis

In this chapter, our purpose is to define the activity of ontology localization as understood in this PhD work (section 8.1). First, we review some translation theories we draw on to address the localization issue (section 8.2). Then, we identify the different dimensions that interact in the performance of the localization activity (section 8.3). After that, we are in the position of characterizing the problem of ontology localization (section 8.4). Next, we devote section 8.5 to spelling out the different layers of the ontology that may be affected by the localization process. Finally, section 8.6 aims at giving an overview of the strategies that can be applied

to ontology localization depending on the dimensions previously identified.

8.1 Ontology Localization: Definition and Baselines

The term localization is derived from the word "locale", which traditionally means small area or vicinity, as (Esselink, 2000: 1) explains in his book *A Practical Guide to Localization*. According to the Shorter Oxford English Dictionary on Historical Principles, in the 17th century the verb "localize" was used in English with the meaning of "to act in accordance with the custom of the place (1600, G. Harvey)", which then fell into disuse, as reported in Aguado de Cea and Lorente Enseñat (1997).

The Merriam-Webster Online Dictionary currently offers a general definition of localize as "to make local" or "to orient locally". This is nowadays the most common use of the word, which has been adopted by economic and marketing theories as a response to the contrary effect of globalization. Or as (Pym, 2002) puts it, "Globalization, of course, requires both processes: English for the centralized production culture, and local languages for the marketing of products locally". In commercial settings, for example, localization is defined as the way to "adapt products for non-native environments"[1], in order to guarantee their acceptability in foreign markets. The degree of adaptation may vary from product to product. Some products may just require a translation of their instructions for use, and others may need physical modifications. An example of physical localization would be the fact that automobiles sold in Australia, the United Kingdom, India, Japan and much of southern Africa need to have their steering wheels on the right side of the vehicle.

A field in which physical modification is usually not so starkly required is web design and software. This is due to the *internationalization* activity previously undergone by the product, that consists in designing the product in such a way, that some of its parts are already prepared to be replaceable, mostly text. In this scenario, the process of localization consists in the "translation and adaptation of a software or web product, which includes the software application itself and all related product documentation" (Esselink, 2000: 1-24). According to this definition, localization can affect the "surface" of a software product (graphical user interfaces, online help, product documentation), or the actual functionality and behavior of the software to comply with the different processes and rules in place in another country.

In Ontological Engineering, the localization of ontologies could be considered as a subtype of software localization in which the product is a shared model of a particular domain, i.e., an ontology, to be used by a certain application. In this context, Ontology Localization has been defined as "the adaptation of an ontology to a particular language and culture" (M. C. Suárez-Figueroa and Gómez-Pérez, 2008). This definition has been subsequently revisited in Cimiano et al. (2010) to refer

[1] http://en.wikipedia.org/wiki/Internationalization_and_localization [Accessed in January 2009].

to "the process of adapting a given ontology to the needs of a certain community, which can be characterized by a common language, a common culture or a certain geopolitical environment".

The first definition is more general and refers to "language and culture" as a one entity, in the sense that any differences in the categorization of reality because of cultural discrepancies will be inevitably mirrored in the language[2]. The second definition puts the emphasis on the adaptation of the ontology to the needs of a target community that can be characterized (a) by speaking a different language from that of the original community of users, or (b) by speaking the same language of the original community of users, but belonging to a different culture or geopolitical environment. In any case, both definitions regard Ontology Localization as the adaptation of the ontology and its natural language documentation to the needs of the target users.

Both approaches, Software Localization and Ontology Localization, have a very pragmatical and economical orientation, since the idea is to reuse software products or ontologies already available instead of developing them from scratch. And in both approaches, the starting point is a "product" created within a certain culture and in a certain language, i.e., a monolingual product.

In Software Localization, the original product is adapted to different cultural communities and the result will be normally used independently from the original product. In Ontology Engineering, the localized ontology may be used independently from the original ontology, or it may also happen that the ontology is expected to support an application in which several natural languages need to interoperate. In the latter case, the output will be a *multilingual ontology*. But before analyzing the different functions that localized ontologies may serve, we will briefly refer to translation, and the translation theories in which our understanding of the localization activity has its roots.

8.2 Translation Theories in Ontology Localization

Translation may be considered the *mother* activity that encompasses Software and Ontology Localization. As it has been claimed in (Pym, 2002), software localization is a more general process than translation because of several aspects that are not assumed to happen in a traditional translation process. Firstly, because software localization starts right after the internationalization phase of a product. Secondly, because it may require the adaptation of some technical aspects of the product. And, finally, because the activity of software localization also involves the development of marketing support for the product. Although accepting these additional tasks, it may as well be argued that the translation activity is at the core of any localization process and that the rest of tasks are circumstantial factors due to the nature of the object to be translated. Because of this, we deem it necessary to care-

[2] See chapter 2 for an argumentation in favor of functional and cognitivist approaches to language in which this view is propounded.

fully analyze the different dimensions involved in the translation process, and draw a similar scenario for ontology localization.

Translation is an ancient activity, whose first evidences date back to the 18th century BC (Hurtado Albir, 2001: 99). Following Functionalists theories of translation, specifically Vermeer's *Skopostheorie*, translation is considered a special type of *intentional transfer*, or what is the same, a special type of action or *aktionales Handeln* (Vermeer, 1978), in which "communicative verbal and non-verbal signs are transferred from one language into another" (Nord, 1997: 11). *Actions* are regarded to take place in a situation, be part of the situation, and, at the same time, modify the situation (Vermeer, 1978). This modification of the final situation can end up in a new product, no matter if it is a text, a speech or a web page.

Functionalists put the emphasis on the fact that every translation is intended to fulfill a specific ***function*** in a specific target culture, and that the function is what guides the translator along the translation process. Furthermore, they argue that translation cannot be reduced to a word-for-word transfer, as in every translation process many aspects need to be considered, which eventually influence the translator in the decision making process (that can eventually be a word-for-word translation or not). The main factors proposed by Nord (1997: 60) are the following:

- Intention of the text - to inform, to convince, to give orders
- Target-text addressee(s) - adults, children, experts, scientists
- Time and place of the text reception - a company, a country, for one year, for a month
- Medium over which the text will be transmitted - monolingual or bilingual web pages, brochures
- Motive for the production or reception of the text - to present a new product, to teach about a new European normative, to help users manage a computer program

According to this, translators will be faithful to the source text, or free to adapt as many textual and non-textual elements as they consider necessary to achieve a certain purpose. This variety of translation decisions can be systematized in a dual typology of translations (adapted from Nord (1997: 49 ff.)):

1. Translations in which the purpose of the translation is to **document** or inform the target reader about a situation in the original culture. These translations normally reproduce form and/or content of the original document, and may result in a text with a foreign flair for the target reader, so that (s)he is conscious of the character of a translation of the text.
2. Translations in which the purpose is to produce in the target reader the same effect the original text produced in the original reader. These translations

> usually aim to achieve the same **functions** of the original. In this case, the translator may have to adapt many aspects of the text, or even change or omit facts, so that the target reader feels the text as original of his or her own culture.

Many practitioners and translation theorists agree on this difference and speak about ***overt vs. covert*** translation (House, 1977: 188 ff.), or ***documentary vs. instrumental*** translation (Nord, 1989: 47 ff.), respectively.

If we apply these theories to the localization in software industry, we may agree on the fact that the translation carried out in Software Localization is an *instrumental translation* in which the aim is to offer target users a product that fits in their knowledge structures and addresses their expectancies, as happened in the case of the original users. As already outlined, this may involve pure textual translation or also the adaptation of some technical aspects or processes. In the next section, we will try to extrapolate these considerations to the localization of ontologies.

8.3 Dimensions in Ontology Localization: Function and Domain Type

As in any process involving translation or transfer from one culture to another, the **purpose** or **function** of the output will guide the localization process. The same principle applies to the localization of ontologies, in which the ontology might be localized with different goals in mind.

Function

On the one hand, the goal of the localized ontology may be to fulfill the same function in the target community as the original ontology had in the source community. Let us imagine that we have an ontology of fish species in English that has been used by an application to index documents reporting about the situation of the fishery stock in British waters. If we now want to do the same but for documents in Spanish reporting about Spanish waters, we will need to localize the ontology for the Spanish culture taking into account that certain categories may be differently understood or that certain species may have to be added or removed. In any case, we would be applying certain strategies required by the ***instrumental*** purpose of the localization activity.

In the ***documentary*** localization, on the other hand, the purpose is to support the use of the original ontology by members of another community. This means that the localized ontology will be documented in the language of the target culture but will not be used in an equivalent situation in the target community, but in a different one. If we take the example above, our ontology of fish species in English would be localized into Spanish to annotate documents in Spanish talking about

the situation of the fishery stock in British waters (and not in Spanish ones). This means that the localized ontology and the original one are not used in equivalent situations in their respective target cultures, but the localized ontology in Spanish is used to talk about the original culture. The translation decisions taken in the previous case might not be appropriate in this situation.

Domain Type

The implications that each of the localization approaches will have in the actual localization process are quite different, and will affect different layers in the ontology. But before moving to representational aspects of the localization of ontologies, we still need to refer to a further dimension involved in this process: the type of domain being represented in the ontology. Here, we make a distinction between ***internationalized*** or ***standardized*** domains, and domains more prone to reproduce the vision of the world of a certain community, the so-called, ***culturally-influenced*** domains.

As already introduced in chapter 2, by ***internationalized*** or ***standardized*** domains we understand those technical or specialized domains of knowledge in areas such as engineering, economy, or medicine that have standards for processes and descriptions, and whose categorizations usually reflect the common view of all the cultures represented in the localization project. These could be even deemed as *artifactual* domains of knowledge.

On the contrary, what we have called ***culturally-influenced*** domains refer to those domains of knowledge dealing with "anthropological matters" and that tend to be conceptualized in a different manner by different groups of people that share a certain vision of the world. For instance, under this group we include the judicature, geography or the political and administrative organization of countries, universities, and so on.

Given these two dimensions of the localization activity, namely, the **function** of the final ontology and the **domain type**, four combining possibilities arise, as illustrated in table 8.1:

- The function of the final ontology is to *document* the ontology in the target culture, and the ontology represents an *internationalized* domain (n.a.).
- The function is *instrumental*, and the ontology represents an *internationalized* domain (Use Case 1).
- The function of the final ontology is to *document* the original ontology so that it can be used by members of the target community, and we are dealing with a *culturally-influenced* domain (Use Case 2).
- The function of the final ontology is to be used in an equivalent situation in the target culture, i.e., *instrumental* function, and the domain type is a *culturally-influenced* domain (Use Case 3).

Function / Domain Type	Internationalized	Culturally-influenced
Documentary	n.a.	Use Case 2
Instrumental	Use Case 1	Use Case 3

Table 8.1: Combination options between function and domain type

In the case of an *internationalized* domain, there would be no differences between using the ontology in an equivalent situation (*instrumental* function) or in a different one (*documentary* function), and, therefore, we could consider them as the same case. This is why we regard the case of an internationalized domain and documentary function as not applicable in table 8.1, and merge both cases in **Use Case 1**. Then, with the aim of exemplifying the other two cases, we include some examples of real use cases (**Use Case 2** and **Use Case 3**) of localization projects in which these dimensions are decisive in determining the translation strategies to be employed, and how to represent multilingualism in ontologies.

Use Case 1: GenomaKB. In the GenomaKB project[3] (Cabré et al., 2004), terminology experts of the Institute of Applied Linguistics at the Universitat Pompeu Fabra in Barcelona, Spain, created a biomedical knowledge base of the human genome in three languages (Spanish, English and Catalan) to assist terminologists, translators and scientific journalists working in this domain.

The starting point was a knowledge base modeling the domain of genomics with links to three further modules on terminological, textual and factographic information. Domain experts from the three linguistic communities worked together to come up with a common and consensual conceptualization of the domain. Once the ontology was stable, its concepts were linked to the terms in English, Spanish and Catalan, and stored in the terminological module. Here the conceptualization is a good example of what we understand as an *internationalized domain*, reflecting the common view of all the cultures represented in the project. Regarding the functionality aspect, it is regarded as *instrumental*, since the three versions of the ontology are to be used in equivalent situations.

Use Case 2: New to Holland. The New to Holland project website[4] concerns an ontology driven application developed by the BeInformed[5] company for the Dutch government on informing immigrants, e.g., on the process of applying for an immigration permit.

The underlying conceptualization of the New to Holland ontology reflects certain specific characteristics of Dutch immigration procedures that need to be localized into several other languages. In this scenario, the ontology is modeling what we have called a *culturally-influenced* domain and the purpose of localization is to

[3] http://genoma.iula.upf.edu:8080/genoma/index.jsp
[4] http://www.newtoholland.nl
[5] http://www.beinformed.nl

document specifics of Dutch administration services into several other languages. This is therefore clearly a case of localization for documentary purposes, i.e., for the purpose of explaining the meaning of concepts and procedures in the language of target users of applications that build on the adapted ontology.

Use Case 3: WordNet related projects (EuroWordNet[6][7], Meaning[8], GlobalWordNet[9], Kyoto[10]). The different projects that have been running since the beginnings of the EuroWordNet project for linking WordNets in different languages to the Princenton English WordNet (Miller, 1990; Miller et al., 1999), are a good exponent of the difficulties in building interoperable multilingual lexicons.

Although WordNet cannot be considered an ontology in a strict sense, we believe that these projects better reflect the difficulties of having to perform a *functional* localization of a *culturally-influenced domain* represented here by general lexicons in different target languages. The objective of each lexicon is to capture the specificities and particularities of each language, thus its characterization as *culturally-influenced* domain, and the resulting lexicons are to be used in equivalent situations in their respective target cultures, i.e., with an *instrumental* function.

Finally, the last dimension worth mentioning is *interoperability*. Interoperability has to do with the exchange of data between the localized ontology and the original one. If the different versions of the ontology are expected to be used in a multilingual environment some concessions will be made in favor of interoperability. Concessions could be related with the fact of representing in the ontology only those concepts that are common to all cultures, and leaving aside those that are specific of one culture. However, if the localized ontology is to be used as an independent ontology, then the needs of the target culture or the final function of the ontology will take priority over interoperability with the original ontology. This issue will be given more attention in section 8.6.

8.4 Characterization of the Localization Problem in Ontologies

When dealing with what we have called *internationalized* domains, the strategies to be followed in the localization activity are usually restricted to the search for term equivalents in the target culture. The concept of equivalence in translation, though being of central importance, has been the source of much controversy among translation scholars (for more on this see Hurtado Albir (2001: 203)).

[6] http://www.illc.uva.nl/EuroWordNet

[7] We refer the interested reader to (Vossen, 2004) for a detailed description of the EuroWordNet project.

[8] http://www.lsi.upc.edu/ñlp/meaning/

[9] http://www.globalwordnet.org/

[10] http://www.kyoto-project.eu/

Finding equivalences among languages and cultures is seen as a tough task, and most of the investigations on translation try to avoid that term or replace it by others such as translation *adequacy* (Reiss and Vermeer, 1984: 124 ff.), (Nord, 1997: 34 ff.). The reason for this may be that for some time *equivalence* was related to the similarities between linguistic structures in different languages. Later on, the notions of function and context gained relevance and showed that two linguistic structures that could be said equivalent in one specific context, would not be necessarily equivalent in other contexts. In spite of this, we argue that the notion of equivalence in ontology localization is appropriate because it designates concepts or "visions of the world" that are shared among different cultures, no matter how different the linguistic structures that express them in each language are. In this sense, looking for equivalents in *internationalized* domains is quite straightforward because the same category (or a very similar one) exists in the target culture.

Unlike internationalized domains, *culturally-influenced domains* of knowledge pose major problems in the search for equivalents. This does not mean that the two cultures are looking at a different reality, or that one culture cannot understand how the other categorizes reality, it is just that some aspects or features of reality are more relevant for certain cultures and go unnoticed for others. This basically derives in a categorization mismatch, i.e., in an inexact correspondence between the categorization of reality that two cultures make.

For the purposes of this research, we classify categorization relations in the following way:

1. near-equivalence relation
2. subsumption relation
3. many-to-many equivalence relation

We try to illustrate these cases with some examples of categorizations of the *hydrography* domain representing the French and North-American culture[11].

Near-equivalence relation

This would suggest that when two cultures share the same vision of the world, they categorize reality with the same granularity level. This is then normally reflected in the language by a word or term for designating that same concept. In figure 8.2, we illustrate the subclass-of relation holding between watercourses and natural channels. In this example, *watercouse* in the North-American culture and *course d'eau* in French would be understood as equivalent categories.

According to the guidelines for the development of multilingual thesauri in (*ISO 5964:1985 - Documentation - Guidelines for the establishment and development of multilingual thesauri*, 1985), we could talk about *exact equivalents*, but

[11] This categorizations may be approximate but have been included for the sake of illustration.

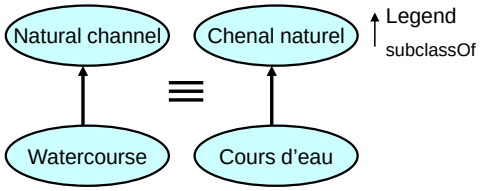


Figure 8.2: Example of near-equivalence relation between concepts

we prefer the use of the term *near-equivalents*, as also proposed in that document, because as observed in (Edmonds and Hirst, 2002) "identical meaning is rarely the case".

Subsumption relation

In this case, the target culture makes a more fine-grained or coarse-grained distinction of a certain reality that does not correlate with the granularity level of the categorization made in the original ontology. That normally derives in one of the cultures having one term for designating one concept, whereas in the other culture several terms are available to designate more fine-grained concepts.

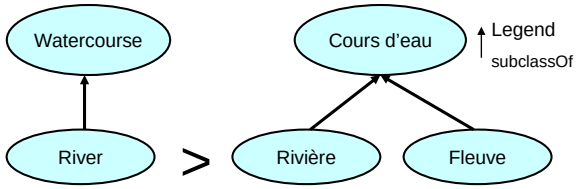


Figure 8.3: Example of subsumption relation among concepts

In (*ISO 5964:1985 - Documentation - Guidelines for the establishment and development of multilingual thesauri*, 1985), they talk about *partial equivalence* to refer to the existence of only one term in each language, in which one of them has "a slightly broader or narrower meaning than the preferred term in the other language". According to our experience, this situation usually implies the existence of several terms in the culture that understands that concept with a higher granularity level.

In figure 8.3, we depict a subsumption relation between the category representing *river* in a North-American conceptualization, and the more fine-grained distinction that the French culture makes of that category.

Many-to-many equivalence relation

This categorization situation refers to the relation among terms in one culture corresponding to an unequal number of equivalents in the other culture, i.e., a

many-to-many equivalence relation.

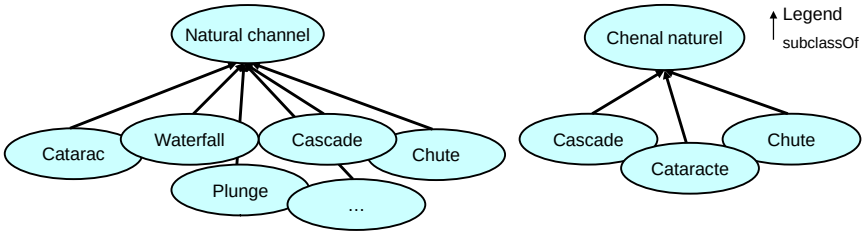


Figure 8.4: Example of many-to-many equivalence relation among concepts

Figure 8.4 shows a categorization mismatch between the North-American understanding of waterfalls, and the French one. It can be observed that the North-American categorization is more fine-grained than the French one, which only makes a distinction between three categories. This corresponds to a many-to-many equivalence relation, in which the North-American culture makes a finer-grained distinction of waterfalls.

At this stage, it should also be noted that even when two cultures share the same language, this does not mean that they share the same vision of the world, which is why we prefer to talk about cultures. If we take for example the Spanish language, which is spread over many countries, we immediately spot differences in the categorization and the vocabulary used to refer to categories (e.g. *piscina* in Spanish for swimming pool, and *alberca* in Mexican). For this reason, when reusing and ontology in Spanish, we will have to check in which Spanish-speaking culture the ontology is going to be reused to carry out all the necessary adaptations.

8.5 Ontology Layers involved in the Localization Activity

Before dealing with the different strategies that can be followed in ontology localization, we need to refer to the different layers in which an ontology can be divided. Depending on the dimensions defined in section 8.3, i.e., localization function, domain type and interoperability, the strategies followed to perform the localization of an ontology will affect some ontology layers or others.

According to (Barrasa, 2007), the following division of ontology layers can be made:

I. Lexical layer: characters and symbols that make up the syntax (ASCII encoding, UNICODE, etc.)

II. Syntactic layer: structure of characters and symbols, i.e., the grammar. It embraces different representation languages (e.g. RDF(S)[12], OWL, etc.)

[12]RDF(S) stands for Resource Description Framework Schema, and it is a knowledge representation language for the authoring of ontologies, also endorsed by the W3C (see note 9).

III. Representation paradigm layer: paradigm followed in the representation of the ontology (frames, semantic networks, Description Logics, etc.) that allows for certain ways of expressing and structuring knowledge

IV. Terminological layer: terms or labels selected to name ontology elements

V. Conceptual layer: related to conceptualization decisions, such as granularity, expressiveness, perspective, etc.

VI. Pragmatic layer: final layout of the model according to the user's needs

Taking Barrasa's classification, we may state that the *terminological*, *conceptual*, and *pragmatic* layers are the ones that will most probably be involved in the Ontology Localization activity.

- The *terminological* layer plays a decisive role in the localization activity since it refers to the vocabulary or the names we give to the different ontology elements. As a result of this activity, ontology labels will be expressed in a different natural language, or new labels may be assigned in the same natural language.
- Regarding the *conceptual* layer, certain ontologies may require the adaptation of their conceptual structure in order to fit in the thoughts of reference of a specific cultural community, as already illustrated in section 8.4.
- As for the *pragmatic* layer, the needs of the final application will determine the type and quantity of linguistic information that is to be related to the ontology.

Since our objective is to generalize as much as possible and offer a broad picture of the ontology localization activity, we will not focus on any specific application, and therefore, we will not consider the *pragmatic* layer in our analysis. The rest of the layers -*lexical*, *syntactic* and *representation paradigm* layers- should not be so strongly affected by the localization activity, and will also be left aside.

As a result of that, we will focus on two ontology layers: the **terminological layer** and the **conceptual layer**, and will try to describe the interaction between them. Once we have assumed that semantics cannot be separated from syntax, it is not hard to imagine that changes to one layer will inevitably affect the other one. If we refer to this in terms of changes to the conceptualization layer, these will be inevitably reflected at the terminological layer. However, modifications in the conceptualization layer will only make sense in *culturally-influenced* domains, whereas *internationalized* domains only require modifications in the terminological layer with no impact on the conceptualization, since the latter is valid and shared among the cultures involved. Let us illustrate this layer interaction with some examples.

Consider an ontology about political functions and offices. Most democratic systems distinguish for example the role of *head of government* in the sense of

head of the executive power vs. the role of *head of state* with mainly representative function.

An ontology designed to model political functions and offices in Germany would further distinguish between the *Bundeskanzler (federal chancellor)* playing the role of the *head of government* and the *Bundespräsident* playing the role of the *head of state*. If we want to use this ontology about political officers engineered for the German culture in applications that concern (also) other countries, e.g., the UK or Spain, we will need to adapt the conceptualization expressed by the ontology. In the case of the UK, we would introduce the class of *prime minister* as *head of government* and the *queen* as *head of state*. In the case of Spain, we would introduce the class of *presidente* (president) as *head of government* and the *monarca* (monarch) as *head of state*. While one could argue that this adaptation can also be achieved at the terminological layer, e.g., by adding additional labels *prime minister*, *presidente* for the class *Bundeskanzler*, or *queen*, *monarca* for the class *Bundespräsident*, this is clearly insufficient as these concepts will have different extensions and even intensions. In this case, adapting to a different cultural reality may require further adaptations of the underlying conceptualization.

It is important to emphasize that adapting the conceptualization layer will be primarily driven by the inexistence of conceptual equivalents (or concepts with the same granularity level) in the target community, whenever the final purpose of the ontology is to be equally valid in source and target culture, i.e., when the function of the localization activity is *instrumental*.

If the concept of *Bundeskanzler* serves the function of head of government in the German culture, and we aim at reusing the ontology in the British language, we should not translate it as *federal chancellor*, just because the word exists in the English language, unless the purpose of the localization is to *document* in English how the German political structure is organized.

In the next section we try to systematize the different strategies than can be followed in the localization of ontologies taking into account the dimensions identified in section 8.3.

8.6 Translation Strategies in Ontology Localization

According to the dimensions identified in section 8.3, the activity of Ontology Localization was divided in three use cases depending on the domain type and the function of the localized ontology. For each of these use cases, different strategies will be available involving different ontology layers.

First Use Case. The function of the final ontology is to be used in an equivalent situation in the target culture, i.e., **instrumental** function, and the domain type is a **culturally-influenced** domain. In this case we will probably come across categorization mismatches that will have to be solved.

If the original ontology makes a more coarse-grained categorization than the

target culture, we can account for the more fine-grained distinctions at the terminological layer by including the several terms that are subsumed by the original concept. However, we may as well decide to include those more fine-grained distinctions in the conceptualization layer, and modify in this way the original conceptualization by performing a re-engineering process.

One further dimension will be determinant in this sense: interoperability according to the needs of the final application. Does the localized ontology need to interoperate with the original ontology? If each ontology is to be used independently, the target ontology can undergo as many modifications as needed. In this case, this means that the more fine-grained distinctions could be captured at the conceptualization layer. If original and localized ontologies are required to interoperate and work as a multilingual ontology, some concessions will have to be made for the sake of interoperability. This could be interpreted as maintaining the more coarse-grained categorization at the conceptualization layer, and reflect the categorization discrepancies at the terminological layer.

If the original ontology were to reflect a more fine-grained categorization, the strategy would be the same, but exactly the other way round. This means that in the terminological layer the several terms identifying a more fine-grained categorization would be subsumed by a more general term, in case of having interoperability as one of the needs of the final application. And in case that interoperability would not be an issue, the localized ontology could remove the fine-grained distinctions from the conceptualization layer.

Second Use Case. In the second use case, the function of the final ontology is to **document** the original ontology so that it can be used by members of the target community, and the ontology represents a **culturally-influenced** domain. In such a use case, no modifications of the conceptualization layer would apply, and categorization discrepancies would have to be explained for the target culture at the terminological layer.

The first and second use cases have been illustrated in section 8.5 with examples of an ontology about political officers. Thus, taking the same ontology of German political officers as input of the localization activity, different strategies would be followed according to the final function and the needs for interoperability.

Third Use Case. In the third use case, the input is an ontology representing an **internationalized** or **standardized** domain, and in this case no modifications of the conceptualization layer are needed since the same categorization or vision of the world is shared. Here the strategy is to modify the terminological layer to account for the equivalents in the target culture. Interoperability is not an issue either.

8.7 Summary

In this chapter, we have firstly defined the concept of localization, offering a briefly account of its etymology, and ending up with the definition of Ontology Localization. Then we also provide a short account of the translation theories we draw on to approach the ontology localization problem. This allows us to identify the two main dimensions that are involved in any translation and/or localization process, and which determine the type of strategies to use in each case. We are referring here to the function of the localization (instrumental vs. documentary) and the domain type of the ontology (internationalized vs. culturally-influenced). We also provide some examples of localization projects involving different types of dimensions to better illustrate all possible cases.

Then, we characterize the localization problem in ontologies, which is mainly related with categorization mismatches between or among cultures. In this context, we provide a classification of categorization relations that, to the best of our knowledge, would account for most cases.

The next step is to find out how the different ontology layers are influenced by the localization activity. In this sense, we devise some strategies to solve localization issues that involve the conceptual and/or terminological layers in an ontology.

Chapter 9

Modeling Multilingualism in Ontologies

In the previous chapter, we characterized the problem of localizing ontologies by spelling out the three dimensions that may play a role in that activity, namely, function, domain type, and interoperability requirements. In the present chapter we want to analyze the state of the art on models or formalisms to represent multilingual information in ontologies. The ultimate goal of such an analysis is to find out the main strengths and drawbacks of each modeling modality, to propose a model to store multilingual information in ontologies.

Once we have characterized the problem of Ontology Localization and have outlined the different strategies that can be employed in each scenario, our aim is to analyze the current state of the art on models to represent multilingual information in ontologies. If the original ontology and the localized one are not required to interoperate, the resulting ontology will behave as a monolingual ontology, and the way of storing the linguistic information will probably coincide with that of the original ontology. However, if the original and localized ontology need to interoperate, we will be dealing with a multilingual ontology. In the state of the art we have identified three modeling approaches that permit to account for multilingualism in ontologies, and our aim is to analyze the strengths and drawbacks of each one to identify open research problems and work assumptions.

Up to now, the number of multilingual ontologies is still quite small compared to the total amount of ontologies available in the Web. The Semantic Web search engine Watson[1] provides some data about the language of ontology labels, and says that around 80% of ontologies in the Web have literals in English[2]. We identify three main ways of obtaining a multilingual ontology, depending on the layer(s) involved in the Localization Activity (Aguado de Cea et al., 2007), (Montiel-Ponsoda et al., 2010):

- **Including multilingual labels in the ontology**: this implies localization at

[1] http://kmi-web05.open.ac.uk/WatsonWUI/

[2] See http://watson.kmi.open.ac.uk/blog/2007/11/20/1195580640000.html

the terminological layer (see section 8.4), and the ontology conceptualization remains unmodified. Linguistic information in multiple languages is included in the ontology.

- **Combining the ontology with a mapping model**: this allows localization at the conceptual layer since conceptualizations in different languages are mapped to each other. Linguistic information is also included in the ontology. The mappings establish links or equivalence relations among the various conceptualizations.

- **Associating the ontology with an external linguistic model**: localization is performed at the terminological layer, which is now represented by a complex external model that stores linguistic information. Conceptual layer adaptations are also foreseen.

