UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN

Ph.D. Thesis

A formalization for mapping discourses from business-based technical documents into controlled language texts for requirements elicitation

Bell Manrique Losada

Medellín, Octubre de 2014

SUMMARY OF Ph.D. THESIS

Tittle:	A formalization for mapping discourses from business-based technical documents into controlled language texts for requirements elicitation
Director	Ph.D. Carlos Mario Zapata Jaramillo Email: cmzapata@unal.edu.co
Author:	MSc. Bell Manrique Losada Email: bmanriq@unal.edu.co, belleta55@gmail.com
Program:	Doctorado en Ingeniería
Area:	Sistemas e Informática
Keywords:	Requirements Elicitation, Natural Language, Controlled Language, Technical Document, Natural Language Processing, Linguistics Engineering, Domain Knowledge
Dissertation Committee:	Ph.D. René A. Venegas Velásquez Ph.D. Diego A. Burgos Herrera Ph.D. Jairo Hernán Aponte Melo
Dissertation Date:	December 11, 2014

© Bell Manrique-Losada, 2014. All rights reserved.

CONTENTS

	Page
LIST OF TABLES	5
LIST OF FIGURES	6
ABSTRACT	8
CHAPTER 1 INTRODUCTION	9
1.1 PROBLEM STATEMENT	9
1.2 RELATED WORK AND MOTIVATION	10
1.3 OBJECTIVES AND JUSTIFICATION	10
1.4 THESIS FOCUS AND KEY CONTRIBUTIONS	11
1.5 THESIS OUTLINE	14
CHAPTER 2 CONCEPTUAL FRAMEWORK	16
2.1 REQUIREMENTS ENGINEERING AND REQUIREMENTS ELICITATION	16
2.1.1 The Requirements Elicitation (RE) process	17
2.1.2 Natural and Controlled Language in RE	18
2.1.3 RE Techniques	20
2.2 DOMAIN KNOWLEDGE ACQUISITION AND REPRESENTATION	21
2.2.1 Knowledge Acquisition (KA)	22
2.2.2 Role and challenges of KA in the RE process.	22
2.3 NATURAL LANGUAGE PROCESSING	23
2.3.1 Linguistic Analysis	24
2.3.2 Text Analytics	27
2.4 DISCOURSE ANALYSIS AND CORPUS LINGUISTICS	28
2.4.1 Discourse Analysis	28
CHAPTER 3 STATE-OF-THE-ART REVIEW	34
3.1 MODELING DOMAIN INFORMATION IN THE RE PROCESS	34
3.1.1 Discourse Modeling based on natural language descriptions	34
3.1.2 Modeling and representing knowledge in RE process	37
3.2 SYSTEMATIC REVIEW FOR MAPPING NLCL INTO THE RE PROCESS	40
3.3 KNOWLEDGE AND PATTERN IDENTIFICATION OF LANGUAGE IN THE RE PROCESS	43
3.3.1 Document Processing by applying NLP	43
3.3.2 Knowledge Identification from Business Processes	45

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 4 FRAMEWORK FOR THE FORMALIZATION 50
4.1 CHARACTERIZATION OF THE ELICITATION TECHNIQUE 50
4.1.1 Referents of the Elicitation Technique 50
4.1.2 Background Reading 51
4.2 FOUNDATIONS AND ASSUMPTIONS FOR THE MAPPING FORMALIZATION 52
4.2.1 Linguistics Foundations and Assumptions 52
4.2.2 Foundations and assumptions of Business-based Technical Documents 53

CHAPTER 5 FORMALIZATION OF THE MAPPING PROCESS 56
5.1 RHETORICAL ORGANIZATION MODEL 56
5.1.1 Corpus definition 56
5.1.2 Analysis of the digital corpus 57
5.1.3 ROM Design 59
5.2 BEHAVIORAL MODEL 69
5.2.1 Processing Model 69
5.2.2 Method for Mapping 78
5.3 ARCHITECTURAL MODEL 80

CHAPTER 6 MODEL VALIDATION 82
6.1 EMPIRICAL VALIDATION 82
6.1.1 Hypotheses for the validation 82
6.1.2 Empirical validation strategies 82
6.1.3 Organizing and conducting the case study 84
6.1.4 Description and Results of the Validation Studies 89
6.2 DEVELOPMENT PROCESS OF THE SOFTWARE PROTOTYPE 101
6.2.1 Approach 1: Software app for language pattern recognition 101
6.2.2 Approach 2: Prototype of the Software System NAHUAL 108

CHAPTER 7 PUBLICATIONS 116

CHAPTER 8 CONCLUSIONES AND FUTURE WORK 119
8.1 CONCLUSIONS 119
8.2 FUTURE WORK AND OPEN ISSUES 121

APPENDICES (ATTACHMENT) 123

© Bell Manrique-Losada, 2014. All rights reserved.

LIST OF TABLES

	PAGE
TABLE 3.1. SUMMARY OF FINDINGS FOR DISCOURSE MODELING BASED ON NL DESCRIPTIONS	36
TABLE 3.2 RESULTS FROM THE 1ST SEARCH PHASE	40
TABLE 3.3 RESULTS FROM THE 2ND SEARCH PHASE	41
TABLE 3.4 SYNTHESIS OF PRIMARY STUDIES IN CATEGORY A	41
TABLE 3.5 SYNTHESIS OF PRIMARY STUDIES IN CATEGORY B	42
TABLE 3.6 SYNTHESIS OF PRIMARY STUDIES IN CATEGORY C	43
TABLE 3.7 SYNTHESIS OF PRIMARY STUDIES	45
TABLE 5.1 SAMPLE OF RESULTS FOR WORD FREQUENCY	58
TABLE 5.2 PROPOSAL OF MORPHOSYNTACTIC FEATURES	63
TABLE 5.3 PROPOSAL OF FUNCTIONAL AND LEXICAL FEATURES	64
TABLE 5.4 RESULTS AND ACTIONS ON THE REFERENCE MODEL	66
TABLE 5.5 PROPOSAL OF RHETORICAL ORGANIZATION MODEL (ROM)	67
TABLE 5.6 SAMPLE OF PRIORITIZED VERBS FOR VERB-CENTERED ANALYSIS	71
TABLE 5.7 SAMPLE OF PRIORITIZED VERBS FOR VERB-CENTERED ANALYSIS	72
TABLE 5.8 PROPOSAL OF RULES PATTERN FOR MAPPING	75
TABLE 5.9. SECTIONS OF JSD	76
TABLE 5.10. EXTRACTED PES	77
TABLE 6.1 SAMPLE OF EXTRACTED PHRASES BY WORK TEAMS	90
TABLE 6.2. LIST OF CONCEPT AND CONCEPTS INTERACTION	94
TABLE 6.3. LIST OF PHRASES WITH CONCEPT INTERACTION	94
TABLE 6.4. SAMPLE OF FIRST 10 RESULTS OF THE MAPPING	98
TABLE 6.5. SUMMARY OF RELATIONS COMPARISON OF THE MAPPING	98
TABLE 6.6. SAMPLE LIST OF THE IDENTIFIED RELATIONS	99
TABLE 6.7. SAMPLE LIST OF THE IDENTIFIED RELATIONS	99
TABLE 6.8. GENERATED DISCOURSE IN UN-LENCEP FROM ORIGINAL PROCEDURE	106
TABLE 6.9. GENERATED DISCOURSE IN UN-LENCEP FROM THE PROPOSED PROCEDURE	107
TABLE 6.10. GENERATED DISCOURSE IN UN-LENCEP FROM PROPOSED PROCEDURE	109
TABLE 6.11. MAPPING RULES IMPLEMENTED IN NAHUAL	112

© Bell Manrique-Losada, 2014. All rights reserved.