The appropriateness of each approach will be principally determined by the domain type of the ontology and the final function of the resulting ontology. We will only take into account those ontologies in which the different linguistic versions need to interoperate resulting in a multilingual system.

9.1 Including Multilingual Labels in the Ontology

Including multilingual labels in the ontology is the most widespread modeling option within the ontological community nowadays, because it is well supported by the most popular ontology development languages: RDF(S) and OWL. It consists of making use of the labeling functionality of RDF(S) and OWL ontology representation languages[3]. This mainly relies on two RDF(S) properties, `rdfs:label` and `rdfs:comment` that permit to associate word forms and descriptions to ontology elements. The language of labels and definitions can also be specified using the "language tagging" facility of RDF literals (e.g., xml:lang="es"). In the following, we include an example of the ontology code for the class *Rio* (Ontology1175677975;Rio), in which two labels (*Río* and River) and one comment in Spanish are associated to the ontology class.

```
   <owl:Class rdf:about= "&Ontology1175677975;Rio">
<rdfs:label xml:lang="es">Río</rdfs:label>
<rdfs:label xml:lang="en">River</rdfs:label>
<rdfs:comment xml:lang="es">Masa de agua continental que
fluye en su mayor parte sobre la
superficie del suelo</rdfs:comment>
```

[3]Properties of the RDF Schema vocabulary, as recommended by the W3C consortium (http://www.w3.org/TR/rdf-schema/).

These RDF(S) properties can be complemented by Dublin Core metadata[4] that have been created to describe resources of information systems. Examples of the Dublin Core Metadata elements are: title, creator, subject, source or description. Figure 12.5 shows how this is visualized in the ontology editor Protégé.

Disadvantages: All annotations are referred to the ontology element they are attached to, but it is not possible to define any relation among the linguistic annotations themselves (e.g., saying that one is synonym or translation of the other). This results in a bunch of unrelated data whose motivation is difficult to understand even for a human user.

When different labels in the same language are attached to the same ontology element, absolute synonym or exact equivalence is assumed among the labels. As reported in (Edmonds and Hirst, 2002) "identical meaning" among linguistic synonyms is rarely the case. It could be argued that in technical or specialized domains, absolute synonymy exists, but even in those domains, labels usually differ in "denotation, connotation, implicature, emphasis or register (Dimarco et al., 1993), what sometimes is reflected in the subcategorization frames they select (syntactic arguments they co-occur with).

A similar situation arises when labels in different languages are attached to the same ontology element. In some cases, they will share the common meaning represented by the ontology element, for example in *internationalized* domains. However, the problem appears when a language understands a certain concept with a different granularity level to the one represented by the ontology concept, as may happen in *culturally-influenced* domains. In this case, if more fine-grained equivalents exist in one of the languages represented by several labels, it is not possible to make those differences explicit in the model for a suitable treatment of multilingualism.

Finally, scalability issues should also be mentioned. If only labels in different languages are needed, the RDF(S) properties can suffice. But if additional linguistic information is needed, such as several labels, comments and further annotations in different languages, linguistic information would become unmanageable since no relations can be established among the different annotations.

Advantages: Labels can be integrated in the ontology in as many languages as the user wishes. In the same sense, by making use of additional annotations (such as those provided by Dublin Core), further information can also be included to document the ontology in natural language.

This system allows localization at the terminological layer, as labels for ontology classes can be expressed in various natural languages (see Figure 9.1 for a simplified representation). This model has proved more suitable for *internationalized* or *standardized* domains of knowledge in which the function of the Localization Activity is *instrumental* (Use Case 1 in section 8.2). This implies that no catego-

[4] http://dublincore.org

rization mismatches will require documentation in the ontology, nor modifications of the conceptualization layer. Nevertheless, linguistic information would have to be restricted to labels and comments. If a richer amount of linguistic description is needed, we should look for a different option (see section 9.3).

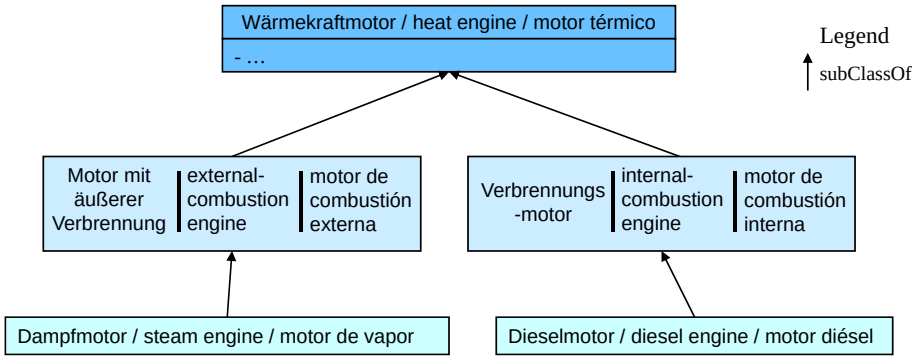


Figure 9.1: Multilingual labels included in the ontology

9.2 Combining the Ontology with a Mapping Model

This approach assumes the existence of an original ontology and one or several ontologies localized to different natural languages, all of them represented as independent ontologies. The localized (monolingual) ontologies may have been obtained after performing the localization activity on the original ontology. A further scenario may consider the possibility of having a set of ontologies in different languages on the same domain and of similar extension. According to this approach, there are various modeling ways depending on the mapping arity and the graph form. The two main representation forms are:

- Binary mappings in an orthogonal graph. In this case, each monolingual ontology organizes knowledge of a certain culture, and is mapped to the rest of ontologies in a pair-wise fashion.

- Binary mappings in a radial graph. In this option, monolingual ontologies are mapped to each other through an interlingua consisting of a set of common concepts that allow establishing equivalences. See Figure 9.2 for a highly simplified representation of this modeling modality.

Disadvantages: The establishment of mappings or alignments among conceptualizations in different languages is by no means trivial, since mismatches arise due to each conceptualization capturing the cultural specificities of each language. Regarding the quantity of linguistic information embedded in the ontology, it is often limited to labels and definitions associated with ontology classes making use of the RDF(S) properties. In this sense, this representation would have the limitations

of the model presented in section 9.1

Advantages: This option enables independent conceptualizations in each language, what may better capture the specificities of each culture.

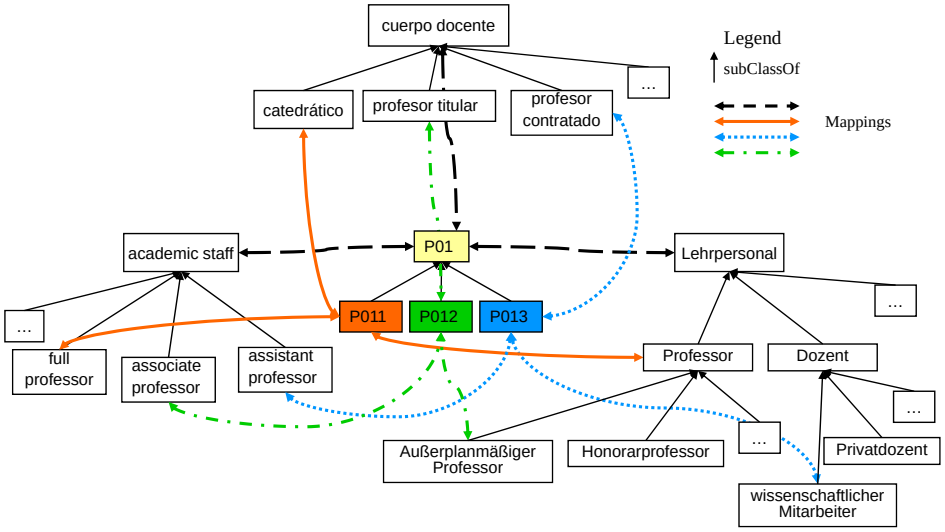


Figure 9.2: Binary mapping in a radial graph

As already introduced in the use case description in section 8.3, one of the most representative applications following this approach is EuroWordNet (EWN) (Vossen, 1998) and the rest of follow-ups of this project, namely, GlobalWordNet[5], Meaning (Atserias et al., 2004), and Kyoto (Vossen et al., 2008). However, the interest of mapping or aligning ontologies documented in different natural languages following this approach is increasing as reported in (Euzenat et al., 2009). The main motivation behind this is the existence of ontologies in different languages describing the same domain of knowledge, and the need for interoperability among them.

The EWN multilingual general lexicon consists of monolingual wordnets, each one reflecting the linguistic and cultural specificities of a certain language, linked to each other through an interlingual set of common concepts that caters for equivalences among ontologies. As already mentioned, the crucial issue in the development of such multilingual models is the establishment of mappings among concepts in the different conceptualizations. Being aware of this problem, EWN developers took as starting point the English wordnet (WordNet1.5) (Fellbaum, 1998; Miller et al., 1999) in order to guarantee a minimal level of compatibility between the independent wordnets. The risk of this option, as the same authors anticipated, was that the resulting conceptualizations could be biased by the English one (Vossen,

[5]http://www.globalwordnet.org/

2004).

This model allows localization at the conceptual layer. It would not make sense in the case of *internationalized* domains, because the conceptualization would be exactly the same (unless the different conceptualizations were already available). In the case of *culturally-influenced* domains, having one conceptualization per culture would allow to better capture the *vision of the world* that each culture makes of the same reality. The key element would be the mappings that link categories in different languages. These should be powerful enough so as to capture the relations of (near-)equivalence, subsumption and many-to-many equivalence existing among the different categorizations (as identified in section 8.4). This mapping technology is already available and has been applied to automatically aligning ontologies in the same language (Euzenat et al., 2009). Experiments are starting to be made with ontologies in different languages Fu et al. (2010), although it still represents an open research question.

9.3 Associating the Ontology with an External Linguistic Model

In this modeling option, the elements of the ontology have links to linguistic data stored outside the ontology (see Figure 9.3). This allows a separation of conceptual and terminological layers, and the localization activity is mainly carried out at the terminological layer. However, the ontology conceptualization layer can also undergo modifications, such as the creation of language specific ontology modules, if so required by the final application. The linguistic needs of the final task or application will determine the quantity and type of linguistic information to be captured in the external model, which represents the terminological layer.

Disadvantages: Since there is just one conceptualization, this model is not as flexible as the one described in section 9.2, in which cultural specificities were captured at the conceptual layer (despite the limitations imposed by interoperability and mapping discovery). This means that cultural specificities have to be accounted for at the terminological layer.

Advantages: This type of representation allows the enrichment of domain ontologies with linguistically rich and complex models. Since these are external portable models, they can be associated to any domain ontology and published with them. Explicit links can be established among the different linguistic categories that compose the models. In this sense, it is possible to build links between lexicalizations, senses, definitions, provenance sources, and so on. Regarding conceptualization mismatches between languages, these can be explicitly captured in the external model. If additional linguistic information is required by the final application, the models can be extended with further linguistic classes or by interoperating and navigating other models.

This modeling modality is appropriate for *culturally-influence* domains in which the ontology localization activity has not only an *instrumental* function, but also a *documentary* one. In localization processes with instrumental function, cultural discrepancies can be captured at the terminological layer, since an external model would permit the inclusion of as much linguistic information as needed. And, in the same way, the terminological layer would have to account for all the necessary explanations and paraphrases in case of a documentary function.

Finally, an additional feature needs to be emphasized, namely, the possibility for terminologists, translators or linguists to work on the linguistic model without interfering with the ontology model.

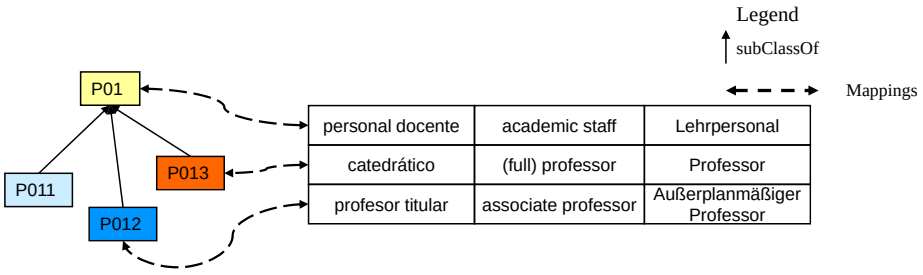


Figure 9.3: Ontology associated with external linguistic model

In the literature, this idea has been realized in the *Termontography* framework (Kerremans and Temmerman, 2004; Kerremans et al., 2004), which provides methodological guidelines for the development of ontologies in a multilingual and multicultural scenario. This approach proposes to capture culture-independent categories or concepts in the ontology, and describe culture differences in a culture-specific layer. However, they do not provide a model for implementing the so-called *cultural-specific layer*. Moreover, this approach assumes the *internationalization* of the ontology being build, as in the case of software products. This means that *only* those concepts that are common to all the cultures involved in the localization project will be captured in the ontology.

We also find some models created with the objective of linguistically enriching ontologies, although conceived for different purposes. The distinguishing aspect among them is determined by the kind of linguistic classes that make up the models. In this sense we find two main trends: (a) models whose aim is to document ontologies in different languages with multilingual or translational purposes, and (b) models that try to account for the morpho-syntactic realizations of ontology classes and relations in language. The models we are referring to are OntoTerm, on the one hand, and the LexInfo family of models, on the other.

OntoTerm[6] is a terminological management system that allows to associate linguistic information stored in a data base to ontology concepts. Two applica-

[6] http://ontoterm.com

tions that have been built on OntoTerm are *The Human Genome Knowledge Base* GENOMA-KB[7] and OncoTerm[8] (see also Use Case 2, in section 8.4). Both applications pursue terminological or translational objectives by linking a terminological multilingual database to highly specialized ontologies of the biology and oncology domains, respectively. The lexical and terminological information associated to ontology elements relates to terms and definitions in different languages accompanied by basic morphological information (part of speech, gender and number), and illustrative examples of sentences in which the terms appear.

Then, we have to refer to LingInfo (Buitelaar, Declerck, et al., 2006), (Buitelaar, Sintek, and Kiesel, 2006), LexOnto (Cimiano et al., 2007) and its merged version: LexInfo (Buitelaar et al., 2009). These three models capture linguistic descriptions in an ontology, and have been thought to be associated to arbitrary domain ontologies. The LingInfo model focuses on the representation of the morphological and syntactic structures (segments, head, modifiers) of a term. LexOnto goes one step further in that it pursues to represent linguistic realizations of ontology elements. This model builds on the notion of subcategorization frames, i.e., linguistic predicate-argument structures that represent how an ontology label (noun, adjective or verb) is syntactically realized in a certain linguistic structure. These models have been designed with the aim of improving tasks such as ontology learning or ontology population from text, and this has determined the set of linguistic information captured in the model. In chapter 10, we will analyze these models in more detail, focusing on those aspects that could contribute to the localization of ontologies.

To sum up, we include a figure that illustrates the appropriateness of the modeling options analyzed so far according to domain type and function of the localization activity, the two dimensions that were described in section 8.3. As has been represented, the third modeling option, *Associating the ontology model with an external linguistic model*, is the most flexible one because it permits ontologies to interact with complex models of linguistic information. Depending on the kind of information captured in the external model, it is able to account for both internationalized as well as culturally-influenced domains of knowledge, and it can also serve both functions, instrumental as well as documentary. It is also the most practical approach since it allows to reuse not only available ontologies, but also available models that represent lexical and/or teminological information. For more on this see chapter 10.

Next, before moving to the specification of the requirements that a model that aims at contributing to the localization of ontologies should comply with, we summarize the main open research problems and assumptions made for this work.

[7] http://genoma.iula.upf.edu:8080/genoma

[8] http://www.ugr.es/õncoterm/alpha-index.html

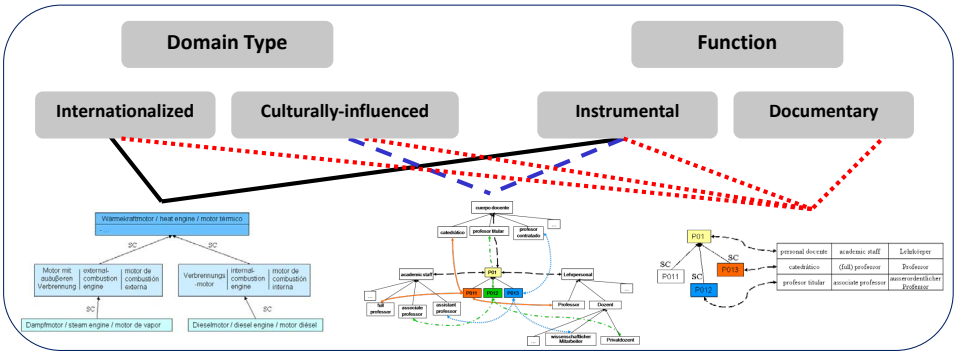


Figure 9.4: Appropriateness of modeling option according to domain type and function

9.4 Open Research Problems and Work Assumptions

After having analyzed the state of the art on approaches to represent linguistic information in ontologies, we identify some open research problems that need to be addressed if our aim is to propose a model that helps in the localization of ontologies (independent of domain type and function).

I. The RDF(S) and OWL modeling option, *Including multilingual labels in the ontology*, suffers from two main drawbacks

- Complex models of linguistic description cannot be linked to ontologies because this option **does not allow to capture relations** between different types of linguistic description elements.
- Cultural discrepancies between the terms associated to ontological entities cannot be accounted for, because **total equivalence is assumed** between the terms associated to the same ontological entity.

II. The option of *Combining the ontology with a mapping model* also poses some problems

- **Ontologies describing the same domain of knowledge** have to be available, otherwise they need to be developed.
- Depending on the domain type, the **mapping establishment may not be so trivial** (mappings need to capture cultural specific relations between concepts, which may differ from the equivalence relation).
- Linguistic information is usually represented by means of RDF(S) and OWL labels.

III. The proposal we make in this PhD work goes in line with the third modeling option: *Associating the ontology with an external linguistic model* because of the advantages mentioned in section 9.3. However, the approaches and

models devised so far do not take into account some of the dimensions identified for the localization of ontologies.

- Approaches such as Termontography assume the development from scratch of an ontology that will only capture those concepts that are common to all the cultures involved in the localization project.
- The models developed so far have been applied to ***internationalized*** **or** ***standardized*** **domains of knowledge**, and do not have mechanisms to account for cases of *culturally-influenced* domains.
- Other models that offer complex descriptions of linguistic information have focused on the **morpho-syntactic description** of ontological entities, and they neglect other aspects of lexical and terminological descriptions that are needed in a multilingual and multicultural context.

We believe that for a model to contribute to the localization of ontologies so that they can be used in a multilingual scenario, several issues still need to be taken into account:

- A complex set of linguistic description elements needs to be provided, and relations have to be established between elements in the same language and also across languages. This is needed for a proper description of the ontology in multiple natural languages so that it can be reused in several cultural contexts.

- Since many ontologies organize the knowledge of specific domains, we argue that our linguistic model should represent not only linguistic and lexical information, but also description elements of terminological resources.

- It cannot always be assumed that several ontologies are available in the same domain of knowledge and with similar extension, and that these can be combined with a mapping model. We cannot either assume that users will be ready to create one ontology for each of the languages involved in a localization project, because this can be a highly time and resource consuming process. Therefore, from a practical viewpoint, we argue that models have to be designed to provide multilingualism to available monolingual ontologies.

- In the case of culturally-influenced domains, our hypothesis is that some of the cultural discrepancies can be captured in an external model, so that the ontology remains unmodified. This means that the external linguistic model has to be powerful enough so as to account for categorization mismatches between or among cultures. We also contemplate the idea that some cultural discrepancies may need to be included in the ontology, but this option is not further investigated in this work.

- We are in favor of a model that follows current standardization trends for the representation of linguistic information on the Web, so that not every linguistic description element is captured in the model, but linguistic elements from other models can be reused according to the specific needs of each localization project. In this sense, the model should be extensible to accommodate additional linguistic descriptions.

It is important to note that since our objectives is to account for multilingualism in ontologies, we will leave out of the scope of our model the morphological and syntactic decomposition of terms. In this sense, we argue that there are some models that already account for these aspects, and that our model should interoperate with them and import those descriptions when needed. We come back to this issue in section 10.1, chapter 10.

Chapter 10

Requirements for an Ontology Localization Model

The model we propose in this PhD work for representating multilingual information in ontologies has the main purpose of enriching an ontology with lexical and terminological information to allow the localization of an ontology to one or several target languages. As outlined in chapter 9, we have opted for associating the ontology with an external model, because this representation modality offers the most advantages for the localization activity, being it both of an *instrumental* or a *documentary* nature. These advantages are summarized below:

- The possibility of providing independent and complex models of linguistic information that can be self-contained and from which information can be inferred. The independence between the ontology and the linguistic model guarantees the full development of both without one restricting the other. In particular, in the case of the linguistic model this allows the existence of a complex model that contains as much linguistic information as required by the final application, and, additionally, in different languages.

- The flexibility for interoperating with existing standards for the representation of lexical and terminological information. By interoperating with those models, there also exists the possibility for the model of interchanging knowledge with the standards, and being extended with further linguistic description elements if so required by the final application.

- The capability for solving the needs of the localization activity depending on three factors: (1) the *domain* of knowledge represented in the ontology (internationalized vs. culturally-influence domains), (2) the *function* of the localized ontology (instrumental vs. documentary), and (3) the *linguistic needs* of the final application.

The first aspect has been already dealt with in chapter 9. In the present chapter, thus, we want to analyze in more detail the interoperability and localization

requirements. Finally, we will refer to the accessibility of the model to external resources, and the searching and navigation possibilities of the model by committing to representation standards.

10.1 Resource Interoperability

Linguistic knowledge should be encoded following standard models in order to guarantee interoperability, reuse, and commitment to best practices. The potential integration of terminological and lexical knowledge bases into our model requires interoperability with existing and proposed standards. This integration supports knowledge exchange between heterogeneous sources, and mappings between them provide assistance with re-engineering activities.

In the state of the art we come across some standardization initiatives that have been developed in order to capture linguistic information that can be reused for various purposes. As the most important initiatives we mention a number of standards from the International Organization for Standardization (ISO) and the World Wide Web Consortium (W3C), which capture terminological and lexical information, and need to be taken into account:

The **Terminological Markup Framework** (*ISO 16642 - TMF-Terminological Markup Framework, Computer applications in terminology*, 2003) (and the associated TermBase eXchange format, TBX[1]), which captures the underlying structure and representation of computerized terminologies. The main structure and classes that make up this representation standard are illustrated in figure 10.1.

The Terminological Data Collection is a top level container for all the information contained in the terminology system. The Global Information Section class contains any kind of general information referring to the terminology system, for instance, title, institution developing the resource, or date of creation. Complementary Information can be included to account for bibliographical or administrative information. The Terminological Entry (TE) class contains information assigned to a single concept, i.e., terms, descriptive information pertinent to a concept, and administrative information concerning the concept. Depending on whether the termbase is monolingual or multilingual, a Language Section (LS) applies. In the case of a multilingual terminology, the information about terms under this class will be contained in different languages, as shown in figure 10.2.

Term Section (TS) is the class that contains the terms *per se* in a certain language. Terms are understood here as "designations of a defined concept in a special language by a linguistic expression". Finally, the Term Component Section class stores information about morphemic elements, words, or contiguous strings from which a polymorphemic term is formed.

In summary, we can state that this representation assumes that a Terminological Entry, which is regarded as the concept, has designations or labels assigned to

[1] http://www.lisa.org/fileadmin/standards/tbxISO_final.html

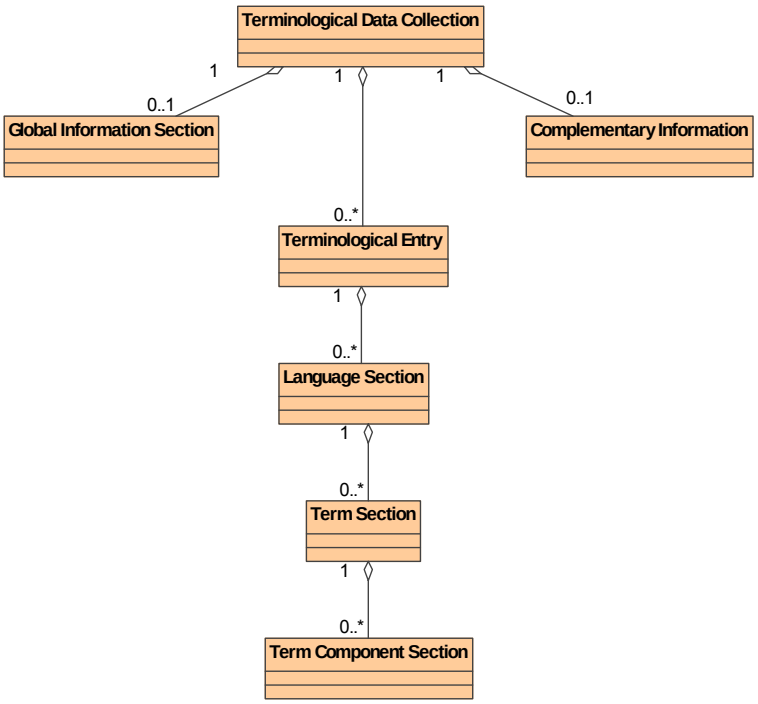


Figure 10.1: TMF structural representation

it in different languages. This representation cannot be said to foresee the case of conceptualization mismatches between languages, for instance when one language categorizes a certain parcel of reality with a higher degree of granularity, and has more specific terms referring to that concept. As a consequence of this, such a representation will not be suited for a model that intends to account for those mismatches among different languages. Nevertheless, it may suffice for highly specialized domains of knowledge, or what we have denominated *internationalized* domains.

Both formats, TMF and TBX, make use of the so-called ISO 12620 data categories (*ISO 12620 - Data Categories, Terminology and other language resources*, 2003). ISO 12620 is an inventory of data categories or data category registry (DCR) that contains *elementary descriptors of a linguistic structure or an annotation scheme*[2]. Examples of data categories are: definition, term type, context or language identifier.

For the purposes of a model that aspires to enrich domain ontologies with linguistic information, we are interested in the term-related information included in section 2 of the ISO 12620 DCR[3]. Our standpoint here is that accounting for the terminology of a certain domain, and more specifically, for the relations among the

[2]http://www.clarin.nl/system/files/ISOcat-introduction.pdf
[3]http://www.ttt.org/clsframe/datcats02.html

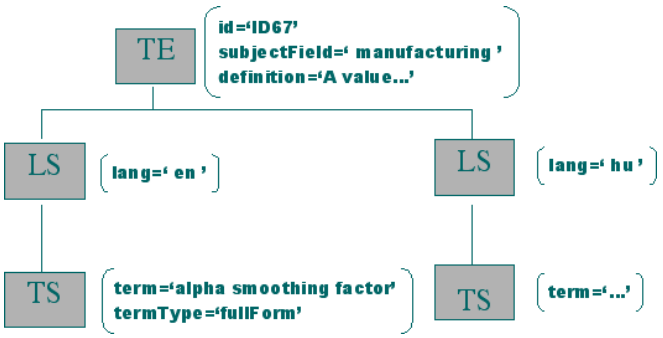


Figure 10.2: Multilingual term entries in TMF

terms used in that domain, is of special relevance for such a model. In this regard, the term-related information section contains attributes of term type such as:

- *main entry term*, described as "the concept designation that has been chosen to head a terminological record".
- *synonym*, "any term that represents the same or a very similar concept as the main entry term in a term entry".
- *international scientific term*, "a term that is part of an international scientific nomenclature as adopted by an appropriate scientific body".
- *common name*, "a synonym for an international scientific term that is used in general discourse in a given language".
- *full form*, "the complete representation of a term for which there is an abbreviated form".
- *variant*, "one of the alternate forms of a term".

By reusing the ISO 12620 categories to identify the linguistic elements and attributes in our model, interoperability with other standards committing to ISO 12620 will be made easier.

SKOS (Simple Knowledge Organization Systems) (Miles et al., 2005) is a W3C proposed standard that provides a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, taxonomies, folksonomies, other types of controlled vocabulary, and also concept schemes embedded in glossaries and terminologies. The practical goal of SKOS is exploiting RDF(S) and OWL data models to model thesauri typical relations with a formal language. At the moment, SKOS core covers the following data objects for handling labels:

I. prefLabel: a preferred label

II. altLabel: an alternative label

III. hiddenLabel: a hidden label (not exposed to any search methods)

And the following ones for handling semantic relations between concepts:

- broader: a more general concept
- narrower: a more specific concept
- related: a concept that states in a certain relation to another

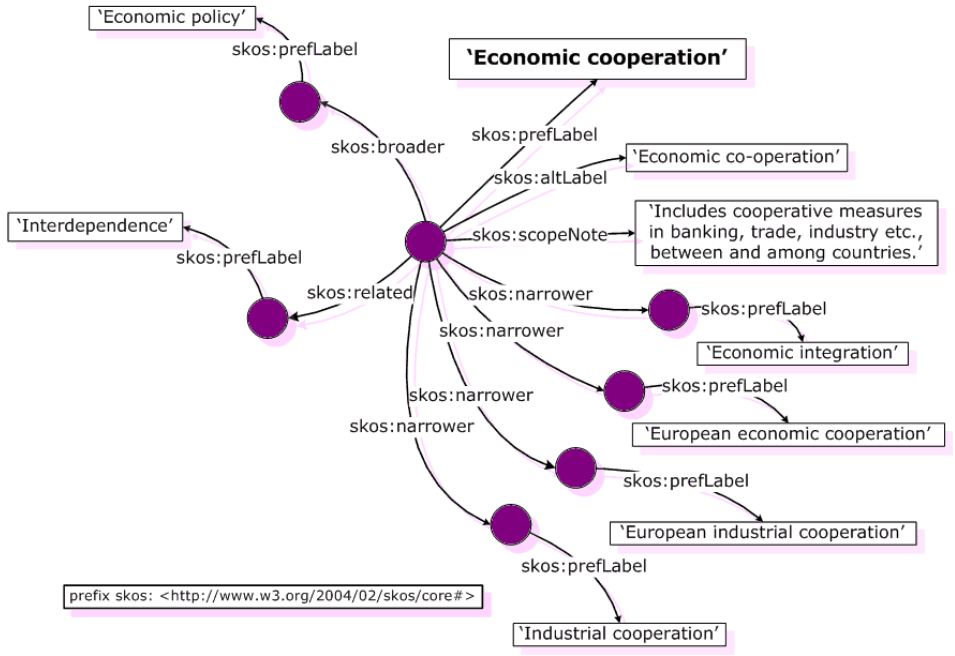


Figure 10.3: RDF graph illustrating terminological and semantic relations in SKOS

Figure 10.3 illustrates the different types of linguistic and semantic relations accounted for in SKOS. Although SKOS has not be designed to specify the meaning of linguistic constructs with respect to an ontology, we are interested in the representation of terminological relations, and not so much in the semantic ones, since these are expressed in our case in the ontology. Regarding the terminological relations, we think that our model should also be able to specify which the preferred label is against alternative labels. This is common practice in organizations or institutions relying on terminology-based resources for tasks such as indexing or information retrieval. In this sense, we believe that our model should also cater for this type of information.

The **Lexical Markup Framework** (LMF; ISO 24613) (*ISO 24613 - LMF- Lexical Markup Framework , Language Resource Management*, 2006) is a meta-model

that provides a common, standardized framework for the construction and use of computational lexicons. This allows interoperability and reusability across applications and tasks. It provides a common, shared representation of lexical objects, including morphological, syntactic, and semantic aspects. The LMF consists of a core package and several extensions that can be added if required by the final application. Each extension focuses on one aspect of language: morphology, syntax and semantics, and there are special extensions for Machine Readable Dictionaries, and for Multilingual lexicons. Figure 10.4 shows the dependencies between the core package and the extensions.

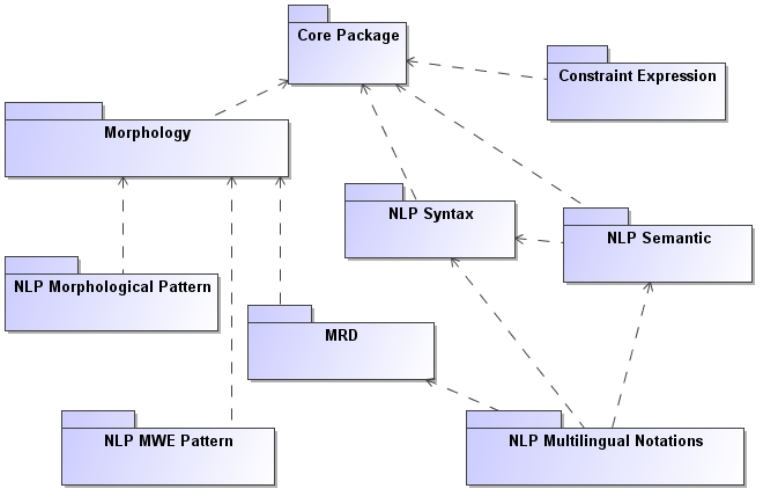


Figure 10.4: Dependencies between the LMF core and extension packages

Committing to this standard is of high interest to us, because it would allow our model not only to interoperate with lexicons following this standard, but also to extend the model with additional packages in case further linguistic information would be needed.

The core package of LMF can be seen in figure 10.5. The upper structure coincides with TMF, containing a class representing general information about the lexicon. However, it differs from TMF in that any specifications to the language of the lexicon have to be made already at this level. This means that a Lexical Resource can consist of several language-specific Lexicons. A Lexicon consists in its turn of several Lexical Entries. Lexical Entry is defined as a lexeme, i.e., *an abstract unit generally associated with a set of forms sharing a common meaning* (*ISO 24613 - LMF- Lexical Markup Framework , Language Resource Management*, 2006). Each Lexical Entry in LMF can contain one to many different forms, and can have from zero to many senses. The Form class is an abstract class that manages one or more orthographical variants as well as lemmas, and Definition contains a description of a sense for human understanding of meaning.

At first sight, the structure proposed by the LMF standard would serve our

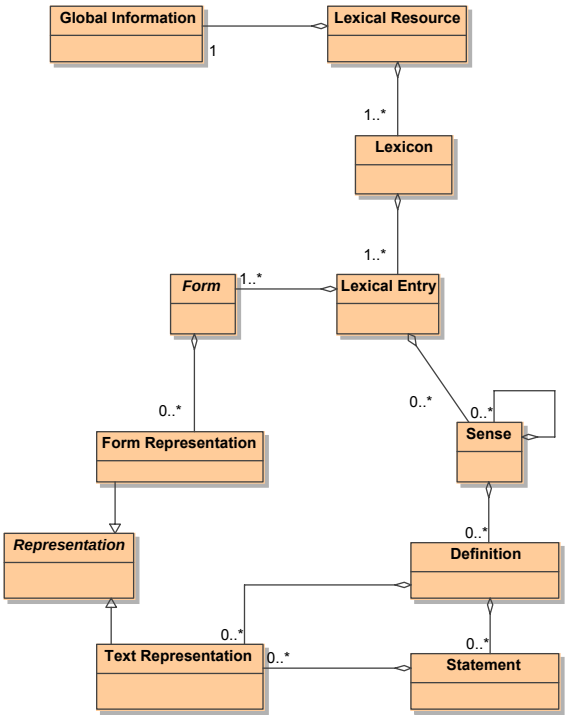


Figure 10.5: LMF core package

purposes of representing word forms associated to a common meaning. However, our intention would not be that of representing all possible senses of a word form -which is the case in a lexicon- but only those meanings or senses that would be equivalent to the concepts captured in the conceptualization, or at least have a higher level of overlap. Apart from that, it is important to mention that the LMF also relies on the ISO 12620 data categories, as was the case of TMF.

Regarding the NLP multilingual extension, we are also interested in the representation of equivalents for Senses between or among two or more natural languages. This is achieved in LMF by means of the so-called Sense Axis, which implements an approach based on an interlingua, i.e., an intermediate general conceptualization that encompasses the meanings of the languages in question. The relation between senses in the different languages is defined by means of the Sense Axis Relation class, but the different types of relations are not further specified. So, for instance, in figure 10.6, the Sense Axis Relation defines the relation existing among different senses as "more general", which would be a quite fuzzy relation only understandable to humans, but not to the machines processing that lexicon. And in fact, as said in (*ISO 24613 - LMF- Lexical Markup Framework , Language Resource Management*, 2006), the goal of the Sense Axis Relation "is not to code a complex knowledge organization system". For all these reasons, the NLP multilingual extension of LMF would not completely satisfy the purposes of

our model of being able to account for conceptualization mismatches between or among different languages.

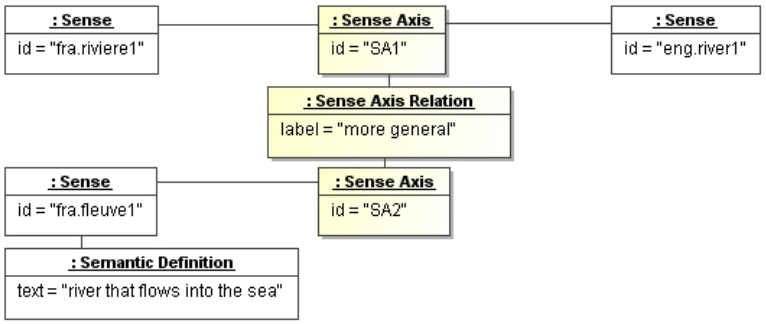


Figure 10.6: Instantiation of sense axis and sense axis relation

All in all, from all the standards taken into account, LMF is closer to our objectives that other models. We argue that by committing to the core classes of LMF, our model will be able to account for the different meanings of a concept in the languages involved in the localization activity. The TMF standard and SKOS will allows us to capture lexical variants and terminological relations. And last but not least, by reusing the linguistic descriptions captured in the three models analyzed, we will be committing to the ISO 12620 categories.

10.2 Localization Requirements

The key requirement of the model that we propose in this work relates to the ability of accounting for the linguistic realization of the knowledge represented in the ontology in a language different from the one in which the ontology has been originally expressed. This has been the main motivation for our research, in which we have tried to design a model that captures linguistic data in such a way that it permits, on the one hand, to maximize the correspondence between ontological conceptualization and lexical/terminological standardization, and, on the other hand, to enrich the ontology with natural language information in order to localize the ontology and make it suitable for a specific cultural and linguistic community.

In this section, our purpose is to spell out the requirements that we consider a model for the localization of ontologies should satisfy.

I. The model should be able to account for **conceptualization mismatches** between or among languages. This situation occurs when dealing with so-called *culturally-influenced domains* in which the function of the localization activity is *instrumental* (see section 8.3). As a result of that, it may happen that the conceptualization represented by the ontology does not exactly correspond with the conceptualization that the target culture would make of the same domain parcel. The typology of cases that may arise in this situation

have been dealt with in section 8.4. This has a straightforward impact in the linguistic realizations of concepts in a language, in that there may be no direct lexicalizations in the target language in terms of one designation that can be used as label for each concept. Otherwise, when dealing with *culturally-influenced* domains in which the localization function is *documentary*, or when dealing with *internationalized* domains, the model has to offer the necessary machinery to explain the original conceptualization to the target communities involved.

II. The model should be expressive enough so as to capture **well-defined relations between or among lexicalizations in the same language**, including lexical and terminological variants, geographical or dialectal variants, formal and informal variants, synonyms, antonyms, and so on. This is important to capture the full expressiveness of a language in a certain setting.

III. We require that the model captures **relations between or among lexicalizations in different languages**, i.e., translation or equivalence relations. Additionally, we also aim at representing the type of equivalence relation existing between the senses that correspond to those lexicalizations.

As already described in section 9.3, the state of the art in the representation of linguistic and multilingual information in ontologies fell short of addressing the needs of a portable model for localizing ontologies. At the time of designing the model, the closest approaches were the ones represented by the Termontography approach, the terminology management system OntoTerm (used in the implementation of the GENOMA Knowledge Base and the OncoTerm terminology) and the LingInfo model, all of them outlined in section 9.3.

A further model, LexOnto, was practically developed in parallel to our model[4], and its follow-up, LexInfo, combines the linguistic elements represented in LingInfo and LexOnto. This latter model is out of the scope of this dissertation work because of being on-going work at the time of writing. In the following we provide a detailed description of OntoTerm, LingInfo and LexOnto.

OntoTerm[5] is a Terminology Management System developed in the late 90s that consist of two fundamental modules, namely, an ontology editor and a termbase editor. This system allows the integration at the same time of the ontology and its related terminological database. The ontology editor provides a core ontology with the 21 basic concepts from Mikrokosmos (ALL, OBJECT, EVENT, PROPERTY, etc.) (Kavi and Nirenburg, 1995). Mikrokosmos aims at organizing general knowledge in a way which is independent of any language, by classifying the knowledge of the world, i.e., all entities, into objects, events, and properties.

[4]Both models were presented for the first time at the OntoLex Workshop in Busan, South Corea, 2007.

[5]http://www.ontoterm.com/

The OncoTerm tool permits then to link the specialized knowledge of a domain to the upper level ontology.

The construction of the ontology has to be prior to the language-specific terms, and no term can be included in the term base if the corresponding concept has not been created first. Terms in different languages can be associated to the ontology concepts. The OntoTerm terminological database follows the term base data model of the CLS Framework[6] and the relational database manager Reltef[7]. The CLS Framework was designed in order to deal with the structure and content of terminological databases. It is based on some ISO 12620 data categories considered relevant for representing terminological information. This data category selection resulted in the development of the MARTIF ISO standard (*ISO 12200:1999 - Computer applications in terminology - Machine-readable terminology interchange format (MARTIF)*, 1999) that, in its turn, enables the exchange of data among terminological resources. The CLS Framework includes the application Reltef, a model consisting of an Entity Relation diagram and a set of tables and relationships, which is in charge of the data recovery and maintenance of the database.

Figure 10.7 illustrates the OntoTerm TermBase editor that consists of the terminological elements included in the database on the left, and the data categories on the right. The data categories selected for the description of the terminological element MYELOID-LEUKEMIA are included in the terminological database. It is also worth mentioning that OntoTerm allows to make explicit references to the documents from which the terms have been extracted. In fact, the corpus of selected documents for terminology extraction are compiled and stored in a Corpus module also associated to the TermBase editor. In short, the three main modules managed by OntoTerm are the Bibliographical, the Lexicographical, and the once containing the Specialist Data or concepts. The modules and its inter-relationships are illustrated in figure 10.8 (from Feliu and Cabré (2002)), representing the architecture of the knowledge base GENOMA.

The main similarities between this system and the model that we intend to propose for the localization of ontologies are listed below:

- Independence between conceptual and terminological knowledge
- Compliance with existing standards for the representation of terminological information
- Onomasiological approach, in which the conceptual knowledge is introduced prior to the terminological knowledge
- Support for multilingualism
- Reference track

[6] http://www.ttt.org/clsframe/

[7] http://www.ttt.org/clsframe/reltef.html

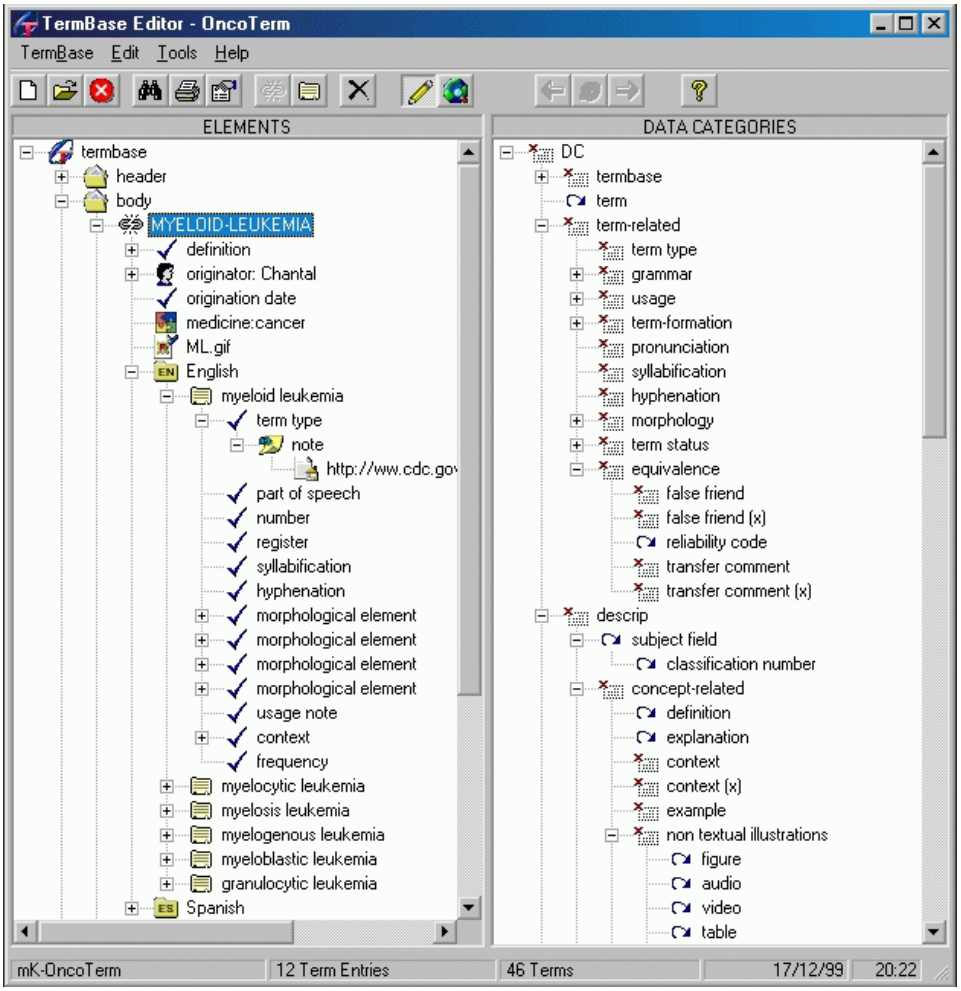


Figure 10.7: Snapshot of the TermBase editor view in OntoTerm

However, there are also important discrepancies that prevented us from reusing this system for our needs, among others, that OntoTerm uses a proprietary system. This was a major drawback, since we wanted to propose a portable model that could be associated to available ontologies in the Web. Taking into account that OWL and RDF(S) have become the most popular knowledge representation languages for ontologies and *de facto* standards, we decided to commit to those languages and propose a model that could be associated to ontologies following these standards. We also decided to make use of the OWL language for our representation purposes, and implement the linguistic model in OWL. This would also allows us to take advantage of the whole machinery at our disposal regarding ontology editors, reasoners, and so on.

A further handicap of the OntoTerm system was the fact that it was too centered on terminological needs, and neglected some semantic aspects that we wanted to

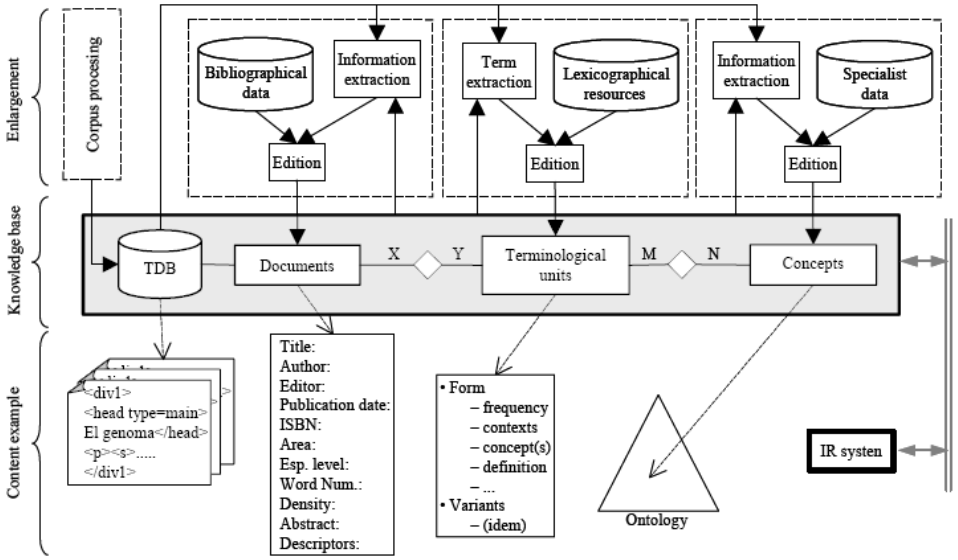


Figure 10.8: Architecture of the GENOMA-KB implemented in OntoTerm

capture in our model. Basically, it assumed an exact correspondence between a concept in the ontology and the set of terms associated to it. While this could be assumed for *internationalized* domains, as was the case of the Genoma-KB and OncoTerm applications, we argue that such a representation would fall short of supporting the localization of *culturally-influenced* domains.

LingInfo, (Buitelaar, Declerck, et al., 2006) and (Buitelaar, Sintek, and Kiesel, 2006), is a lexicon model for the representation of terms associated to classes and properties of an ontology. The lexicon information is included in the ontology, but forms part of a module that extends the lexical information associated to ontology elements, by means of a meta-class (ClassWithLingInfo) and a meta-property (PropertyWithLingInfo). In this way, the instances of the LingInfo classes are linked to classes and properties. The link connecting the ontology and the lexicon model is illustrated in figure 10.9.

LingInfo principally accounts for the representation of inflection and morphological decomposition of ontology labels for classes. The main classes contained in the model are: language identifier, part of speech information, morphological decomposition (modifier, head), and syntactic decomposition (phrase category: noun phrase, verbal phrase, etc.).

This model also inspired us in the design of our model for the localization of ontologies, although we were not so interested in the morpho-syntactical decomposition of ontology labels, at least in a first stage. However, a possible extension of the model to cover those properties had to be foreseen, and could be achieved by committing to the LMF standard, for instance, as already mentioned in section 10.1.

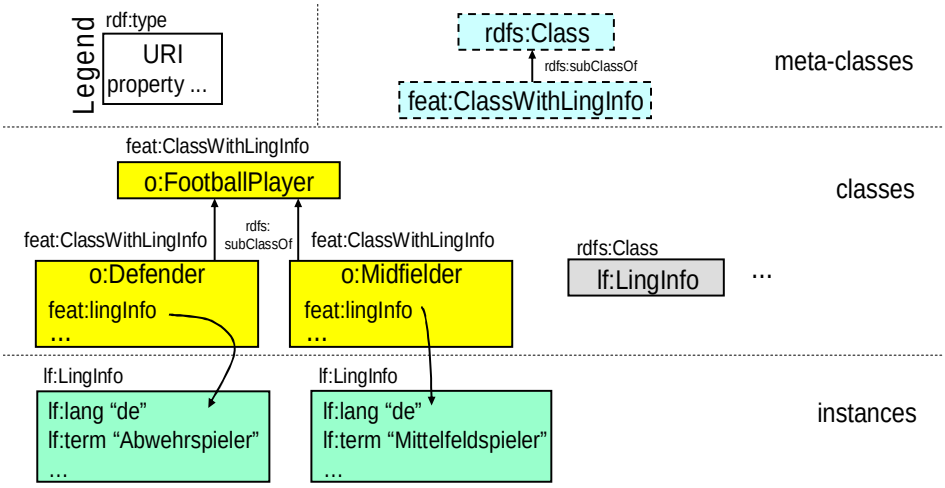


Figure 10.9: LingInfo model with multilingual instances

Regarding the modular integrated approach followed by LingInfo, we were in favor of a more strict separation between the conceptual knowledge and the linguistic multilingual representation.

LexOnto (Cimiano et al., 2007) is a lexicon model that was born with the purpose of capturing the "syntactic behavior" of words in relation to their counterparts in the ontological representation. This means that the information in the lexicon pursues to capture more complex structures, i.e., sentences, in which ontology labels take place, and then relate the different parts of the sentence to the corresponding ontology elements.

The classes that make up this model principally account for the predicate-argument structures that nouns, verbs and adjectives project. As can be seen in figure 10.10 (from Buitelaar et al. (2008)), Lexical Elements can be anchored to ontology classes and properties. A Lexical Element in LexOnto consists of WordForms (verbs, nouns, or adjectives) participating in a predicate-argument structure or subcategorization frame (e.g., nominal prepositional phrases, transitive verbal phrases, scalar adjective phrases, etc.).

An example taken from Buitelaar et al. (2008) illustrates the purposes of this model. If the ontology models the relation between countries and their capital cities `(capital(Country, City))`, the following linguistic structures could be linked to it: *City is capital of Country*, or *Country has capital City*. This is the main contribution of the model, namely, the ability of explicitly representing *verbal* argument structures and map them to ontology structures in a sort of three-element triple. In the case of *nominal* or *adjectival* argument structures, LexOnto also allows to derive the internal structure underlying these word forms. Let us take the example of the noun phrase *great vein of the heart*. This word form would categorize a *nominal prepositional phrase* whose elements (noun phrase, preposition,

determiner, noun) would be explicitly represented in the model.

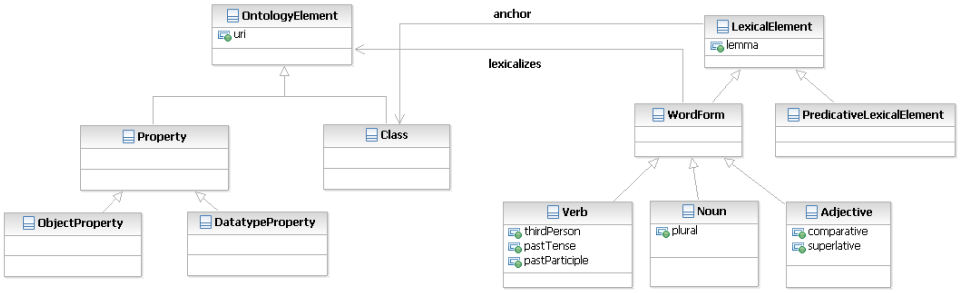


Figure 10.10: Main elements of the LexOnto lexicon model

In summary, this model can be considered orthogonal to the model we need to propose for the localization of ontologies. It provides a very complete grounding of the ontological representation in the language, but its objective is not to account for multilingualism or making a certain conceptualization suitable for a target linguistic community. Still, all the information captured in the model could be complementary to the information we aim at representing in our model for ontology localization. Furthermore, the core structure of the LexOnto model is also based on the LMF standard, so that interoperability or reuse would be feasible.

10.3 Accessibility Requirements

As already outlined in the previous section, accessibility (knowledge interchange, querying, navigation, import and export) was a central requirement for a linguistic model expected to localize ontologies. In this sense, accessibility had to be seen from two perspectives:

I. Accessibility from the point of view of the language representation selected to implement the model, and the corresponding tool support available to manage it.

II. Accessibility to external resources from which to obtain the information to (automatically or semi-automatically) populate the model.

The first requirement would be satisfied by committing to the OWL and RDF(S) knowledge representation languages that are currently supported by the most popular ontology editors and ontology management systems (Protégé[8], NeOn Toolkit[9], TopBraid Composer[10], etc.)

The second requirement is to be fulfilled by committing to standards for the representation of lexical and terminological information, on the one hand, and on

[8] http://protege.stanford.edu
[9] http://neon-toolkit.org
[10] www.topbraidcomposer.com

the other hand, by relying on a tool that would access external multilingual linguistic resources to help populating the model. We will come back to this issue in section 11.2.

10.4 Summary

In this chapter, our aim was to spell out the requirements that our model should comply with. To the best of our knowledge, the requirements addressed in this chapter cover the basic needs of a model that aims at associating multilingual information to ontologies with the objective of making the conceptualization reusable in different cultural settings. Requirements can be divided in four groups: representation, interoperability, localization, and accessibility. Those requirements that have to do with representation options were described in chapter 9, the rest of them have been reported in this chapter. We have summarized all of them in figure 10.11.

Taking into consideration all the requirements, standardization initiatives, and previous (and parallel) work on linguistic models designed to be associated to ontologies, we came up with a proposal for a model to enrich domain ontologies with multilingual information. The model has been named Linguistic Information Repository, or its acronym LIR, and will be described in chapter 11.

Requirements for an ontology localization model		
Representation	**R1**	Independence between the ontology and the linguistic model
	R2	Complex and rich representations of linguistic information
	R3	Commitment to ontology web languages (specifically, OWL)
Interoperability	**R4**	Commitment to Data Category Registry (ISO 12620)
	R5	Representation of term-related information accroding to TMF (e.g., main entry term, international scientific term vs. common name, full form vs. short form, term variants, etc.)
	R6	Representation of terminological relations according to SKOS (e.g., preferred labels vs. alternative lables)
	R7	Representation of lexical relations according to LMF (e.g., lexical entry, word form, sense, definition)
	R8	Commitment to LMF to interoperate with LMF lexicons and extend the model with additional packages, if required
Localization	**R9**	Representation of categorization mismatches between or among languages
	R10	Establishment of explicit links between or among lexicalizations within the same language
	R11	Establishment of explicit links between or among lexicalizations in different languages
	R12	Description of the meaning or semantics of lexical and and terminological elements
	R13	Explicit reference to source provenance as in OntoTerm
	R14	Explicit reference to the language to which the linguistic descriptions belong (by means of a language identifier)
Accessibility	**R15**	Integration in an ontology editor or plug-in to provide access to external resources that populate the model

Figure 10.11: Summary of requirements for an ontology localization model

Chapter 11

Linguistic Information Repository: a Model for Ontology Localization

The Linguistic Information Repository (LIR[1], henceforth) has been created with the twofold purpose of fulfilling the needs of portability and association of multilingual information to domain ontologies, on the one hand, and adapting ontologies to the needs of the languages involved in the localization activity, on the other.

In this chapter our objective is to describe the different classes that compose the model, justifying the inclusion of them (section 11.1). In section 11.2, we report on the implementation of the LIR in the LabelTranslator system, a plug-in of the NeOn Tooltik that relies on the LIR model to store the linguistic information obtained from the translation of ontology labels (Espinoza et al., 2008a), (Espinoza et al., 2008b), (Espinoza, Gómez-Pérez, and Montiel-Ponsoda, 2009), and (Espinoza, Montiel-Ponsoda, and Gómez-Pérez, 2009). Finally, in section 11.3, we present an extension to the Ontology Metadata Vocabulary (OMV) (Hartmann et al., 2006), the so-called LexOMV, a vocabulary that allows to report about multilingualism at the ontology meta-data level.

In order to guarantee interoperability with existing standards for the representation and integration of terminological and lexical knowledge, the LIR adopts a number of ISO data categories for linguistic description, mainly present in the ISO standards Terminological Markup Framework (TMF) and Lexical Markup Framework (LMF), as explained in section 10.1, chapter 10. In this way it commits to the interoperability requirements R4 to R8 identified section 10.4. Its design is mainly based on the core package of LMF. In LMF, a Lexicon comprises Lexical Entries that are linguistically realized by word forms related to the different senses a word

[1]This model has been designed in the framework of the NeOn project, as reported in the literature ((Peters et al., 2007), (Montiel-Ponsoda, Aguado de Cea, Gómez-Pérez, and Peters, 2008), or (Montiel-Ponsoda, Peters, et al., 2008)). We are greatly indebted to Wim Peters and Margherita Sini for their contributions.

can have, as happens in WordNet[2]. However, the rationale underlying the LIR is not to design a lexicon for different natural languages and then establish links to ontology concepts, but to provide a linguistic layer in different natural languages that captures the conceptual knowledge represented in a specific domain ontology.

In the LIR, each lexical entry belonging to a certain language can be realized by different word forms linked to a sense, which is constrained by the knowledge represented in the ontology. This is assumed for practical reasons, although word senses and concepts can not be said to overlap (Hirst, 2004). The reason for this is that word senses are tightly related to the particular vision of a culture, whereas ontology concepts try to capture objects of the real world in a formal way (i.e. in a machine-understandable way), and are defined according to expert criteria agreed by consensus. These criteria need not fully reflect the lexical meaning of the natural language label that lexicalizes the concept. However, by keeping the sense of the lexical entry independent from the concept in the ontology, the LIR can account for the way in which a certain cultural and linguistic community understands the bit of reality captured in the concept. As will be shown in section 11.2, this assumption allows us to comply with one of the most important localization requirements, namely, the possibility to account for conceptualization mismatches (see localization requirements R9 to R14 spelled out in section 10.4).

It could be stated that the LIR goes more in the line of what Pustejovsky defined as Sense Enumeration Lexicon (Pustejovsky, 1995: p.47), in which a unique sense is associated to a word string. As the author says to this respect "Even if we were to assume that sense enumeration were adequate as a descriptive mechanism (...), it is not always obvious how to select the correct word sense in a given context (...)". We agree with this approach and admit that this theory would not be adequate if our purposes were to design a lexicon for a language. However, we argue that this can be a suitable approach to enrich domain ontologies with multilingual information, because our objective is to associate lexical semantic meaning to ontological knowledge, which already provides us with the needed context so as to restrict a certain reading of a word.

According to the needs of the final application, the LIR could be extended with further linguistic knowledge, such as morphological decomposition and syntactic complementation, as modelled in LMF or LexOnto, and could be obtained by navigating those models after establishing a connection between them. In fact, at the time of writing this document, discussions are being hold to integrate LIR with the LexInfo model, which is in its turn an integration of LingInfo and LexOnto (described in section 10.2) in the framework of the European project Monnet[3].

The LIR also serves the objective of integrating and aggregating multilingual information contained in heterogeneous and distributed lexical sources by guaranteeing a homogeneous access to the information and keeping track of the sources of provenance.

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

[3] http://www.monnet-project.eu/Monnet/Monnet/English?init=true

In the following, our purpose is to describe in more detail the classes and properties that make up the LIR, as represented in figure 11.1.

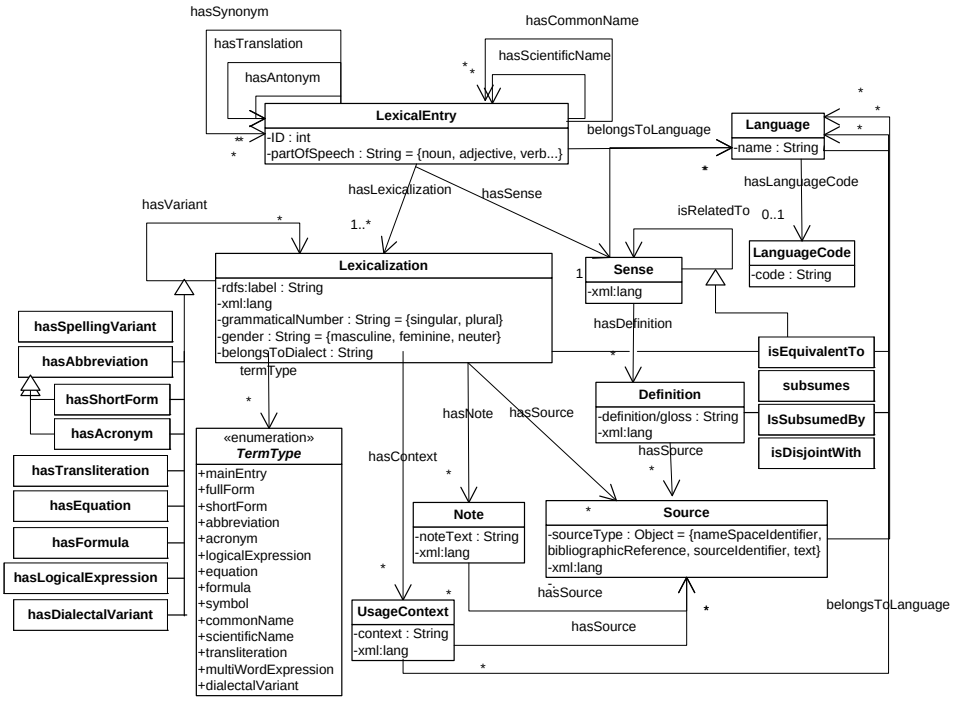


Figure 11.1: The LIR model

11.1 Description of the LIR Model

The linguistic information captured in the LIR is organized around the `LexicalEntry` class. A lexical entry is considered a unit of form and meaning in a certain language. Therefore, it is associated to the classes `Language`, `Lexicalization` and `Sense`. A set of related lexicalizations or term variants shares the same meaning (represented by the sense) within the specific context of a certain cultural and linguistic universe. This is the core of the LIR, which is LMF compliant. In the rest of this section, we will analyze the classes and properties that compose the LIR with more detail.

LIR Classes

1. LexicalEntry: a lexeme in the sense of LMF, which is a unit of form and meaning. A lexeme is an ordered collection of related word forms, having the same lexical meaning. Please note that the meaning shared by the word forms is lexical, not grammatical. In other words, meaning differences between e.g. singular/plural

are not covered by lexical meaning. The `LexicalEntry` class manages the link between the classes `Sense` and `Lexicalization`. It is an abstract class, of which each instance is a combination of a set of lexicalizations and one sense.

The `LexicalEntry` class has the following attributes or data type properties: `ID` and `partOfSpeech`.

- **ID** gets assigned a number according to the number of lexical entries previously associated to the ontology element. If the ontology element has only one lexical entry associated to it, the ID will be 1. This identification is included just for organizing purposes.

- **PartOfSpeech** is defined as the the grammatical class of the `LexicalEntry`. Traditionally, members of the set of word forms incorporated into a particular lexeme are selected on the basis of part of speech, inflectional behaviour and meaning. Within the LIR, this means that lexemes are prefiltered by the major syntactic class by means of the `partOfSpeech` attribute. This corresponds with the encoding of part of speech in LMF. By doing so, the repetition of `partOfSpeech` for all Lexicalization instances is avoided, since Lexicalizations are deemed to belong to the same major part of speech. Synonymy relations across major part of speech boundaries will need to be implemented at the `LexicalEntry` level.

2. Sense: a language-specific unit of intensional lexical semantic description. In is an abstract or empty class "materialized" through the `Definition` class. The purpose of being an abstract class is that it is also considered a pointer to the resource in which the same sense is contained. The provenance source is then made explicit through the `Source` class. It only contains the attribute `xml:lang`.

- `xml:lang`: reflects the language code from ISO639-2 (*ISO 639 - Codes for the representation of names of languages*, 2002) associated with the range of the `belongsToLanguage` relation or object property (see more details in the description of the `Language` class). This allows us to model idiosyncratic differences between language specific meanings.

Within LIR, there are two possible ways to model language specificity:

a) Based on the principled viewpoint that lexical entries by default express language specific notions, `Sense` is necessarily considered language specific as well. This is assumed in e.g. the EuroWordNet model (Vossen, 2004). Besides, two lexical entries in different languages are associated with different senses. If the lexical entries mean the same, we link them with the `hasTranslation` relation. Other types of equivalence relations between lexical entries, such as equivalence, subsumption and disjointness, can be modeled in LIR by postulating sub-relations of the `isRelatedTo` relation between senses.

b) Terminological entries in e.g. TMF and TBX define one sense for a multilingual set of terms. The assumption behind this is that terms (as opposed to

lexical items in general) have a very precisely defined meaning within a domain. In order to model this terminological case, we can either apply a), or link each LexicalEntry to one and the same Sense, i.e. the meaning of the terminological entry. This would represent a variant of an interlingua approach assuming equivalence between language-specific lexical entries.

The LIR is capable of modeling both options. A choice needs to be made for each use case. The translational or conceptual equivalence between lexical entries is expressed by the relations `hasTranslation`, between lexical entries in different languages, and `hasSynonym` between lexical entries in the same language (see points 2 and 3).

3. Lexicalization: a word form. This class corresponds with the LMF `Form Representation` class, defined as a class representing one variant orthography of a `Form` (*ISO 24613 - LMF- Lexical Markup Framework , Language Resource Management*, 2006). The choice of this data category means that the lexicalizations of concepts are deemed word forms rather than lemmas or citation forms, and therefore also include inflected forms, such as plurals.

The class Lexicalization has the following attributes:

- `rdfs:label`: string representing the word form.
- `xml:lang`: (optional) language code from ISO639-2 associated with the range of the `belongsToLanguage` object property. The reason for including the language of the `Lexicalization` as an attribute and as a relation or object property with range `Language` is motivated by the posibility of representing loanwords, words borrowed from a foreign language that have been incorporated in another language. This apparent redundancy is also present in the `Definition`, `Source`, `Note` and `UsageContext` classes as will be explained below.
- `grammaticalNumber`: captures the morphosyntactic features of the lexicalization, and can take the following values: “singular”, “plural” and “other”.
- `gender`: captures grammatical and inflectional features of the lexicalization, and can take the following values: “masculine”, “feminine”, and “neuter”.
- `belongsToDialect`: (optional) the dialect name to which the `Lexicalization` belongs. This is an optional attribute that can be used to further specify the `xml:lang` attribute, in case we need to account for geographical or dialectal variants.

Further, it contains a set of descriptions for term types taken from TMF and TBX, split up into: **a) Term type attributes** represented as a set of Boolean attributes or values of the `termType` attribute in itself that describe a number of term types (sub-properties of the attribute or datatype property `termType`):

- `mainEntry`: the concept designation that has been chosen to head a terminological record (ISO 12620: section 02.01.01).

- `formula`: figures, symbols or the like used to express a concept briefly, such as a mathematical or chemical formula (ISO 12620: section 02.01.14).
- `equation`: an expression used to represent a concept based on the statement that two mathematical expressions are, for instance, equal as identified by the equal sign (=), or assigned to one another by a similar sign (ISO 12620: section 02.01.15).
- `symbol`: a designation of a concept by letters, numerals, pictograms or any combination thereof (ISO 12620: section 02.01.13).
- `logicalExpression`: an expression used to represent a concept based on mathematical or logical relations, such as statements of inequality, set relationships, boolean operations, and the like (ISO 12620: section 02.01.16).
- `scientificName`: a term that is part of an international scientific nomenclature as adopted by an appropriate scientific body (ISO 12620: section 02.01.04).
- `commonName`: a synonym for an international scientific term that is used in general discourse in a given language (ISO 12620: section 02.01.05).
- `fullForm`: the complete representation of a term for which there is an abbreviated form (ISO 12620: section 02.01.07).
- `acronym`: an abbreviated form of a term made up of letters from the full form of a multiword term strung together into a sequence pronounced only syllabically (ISO 12620: section 02.01.08.04).
- `shortForm`: an abbreviated form that includes fewer words than the full form, e.g. “Intergovernmental Group of Twenty-four on International Monetary Affairs” vs. “Group of Twenty-four” (ISO 12620: section 02.01.08.02).
- `abbreviation`: a term resulting from the omission of any part of the full term while designating the same concept, e.g. adjective vs. adj. (ISO 12620: section 02.01.08).
- `transliteration`: a form of a term resulting from an operation whereby the characters of an alphabetic writing system are represented by characters from another alphabetic writing system (ISO 12620: section 02.01.10).
- `multiWordExpression`: this attribute is equivalent to ISO 12620 `Phrase`, defined as a phraseological unit containing any group of two or more words that are frequently expressed together and that comprise more than one concept (ISO 12620: section 02.01.18).
- `dialectalVariant`: this attribute indicates whether a word form originates from a dialect.

b) A number of relations between Lexicalization classes expressed by the object property `hasVariant` and its following sub-properties. In TMF and TBX, these term types are represented as attributes rather than relations. However, representing them as relations rather than as Boolean attributes ensures the proper link between unique source and target lexicalizations where term type attributes allow multiple derivations of relations.

The reason for using both a set of Boolean attributes and a set of relations is that relations cannot always be deduced from a set of attributes. For instance, if two lexicalizations are associated with a `LexicalEntry`, one of them as a full form, and one as an abbreviation, then it is impossible to determine with certainty if, on the basis of Boolean attributes, the full form lexicalization is related to the abbreviation.

Also, if a `LexicalEntry` contains two full form lexicalizations and one acronym, it is impossible to determine which full form is in the domain of the `hasAcronym` object property on the basis of attributes alone. For instance, the WordNet synset (*J, Joule, watt second (unit of electrical energy)*) contains three Lexicalizations, of which two are full forms, and one is an acronym. Using attributes alone will not enable the user to establish the right `hasAcronym` relation between any pair wise combination of these `Lexicalizations`.

Conversely, in cases where a `LexicalEntry` occurs in isolation, it is impossible to determine the term type of the `Lexicalization` on the basis of relations, because there are not any available. For instance, when there is only one `LexicalEntry` containing a scientific name, the relation `hasScientificName`, which holds between `LexicalEntries` (see below), cannot be used to characterize the `Lexicalizations` contained by the `LexicalEntry` as scientific name. In order to be able to do this, this `hasScientificName` relation needs at least a pair of lexical entries one of which contains a lexicalization with a `scientificName` attribute value "true", whereas the other needs a `Lexicalization` with the `commonName` attribute value "true". The attribute `ScientificName` is necessary to characterize the lexicalization from this isolated `LexicalEntry` as a scientific name.

The relations or object properties specializing the `hasVariant` relation are:

- `hasSpellingVariant` (inverse: `isSpellingVariantOf`)
- `hasAbbreviation` (inverse: `isAbbreviationOf`)
- `hasAcronym` (inverse: `isAcronymOf`)
- `hasShortForm` (inverse: `isShortFormOf`)
- `hasTransliteration` (inverse: `isTransliterationOf`)

The relations `hasAcronym` and `hasShortForm` are subtypes of `hasAbbreviation`. Although both have been officially disallowed in TMF, and the

use of the more general attribute `Abbreviation` is prescribed, we argue that this may be useful for some applications.

4. Language: this language concept has been imported from the *language-code ontology* of the Food and Agriculture Organization (FAO[4]). This ontology contains multilingual language names and ISO639 codes. It is linked to various LIR classes through the object property `belongsToLanguage` and its inverse `hasLinguisticExpression` (see below).

5. LanguageCode: the ISO 639-1 and 639-2 codes are standard labels for languages, which have been incorporated into FAO's languagecode ontology. ISO 639-1 is the alpha-2 code (codes composed of 2 letters of the basic Latin alphabet). Multiple codes for the same language are to be considered synonyms. ISO 639-2 is the alpha-3 code (codes composed of 3 letters of the basic Latin alphabet). Both ISO639-1 and ISO639-2 are subclasses of `LanguageCode`. `Language` and `LanguageCode` are related through the object property `hasLanguage-Code` (see below).

6. Definition: a statement that describes a concept and permits its differentiation from other concepts within a system of concepts. (ISO 12620: section 05.01). The Definition class has the following attributes:

- `definition/gloss`: string.
- `xml:lang`: optional attribute to indicate the language in which the definition is written. It reflects the language code from ISO639-2 associated with the range of the `belongsToLanguage` object property.

7. Source: the provenance of the linguistic/terminological information. The class Source contains the following data properties:

- `sourceType`, which itself has the following sub-properties:
  - `nameSpaceIdentifier`: URL/URI (see ISO12620: section 10.21).
  - `bibliographicReference`: a complete citation of the bibliographic information pertaining to a document or other resource (see ISO12620: section 10.19).
  - `sourceIdentifier`: the code assigned to a document in a terminological collection and used as both the identifier for a bibliographic entry and as a pointer in individual term entries to reference the bibliographic entry identified with this code (see ISO12620: section 10.20).
  - `text`: e.g. a textual description of the resource, or maybe a unique key into the resource specific information structure (for instance, in the case of a dictionary, the composite key lemma, part of speech and sense number).

[4] http://www.fao.org/aims/aos/languagecode.owl

- `xml:lang`: optional attribute to reflect the language code from ISO639-2 associated with the range of the `belongsToLanguage` object property.

8. UsageContext: a text or part of a text in which a term occurs (ISO12620: section 05.03). TBX describes this class as follows: context sentences serve the following purposes

- They prove that the term actually exists in real language.
- They can shed light on the meaning of the term.
- They can provide additional "encyclopedic" information about the term that is not in the definition (the who, why, when, where, how).
- They can illustrate how the term is used in discourse (collocations, register, etc.). For instance, a context sentence could alert the translator that the term is colloquial.
- They can provide grammatical information (such as gender), stylistic clues (such as hyphenation or capitalization) as well as alternate forms (abbreviations and so forth).
- The requirement to include a context sentence for the target language term helps to prevent the terminologist from simply translating the source language term, by requiring him or her to find an equivalent designation of the concept actually in use in the target language. This helps to ensure authenticity of the target language term and helps to reduce influence of the source language on the target language

Usage contexts can consist of plain text, and therefore be associated with `Lexicalization` in order to model the occurrence of word forms in context. One could argue that the information captured in this class can be very useful for humans but not for machines. And it is certainly so. A further processing would be needed to discover the syntactical behavior of the lexicalization in question. At this stage, the LIR could be extended with other models covering and formalizing this specific information such as LMF, LingInfo, or LexOnto.

The class `UsageContext` has the following attributes:

- `context`: the textual context in string format.
- `xml:lang`: optional attribute reflecting the language code from ISO639-2 associated with the range of the `belongsToLanguage` object property.

9. Note: supplemental information pertaining to any other element in the data collection, regardless whether it is a term, term-related, descriptive, or administrative (ISO12620: section 08). This class can be linked to any class from the LIR model. For the moment, this sort of supplemental information envisages to be captured in a non-formal way through free text. It is possible that in a later stage these

differences can be formalized to a greater extent. The `Note` class will function as an extension point for this potential further formalization. The class Note has the following attributes:

- `noteText`: the content of the Note in string format
- `xml:lang`: optional attribute reflecting the language code from ISO639-2 associated with the range of the `belongsToLanguage` object property.

In the following we describe the relations used to link classes in the LIR model.

LIR Relations

1. hasLexicalEntry: the link between the ontology and the LIR, as shown in figure 11.2. This relation has, as yet, no semantic characterization apart from "is lexicalized by".

- Domain: `OntologyElement` external to the LIR[5]
- Range: `LexicalEntry`
- Inverse: `isLexicalEntryOf`

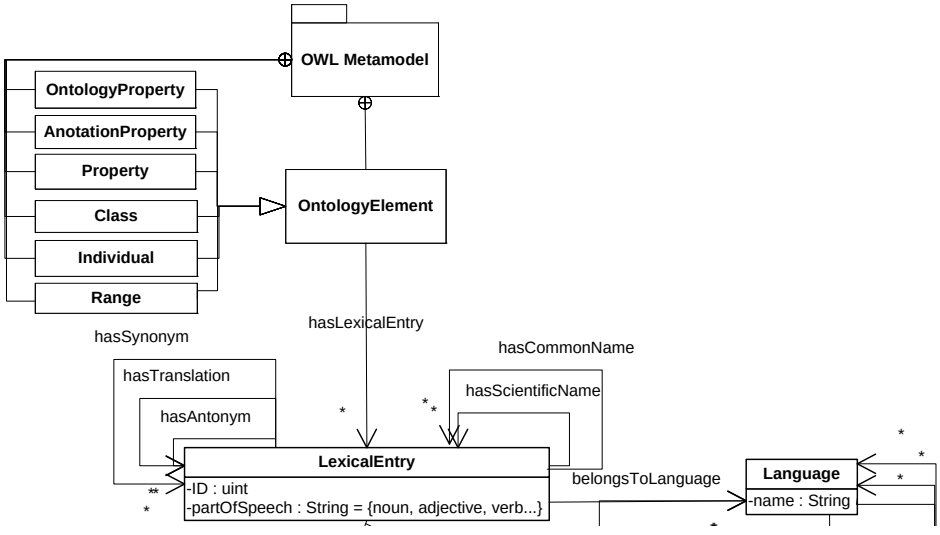


Figure 11.2: Link between ontological and lexical knowledge

The upper part of the figure 11.2, shows the central part of the OWL metamodel, which follows the Description Logic (DL) paradigm. `OntologyProperty`, `AnnotationProperty`, `Property`, `Class`, `Individual` and `DataRange`

[5]The class Ontology Element is part of OWL ontology metamodel, see http://owlodm.ontoware.org/OWL1.0

are all ontology elements. It has been our aim to design a linguistic model that allows the association of lexical and terminological data with each `OntologyElement`.

As mentioned above, by representing conceptual and lexical knowledge in two separate models we are complying with one of the requirements to which the model was subject to: independence of conceptual and linguistic information. Although a relation has been established between the `OntologyElement` of the OWL meta-model, for the time being we only pursue the linguistic enrichment of classes, properties and individuals of the ontological meta-model.

2. hasSynonym: lexical semantic equivalence relation between lexical entries. WordNet distinguishes between the lexical relations synonymy and antonymy (for the latter see point 18) on the one hand, which depend on the lexemes involved in the relation, and conceptual relations between synsets on the other, which do not depend on the lexemes that constitute the synsets. The decision whether two lexical entries in different languages are synonyms, depends on the characterization of the synonymy relation. Since labels are elements from natural language, the logical notion of synonymy (the preservation of truth conditions in all contexts) is hardly ever applicable. Because of this fact, Miller (1990) suggest using a weaker notion of synonymy, namely 'semantic similarity', which is defined as "two expressions are synonymous in a linguistic context C if the substitution of one for the other in C does not alter the truth value".

In the LIR model we are concerned with capturing lexical knowledge, which is connected, but not equivalent to, ontological knowledge in our model through the `hasLexicalEntry` relation (see above). Therefore we follow this lexical, rather than logical, notion of synonymy.

- Domain: `LexicalEntry`
- Range: `LexicalEntry`
- Inverse: `isSynonymOf`

3. hasTranslation: translation equivalence relation between LexicalEntries from different languages.

- Domain: `LexicalEntry`
- Range: `LexicalEntry`
- Inverse: `isTranslationOf`

4. hasVariant: this property and its sub-properties (points 5-9 below) reflect the `termType` data property associated with `Lexicalization`. The reason for this redundancy is given in the Lexicalization section above (point 3).

- Domain: `Lexicalization`

- Range: `Lexicalization`
- Inverse: `isVariantOf`

5. hasSpellingVariant: a relation between lexicalizations describing variance in orthographic representation.

- Domain: `Lexicalization`
- Range: `Lexicalization`
- Inverse: `isSpellingVariantOf`

6. hasTransliteration: it is related to the `Transliteration` data type property described above.

- Domain: `Lexicalization`
- Range: `Lexicalization`
- Inverse: `isTransliterationOf`

7. hasAbbreviation: it is related to the `Abbreviation` data property described above. This in turn subsumes the following relations: `hasShortForm` and `hasAcronym`, which are related to the attributes `ShortForm` and `Acronym` described above.

- Domain: `Lexicalization`
- Range: `Lexicalization`
- Inverse: `isAbbreviationOf`; `isShortFormOf`; `isAcronymOf`

8. hasScientificName and hasCommonName: both relations have been defined as inverse relations between lexical entries. This gives us a more economical representation of this information, because it reduces the reduplication of this information at the lexicalization level. If we maintain the `hasScientificName` relation as a relation between lexicalizations, we need to encode this relation between each common name lexicalization within each `LexicalEntry` and each scientific name lexicalization, not only within a language, but also across languages, since the scientific name is the same for each language specific common name.

- Domain: `LexicalEntry`
- Range: `LexicalEntry`
- Inverse: `isScientificNameOf`, `isCommonNameOf`

9. hasDialectalVariant: it indicates whether a word form originates from a dialect. The name of the dialect is encoded by the `belongsToDialect` attribute.

- Domain: `Lexicalization`
- Range: `Lexicalization`
- Inverse: `isDialectalVariantOf`

10. hasNote: relation between any `OntologyElement` and `Note`.

- Domain: `LexicalEntry`, `Lexicalization`, `Sense`, `Source`, `Definition`, `UsageContext`
- Range: `Note`
- Inverse: `isNoteOf`

11. hasSource: it associates various classes with Source. Domain: LexicalEntry, Lexicalization, Sense, Note, Definition, UsageContext Range: Note Inverse: isSourceOf

12. hasDefinition: it associates Sense with Definition.

- Domain: `Sense`
- Range: `Definition`
- Inverse: `isDefinitionOf`

13. hasSense: it associates LexicalEntry with Sense.

- Domain: `LexicalEntry`
- Range: `Sense`
- Inverse: `isSenseOf`

14. belongsToLanguage: it associates language origin with a number of classes.

- Domain: `LexicalEntry`, `Lexicalization`, `Sense`, `Source`, `Definition`, `UsageContext`, `Note`
- Range: `Language`
- Inverse: `hasLinguisticExpression`

15. hasContext: it links contextual information with word forms and lexemes.

- Domain: `LexicalEntry`, `Lexicalization`
- Range: `UsageContext`
- Inverse: `isContextOf`

16. isRelatedTo: this property denotes a general notion of lexical semantic relatedness between senses.

- Domain: `Sense`
- Range: `Sense`

This relation has been further specified in order to capture more fine-grained distinctions between senses within and across languages. The subtypes of `isRelatedTo` relations are the following:

- `isEquivalentTo`: to identify near-equivalent senses
- `subsumes` and `isSubsumedBy`: to represent partial equivalence between senses, in which one of them makes a more fine-grained or coarse-grained description of the same concept
- `isDisjointWith`: to define the relation between senses that are intensionally very similar, whereas extensionally they apply to different referents.

17. hasLanguageCode: this relation has been imported from FAO's language-code ontology (http://www.fao.org/aims/aos/languagecode.owl). It links the FAO `Language` class to the FAO `LanguageCode` class with its subclasses ISO639-1 and ISO639-2.

- Domain: `Language`
- Range: `LanguageCode`
- Inverse: `isCodeOf`

18. hasAntonym: lexical semantic relation between the lexical entries expressing semantic opposition. WordNet distinguishes between the lexical relations synonymy (see no. 2) and antonymy on the one hand, which depend on the lexemes involved in the relation, and conceptual relations between synsets on the other, which do not depend on the lexemes that constitute the synsets.

- Domain: `LexicalEntry`
- Range: `LexicalEntry`
- Inverse: `isAntonymOf`

11.2 LIR Technological Support

The LIR model has been implemented in OWL in a joint effort of researchers at the Natural Language Processing Group of the University of Sheffield and researchers at the Ontology Engineering Group of the Universidad Politécnica de

Madrid. The OWL code is available in the following URL http://gate.ac.uk/gate-extras/neon/ontologies/lir1.7.owl

This version of LIR is supported by the LabelTranslator system[6], a plug-in of the NeOn Toolkit[7]. The version of the LIR implemented in LabelTranslator contains all the classes and properties described in the previous section, except for the subtypes of the `isRelatedTo` relation between senses, which have not been updated at the time of writing this document. In this way, the LIR model complies with the accessibility requirement (R15) in section 10.4.

Figure 11.3 shows a snapshot of the NeOn Toolkit ontology editor. The ontology tree can be visualized on the left hand side of the window. The right hand side part is reserved to the different Entity Properties views (Class Restrictions, Taxonomy, Annotation, Source View, and Linguistic Information). The one that interests us is the Linguistic Information Entity Properties view, which is the one activated in the image, because it contains the whole set of classes and properties that make up the LIR.

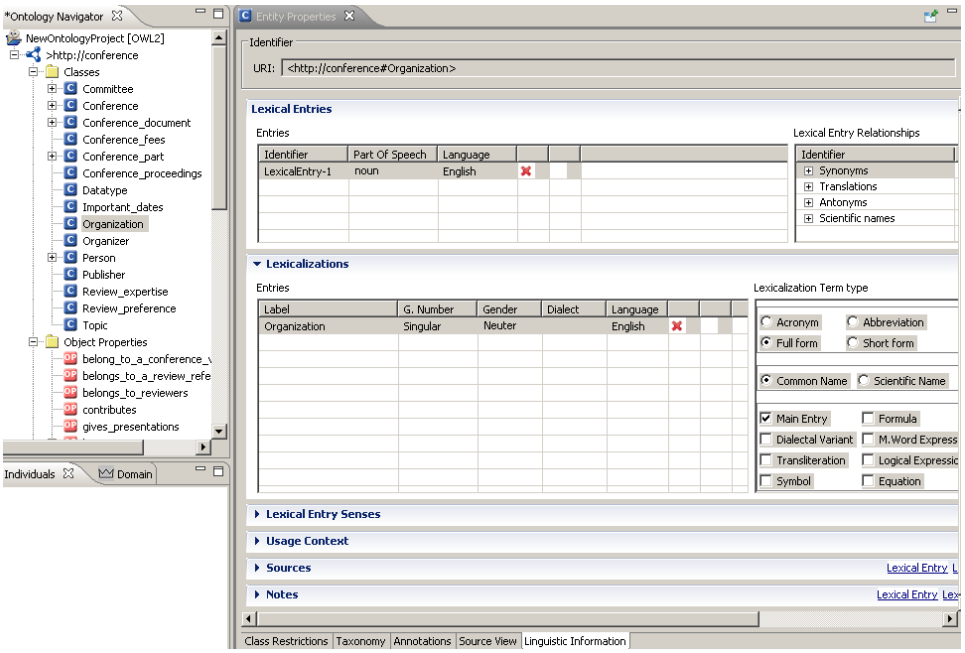


Figure 11.3: LabelTranslator linguistic information entity properties view

When the ontology user creates/imports a new OWL ontology in the NeOn Toolkit, the LabelTranslator plug-in automatically builds an empty linguistic model associated to the ontology under consideration.

LabelTranslator has been created with the aim of automating the process of ontology localization, and is accurately described in (Espinoza et al., 2008a), (Espinoza

[6] http://neon-toolkit.org/wiki/LabelTranslator

[7] Version 2.3 of the NeOn Toolkit can be downloaded from http://neon-toolkit.org/wiki/Download

et al., 2008b) and (Espinoza, Gómez-Pérez, and Montiel-Ponsoda, 2009). The languages supported by the current version of the plug-in are Spanish, English and German. LabelTranslator takes as input an ontology whose labels are described in a source natural language and obtains the most probable translation of each ontology label in a target natural language. Basically, the system relies on a translation component which automatically obtains translations for each ontology label (name of an ontology term) by consulting different linguistic resources. In its current version, LabelTranslator accesses multilingual lexical databases (EuroWordNet), bilingual dictionaries (Wiktionary[8], IATE[9]), translation services (GoogleTranslate[10], BabelFish[11], FreeTranslation[12]), and other ontologies available on the Web. After that, a ranking method is used to sort each candidate translation according to the similarity with the lexical and semantic context of the original ontology label. This means that the ranking method compares the resulting translations and associated definitions (synsets, etc.) with the semantic context of the ontology label in question, i.e., with the labels of its superclasses, subclasses, attributes, sibling concepts, and any additional descriptions or comments in natural language. In short, it can be stated that the LIR is used by LabelTranslator to store the linguistic information it obtains as a result of the translation process of ontology labels.

Additionally, the LabelTranslator plug-in provides the LIR model with accessibility to external resources from which information can be automatically obtained to populate its classes. However, it should be noted that in the present version LabelTranslator only obtains translations for the labels in the original ontology, and definitions related to those translations, whenever they are available in the accessed resources. Finally, the translation candidates automatically selected by the system (or in a supervised scenario, by the human translator) are stored in the LIR model.

The rest of linguistic information captured in the LIR has to be manually introduced by the user or translator, if so required by the final application. In the same sense, if the system does not support the language combination we are interested in, we can still use LabelTranslator to take advantage of the LIR API implemented in the NeOn Toolkit. This means that we can manually introduce linguistic information in the LIR in any language[13].

Figure 11.4 shows the Linguistic Information Entity Properties view associated to the sample ontology label *FAO*. LabelTranslator completes in runtime the fields of the Linguistic Information view according to the information obtained by the system in the translation process.

Initially, the linguistic information page shows five sections that correspond to

[8] http://en.wiktionary.org/wiki

[9] http://iate.europa.eu

[10] http://www.google.com/translate t

[11] http://babelfish.altavista.com

[12] http://ets.freetranslation.com

[13] It should be noted here that in the development of the LIR model only European languages have been taken into consideration. Some properties of other languages may be missing and would require an extension of the current model.

the lexical entries of the selected ontology element (FAO in our example) and the associated information of each lexical entry: *lexicalizations*, *lexical entries senses*, *usage contexts*, *sources*, and *notes*.

In the example given, three lexical entries (`LexicalEntry-1`, `LexicalEntry-2`, and `LexicalEntry-3`) are associated with the same concept (`FAO:Class`). Two lexical entries (`LexicalEntry-1` and `LexicalEntry-2`) belong to the same `Language` (English), whereas the third lexical entry (`LexicalEntry-3`) belongs to Spanish. The two English lexical entries are considered synonyms, and translations of the Spanish lexical entry. This information is shown in the field *Lexical Entry Relationships*.

Of course, every time that the user chooses a new entry, the interface automatically displays the information correlated in the different sections. Thus, in our example `LexicalEntry-1` includes two lexicalizations whose labels are *FAO* and *Food and Agriculture Organization*, respectively. *Food and Agriculture Organization* has acronym *FAO*, and, moreover, it is considered a *common name* (in opposition to *scientific name*) and a *multi-word expression*. This information is shown in the fields *Lexicalization Term Type* and *Lexicalization Variants*.

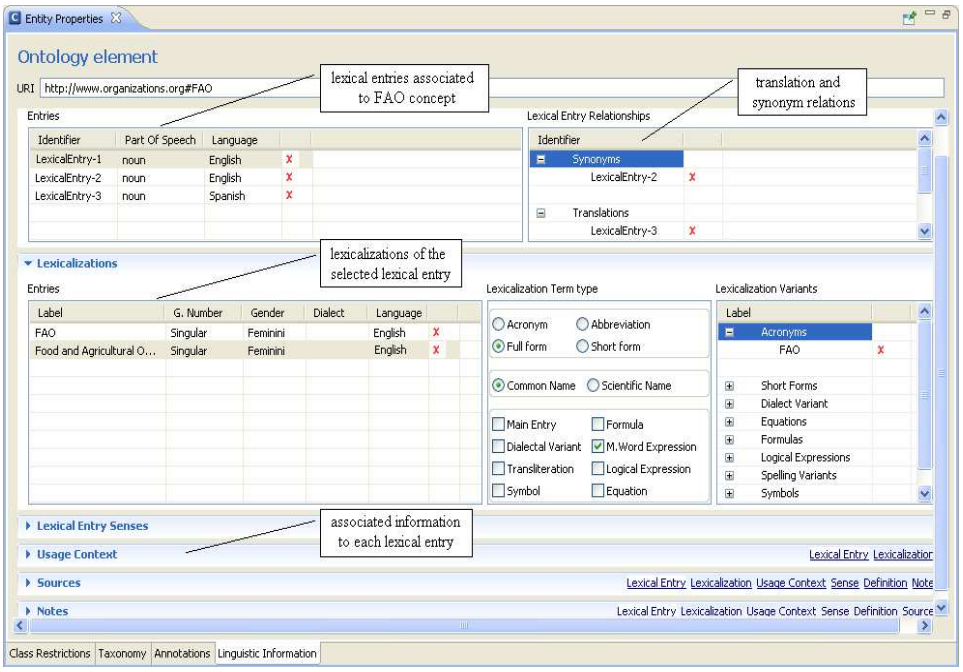


Figure 11.4: Instantiation of the LIR in LabelTranslator

11.3 LexOMV: Multilingualism at the Metadata Level

In this section we present LexOMV, a set of terms and descriptions that serve the objective of describing the linguistic and multilingual information associated to

ontologies at the metadata level (Montiel-Ponsoda et al., 2007). LexOMV was proposed as an extension to OMV, the Ontology Metadata Vocabulary, a vocabulary which consists of a common set of terms and definitions for the description of ontologies with the aim of improving the search, accessibility and reuse of ontologies for the Web.

The OMV is a standard for describing ontologies developed by the joint work of researchers at the AIFB Institute, University of Karlsruhe, and at the Ontology Engineering Group, Universidad Politécnica de Madrid. The main purpose of this research was to create a metadata vocabulary "reflecting the most relevant properties of ontologies for supporting their reuse" (Hartmann et al., 2006). By means of this standard, ontologies are annotated, which in turn implies the existence of tools and metadata repositories that support the "engineering process, maintenance and distribution of ontologies" (*ibidem*).

As in every process of proposing and approving a standard, the requirements the ontology metadata should comply with were analyzed in the first place. Those requirements took into consideration that the metadata should be *understood* by humans (by usage of natural language concepts) as well as by machines (by usage of Semantic Web languages). It should cover the needs of the majority of ontologies without loosing sight of particular application scenarios in which extensions should also be possible. Furthermore, in order to make the reuse and exchange of ontologies effective and efficient, the ontology metadata should provide not only general information of the ontology (e.g. name, description, date of creation, etc.) but also statistical metrics such as the size and structure of the ontology, applicability information (i.e. intended usage or scope), location (e.g. URL), information about the physical representation such as the language and syntax of the formalization, provenance and information about relationships with other resources (e.g. import ontology). Finally, to ensure and facilitate the interoperability of OMV among machines and applications, it is represented as an ontology in OWL.

Therefore, and taking all these requirements into account, OMV was designed modularly. It defines a core and allows the creation of various extensions. Some of the main classes and properties of the OMV Core can be observed in figure 11.5. As we can see in that figure, OMV provides information about the `Person` or `Organization` that created the ontology, the `Type` of ontology, the `OntologyLanguage` or the `Methodology` followed for its development, as well as data about the `KnowledgeRepresentationParadigm` it uses, the `EngineeringTool` with which it was created, or the `Task` for which the ontology was originally conceived.

The OMV Core covers the majority of available information about ontologies. Nevertheless, OMV can also reflect the specificities of a particular ontology task or application by the development of OMV extension modules. According to this, we proposed one of such extensions for covering information about linguistic and multilingual data contained or associated to ontologies. It was ratified and accepted by

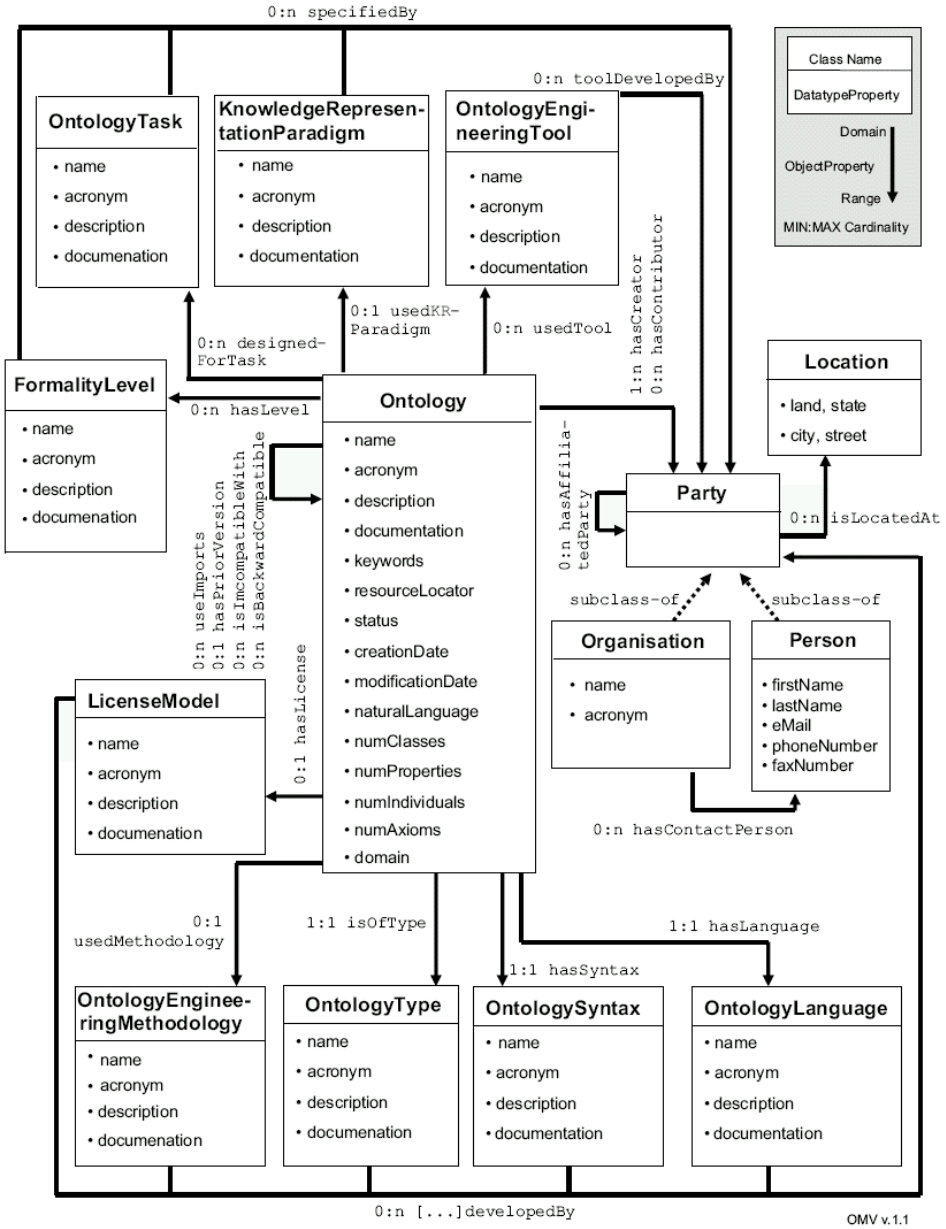


Figure 11.5: OMV core v1.1

OMV authors and implemented in OWL as one of the official OMV extensions[14].

According to the OMV philosophy, the purpose of the metadata collected in the OMV is to offer ontology users a general description of available ontologies to enable an efficient identification of what they are looking for. In that sense, the

[14]http://sourceforge.net/projects/omv2/files/OMV%20Extensions/lexomv_v0.9.owl/download for downloading version 0.9

foreseen increase of multilingual ontologies needs also to be reflected at this metadata level. Hence, our proposed extension to the OMV Core, LexOMV, in which we aim at capturing the general information about the linguistic and multilingual data present in the ontology.

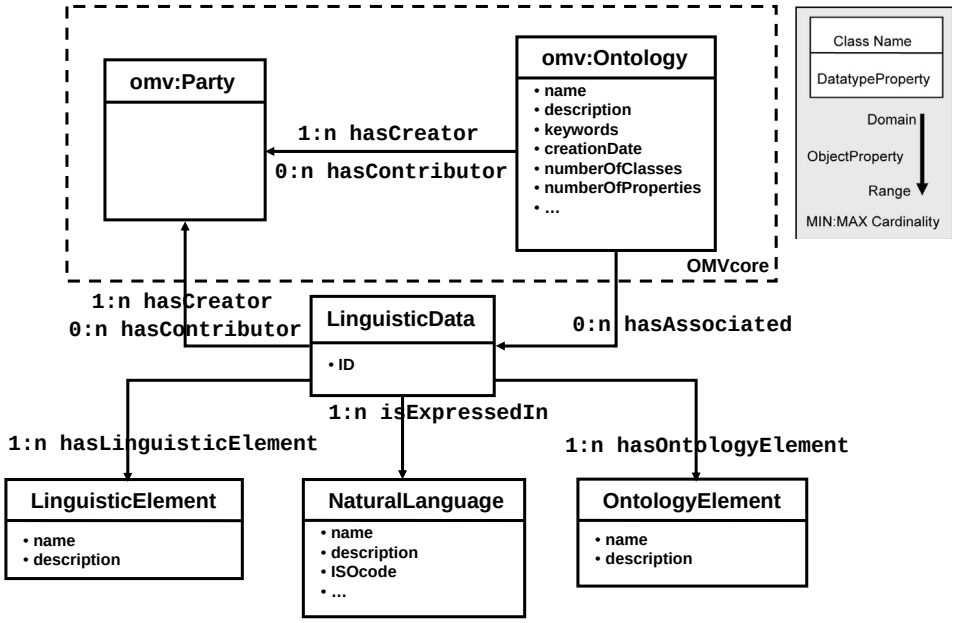


Figure 11.6: LexOMV

In the following we provide a description of the classes and relations created for this purpose:

1. LinguisticData: this is the class that connects the LexOMV extension with the OMV core. This class has one property, which is an identifier, `ID`.

2. OntologyElement: the class `OntologyElement` allows us to make separate statements about the different elements in an ontology that have linguistic information associated to them. The properties `name` and `description` refer to the type of the ontology elements (*classes*, *properties*, *individuals*, etc.). In this way, the model foresees the description of ontologies following different ontology representation paradigms.

3. NaturalLanguage: this class allows us to identify the different natural languages in which the linguistic elements are expressed. For this aim it contains the attributes `name`, `description` and `ISOcode`.

4. LinguisticElement: the class `LinguisticElement` defines the linguistic descriptions associated to the ontological entities. It includes the property `name`, referring to the name of the linguistic element, for example, *definition*, *lex-*

icalization, or *part of speech*. It also allows to account for a description of the linguistic element in question, i.e., what is understood under *definition*, *lexicalization*, or *part of speech* in a certain linguistic model.

Therefore, in order to express that the piece of linguistic data in question (let us say, *Definition*) is expressed in three languages (e.g. *English, Spanish and French*) for a certain type of ontology element (e.g., *Class*) in a given ontology, we link the ontology (described in the OMV Core) via the `hasAssociated` relation to the `LinguisticData` class where we integrate all the necessary information using: `hasOntologyElement` property to relate the `Class` ontology element, `hasLinguisticElement` property to relate the `Definition` linguistic element and `isExpressedIn` to relate the English, Spanish and French languages.

The description property of the `LinguisticElement` class offers the possibility of defining the quantity and quality of linguistic data provided by the linguistic element in question. For instance, and following with our example of the `LinguisticElement` *Definition*, it could be defined as "a language-specific unit of intensional lexical semantic description" in a certain linguistic model. In the same sense, part of speech could be defined as "the grammatical class of the lexicalization", and so on. By means of that description property in natural language, the user is made aware of the scope and coverage of the linguistic information offered by the `LinguisticElement` class.

Thanks to LexOMV, we inform the user searching for ontologies with linguistic information, of the various types of linguistic data included in the ontology in different languages. Furthermore, our extension allows us to describe who the authors and contributors of those linguistic data were by relating the `LinguisticData` class to the `Party` class of the OMV Core. According to this extension, we can now capture the author name or date of creation of the ontology next to information like "this ontology includes lexicalizations and definitions of ontology classes in English, Spanish and French". Moreover, and as a result of the general approach of this extension, we are able to capture any kind of linguistic information depending on the linguistic model adopted for the ontology.

11.3.1 Closing the circle: multilingualism at data, knowledge representation and metadata levels

Figure 11.7 illustrates the different levels at which multilingualism can be present. In this figure we first identify the two levels at an ontology-based application affected by the inclusion of multilingual data: *knowledge representation* and *data* levels; and, second, at a higher level, the *metadata* level that reports about the data in the ontology.

Depending on the layers implied in the localization activity, the knowledge representation level will be modeled in a different way. In our illustration, we have represented the modeling option in which an external model is associated to the ontology, by including a sample of LIR.

The figure explains graphically how LIR is instantiated for a given domain on-

tology (*GeographyOnto*, in our example) and also for its instances. The upper level of the figure represents how OMV Core and LexOMV are instantiated taking into account the information present in the lower part of the figure. Therefore, LexOMV allows us to make the following assertions about the multilingual data included in the ontology: the GeographyOnto domain ontology has some linguistic elements (specifically lexicalizations and definitions) expressed in Spanish, associated to the ontology element class.

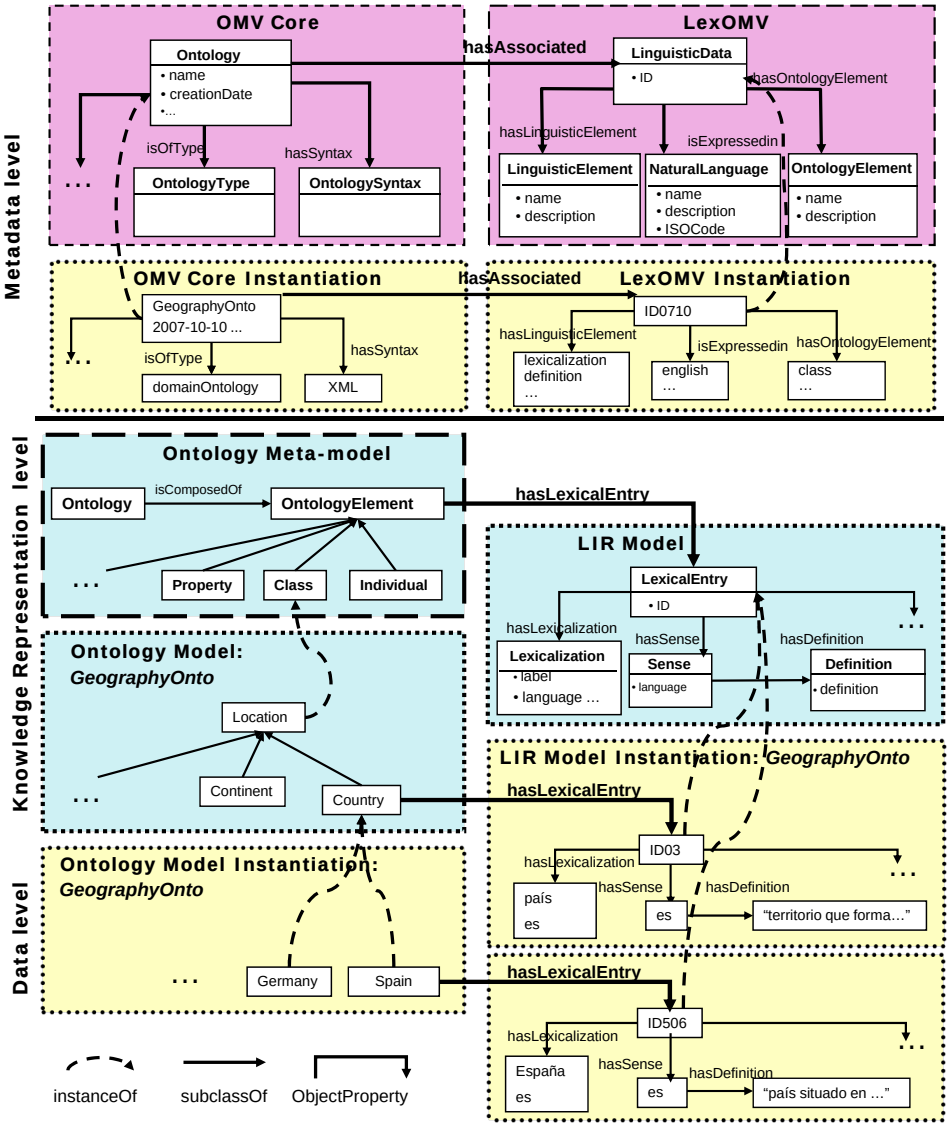


Figure 11.7: Ontology structure levels affected by multilingualism

11.4 Summary

This chapter starts with a detailed definition of the classes and relations that compose the LIR model. We also point to the standards (Data Category Registry, LMF, TMF, etc.) they have been obtained from, and justify its inclusion to comply with the requirements.

Then, we devote some time to present the LabelTranslation system, a plug-in of the ontology editor NeOn Toolkit, in which the LIR has been implemented. LabelTranslator relies on the LIR to store the linguistic information obtained as a result of the localization process that the system supports. This achieves the purpose of associating linguistic information to ontologies in several natural languages.

To conclude the chapter we present LexOMV, an extension to the Ontology Metadata Vocabulary (OMV) that serves the objective of reporting about the multilingual information associated to ontologies at the ontology metadata level. The purpose of this contribution is to guarantee ontology search results according to the linguistic information related to ontological entities.

Chapter 12

LIR Validation

In this chapter, our aim is to describe some experimental initiatives performed with the aim of assessing the validity of the LIR according to the requirements spelled out in figure 10.11, chapter 10.

Two tests have been conducted in this sense. Firstly, the LIR has been validated against the multilingual requirements of an international organization, the Food and Agriculture Organization of the United Nations (FAO), in the framework of the NeOn project. This evaluation is reported in section 12.1.

Secondly, the modeling option offered by the LIR has been compared against the RDF(S) and OWL labelling option (described in section 9.1) by means of an ontology of the Hydrographical domain. With the aim of carrying out this comparison, the same ontology has been implemented according to the two modeling options, as will be explained in section 12.2.

In both cases, the LIR has proven to solve multilingual representation problems related with the establishment of well-defined relations among lexicalizations within and across languages, as well as conceptualization mismatches among different languages.

12.1 Compliance of the LIR against FAO Requirements

The FAO, as many other organizations and institutions operating at an international level, has principally relied on glossaries and thesauri to manage multilingual information with translational and document indexing purposes. In the specific case of the FAO, one of the most used and updated lexical resources has been the AGROVOC thesaurus[1]. AGROVOC can be defined as a controlled vocabulary designed to cover the terminology of all subject fields in agriculture, forestry, fisheries, food and related domains. It was developed by the FAO and the Commission of the European Communities in the early 1980s, and first published in 1982 in three languages: English, Spanish and French. Nowadays, it contains information in more than a dozen languages (English, French, Spanish, Arabic, Chinese,

[1] http://aims.fao.org/website/AGROVOC-Thesaurus/sub

Czech, Japanese, Portuguese, Thai, Slovak, Lao, Hindi, German, Italian, Hungarian), and is under development for some more (Marati, Polish, Korean, Farsi, Malay, Amharic, Catalan and Russian).

In 2003, the FAO initiated the development of the AGROVOC Concept Server (CS) (Liang et al., 2008), an ontology created *ad hoc* from the original thesaurus to add semantics to the information contained in AGROVOC and overcome in this way some of the main deficiencies of thesauri. Although the CS solved some immediate needs, as reported in (Liang et al., 2008), the requirement of a portable model that would enrich any domain ontology created within the organization with multilingual information remained unsatisfied[2]. Thesauri drawbacks experimented by FAO knowledge management experts are listed in the following:

- Thesaurus relationships (Broader Term, Narrower Term, Related Term, USE and UsedFor) fall short of expressing semantic and lexical relations in a refined and precise way.
- Thesaurus relationships do not cover all possible associations between terms in the sense that it is not possible to retrieve and distinguish an *acronym* from a *full form* description, a *synonym* from a *translation*, or a *scientific name* from a *common name*.
- Thesauri do not specify lexical variants for dialects or local languages for a geographical region, such as the ones we could find between Spanish used in Spain and Spanish used in Latin America.
- Thesauri do not allow more than one translation per term to be set. According to this, for example, the English term *Field size* can be translated in French as *Taille des parcelles* or *Dimension des parcelles*. In the current AGROVOC thesaurus one of the translations is assigned as the translation of the descriptor, and the other as an associated non-descriptor. Any divergences or discrepancies in meaning remain hidden, and cannot be explicitly accounted for.

Most of the drawbacks identified here coincide with the localization and interoperability requirements (R4-R14) that we set down for a localization model in figure 10.11, chapter 10. Therefore, we could assume that the LIR model could overcome some of the major limitations of thesauri, on the one hand, and fulfill the needs of portability and association of multilingual information to domain ontologies, on the other. This means that in FAO, not only could several resources such as AGROVOC or the Concept Server benefit from the LIR paradigm, but also recently developed domain-specialized ontologies could take advantage of this model.

In the following we describe with real examples from the AGROVOC Thesaurus how the LIR could solve FAO multilingual needs.

[2]For a description of the alignments between the AGROVOC CS and the LIR model to enable an automatic population of the LIR with the AGROVOC CS data see (Peters et al., 2009).

- Establishment of well-defined relations within lexicalizations in one language (R4, R5, R7, R10)
- Establishment of well-defined relations within lexicalizations across languages (R4, R5, R11, R12)
- Conceptualization mismatches among different cultures and languages (R4, R7, R9, R11, R12, R13, R14)
- Representation of non-native language expressions (R4, R7, R14)

Example 1: Establishment of well-defined relations within lexicalizations in one language. The example in Figure 12.1 concerns the establishment of relations among term variants belonging to the same language. Specifically, this case exemplifies the use of various *acronyms* and *full forms* attached to one and the same concept.

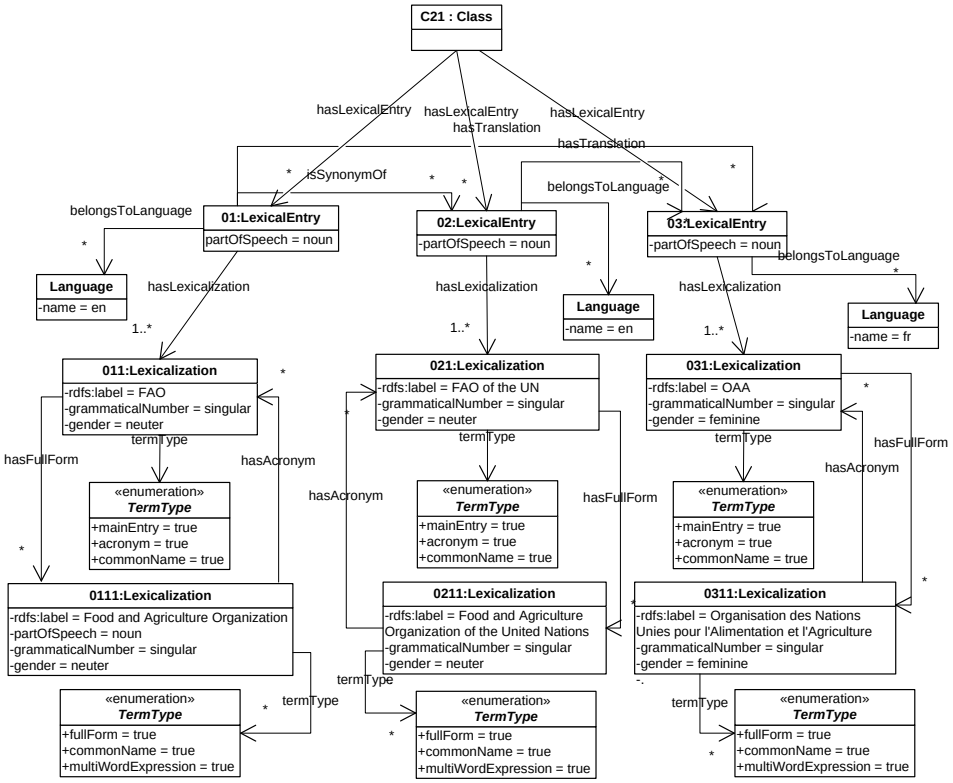


Figure 12.1: Representation of acronyms and full forms within a language

Three lexical entries (`01:LexicalEntry`, `02:LexicalEntry` and `03:LexicalEntry`) are associated with the same concept (`C21:Class`), which means that they are terms that identify one and the same concept. Two lexical entries (`01:LexicalEntry` and `02:LexicalEntry`) belong to English, whereas the

third lexical entry (`03:LexicalEntry`) belongs to French. The two English lexical entries are considered synonyms, and both are translations of the French lexical entry. Each lexical entry contains two lexicalizations. For example, `01:LexicalEntry` includes `011:Lexicalization` and `0111:Lexicalization`, whose labels are *FAO* and *Food and Agriculture Organization*, respectively. FAO is the acronym for Food and Agriculture Organization, and, moreover, it is considered the main entry. *FAO of the UN* and *Food and Agriculture Organization of the United Nations* are deemed synonyms of FAO and Food and Agriculture Organization. Both lexical entries (`01:LexicalEntry` and `02:LexicalEntry`) are translations of *OAA* and *Organisation des Nations Unies pour l'Alimentation et l'Agriculture* in the French language.

Thanks to LIR it is possible to retrieve synonyms within the same language associated with the same concept, and distinguish different term types such as acronyms and full forms.

Example 2: Establishment of well-defined relations within lexicalizations across languages. The second example highlights the possibility given by the LIR model to represent scientific names and use them across languages (scientific names are in Latin and are internationally accepted over scientific communities).

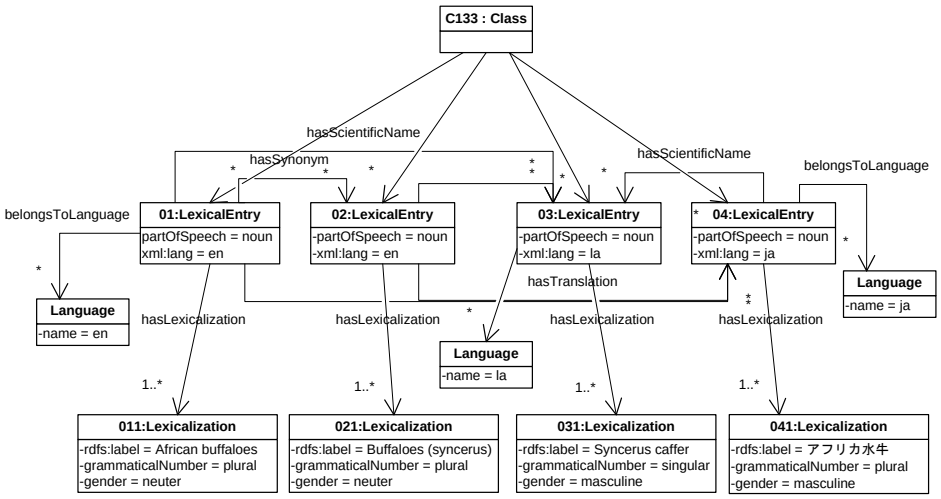


Figure 12.2: Representation of scientific names and common names across languages

Variants in the same language (e.g. *Buffaloes (syncerus)*) can therefore be connected to the same scientific term, such as the English and Japanese translations. We have illustrated in Figure 12.2 how the concept *buffaloes* (`C133:Class`) has four lexical entries associated (`01:LexicalEntry`, `02:LexicalEntry`, `03:LexicalEntry`, `04:LexicalEntry`). Two of them belong to the English language and contain synonymous lexicalizations (`011:Lexicalization` and

021:Lexicalization). Then, we have a lexicalization in Latin that represents the scientific name, and it is accordingly related with the rest of lexical entries by means of the object property hasScientificName. Finally, 04:LexicalEntry belongs to the Japanese language, which is also the common denomination in Japanese of the *Syncerus caffer* scientific name, and, at the same time, the translation of the two lexicalizations in English.

Example 3: Conceptualization mismatches among different languages. More often than not, conceptualizations of the same domain coming from different communities show important discrepancies, because the granularity level with which some concepts are understood may not be the same. This results in a mismatch of terminological equivalents, as reported in section 8.4. The situation can be summarized in two cases: (a) one in which a culture makes a more fine-grained distinction of a certain reality parcel than the other, or (b) the opposite situation, in which a culture does not make so fine-grained distinctions but remains at a more underspecified level.

In order to explicitly express that kind of specificities among cultures, LIR has foreseen the classes Sense, Definition and Note, as well as the relations that specify the isRelatedTo relation among senses (isEquivalentTo, subsumes, isSubsumedBy, and isDisjointWith). Let us imagine the case in which our ontology contains the class *river*. In English, river is defined as a *natural stream of water of usually considerable volume*. To the best of our knowledge, the French language has no exact equivalent, but a different granularity level represented by different terms. On the one hand, the term *course d'eau*, which is slightly more general, and could be considered a translation of *stream of water or watercourse*, and on the other hand, the terms *fleuve* and *rivière*, which are more specific. Broadly speaking, *fleuve* is a river that flows into the sea, whereas *rivière* is a river that can flow into the sea or into another stream.

We have tried to represent the following scenario in Figure 12.3. In this case, the ontology concept, river (C2321:Class), has three lexical entries associated with it (033:LexicalEntry, 031:LexicalEntry, and 030:LexicalEntry). The lexicalization related to the English language is river, whereas there are two lexicalizations in French, *fleuve* and *rivière*. Basically, the three lexical entries correspond to the same object in the real world, as described in the ontology concept. However, LIR captures cultural specificities in the terminological layer by means of a more complex machinery of linguistic classes. In the first place, each lexical entry is assigned to a different Sense class, and a definition in natural language in the Definition class. At the linguistic level, these lexical entries are related by the hasTranslation relation, but at the semantic level the two French senses are related to the English sense by the subsumes relation. This means that the French lexical entries are more specific than the English one. Between them, the two lexical entries are related by the isDisjointWith relation, which means that the individuals that are related to one cannot be related to the other. Finally, the Note class is used to make some comments about the use of

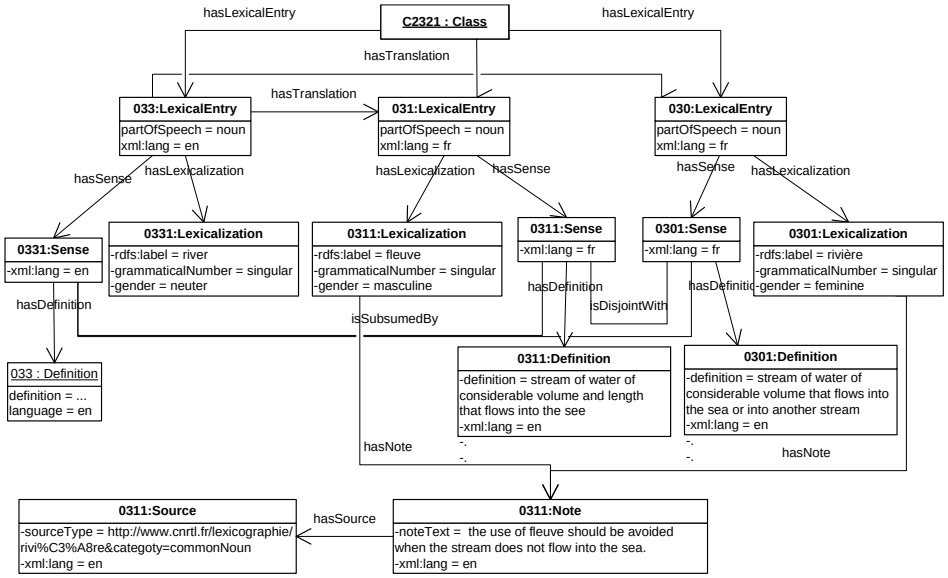


Figure 12.3: Representation of conceptualization mismatches

the lexicalizations.

We should note here that our starting point is a given conceptualization that reflects how a certain community classifies reality. Then, by means of LIR we try to define translations or equivalences of those concepts in other languages. Considering our example of the concept *river*, it would be possible to modify the ontology on the basis of the linguistic information contained in LIR, if deemed necessary by the final application. In this case, two additional classes underlying *fleuve* and *rivière* would be added as subclasses of the concept river. Then, in the English language, we could describe those concepts as "rivers that flow into the sea" or "rivers that can flow into the sea or into other rivers", or we could simply associate the three concepts to the lexicalization *river*. The decision would depend on the needs of the final application.

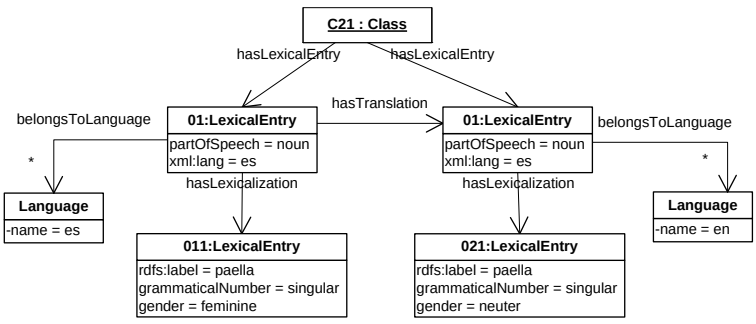


Figure 12.4: Representation of non-native language expressions

Example 4: Representation of non-native language expressions. The last example we want to include here is related to the possibility offered by LIR of expressing that certain lexicalizations belonging to a specific language can be used in another language. This is the case of the Spanish word *paella*, a word also used in other languages such as English and Italian. By using the `belongToLanguage` link provided by the LIR model, we can express that a term is used in a specific country or a specific culture, and using the `xml:lang` attribute we can identify the real language of the term (see Figure 12.4).

12.2 Comparison of the LIR against the RDF(S) and OWL Modeling Option

In this section our aim is to compare the LIR model against the modeling option presented in section 9.1, namely, the RDF(S) and OWL labeling option that permits to include multilingual labels in the ontology model. As already said, the labeling functionality offered by the ontology representation languages OWL and RDF(S) is the most used modeling modality nowadays to document ontologies in natural language. Since the language of the properties can also be specified using the "language tagging" facility of RDF literals (e.g., label@en), ontologies can be enriched with linguistic information in different natural languages, becoming "multilingual ontologies".

In order to demonstrate how some of the major drawbacks of this modeling modality could be overcome by a more complex model of linguistic information such as the LIR, we compared two versions of the same multilingual ontology making use of these two modeling modalities, namely, RDF(S) labeling functionality and the LIR model.

The ontology used for our purposes is *hydrOntology*[3], an ontology of the hydrographical domain developed by researchers of the Ontology Engineering Group at the Universidad Politécnica the Madrid, together with experts of the Spanish National Geographic Institute[4] (IGN-E).

hydrOntology (Vilches-Blázquez et al., 2009) is an ontology in OWL developed with the aim of harmonizing heterogeneous information sources coming from several cartographic agencies and other international resources. Initially, *hydrOntology* was created as a local ontology that established mappings between different data sources (feature catalogues, gazetteers, etc.) of the IGN-E. Firstly, its purpose was to serve as a harmonization framework among Spanish cartographic producers. Later, the ontology evolved into a global domain ontology that attempts to cover most of the concepts of the hydrographical domain.

hydrOntology was originally developed in Spanish, and therefore, the `labels` given to the concepts in the original ontology were in Spanish. Later on, English labels were also related to ontology concepts, and the language of those labels

[3]The OWL code of the ontology can be accessed in http://geo.linkeddata.es/web/guest

[4]http://www.ign.es/ign/es/IGN/home.jsp

was specified by means of language tags. Definitions or glosses describing the concepts were also included in Spanish and English, if available in the resources accessed, by making use of the `rdfs:comment` property. Finally, one metadata element of Dublin Core (`source`) and one additional annotation (`provenance`) were used to report about the resources from which the different definitions (`rdfs:comment`) and labels (`rdfs:label`) had been obtained, respectively. It must be noted that the process of documentation was not systematically carried out for different reasons, and not all types of annotations were available for every concept.

A snapshot of the class hierarchy of *hydrOntology* in the Protégé ontology editor can be seen in figure 12.5. The concept `Río` (river) has been chosen for illustration. It has nine annotations related to it: three `provenance` annotations, two `rdfs:comment` annotations, three `rdfs:label` annotations, and one `source` annotation.

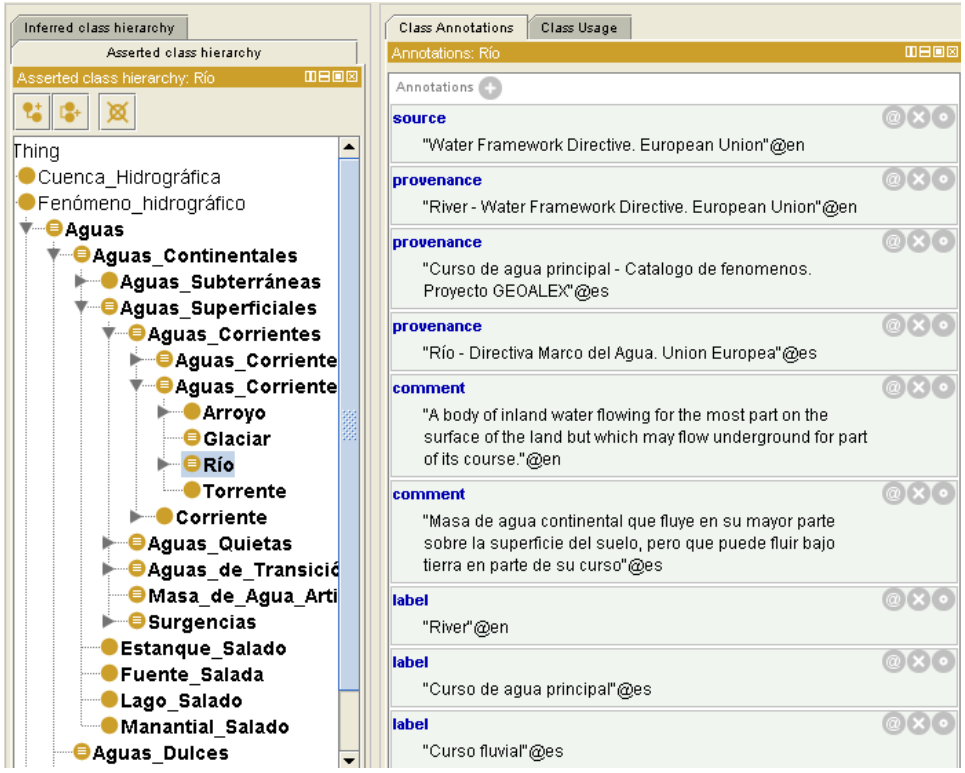


Figure 12.5: Snapshot of the *hydrOntology* hierarchy and class annotation properties in Protégé

As already reported, the provenance annotation gives information about the linguistic resources (glossaries, thesauri, dictionaries, etc.) from which labels have been obtained. Since there are no mechanisms for relating the label (e.g. *River*) with its source of provenance (e.g. *Water Framework Directive*), the authors have

decided to include the label in the provenance text for the sake of clarity (e.g., *River - Water Framework Directive. European Union@en*). Two comments are included, one in Spanish, and one in English, though no relation to any of the labels is given. Finally, three label annotations are given: two in Spanish (in addition to the one given in the URI, i.e., `Río`) and one in English. The two additional labels in Spanish are *Curso de agua principal* (Main Watercourse), and *Curso fluvial* (Watercourse). According to the authors, the main difference among the three synonyms is the discourse register. The label `Río` would appear in general documents, whereas the other two additional labels would only come up in technical documentation managed by experts in the domain. It is worth noting that such fine-grained aspects could be relevant for certain indexing or information extraction tasks, but cannot be made explicit in the RDF(S) labeling functionality.

Regarding the English translation, *River*, it is not possible to know to which of the Spanish labels it is related or of which it is translation. *River* is considered to be in an equivalence (or near-equivalence) relation with `Río`. However, the RDF(S) labeling model does not offer any means to report about those cultural differences that, more often than not, occur between two languages.

The several drawbacks identified in this analysis for an appropriate exploitation of the resulting multilingual ontologies can be summarized as follows:

- All annotations are referred to the ontology element they are attached to, but it is not possible to define any semantic relations among the linguistic annotations themselves. This results in a set of semantically unrelated data.

- When labels within the same language or in different languages are attached to the same ontology element, it is not possible to make explicit which is the relation existing among them.

- Finally, scalability issues will probably arise. If only a couple of languages are involved and not much linguistic information is needed, the RDF(S) properties can suffice. But if a higher number of languages is required, as seems to be the trend in the current demand, the linguistic information will become unmanageable.

The next stage in this comparison was to import the ontology in the NeOn Toolkit to take advantage of the LabelTranslator plugin that stores the linguistic information related to the ontology in the LIR model. In this second version of *hydrOntology*, our purposes were to enrich the ontology already in Spanish and English with two additional languages: French and Catalan. With this aim, we imported the ontology in the NeOn Toolkit, and automatically, all the linguistic classes of the LIR were associated with the concepts and properties in the ontology. However, only the linguistic information of the original ontology in Spanish was automatically stored in the LIR, and not the rest of linguistic information in English that had been included later on. This setback was reported to the developers of LabelTranslator who informed us that only that information associated with the

ontology URIs was automatically instantiated in the LIR. Being that the case, we had to manually introduce the information in English that was already available in the Protégé version of *hydrOntology*.

The following step was the enrichment of the ontology with information in French and Catalan. For this aim we worked together with experts in the domain and resorted to authoritative terminological resources in the domain to manually introduce the information in the LIR by means of the LIR API. For the sake of comparison, we will illustrate the results by taking the concept *river* as example, as in the case of the Protégé version of *hydrOntology*. As shown in figure 12.6, now seven lexical entries with part of speech noun were associated with the concept `Río`: three in Spanish, one in English, one in Catalan and two in French. By clicking on each Lexical Entry we are able to visualize the rest of the linguistic information associated with it: *lexicalizations*, *senses*, *usage contexts*, *sources* and *notes*.

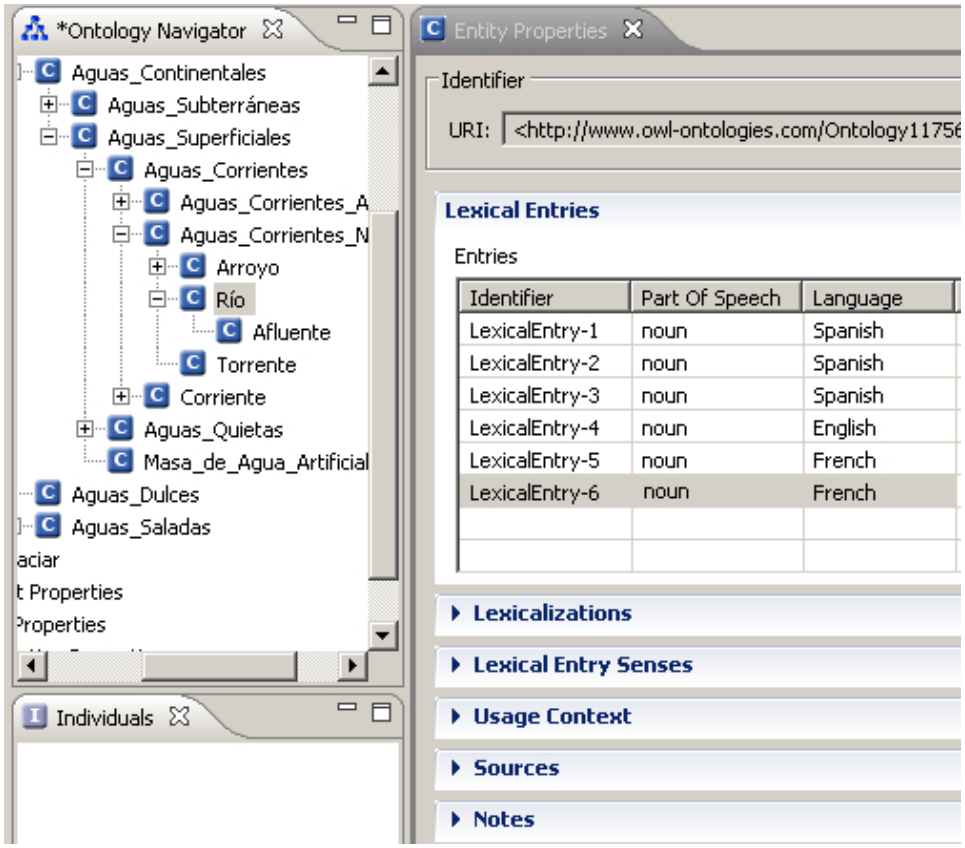


Figure 12.6: Linguistic information associated with *Río* in the LIR model

The three Lexical Entries in Spanish (*Río*, *Curso de agua principal*, and Curso fluvial) are related by means of the `hasSynonym` relation (see figure 12.8 for *Lexical Entry Relationships*). The differences in use depending on register (formal

vs. informal) are explained in the Note class. The Senses of these lexical entries are related by means of the `isRelatedTo` relation, although in future versions of the LIR we expect this to be done with the `isEquivalentTo` subtype of the relation. Then, the three Lexical Entries in Spanish are related to the Lexical Entry in English (*River*), the one in Catalan (*Riu*), and the last two in French (*Rivière* and *Fleuve*) by means of the `hasTranslation` relation (see figure 12.8). The lexical entry in English and the lexical entries in Spanish are considered equivalents in meaning, and the same happens with the Catalan equivalent. Therefore, their senses could also be related by the equivalence relation `isEquivalentTo`.

The two French lexical entries represent two more specific concepts, as already reported in section 8.4, which would stay in a relation of subsumption with the Spanish *Río*, the Catalan *Riu*, and the English *River*. This is an example of conceptual mismatch. The French understanding of river has a higher granularity level and identifies two concepts which are intensionally more specific, and extensionally do not share instances. Therefore, in order to make explicit those differences in meaning, the two lexical entries would be related to two different senses, and definitions in natural language would also be provided for each of them. Figure 12.7 shows some elements of the lexical information that can be related to each lexical entry. In this example, one lexical entry in French (`LexicalEntry-5`), whose lexicalization is *Rivière*, has one sense related to it (`Sense-1`), and its corresponding definition in French.

And, finally, Figure 12.8 shows how the relations of synonymy and translation are explicitly established among lexical entries within the same language and across languages.

By means of the further specifications of the `isRelatedTo` relation among senses we would account for categorization discrepancies among languages, which are not simply motivated by the fact that there are more lexicalizations in one language than in another, but by the different granularity levels that cultures make of the same world phenomenon. One could argue that these language specificities are only captured in the terminological layer of the ontology, but not in the conceptual model. However, this may suffice for certain ontology-based tasks such as information extraction or verbalization, whereas it may be insufficient for others. In that sense, a modification of the conceptualization to adapt the specificities of a certain language could be directly carried out by considering the lexical and terminological information contained in LIR.

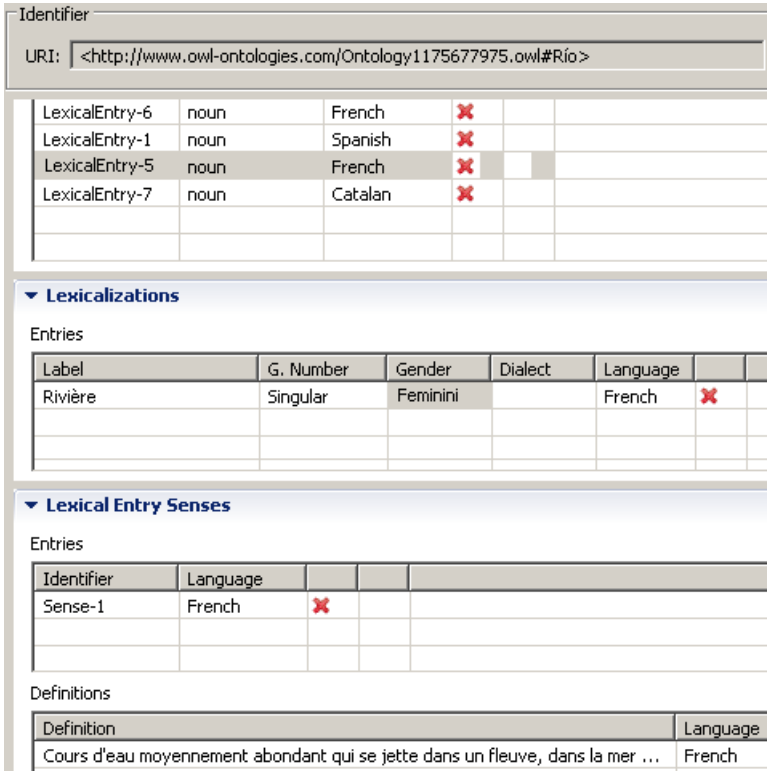


Figure 12.7: Linguistic information associated with the lexical entry *Rivière*

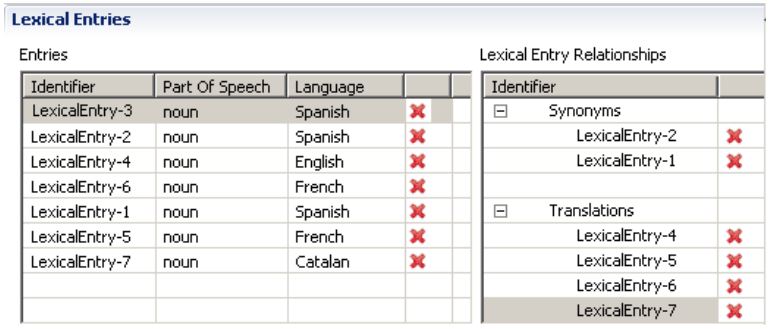


Figure 12.8: Relations of synonymy and translation among labels

12.3 Summary

The objective of this chapter was to show how the LIR satisfies the requirements laid down for an ontology localization model by means of two validation tests.

The first one involved an analysis of the requirements of an international organization that deals with terminological resources in multiple natural languages, the Food and Agricultural Organization or FAO. By means of some specific examples, we demonstrate how the LIR would handle the representation problems faced by

such organizations in regard to the establishment of well-defined relations in lexicalizations within and across languages, as well as conceptualization mismatches.

The second test described the specific case of an ontology of the hydrographical domain, *hydrOntology*, which additionally makes use of the technological support provided by the LabelTranslator NeOn Toolkit plug-in to illustrate how multilingualism issues are represented in the LIR.

Chapter 13

Conclusions and Future Research Lines

In this last chapter, we present the conclusions of this thesis. First, we summarize the main contributions with respect to the state of the art, pointing out the features that we consider the most relevant and innovative in our work. This is followed by an evaluation of the results and an account of open research problems, which leads to the proposal of future lines of work.

13.1 Main Contributions

In this thesis we have presented two approaches to deal with multilingualism at two different stages of the ontology development process. The first approach is centered on the knowledge acquisition and ontology modeling activities. In this context, we have proposed a multilingual repository of LSPs associated to ODPs, and a method for the reuse of ODPs to model ontologies. The repository is sustained on a manual analysis of the semantics conveyed by the linguistic structures captured in the LSPs on the light of the LCM. The method is intended for newcomers to Ontology Engineering, and allows users to formulate in NL what they want to model in the ontology.

The second approach takes place once the ontology has been modeled, and deals with associating multilingual information to the original ontology. The model we propose is to be associated with ontologies already developed within a certain linguistic and cultural community. By including information in additional natural languages, the model makes those ontologies reusable in different linguistic and cultural settings.

Each approach has required an independent analysis of the state of the art and has resulted in a set of specific contributions, which have been validated in some preliminary experiments. Each of the contributions will be summarized in separate sections, in which the main innovative features will also be pointed out.

13.1.1 Multilingual LSPs-ODPs Pattern Repository

A repository containing LSPs associated to ODPs has been proposed in this work. It contains patterns in English and Spanish. The main function of this repository is to match NL formulations produced by users while developing an ontology to the ODPs that better model those formulations. This is achieved in a semi-automatic way, by relying on a NLP application. We have particularly worked on:

- The identification of NL expressions that express the knowledge captured in some ODPs. In this sense, we have focused on verbs as main relation conveyors, and have obtained an initial list of candidate verbal patterns. From that initial set of verbs and verbal phrases, we have analyzed the ones that displayed a polysemic behavior with the lexical template we have proposed and that results after combining the lexical template provided by the Lexical-Constructional Model and the Generative Lexicon machinery. This analysis has allowed us to define the deep semantics of the arguments and events involved in the verbal patterns, and to establish a reliable correspondence between LSPs and the ODPs that better model their semantics. The different steps followed in the creation of the LSPs-ODPs pattern repository have been summarized in figure 5.4.

- The implementation of the English LSPs collected in the repository for its use in the processing of user formulations in NL. This implementation has been carried out in GATE, the General Architecture for Text Engineering, and has resulted in a set of JAPE rules. These rules are used by an application created in GATE that we have called LSPs application, and that generates annotations on the sentences produced by the user, and recommends a modeling solution.

- The publication of the English LSPs and their corresponding JAPE rules in the Ontology Design Patterns Portal. The purpose of this is to make LSPs and their corresponding code available for the Ontology Engineering and the NLP Communities.

13.1.2 Method for the Reuse of ODPs

After having analyzed several approaches on knowledge acquisition based on linguistic patterns, and other approaches on CLs to facilitate untrained users the process of ontology modeling, we have proposed a method that can guide novice users in the acquisition of knowledge and in the activity of ontology modeling by reusing ODPs.

The main benefits of this method are:

- It guides novice users in the formulation of the knowledge that they want to include in the ontology, taking as starting point the ORSD, specifically the set of CQs included in that document. In this sense, some recommendations are provided to the user accompanied by examples of sentences.

- It allows users to express what they want to model in the ontology in natural language. As a consequence of that, it prevents users from having to understand logic formalisms or learning a controlled language.

- It enables the reuse of ODPs, which are considered consensual design solutions by the ontology engineering community. In this sense, novice users can rely on best practices when reusing ODPs. This contributes to the quality of the final ontology.

- The method has been thought for users without much experience in ontology modeling. Methods intended for non-experienced users are more and more demanded by the ontology engineering community with the final aim of bringing ontologies closer to the average user. This would have a great impact in the consolidation of the Semantic Web, because it would contribute to the adoption of ontologies by wider communities of users.

13.1.3 Ontology Localization

We have explored the impact of the localization activity in ontologies, and have analyzed different theoretical and practical issues involved in this activity.

- We have applied functional theories to translation to the characterization of the ontology localization problem. As a result, we have identified three dimensions that need to be considered before starting any ontology localization process. These are: (a) function of the localized ontology, (b) domain type represented in the ontology, and (c) interoperability issues.

- Depending on the dimensions identified for each localization process, we propose a set of translation strategies for the localization of ontologies.

- The layers that can be involved in the localization activity have as well been identified. This has enabled a systematic analysis of several possibilities for representing multilingual information in ontologies depending on the needs of the localization process.

- We have detailed the requirements that, to the best of our knowledge, a model for the localization of ontologies should have. These requirements take into account representation, interoperability, localization and accessibility issues. These requirements could be taken as starting point for the design of any model that aspires to associate multilingual information to ontologies.

13.1.4 LIR Model

In this thesis we have proposed a model that is to be published with domain ontologies and that provides linguistic description elements to ontology classes and properties. The model complies with the following requirements:

- The LIR model is kept separated and independent from the conceptualization. Both, linguistic model and conceptualization are self-contained and can be fully developed. In the case of the linguistic model, this means that it can contain as much linguistic information as required by the final application.

- The model is interoperable with existing standards for the representation of lexical and terminological information, namely, TMF, LMF and SKOS. This means that it can be instantiated with information encoded by those standards, and that can be extended with further description elements captured in those standards.

- Regarding linguistic and localization issues, the main features of this model are summarized in the following:

  - It allows the establishment of well-defined relations within lexicalizations in the same language.
  - It allows the establishment of well-defined relations between lexicalizations across languages.
  - It accounts for the representation of culturally-dependent senses, that do not completely overlap with the concept as represented in the ontology.

- The LIR model has been implemented in the LabelTranslator plug-in of the NeOn Toolkit ontology editor, as reported in section 9.2, chapter 9.

- The validity of the LIR model has been assessed against the multilingual requirements of the FAO, an international organization with multilingual needs, and by comparing it against the RDF(S) and OWL labeling option by means of an ontology of the hydrographical domain.

- An extension to the OMV has been proposed to the authors of the OMV to account for multilingualism at the ontology metadata level. This extension comes to palliate the lack of reporting possibilities offered by OMV with regard to the linguistic and multilingual information associated to ontologies. The proposed extension has been termed LexOMV. With LexOMV and the LIR we claim that we provide multilingualism at the three levels of an ontology-based application, namely, metadata, knowledge representation and data.

13.2 Evaluation Results

In this section our aim is to comment on the results we have obtained from the experimental evaluations carried out on the most important contributions of this work. This analysis will enable us to suggest some future research lines.

13.2.1 Method for the Reuse of ODPs

In this section we will make a distinction between the methodological approach adopted for the reuse of ODPs (chapter 6) and its technological support, the LSPs application created in the GATE Architecture (chapter 7). Both contributions have been validated in a hands-on activity with students attending a course on "Ontologies and the Semantic Web" at the Universidad Politécnica de Madrid, as reported in section 7.3.

I. **Methodological approach.** Regarding the methodological guides provided to novice users, we believe that the results from the experiment reported in section 6.3 encourage us to refine and enhance the method in future work.

- The reaction of the participants was positive regarding the help provided by the guides, which proves that methodologies are effective *devices* for improving the performance of unexperienced users in any activity.
- Users agreed that starting from CQs was very useful. They found that in this way the NL formulation of the domain aspect to be modeled was highly simplified. This confirms our assumption that the formulation of CQs within the Ontology Requirements Specification activity prior to the employment of our method is considered particularly helpful for this task. As already described in section 4.3, CQs are formulated by domain experts and ontology engineers. This helps particularly domain experts to parcel their domain of knowledge in small and manageable bits that are to be modeled in the ontology. In this sense, we argue that some guidelines should also be provided for novel users to formulate CQs on their own. As reported in section 4.2.3, the XD method for the reuse of ODPs aimed at ontology engineers in general provided some subtasks for the formulation of CQs. These subtasks propose users to start from "requirement stories" as a first step in the formulation of CQs. We also believe that it would be convenient to provide some guidelines in this regard.
- Additional help in the NL formulation task was provided by the *Recommendations* table (see section 6.1). The participants of our experiment found them very useful but demanded more illustrative examples of sentences that should be produced by them from CQs. We should work on this in the future.
- For the rest of tasks envisioned in our method, namely, Task 2. Input Refinement and Task 3. Pattern Validation, we may need to work on additional guidelines. These tasks could not be evaluated because of the lack of technical support. Further technological and methodological support will be required in this sense.

- In the case of Task 2. Input Refinement, some strategies have been envisioned for providing support in the performance of this task. However, we are in favor of investigating automatic or semiautomatic techniques for identifying correct modeling solutions when several ODPs match the input sentences. We have already outlined the possibility of accessing external resources such as lexicons (WordNet) or ontologies (through the Watson Semantic Web search engine) available on the Web to find out how the same modeling issue has been solved in other resources. This should be further investigated in future work.

II. **LSPs application.** This application was developed in the GATE Architecture and made use of some processing resources, as detailed in section 7.1.

- The results obtained from the evaluation of the sentences produced by the participants of our hands-on activity are encouraging, since we obtained 86.2% of good matchings. Taking into account the difficulties involved in Natural Language Processing (language ambiguities, anaphora, etc.), the performance of our LSPs application was good.
- Wrong annotations accounted for 13.8% of the total amount. The causes for the wrong annotations produced by the application are mainly three: (a) the user produces incorrect input sentences (misspellings, grammatical errors, etc.), (b) the employed processing resources produce wrong annotations or no annotations, and (c) no matching is possible because input linguistic structures have not been identified and formalized in the repository. In order to improve the performance of the application, initiatives need to be taken to approach the different causes of wrongly or absent annotations. Regarding cause (a), we argue that an automatic correction of input sentences would be required to avoid errors. As far as (b) is concerned, sound processing resources are needed for the languages supported by the application. As regards (c), more effort needs to be put in the extension of the *multilingual LSPs-ODPs pattern repository*.
- Regarding the enlargement of the *multilingual LSPs-ODPs repository*, the Ontology Design Patterns Portal represents a very important initiative because of two reasons. On the one hand, the Portal will allow other users to reuse our patterns. On the other hand, it will also encourage them to contribute to the repository.
- The LSPs application only supports sentences in English. In this sense, it should be extended to support additional natural languages.

13.2.2 Multilingual LSPs-ODPs Pattern Repository

The validation of the English version of the repository has been indirectly validated through the LSPs application. The satisfactory results of the application confirm

the validity of the patterns contained in the repository, although not all patterns were present in the set of CQs used as starting point in the experiment. In this sense, further experiments need to be performed. However, some improvements can already be suggested as a result of this preliminary experiment.

I. The results of the matching between the sentences produced by the participants and the LSPs-ODPs pattern repository suggest that some sentences were not annotated as a result of the LSPs-ODPs repository containing only a restricted number of patterns. This makes us aware of the effort that is demanded by such an approach. It requires for linguists and knowledge engineers to work together in, first, analyzing the semantics of linguistic structures, and, second, finding the most appropriate correspondence to the design solutions in the form of ODPs.

II. With regard to the linguistic analysis performed on candidate linguistic structures with the mechanisms provided by the LCM and the Generative Lexicon in section 5.3, we believe that it produced valuable insights in the semantics conveyed by those structures. Such linguistic models are intended to investigate meaning construction and provide essential mechanisms to account for the relation between semantics and syntax, specially in the case of polysemic linguistic structures.

13.2.3 Ontology Localization

Ontology Localization is a new research field in Ontological Engineering. The approach we explored in the second part of this thesis is embedded in the new paradigms for ontology modeling that are in favor of reusing available resources instead of starting its development from scratch. In this sense, we adopted a practical approach in which an available ontology is reused for being adapted to other cultural and linguistic environments.

We believe that the definition of the dimensions involved in the localization of ontologies, as well as the characterization of the ontology localization problem lay the foundations for further approaches to this activity. Each ontology localization process will have to take these dimensions into account depending on the final function of the ontology, with the aim of finding out which the best strategies for the localization activity are.

We also argue in favor of the validity and relevance of functionalist theories in ontology localization. In this context, it would be desirable to define some methodological guidelines in order to help users in the definition of the dimensions involved in the localization process, as well as the representation possibilities, depending on the domain of knowledge represented by the ontology, and the final function of the localized ontology.

13.2.4 LIR Model

The LIR model comes to palliate the lack of models that permit to represent the complex relation between multilingual information and ontologies. By means of two validation tests we show that this model solves multilingual representation problems related with the establishment of well-defined relations among lexicalizations within and across languages, as well as conceptualizations mismatches among different languages.

These experiments confirm that by means of the LIR the following issues are solved:

- Possibility of including information in as many languages as needed by the final application
- Definition of synonym relations between the linguistic elements belonging to the same language
- Definition of translation relations between linguistic elements in different languages
- Possibility of capturing categorization mismatches between different languages by means of the `Sense` and `Definition` classes

Regarding the possibility of capturing categorization mismatches between cultures, this is currently done by means of descriptions in NL in the `Definition` class, as reported in section 10.1. However, it may be desirable to represent those discrepancies in granularity level also in the conceptualization layer. This could be achieved by extending the ontology with "language-specific modules" that would capture those cultural specificities. For this aim, we think that the `Sense` class could work as an intermediate class between the concept in the ontology and the rest of linguistic information. These ideas need to be further investigated.

13.3 Future Lines of Work

In this section, we identify some features that can be improved to overcome current limitations of our approaches.

I. *Automatic extraction of LSPs to enhance the multilingual LSPs-ODPs pattern repository.* On the light of the encouraging results of the LSPs evaluation, we would like to analyze strategies to automatically learn new verbal patterns that would come to enlarge our LSPs-ODPs pattern repository. Some of the strategies we have applied on a manual basis could be improved and automated to speed up the identification of LSPs.

   A further strategy that we would like to approach involves the automatic transformation of LCM lexical templates into LSPs. The LCM is currently

getting a lot of attention from many fronts because of the holistic proposal it makes to explain meaning construction (see Butler (2009)). Additionally, their authors are also working on the construction of a lexico-conceptual knowledge base called FunGramKB[1] that integrates semantic and syntactic information of verbs that could be exploited for our purposes (see Mairal Usón and Periñán-Pascual (2009) or Periñán Pascual and Arcas Túnez (2010)). This motivates us to investigate ways of taking advantages of previous studies in which LCM lexical templates have been employed to analyze verbs and verbal phrases.

II. *Improvement of the NL Formulation task in the method for the reuse of ODPs.* Regarding the methodological guidelines provided for the activity of ODPs reuse, currently we strongly rely on the results of the Ontology Requirements Specification activity, specifically on the ORSD and the set of CQs in the framework of the NeOn Methodology. Otherwise, the first task in our method (Task 1. NL Formulation) is left to the user's criteria, only supported by some recommendations (see table 6.1 in section 6.1). In this sense, we are already working on the improvement of the Recommendations table for further experiments.

We believe that in this sense a more direct support is needed to control and asses the validity of the user's input. Actions could already be taken at the CQs formulation task. Specifically for the case of novice users, the formulation of CQs could be enhanced by the use of an interactive learning environment in which pedagogical teaching agents guide users along the knowledge acquisition activity (similar approaches are being investigated in the DynaLern project[2], for example).

After that, in the formulation of the NL expressions that are to be transformed into ontological structures, we could also make use of an interface in which the user's input is incrementally parsed. In this way, suggestions of the type of linguistic structures to be employed could be made to novice users. This would also allow us to make suggestions considering the structure and the vocabulary already available in the ontology. Some of these NL interfaces have been proposed for ontology editing (Kaufmann et al., 2006) or question answering (López et al., 2006).

In the case of polysemous LSPs such as the ones analyzed in this research work, the strategies that we envision, in which users interact with the system to solve ambiguities, could also benefit from teaching agents. A further way to improve the NL Formulation task would consist in checking the validity

[1] http://www.fungramkb.com/

[2] http://hcs.science.uva.nl/projects/DynaLearn/

of the user's input in a semi-automatic way by accessing available resources on the Web, such as ontologies, lexicons, terminological resources, etc. This would also allow us to make suggestions to the user so that (s)he could be made aware of other modeling options. We believe that this kind of interaction would have a didactic nature.

III. *Guidelines for ontology localization projects.* In the present research work, we have centered on spelling out the different dimensions that play a role in the localization of ontologies. We argue that these dimensions should be carefully studied for each new ontology localization project. In this context, we believe that users could highly benefit from some kind of protocol or guidelines that would help them find out the most adequate strategies to follow, and the most appropriate multilingualism representation options available to them. They should be made aware of the advantages and disadvantages of each modeling option, and according to their specific requirements, some suggestions should be made to them.

IV. *LIR in the Web of Data.* The model we have proposed to associate multilingual information to ontologies on the Web has the main advantage of being portable and reusable to provide multilingualism to any available ontology. This is even more important in the context of the Web of Data, which has emerged as a result of the Linked Data initiative[3]. Linked Data is a recent initiative that proposes to connect data, information and knowledge on the Web using URIs and the RDF syntax. The potential of Linked Data is that it allows linking data and knowledge on the Web in the same way as documents are connected by the HTML protocol. The benefits that result from this scenario are huge, because those connections can exploit access and interaction possibilities of users with the information.

In this context, multilingualism is going to play a major role, because the data making use of the Linked Data format will be available in different natural languages, and connections will have to be found among them. It is in this specific scenario where models such as the LIR come into scene. Such models will have a twofold impact in the Web of Data. On the one hand, they can provide multilingualism to extant ontologies, so that connections or mappings can be established to other resources in different natural languages. On the other hand, models such as the LIR, which are compliant with ontology languages, can be published in the Linked Data format and be reused to provide multilingualism to other ontologies on the Linked Data cloud, in its totality, or only making use of some of the information contained in them.

Currently, we are participating in the European Project Monnet[4], in which

[3] http://linkeddata.org/

[4] http://www.monnet-project.eu/

the localization of ontologies is a principal issue. The project is concerned with exploiting ontologies in several NLP tasks, such as Cross-Language Information Extraction, Question-Answering or Machine translation (see Mitkov (2003) for an overview of NLP applications). For this aim, the localization of ontologies is crucial, as well as the association of multilingual information to ontologies. In this context, the LIR model has been merged with the LexInfo model[5] and LMF, and has resulted in a new model called *lemon* (lexicon model for ontologies). The lemon model provides not only lexical and terminological information, but also morphological decomposition and syntactic behavior to arbitrary ontologies on the Web. We believe that this joint effort can definitively contribute to the vision of a truly multilingual Web of Data.

[5] http://lexinfo.net/

References

Adriaens, G., and Schreurs, D. (1992). From COGRAM to ALCOGRAM: Toward a controlled english grammar checker. In *Proceedings of the 14th international conference on computational linguistics (COLING92).*

Agirre, E., Ansa, O., Hovy, E. H., and Martínez, D. (2000). Enriching very large ontologies using the WWW. In S. Staab, A. Maedche, C. Nedllec, and P. Wiemer-Hastings (Eds.), *Proceedings of the Workshop on Ontology Learning of the European Conference on Artificial Intelligence (ECAI00). CEUR Workshop Proceedings* (Vol. 31, p. 25-30). Berlin.

Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., and Suárez-Figueroa, M. C. (2008). Natural language-based approach for helping in the reuse of ontology design patterns. In *Knowledge Engineering: Practice and Patterns, Proceedings of the 16th International Conference on Knowledge Engineering (EKAW08)* (p. 32-47).

Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., and Suárez-Figueroa, M. C. (2009). Using linguistic patterns to enhance ontology development. In J. L. Dietz (Ed.), *Proceedings of the international conference on knowledge engineering and ontology development (KEOD 2009)* (p. 206-213).

Aguado de Cea, G., and Lorente Enseñat, A. (1997). *Software localization: problemas lingüísticos y socioculturales.* Available from `http://www.universoabierto.com/basedatos/ficha.php?id=3959`

Aguado de Cea, G., and Álvarez de Mon, I. (2006). Estructuras de clasificación en español. terminología y adquisición de conocimiento explícito para la web semántica. In *Proceedings of the 5th internacional AELFE conference* (p. 492-: 498). Universidad de Zaragoza, Zaragoza.

Aguado de Cea, G., Montiel-Ponsoda, E., and Ramos Gargantilla, J. A. (2007). Multilingualidad en una aplicación basada en el conocimiento. *TIMM, monográfico para la revista SEPLN*, *38*, 77-97.

Alarcón Martínez, R., and Sierra Martínez, G. (2003). The role of verbal predications for definitional contexts extraction. In *International Conference on on Terminology and Artificial Intelligence (TIA03)* (p. 11-20).

Alarcón Martínez, R., Sierra Martínez, G., and Bach Martorell, C. (2008). ECODE: A Pattern Based Approach for Definitional Knowledge Extraction. In *Proceedings of the 13th EURALEX International Congress* (p. 923-928). Barcelona.

Alvarez de Mon, I., and Aguado de Cea, G. (2006). The phraseology of classification in Spanish: integrating corpus linguistics and ontological approaches for knowledge extraction. In *BAAL/IRAAL Joint International Conference.* Cork, Irlanda.

Atserias, J., Villarejo, L., Rigau, G., Aguirre, E., Carroll, J., Magnini, B., et al. (2004). The MEANING Multilingual Central Repository. In *Proceedings of the 2nd International WordNet Conference-GWC.* Brno, Czech Republic.

Aussenac-Gilles, N. (2005). Supervised text analysis for ontology and terminology

engineering. In *Proceedings of the Dagstuhl Seminar on Machine Learning for the Semantic Web* (p. 35-46).

Aussenac-Gilles, N., Biébow, B., and Szulman, S. (2000). Revisiting ontology design: A method based on corpus analysis. In R. Dieng and O. Corby (Eds.), *Knowledge engineering and knowledge management: Methods, models and tools* (p. 172-188). Berlin: Springer Verlag.

Aussenac-Gilles, N., Despres, S., and Szulman, S. (2008). The TERMINAE Method and Platform for Ontology Engineering from Texts. In P. Buitelaar and P. Cimiano (Eds.), *Ontology learning and population: Bridging the gap between text and knowledge* (p. 199-223). IOS Press.

Aussenac-Gilles, N., and Jacques, M.-P. (2006). Designing and Evaluating Patterns for Ontology Enrichment from Texts. In S. Staab and V. Svatek (Eds.), *Managing Knowledge in a World of Networks, Proceedings of the 15th International Conference EKAW06* (Vol. 4248). Podebrady, Czech Republic: Springer.

Aussenac-Gilles, N., and Jacques, M.-P. (2008). Designing and evaluating patterns for relation acquisition from texts with CAMÉLÉON. In *Terminology. Patterns Special Issue* (Vol. 14, p. 45-73). John Benjamins.

Baader, F., and Nutt, W. (2002). Basic Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (Eds.), *Description Logic Handbook: Theory, Implementation and Application* (p. 47-100). Cambridge University Press.

Barrasa, J. (2007). *Modelo para la definición automática de correspondencias semánticas entre ontologías y modelos relacionales*. Unpublished doctoral dissertation, Facultad de Informatica, Universidad Politécnica de Madrid, Madrid, Spain. http://eprints.eemcs.utwente.nl/7146/.

Berland, M., and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedigs of the 37th annual meeting of the ACL* (p. 57-64).

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34-43.

Bernstein, A., and Kaufmann, E. (2006). Gino - a guided input natural language ontology editor. In I. Cruz et al. (Eds.), *Proceedings of the 6th International Semantic Web Conference (ISWC06) and 2nd Asian Semantic Web Conference (ASWC06)* (p. 144-157). Heidelberg: Springer.

Blomqvist, E., Gangemi, A., and Presutti, V. (2009). Experiments on pattern-based ontology design. In *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP09)* (p. 41-48). Redondo Beach, California, USA: ACM.

Buitelaar, P. (1998). *CoreLex: Systematic Polysemy and Underspecification*. Unpublished doctoral dissertation, Computer Science Departament, Brandeis University, Waltham MA, USA.

Buitelaar, P., Cimiano, P., Haase, P., and Sintek, M. (2008). *Towards linguistically grounded ontologies* (Tech. Rep.). Available from `http://people.aifb.kit.edu/pci/lexinfo_tech_report_08.pdf`

Buitelaar, P., Cimiano, P., Haase, P., and Sintek, M. (2009). Towards linguistically grounded ontologies. In *Proceedings of the 6th European Semantic Web Conference (ESWC09)* (p. 111-125).

Buitelaar, P., Declerck, T., Frank, A., Racioppa, S., Kiesel, M., Sintek, M., et al. (2006). LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies. In *Proceedings of the OntoLex 2006 Workshop: Interfacing Ontologies and Lexical Resources for Semantic Web Technologies.*

Buitelaar, P., Sintek, M., and Kiesel, M. (2006). A Multilingual/Multimedia Lexicon Model for Ontologies. In *Proceedings of the 3rd European Semantic Web Conference (ESWC06)* (p. 502-513).

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1996). *Pattern-oriented software architecture. a system of patterns*. Chichester: John Wiley and Sons.

Butler, C. S. (2009). The Lexical Constructional Model. Genesis, strengths and challenges. In C. S. Butler and J. Martín-Arista (Eds.), *Deconstructing constructions* (Vol. 107, p. 117-152). John Benjamins.

Cabré, M. T. (1999). *La terminología: Representación y comunicación*. IULA. Universitat Pompeu Fabra.

Cabré, M. T., Bach, C., Estopà, R., Feliu, J., Martínez, G., and Vivaldi, J. (2004). The GENOMA-KB project: towards the integration of concepts, terms, textual corpora and entities. In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (Vol. 1, p. 87-90).

Ceusters, W., and Smith, B. (2006). A realism-based approach to the evolution of biomedical ontologies. In *Proceedings of the Annual AMIA Symposium* (p. 121-125). Washington, DC: American Medical Informatics Association.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge: MIT Press.

Cimiano, P. (2006). *Ontology learning and population from text. algorithms, evaluation and applications.* Springer.

Cimiano, P., Haase, P., Herold, M., Mantel, M., and Buitelaar, P. (2007). LexOnto: A Model for Ontology Lexicons for Ontology-based NLP. In *Proceedings of the OntoLex07 Workshop at the ISWC07.*

Cimiano, P., Montiel-Ponsoda, E., Buitelaar, P., Espinoza, M., and Gómez-Pérez, A. (2010). A note on ontology localization. *Journal of Applied Ontology*, *5(2)*, 127-137.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., and Staab, S. (2005). Learning taxonomic relations from heterogeneous evidence. In *Ontology learning from text: Methods, applications and evaluation* (p. 59-73). IOS Press.

Cimiano, P., and Staab, S. (2004). Learning by googling. *SIGKDD Explorations*, *6(2)*, 24-33.

Cimiano, P., and Wenderoth, J. (2005). Learning qualia structures from the web. In *Acl workshop on deep lexical acquisition* (p. 28-37).

Cimiano, P., and Wenderoth, J. (2007). Automatic Acquisition of Ranked Qualia

Structures from the Web. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL07)* (p. 888-895).

Clark, P., Harrison, P., Murray, W. R., and Thompson, J. (2009). Naturalness vs. Predictability: A Key Debate in Controlled Languages. In *Proceedings of the Workshop on Controlled Languages (CNL09).*

Clark, P., and Porter, B. (1997). Building concept representations from reusable components. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI97)* (p. 369-376).

Clark, P., Thompson, J., and Porter, B. (2000). Knowledge Patterns. In A. Cohn, F. Giunchiglia, B. Selman, and C. Kaufmann (Eds.), *Proceedings of the 7th International Conference KR00* (p. 591-600).

Climent Roca, S. (2000). *Individuación e información parte-todo. representación para el procesamiento computacional del lenguaje*. Unpublished doctoral dissertation, Universitat de Barcelona. Available from `http://elies.rediris.es/elies8/`

Cocchiarella, N. (1996). Conceptual realism as a formal ontology. In R. Poli and P. Simons (Eds.), *Formal Ontology* (p. 27-60). Dordrecht/Boston/London: Kluwer.

Cocchiarella, N. (2001). Logical and Ontology. *Axiomathes*, *12*, 117-150.

Condamines, A. (2002). Corpus analysis and conceptual relation patterns. In *Terminology* (Vol. 8, p. 141-162). John Benjamins.

Condamines, A., and Rebeyrolle, J. (2000). Construction d'une base de connaissances terminologiques à partir de textes : expérimentation et définition d'une méthode. In J. Charlet, M. Zacklad, G. Kassel, and D. Bourigault (Eds.), *Ingénierie des connaissances, évolutions récentes et nouveaux défis* (p. 225-242). Paris: Eyrolles.

Cregan, A., Schwitter, R., and Meyer, T. (2007). Sydney OWL Syntax -towards a Controlled Natural Language Syntax for OWL 1.1. In *Proceedings of the OWLED07 Workshop on OWL: Experiences and Directions.*

Croft, W., and Cruse, D. A. (2004). *Cognitive linguistics*. Cambridge: Cambridge University Press.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Annual Meeting of the ACL.*

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Dimitrov, M., Dowman, M., et al. (2009, November). Developing Language Processing Components with GATE Version 5 (a User Guide) [Computer software manual].

Cunningham, H., Maynard, D., and Tablan, V. (2000, November). JAPE: a Java Annotation Patterns Engine (Second Edition ed.) [Computer software manual].

Cunningham, W., and Beck, K. (1987). Using Patterns Languages for Object-Oriented Programs. In *OOPSLA87 workshop on the Specification and Design for Object-Oriented Programming.*

D'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., López, V., et al.

(2008). Toward a New Generation of Semantic Web Applications. *IEEE Intelligent Systems*, *23*(3), 20-28.

Dean, M., and Schreiber, G. (2004). *Owl web ontology language reference* (Tech. Rep.). W3C Recommendation. Available from `http://www.w3.org/TR/2004/REC-owl-ref-20040210/`

Dimarco, C., Hirst, G., and Stede, M. (1993). The semantic and stylistic differentiation of synonyms and near-synonyms. In *Proceedings of the AAAI Spring Symposium on Building Lexicons for Machine Translation* (p. 114-121). Standford, CA..

Dolbear, C., Hart, G., Goodwin, J., Zhou, S., and Kovacs, K. (2007). *The Rabbit language: description, syntax and conversion to OWL* (Tech. Rep.). Ordenance Survey Research.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. In *Terminology* (Vol. 9, p. 99-115). John Benjamins.

Edmonds, P., and Hirst, G. (2002). Near-synonymy and lexical choice. *Computational Linguistics*, *28*(2), 105-144.

Egaña, M., Rector, A. L., Stevens, R., and Antezana, E. (2008). Applying Ontology Design Patterns in Bio-ontologies. In A. Gangemi and J. Euzenat (Eds.), *Proceedings of the 16th International Conference on Knowledge Engineering: Practice and Patterns (EKAW08)* (p. 7-16). Springer.

Egaña Aranguren, M., Stevens, R., and Antezana, E. (2007). Ontology Design Patterns (ODPs) for bio-ontologies (Talk). In *Bio-ontologies SIG at ISMB/ECCB*. Vienna.

Espinoza, M., Gómez-Pérez, A., and Mena, E. (2008a). Enriching an Ontology with Multilingual Information. In *Proceedings of the 5th Annual of the European Semantic Web Conference (ESWC08)* (p. 333-347).

Espinoza, M., Gómez-Pérez, A., and Mena, E. (2008b). LabelTranslator - A Tool to Automatically Localize an Ontology. In *The Semantic Web: Research and Applications* (p. 792-796). Berlin / Heidelberg: Springer.

Espinoza, M., Gómez-Pérez, A., and Montiel-Ponsoda, E. (2009). Multilingual and Localization Support for Ontologies. In *Proceedings of the 6th Conference of the European Semantic Web Conference (ESWC09)* (p. 821-825). Berlin / Heidelberg: Springer.

Espinoza, M., Montiel-Ponsoda, E., and Gómez-Pérez, A. (2009). Ontology Localization. In *Proceedings of the 5th International Conference on Knowledge Capture (KCAP09)* (p. 33-40).

Esselink, B. (2000). *A practical guide to localization*. John Benjamins.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., et al. (2004). Methods for domain-independent information extraction from the web: An experimental comparison. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI04)* (p. 391-398).

Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., et al. (2009). Results of the ontology alignment evaluation initiative 2009. In *Workshop on Ontology Matching (OM09) at the ISWC Conference*.

Evans, V. (2006). Lexical concepts, cognitive models and meaning construction. *Cognitive Linguistics*, *17*(4), 491-534.

Evans, V. (2010). Cognitive Linguistics. In L. Cummings (Ed.), *The pragmatics encyclopedia* (p. 46-49). Routledge. Available from http://www.vyvevans.net/cognitiveLinguisticsPRAG-ENCYC.pdf

Evans, V., Bergen, B. K., and Zinken, J. (2006). The cognitive linguistics enterprise: An overview. In V. Evans, B. K. Bergen, and J. Zinken (Eds.), *The cognitive linguistics reader.* Equinox Publishing Co. Available from http://www.vyvevans.net/CLoverview.pdf

Faatz, A., and Steinmetz, R. (2002). Ontology Enrichment with Texts from the WWW. In *Proceedings of the 2nd Workshop on Semantic Web Mining* (p. 20-35). Helsinki.

Faber, P., and Mairal Usón, R. (1999). *Constructing a Lexicon of English Verbs* (Vol. 23). Berlin; New York: Mouton de Gruyter.

Fauconnier, G. (1985). *Mental spaces*. Cambridge: Cambridge University Press.

Fayad, M., and Srikanth, G. (2007). *Choosing the Right Pattern- Real Challenges.* Available from http://pattern.ijop.org/?p=20

Feliu, J., and Cabré, M. T. (2002). Conceptual relations in specialized texts: new typology and an extraction system proposal. In *Proceedings of the 6th international conference on terminology and knowledge engineering (tke02)* (p. 45-49). Nancy.

Feliu Cortés, J. (2004). *Relacions conceptuals i terminologia: anàlisi i proposta de detecció semiautomàtica*. Unpublished doctoral dissertation, Institut Universitaria de Lingüística Aplicada, Universitat Pompeu Fabra, Barcelona. Available from http://www.tdx.cesca.es/TDX-0520104-111213/index_an.html#documents

Fellbaum, C. (1998). *WordNet. An Electronic Lexical Database* (C. Fellbaum, Ed.). The MIT Press.

Fernández-López, M., Gómez-Pérez, A., Pazos, J., and Pazos, A. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems and their applications*, *4*(1), 37-46.

Fillmore, C. J. (1975). An alternative to checklist theories of meaning. In *Proceedings of the 1st Annual Meeting of the Berkeley Linguistics Society* (p. 123-131).

Fillmore, C. J. (1982). Frame Semantics. In The Linguistic Society of Korea (Ed.), *Linguistics in the Morning Calm* (p. 111-138). Seoul, Hanshin.

Fillmore, C. J., and Atkins, B. T. (1992). Toward a frame-based lexicon: The semantics of RISK and its neighbors. In A. Lehrer and E. F. Kittay (Eds.), *Frames, Fields and Contrasts* (p. 75-102). Hillsdale, NJ: Lawrence Erlbaum.

Finkelstein-Landau, M., and Morin, E. (1999). Extracting Semantic Relationships between Terms: Supervised vs. Unsupervised Methods. In *Proceedings of the International Workshop on Ontological Engineering on the Global Information Infrastructure* (p. 71-80).

Fu, B., Brennan, R., and O'Sullivan, D. (2010). Cross-Lingual Ontology Mapping and Its Use on the Multilingual Semantic Web. In *Proceedings of the 1st International Workshop on the Multilingual Semantic Web (MSW10). CEUR Workshop Proceedings.* (Vol. 571, p. 13-20).

Fuchs, N. E., Kaljurand, K., and Schneider, G. (2006). Bidirectional mapping between OWL DL and Attempto Controlled English. In *Proceedings of the 4th Workshop on Principles and Practice of Semantic Web Reasoning* (p. 179-189). Budva, Montenegro.

Fuchs, N. E., Schwertel, U., and Schwitter, R. (1998). Attempto Controlled English - Not Just Another Logic Specification Language. In *Proceedings of LOPSTR98.*

Funk, A., Davis, B., Tablan, V., Bontcheva, K., and Cunningham, H. (2007). *Sekt project D2.2.2 Report: Controlled Language IE Components version 2* (Tech. Rep.). University of Sheffield. (SEKT EU-IST-2003-506826)

Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., and Handschuh, S. (2007). CLOnE: Controlled Language for Ontology Editing. In *Proceedings of the International Semantic Web Conference (ISWC07).*

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns. Elements of Reusable Object-Oriented Software*. New York: Addison-Wesley.

Gangemi, A. (2005). Ontology Design Patterns for Semantic Web Content. In Y. G. et al. (Eds.) (Ed.), *Proceedings of International Semantic Web Conference (ISWC05)* (p. 262-276). Berlin Heidelberg: Springer.

García-Silva, A., Gómez-Pérez, A., Suárez-Figueroa, M. C., and Villazón-Terrazas, B. (2008). A Pattern Based Approach for Reengineering Non Ontological Resources into Ontologies. In *3rd Asian Semantic Web Conference ASWC08.* Bangkok, Thailand.

Girju, R. A. B., and Moldovan, D. (2003). Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. In *Proceedings of the HLT-NAACL03.*

Gómez-Pérez, A., Ferández-López, M., and Corcho, O. (2003). *Ontological Engineering*. Springer, New York.

Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2009). NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology. In *Proceedings of International Conference on Software, Services and Semantic Technologies (S3T09).* Sofia, Bulgaria.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. Chicago: Chicago University Press.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language*. Oxford: OUP.

Grefenstette, G. (1992). Finding semantic similarity in raw text: the deese antonyms. In R. Goldman, P. Norvig, E. Charniak, and B. Gale (Eds.), *Working notes of the aaai fall symposium on probabilistic approaches to natural language* (p. 61-65).

Grüninger, M., and Fox, M. S. (1994). The role of competency questions in enter-

prise engineering. In *Proceedings of the ifip wg5.7 workshop on benchmarking - theory and practice.*

Guarino, N. (1998). Formal Ontology and Information Systems. In N. Guarino (Ed.), *Proceedings of Formal Ontology in Information Systems (FOIS98)* (p. 3-15). Trento, Italy: IOS Press.

Hahn, U., and Schnattinger, K. (1998). Towards text knowledge engineering. In *Proceedings of the 15th National Conference on Artificial Intelligence and 10th Conference on Innovative Applications of Artificial Intelligence* (p. 524-531).

Hartmann, J., Palma, R., and Bontas, E. P. (2006). *OMV-Ontology Metadata Vocabulary for the Semantic Web.* (OMV Report No. 2.0). Available from http://ontoware.org/frs/download.php/336/OMV-ReportV2.1.pdf

Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING92)* (p. 539-545).

Hearst, M. A. (1998). Automated Discovery of WordNet Relations. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database.* MIT Press. Available from http://people.ischool.berkeley.edu/~hearst/papers/wordnet98.pdf

Hirst, G. (2004). International Handbooks on Information Systems. In S. Staab and R. Studer (Eds.), *Handbook on Ontologies* (p. 209-230). Springer. Available from DBLP, http://dblp.uni-trier.de

Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., and Wang, H. H. (2006). The Manchester OWL Syntax. In *Proceedings of OWLED06.*

House, J. (1977). *A Model for Translation Quality Assessment*. Tübingen: Narr.

Hurtado Albir, A. (2001). *Traducción y Traductología. Introducción a la Traductología*. Madrid: Cátedra.

ISO 12200:1999 - Computer applications in terminology - Machine-readable terminology interchange format (MARTIF). (1999). Available from http://www.iso.org/iso/catalogue_detail.htm?csnumber=21174 (1999)

ISO 12620 - Data Categories, Terminology and other language resources. (2003). Available from http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=32347

ISO 16642 - TMF-Terminological Markup Framework, Computer applications in terminology. (2003). Available from http://www.loria.fr/projets/TMF/

ISO 24613 - LMF- Lexical Markup Framework , Language Resource Management. (2006). Available from http://lirics.loria.fr/doc_pub/LMF%20rev9%2015March2006.pdf

ISO 5964:1985 - Documentation - Guidelines for the establishment and develop-

ment of multilingual thesauri. (1985). Available from http://www.iso.org/iso/catalogue_detail.htm?csnumber=12159

ISO 639 - Codes for the representation of names of languages. (2002). Available from http://www.iso.org/iso/en/commcentre/news/archives/2002/iso639_1.html

ISO/IEC 14977:1996 - Information technology - Syntactic metalanguage - Extended BNF. (1996). Available from http://www.iso.org/iso/catalogue_detail.htm?csnumber=26153

Iwanska, L., Mata, N., and Kruger, K. (2000). Fully Automatic Acquisition of Taxonomic Knowledge from Large Corpora of Texts: Limited-Syntax Knowledge Representation System based on Natural Language. In L. M. Iwanksa and S. Shapiro (Eds.), *Natural Language Processing and Knowledge Processing* (p. 335-345). MIT/AAAI Press.

Johnson, C., and Fillmore, C. J. (2000). The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL00)* (p. 56-62). Seattle, WA.

Johnson, M. (1987). *The Body in the Mind. The Bodily Basis of Meaning, Imagination and Reason*. University of Chicago Press.

Kaljurand, K., and Fuchs, N. (2007). Verbalizing OWL in Attempto Controlled English. In *OWL: Experiences and Directions (OWLED07).*

Kaljurand, K., and Fuchs, N. E. (2006). Bidirectional mapping between OWL DL and Attempto Controlled English. In *Proceedings of the 4th Workshop on Principles and Practice of Semantic Web Reasoning.* Budva, Montenegro.

Kaufmann, E., Bernstein, A., and Zumstein, R. (2006). Querix: A natural language interface to query ontologies based on clarification dialogs. In *Proceedings of the 5th International Semantic Web Conference (ISWC06)* (p. 980Ű981). Athens, Georgia.

Kaufmann, E., and Berstein, A. (2007). How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? In K. Aberer, K.-S. Choi, and N. Noy (Eds.), *Proceedings of the 6th International Semantic Web Conference 2007 (ISWC07)* (p. 281-294).

Kavi, M., and Nirenburg, S. (1995). A situated ontology for practical NLP. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing at the International Joint Conference on Artificial Intelligence (IJCAI95).* Montreal, Canada.

Kerremans, K., and Temmerman, R. (2004). Towards Multilingual, Termontological Support in Ontology Engineering. In *Proceedings of the Workshop on Terminology, Ontology and Knowledge representation.* Lyon, France.

Kerremans, K., Temmerman, R., and Tummers, J. (2004). Discussion on the Requirements for a Workbench supporting Termontography. In *Proceedings of Euralex04.*

Klyne, G., and Carroll, J. J. (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax* (Tech. Rep.). W3C Recommen-

dation. Available from http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

Labov, W. (1973). The boundaries of words and their meanings. In C. J. N. Bailey and R. Shuy (Eds.), *New Ways of Analysing Variation in English* (p. 340-373). Washington, DC: Georgetown University Press.

Lakoff, G. (1987). *Women, Fire and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press.

Lakoff, G., and Johnson, M. (1980). *Metaphors We Live By*. Chicago: Chicago University Press.

Langacker, R. (1987). *Foundations of Cognitive Grammar* (Vol. 1). Stanford: Stanford University Press.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago, IL: University of Chicago Press.

Liang, A., Lauser, B., Sini, M., Keizer, J., and Katz, S. (2008). From AGROVOC to the Agricultural Ontology Service / Concep Server. An OWL model for managing ontologies in the agricultural domain. In *Proceedings of the OWL: Experiences and Directions Workshop.* Manchester, UK.

López, V., Motta, E., and Uren, V. (2006). PowerAqua: Fishing the Semantic Web. In *The Semantic Web: Research and Applications, proceedings of the 3rd European Semantic Web Conference (ESWC06)* (p. 393-410).

Maedche, A., and Staab, S. (2000). Discovering conceptual relations from text. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI00)* (p. 321-325). Amsterdam: IOS Press.

Maedche, A., and Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, *16*(2), 72-79.

Maedche, A., Staab, S., Stojanovic, N., Studer, R., and Sure, Y. (2003). SEmantic PortAL: The SEAL approach. In D. Fensel, J. Hendler, H. Lieberman, , and W. Wahlster (Eds.), *Spinning the Semantic Web* (p. 317-354). Cambridge: The MIT Press.

Mairal Usón, R., and Cortés-Rodríguez, F. J. (2006). An overview of role and reference grammar. In R. Mairal-Usón (Ed.), *Current trends in linguistic theory* (p. 97-176). Universidad Nacional de Educación a Distancia (UNED).

Mairal Usón, R., and Faber, P. (2007). Lexical templates within a functional cognitive theory of meaning. *Annual Review of Cognitive LInguistics*, *5*, 137-172.

Mairal Usón, R., and Periñán-Pascual, J. C. (2009). The anatomy of the lexicon within the framework of an NLP knowledge base. *Revista española de lingüística aplicada*, *22*, 217-244.

Mairal Usón, R., and Ruiz de Mendoza Ibáñez, F. J. (2006). Internal and external constraints in meaning construction: the lexicon-grammar continuum. In *Estudios de filología inglesa: Homenaje a la dra. asunción alba pelayo.* Madrid, Spain: UNED.

Mairal Usón, R., and Ruiz de Mendoza Ibáñez, F. J. (2008). New Challenges for Lexical Representation within the Lexical-Constructional Model (LCM).

Revista Canaria de Estudios Ingleses. Grammar, Construction and Interfaces(57), 137-158.

Mairal Usón, R., and Ruiz de Mendoza Ibáñez, F. J. (2009). Levels of description and explanation in meaning construction. In C. S. Butler and J. Martín-Arista (Eds.), *Deconstructing constructions* (p. 153-198). John Benjamins.

Marshman, E. (2007). Towards strategies for processing relationships between multiple relation participants in knowledge patterns. An analysis in English and French. In *Terminology* (Vol. 13, p. 1-34). John Benjamins.

Marshman, E. (2008). Expressions of uncertainty in candidate knowledge-rich contexts. In *Terminology* (Vol. 14, p. 124-151). John Benjamins.

Marshman, E., and L'Homme, M. C. (2006). Disambiguation of lexical markers of cause and effect. In H. Picht (Ed.), *Modern Approaches to Terminological Theories and Applications. Proceedings of the 15th European Symposium on Language for Special Purposes (LSP06)* (p. 261-285). Bern: Peter Lang.

Marshman, E., Morgan, T., and Meyer, I. (2002). French patterns for expressing concept relations. *Terminology*, *8*(1), 1-29.

Martín Mingorance, L. (1990). Functional grammar and lexematics in lexicography. In J. Tomaszczyk and B. Lewandowska-Tomaszczyk (Eds.), *Meaning and lexicography* (p. 227-253). Amsterdam: John Benjamins.

Martín Mingorance, L. (1998). El modelo lexemático-funcional: El legado lingüístico de Leocadio Martín Mingorance. In A. Marín Rubiales (Ed.), (chap. El modelo lexemático-funcional). Granada: Universidad de Granada.

Mel'Cuk, I. (1988). Semantic description of lexical units in an explanatory combinatorial dictionary: Basic principles and heuristic criteria. *International Journal of Lexicography*, *1*(3), 165-188.

Mel'Cuk, I., and Polguère, A. (1987). A formal lexicon in the Meaning-Text Theory: (or how to do lexica with words). *Computational Linguistics. Special issue of the lexicon*, *13*(3-4), 261-275.

Merrill, G. H. (2010). Ontological realism: Methodology or misdirection? *Applied Ontology*, *5*, 79-108.

Meyer, I. (2001). Extracting knowledge-rich contexts for terminography. a conceptual and methodological framework. In D. Bourigault, C. Jacquemin, and M.-C. L'Homme (Eds.), *Recent Advances in Computational Terminology* (Vol. xviii). Benjamins.

Miles, A., Matthews, B., Beckett, D., Brickley, D., Wilson, M., and Rogers, N. (2005). SKOS: A language to describe simple knowledge structures for the web. In *Proceedings of the XTech Conference.*

Miller, G. A. (1990). WordNet An On-line Lexical Database. *International Journal of Lexicography*, *3*(4), 235-244.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1999). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, *3*(4), 235-244.

Mitkov, R. (2003). *The Oxford Handbook of Computational Linguistics* (R. Mitkov, Ed.). Oxford University Press.

Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., and Peters, W. (2008). Modelling multilinguality in ontologies. In *Proceedings of the 22nd International Conference on Computational Linguistics, Coling08, Companion volume - Posters and Demonstrations* (p. 67-70). Manchester, UK.

Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., and Peters, W. (2010, Juny). Enriching Ontologies with Multilingual Information. *Journal of Natural Language Engineering*.

Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2008). Helping Naive Users to Reuse Ontology Design Patterns. In *Proceedings of the 1st International Workshop on Knowledge Reuse and Reengineering over the Semantic Web (KRRSW08).*

Montiel-Ponsoda, E., Aguado de Cea, G., Suárez-Figueroa, M. C., Palma, R., Peters, W., and Gómez-Pérez, A. (2007). LexOMV: an OMV extension to capture multilinguality. In *From Text to Knowledge, The Lexicon/Ontology Interface, proceedings of the OntoLex07 Workshop.*

Montiel-Ponsoda, E., Peters, W., Aguado de Cea, G., Espinoza, M., Gómez-Pérez, A., and Sini, M. (2008). *Multilingual and localization support for ontologies.* (Tech. Rep.). Technical report, D2.4.2 NeOn Project Deliverable.

Murphy, G. L. (2002). *The Big Book of Concepts. A comprehensive introduction to current research on the psychology of concept formation and use*. The MIT Press.

Nord, C. (1989). Loyalität statt Treue. *Lebende Sprachen*, *34*(3), 100-105.

Nord, C. (1997). *Translating as a Purposeful Activity. Functionalist Approaches Explained*. Manchester: St. Jerome.

Peña Cervel, M. S., and Samaniego Fernández, E. (2006). An overview of congnitive linguistics. In R. Mairal Usón, M. n. Escobar Álvarez, M. S. Peña Cervel, and E. Samaniego Fernández (Eds.), *Current trends in linguistic theory* (p. 229-316). Universidad Nacional de Educación a Distancia (UNED).

Pearson, J. (1998). *Terms in Context*. Amsterdam: John Benjamins.

Periñán Pascual, C., and Arcas Túnez, F. (2010). The architecture of fungramkb. In *7th international conference on language resources and evaluation (lrec), european language resources association (elra)* (p. 2667-2674). Valeta, Malta.

Peters, W., Espinoza, M., Montiel-Ponsoda, E., and Sini, M. (2009). *D2.4.3: Multilingual and Localization Support for Ontologies (v3)* (Tech. Rep.). NeOn Project Deliverable.

Peters, W., Montiel-Ponsoda, E., Aguado de Cea, G., and Gómez-Pérez, A. (2007). Localizing ontologies in OWL. In *From text to knowledge, the lexicon/ontology interface, proceedings of the ontolex07 workshop.* Busan, South Corea.

Pinto, H. S., Tempich, C., and 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 (ECAI04) (p. 393-397).

Prechelt, L. (1997). *An experiment on the usefulness of design patterns: Detailed description and evaluation* (Tech. Rep.). University of Karlsruhe.

Presutti, V., Daga, E., Gangemi, A., and Blomqvist, E. (2009). eXtreme Design with Content Ontology Design Patterns. In *Proceedings of the Workshop on Ontology Patterns (WOP09). CEUR Proceedings* (Vol. 516).

Presutti, V., Gangemi, A., David, S., Cea, G. A. de, Suárez-Figueroa, M. C., Montiel-Ponsoda, E., et al. (2008). *D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies* (Tech. Rep.). NeOn Project Deliverable.

Pulman, S. G. (1996). Controlled Language for Knowledge Representation. In *Proceedings of the 1st International Workshop on Controlled Language Applications* (p. 233-242).

Pustejovsky, J. (1995). *The Generative Lexicon*. Cambridge, Massachusetts, London, England: The MIT Press.

Pym, A. (2002). *Localization and the Training of Linguistic Mediators for the Third Millennium.* The Challenges of Translation and Interpretation in the Third Millennium. Lebanon. Available from `http://www.tinet.org/~apym/on-line/translation/beirut.pdf` (Accessed on January 2009)

Rebeyrolle, J., and Tanguy, L. (2000). Repérage automatique de structures linguistiques en corpus: le cas des énoncés définitoires. In *Cahiers de grammaire* (Vol. 25, p. 153-174).

Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., et al. (2004). OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors and Common Patterns. In E. Motta and N. Shadbolt (Eds.), *Proceedings of the European Conference on Knowledge Acquistion* (p. 63-81). Northampton, England: Springer-Verlag.

Rector, A., and Rogers, J. (2004). Patterns, properties and minimizing commitment: reconstruction of the GALEN upper ontology in OWL. In G. A. and B. S. (Eds.), *Proceedings of the EKAW 2004 Workshop on Core Ontologies in Ontology Engineering (CORONT). CEUR Workshop Proceedings.* Northamptonshire.

Reich, J. R. (2000). Ontological Design Patterns: Modelling the Metadata of Molecular Biological. Ontologies, Information and Knowledge. In *DEXA00.*

Reiss, K., and Vermeer, H. (1984). *Grundlegung einer allgemeinen Translationstheorie*. Tübingen: Nienmeyer.

Rosch, E. (1973). Natural categories. *Cognitive Psychology*, *4*, 328-350.

Rosch, E. (1975). Cognitive representation of semantic categories. *Journal of Experimental Psychology: General*, *104*, 192-233.

Ruiz de Mendoza Ibáñez, F. J., and Mairal Usón, R. (2006a). *How to design lexical and constructional templates: A step by step*

guide. Available from `http://www.lexicom.es/drupal/files/templateDesign.pdf`

Ruiz de Mendoza Ibáñez, F. J., and Mairal Usón, R. (2006b). Levels of semantic representation: where lexicon and grammar meet. *Interlingüística*, *17*, 26-47.

Ruiz de Mendoza Ibáñez, F. J., and Mairal Usón, R. (2008). Levels of description and constraining of factors in meaning construction: an introduction to the Lexical Constructional Model. *Folia lingüística: Acta Societatis Linguisticae Europaeae*, *42*(2), 355-400.

Sabou, M., Aguado de Cea, G., DŠAquin, M., Daga, E., Lewen, H., Montiel-Ponsoda, E., et al. (2009). *D2.2.3 Methods and Tools for the Evaluation and Selection of Knowledge Components* (Tech. Rep.). NeOn Project Deliverable.

Schutz, A., and Buitelaar, P. (2005). RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In *Proceedings of the 4th International Semantic Web Conference (ISWC05)*.

Schwitter, R. (2004). Representing Knowledge in Controlled Natrual Language: A Case Study. In M. G. Negoita, R. J. Howlett, and L. C. Jain (Eds.), *Proceedings of KES04, Part I* (p. 711-717). Springer.

Schwitter, R. (2007). *Controlled natural languages* (Tech. Rep.). Centre for Language Technology, Macquarie University.

Schwitter, R., Kaljurand, K., Cregan, A., Dolbear, C., and Hart, G. (2008). A Comparison of three Controlled Natural Languages for OWL 1.1. In *Proceedings of the 4th OWL Experiences and Directions Workshop (OWLED08)*. Washington.

Schwitter, R., and Ljungberg, A. (2002). How to Write a Document in Controlled Natural Language. In *Proceedings of the 7th Australasian Document Computing Symposium* (p. 133-136). Sydney, Australia.

Scott, M. (1999). *WordSmith Tools version 3*. Oxford: Oxford University Press.

Séguéla, P. (2001). *Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques*. Unpublished doctoral dissertation, Université Toulouse III Paul Sabatier.

Sierra, G., Alarcón, R., Aguilar, C., and Bach, C. (2008). Definitional verbal patterns for semantic relation extraction. In *Terminology* (Vol. 14(1), p. 74-98). John Benjamins.

Smith, B. (2004). Beyond concepts: Ontology as reality representation. In *Proceedings of the 3rd International Conference on Formal Ontology in Informatic Systems (FOIS04)* (p. 73-84). Amsterdam: IOS Press.

Sánchez, D., and Moreno, A. (2008). Learning non-taxonomic relationships from web documents for domain ontology construction. *Data Knowledge Engineering*, *64*, 600-623.

Sánchez Ruenes, D. (2007). *Domain Ontology Learning from the Web*. Unpublished doctoral dissertation, Departament de Llenguatges i Sistemes Imformàtics Universitat Politècnica de Catalunya.

Snow, R., Jurafsky, D., and Ng, A.-Y. (2004). Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*, *17*.

Soler, V., and Alcina, A. (2008). Patrones léxicos para la extracción de conceptos vinculados por la relación parte-todo en español. In *Terminology* (Vol. 14(1), p. 99-123). John Benjamins.

Staab, S., Schnurr, H.-P., Studer, R., and Sure, Y. (2001). Knowledge Processes and Ontologies. *IEEE Intelligent Systems*, *16*(1), 26-34.

Studer, R., Benjamins, R., and Fensel, D. (1998). Knowledge engineering: principles and methods. *Data and Knowledge Engineering*, *25*(1-2), 161-198.

Suárez-Figueroa, M., Blomqvist, E., D'Aquin, M., Espinoza, M., Gómez-Pérez, A., Lewen, H., et al. (2009). *D5.4.2. Revision and Extension of the NeOn Methodology for Building Contextualized Ontology Networks* (Tech. Rep.). NeOn Project Deliverable.

Suárez-Figueroa, M. C. (2010). *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. Unpublished doctoral dissertation, Universidad Politécnica de Madrid, Madrid, Spain.

Suárez-Figueroa, M. C., Brockmans, S., Gangemi, A., Gómez-Pérez, A., Lehmann, J., Lewen, H., et al. (2007). *D5.1.1 NeOn Modelling Components* (Tech. Rep.). NeOn Project Deliverable.

Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2008). Towards a Glossary of Activities in the Ontology Engineering Field. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC08).* Marrakech (Morocco).

Suárez-Figueroa, M. C., Gómez-Pérez, A., and Villazón-Terrazas, B. (2009). How to write and use the Ontology Requirements Specification Document. In R. Meersman, T. Dillon, and P. Herrero (Eds.), *Proceedings of the 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE09)* (Vol. 2, p. 966-982).

Svátek, V. (2004). Design patterns for semantic web ontologies: Motivation and discussion. In *Proceedings of the 7th conference on bussiness information systems.*

Talmy, L. (2000). *Toward a Cognitive Semantics*. The MIT Press.

Temmerman, R. (2000). *Towards New Ways of Termonology Description. The sociocognitive approach* (Vol. 3). John Benjamins.

Temmerman, R., and Kerremans, K. (2003). Termontography: Ontology Building and the Sociocognitive Approach to Terminology Description. In E. Hajicova, A. Kotesovcova, and J. Mirovsky (Eds.), *Proceedings of CIL17.* Prague, Czech Republic.: Matfyzpress.

Van Valin, R. (2004). *Lexical representation, co-composition, and linking syntax and semantics.* Available from `http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg/vanvalin_papers/LexRepCoCompLnkgRRG.pdf`

Van Valin, R. (2005). *An Overview of Role and Reference Grammar.* Available

from http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg/RRG_overview.pdf

Van Valin, R., and LaPolla, R. (1997). *Syntax. Structure, meaning and function*. Cambridge: Cambridge University Press.

Vela, M., and Declerck, T. (2009). Concept and Relation Extraction in the Finance Domain. In *Proceedings of the 8th International Conference on Computational Semantics (IWCS09).*

Vendler, Z. (1967). *Linguistics in Philosophy*. Ithaca, New York: Cornell University Press.

Vermeer, H. (1978). Ein Rahmen für eine algemeine Translationstheorie. *Lebende Sprachen*, *23*(1), 99-102.

Vilches-Blázquez, L. M., Ramos Gargantilla, J. A., López-Pellicer, F. J., Corcho, O., and Nogueras-Iso, J. (2009). An approach to comparing different ontologies in the context of hydrographical information. In P. et al. (Ed.), *Proceedings of the IF&GIS09* (p. 193-207). St. Petersburg, Russia: Springer.

Vivaldi, J. (2003). Sistema de reconocimiento de términos Mercedes. Manual de utilización [Computer software manual]. Barcelona.

Vossen, P. (1998). Introduction to EuroWordNet. In N. Ide, D. Greenstein, and P. Vossen (Eds.), *Special Issue on EuroWordNet* (Vol. 32(2-3), p. 73-89).

Vossen, P. (2003). Ontologies. In R. Mitkov (Ed.), *The Oxford Handbook of Computational Linguistics* (p. 464-482). Oxford University Press.

Vossen, P. (2004). EuroWordNet: a multilingual database of autonomous and language-specific wordnets connected via an Inter-Lingual-Index. *Semi-special issue on multilingual databases IJL*, *17*(2).

Vossen, P., Agirre, E., Calzolari, N., Fellbaum, C., Hsieh, S. kai, Huang, C.-R., et al. (2008). KYOTO: a System for Mining, Structuring and Distributing Knowledge across Languages and Cultures. In N. Calzolari et al. (Eds.), *Proceedings of the 6th International Language Resources and Evaluation (LREC08).* Marrakech, Morocco: European Language Resources Association (ELRA). (/)

Wierzbicka, A. (1996). *Semantics. Primes and Universals*. Oxford, New York: Oxford University Press.

Winston, M. E., Chaffin, R., and Herrmann., D. (1987). A taxonomy of part-whole relations. *Cognitive Science*, *11*(4), 417-444.

Xu, F., Kurz, D., Piskorski, J., and Schmeier, S. (2002). A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping. In *Proceedings of the 3rd International Conference on Language Resources an Evaluation (LREC02)* (p. 29-31). Las Palmas, Canary Islands, Spain.

Appendix

CQs and Questionnaire about the hands-on activity carried out with ATHENS students to evaluate the LSPs application and the proposed method for ODPs reuse in ontology modeling.

Number	Competency Questions (CQs) -	Answers
CQ1	Which are the types of Olympic Games?	Summer Olympic Games and Winter Olympic Games
CQ2	Which are the sports that make up the Summer Olympic Games?	Aquatics, Athletics, Gymnastics, Judo, Archery, Taekwondo, Tennis, Handball, Football, Cycling.
CQ3	In which summer sports can women participate?	Aquatics, Athletics, Gymnastics, Judo, Archery, Tennis, Handball, Football.
CQ4	Which summer sports are only for men?	
CQ5	What is the difference/relation between sports and disciplines?	
CQ6	Which are the disciplines included in the Aquatics sport?	Diving, Swimming, Waterpolo, Synchronized swimming.
CQ7	Which are the types of disciplines into which volleyball is divided?	Volleyball or Beach volleyball
CQ8	Who are the people that form part of the jury?	Peter Parker, John Doe, Mery Green
CQ9	Who are the members of a team?	
CQ10	Who are the winners in a discipline?	Nadia Comaneci, Jesús Carballo, Michael Phelps, Rafa Nadal (the list is not exhaustive)
CQ11	Who are the participants in a discipline?	Nadia Comaneci, Jesús Carballo, Michael Phelps, Rafa Nadal (the list is not exhaustive)
CQ12	Which is the name of the participant?	Nadia Comaneci, Jesús Carballo, Michael Phelps, Rafa Nadal (the list is not exhaustive)
CQ13	Which is the age of the participant?	
CQ14	Which is the identification number of the participant?	
CQ15	Which is the country of origin of the participant?	United States, Finnland, Germany, Greece, France, Spain, Portugal (the list is not exhaustive)
CQ16	Which participant participates in more than one discipline?	
CQ17	Where are Olympic Games organized?	Beijin, Torino, Athens, Salt Lake, Sydney, Nagano, Atlanta
CQ18	Which are the types of medals?	Gold, Silver, Bronze.
CQ19	How many medals did a participant win?	
CQ20	Which are the events or ceremonies that the Olympic Games consist of?	Opening Ceremony, Closing Ceremony, Medal presentation.
CQ21	Which are the constituents of the Olympic Movement?	National Olympic Committees (NOC), International Olympic Committees (IOC) and International Federations.

Figure 13.1: CQs about the olympic games used in LSPs experiment

Questionnaire about the hands-on activity

ATHENS 2009

Questionnaire about the formulation of sentences for the reuse of ODPs basing on LSPs.

Please answer each question with some detail and be honest!

Thank you very much.

1. Was the formulation of the sentences a difficult or an easy task?

---

2. Were the *Recommendations* given easy to understand?

---

3. Do you find the *Recommendations* useful? Why? Why not?

---

4. Did you miss any *Recommendation*? (Feel free to propose some...)

---

5. Did you find the CQs useful for the subsequent formulation of the sentences? Why? Why not?

---

6. Do you think this approach can be useful for users that are not experts in Ontology Engineering?

---

7. How could the approach be improved? Any ideas?

Figure 13.2: Questionaire about the hands-on activity with ATHENS students