LIST OF FIGURES

PAGE

FIGURE 1.1 REQUIREMENTS ELICITATION FRAMEWORK 12
FIGURE 1.2 GENERAL FRAMEWORK FOR MAPPING 12
FIGURE 1.3 ROM FOR MAPPING 13
FIGURE 1.4 PROCESSING MODEL 13
FIGURE 1.5 ARCHITECTURAL MODEL 13
FIGURE 2.1 SPIRAL VIEW OF THE REQUIREMENTS ENGINEERING PROCESSES. 17
FIGURE 2.2 FRAMEWORK OF REQUIREMENTS ELICITATION PROCESS 18
FIGURE 2.3 CATEGORIES FOR RE METHODS. 20
FIGURE 2.4 ROLE OF KA IN THE RE PROCESS. 23
FIGURE 2.5 LINGUISTICS CATEGORIES. 24
FIGURE 3.1. BASED-TASK CONCEPTUAL MODEL 34
FIGURE 3.2. SEMANTIC NET MODEL 36
FIGURE 3.3. THE KAOS META-MODEL 37
FIGURE 3.4. THE RPGS META-MODEL 38
FIGURE 3.5. META-MODEL OF REQUIREMENTS ENGINEERING DOMAIN KNOWLEDGE 38
FIGURE 3.6. META-MODEL OF ONTOLOGY FOR REQUIREMENTS 39
FIGURE 3.7 .MODEL TO GENERATE LEL FROM REQUIREMENTS 39
FIGURE 5.1 EXAMPLE OF STOPWORDS 58
FIGURE 5.2 EXAMPLE OF CONCORDANCES 59
FIGURE 5.3 PRELIMINARY RHETORICAL ORGANIZATION MODEL 61
FIGURE 5.4 SAMPLE OF COLLOCATIONS FREQUENCY 64
FIGURE 5.5 ANALYSIS OF CO-OCCURRENCES FOR THE RELEVANT FEATURE 'MANAGEMENT' 65
FIGURE 5.6 ROM FOR MAPPING 68
FIGURE 5.7 GENERAL FRAMEWORK FOR THE MAPPING PROCESS 69
FIGURE 5.8 PROCESSING MODEL 69
FIGURE 5.8. PROCESS FOR IDENTIFYING PES 76
FIGURE 5.9. SOME FREQUENT VERBS, NOUNS, AND ADJECTIVES. 77
FIGURE 5.10. METHOD FOR MAPPING. 78
FIGURE 5.11. DESIGN PHASE. 78
FIGURE 5.12. PROCESSING PHASE. 79
FIGURE 5.13. MAPPING PHASE. 80
FIGURE 5.14. ARCHITECTURAL COMPONENTS. 80
FIGURE 6.1. DESIGN TO DE VALIDATION STUDY 83
FIGURE 6.2. DESIGN PROCESS OF VALIDATION PROOFS 84
FIGURE 6.3. HIERARCHY OF MANUAL MACRO-GENRE 86
FIGURE 6.4. CLASSIFICATION OF THE PROCEDURE MANUAL GENRE 87
FIGURE 6.5. CLASSIFICATION OF JOB STANDARD DOCUMENT MACRO-GENRE 88
FIGURE 6.6 PHRASES REPRESENTATION FROM WORK TEAM 1 91

© Bell Manrique-Losada, 2014. All rights reserved.

FIGURE 6.7 PHRASES REPRESENTATION FROM WORK TEAM 2 91
FIGURE 6.8 PHRASES REPRESENTATION BY THE EXPERT. 91
FIGURE 6.9 SETUP FOR PRE-PROCESSING OF TESTING-TEXT. 97
FIGURE 6.10 SAMPLE OF TAGGING OF TESTING-TEXT INSIDE PROCESSING. 97
FIGURE 6.11 PRECISION AND RECALL 100
FIGURE 6.12. GENERAL APPLICATION ARCHITECTURE 104
FIGURE 6.13. BACK-END SUB-COMPONENTS FOR THE APP ANTLR 105
FIGURE 6.14. DOMAIN MODEL 109
FIGURE 6.15. PROCESS MODEL 110
FIGURE 6.16. PROCESS MODEL FOR SPECIFIC ACTIVITIES 110
FIGURE 6.17 CLASSES MODEL FOR NAHUAL 111
FIGURE 6.18 NAHUAL COMPONENT MODEL 113

© Bell Manrique-Losada, 2014. All rights reserved.

ABSTRACT

Requirements elicitation for software engineering is a process for obtaining, analyzing, and specifying requirements supported by stakeholders—analysts, clients, domain experts, and final users, among others. In this process we generate either textual or graphical descriptions, reflecting the most relevant concepts of the stakeholder domain for developing a software product and the related domain knowledge. Based on well-known elicitation techniques, the intervention of the stakeholders is overrated, since interviews, dialogues, and inspection are the most commonly used methods on them. Such methods—subjective and directly dependent on human beings—cause loss of sequence and conciseness, and waste of elicitation time and cost. The intervention of the stakeholders in the process leads to problems related to the usage of natural language: a lot of unstructured, redundant information and overuse of synonyms and ambiguities, among others.

Several approaches have been developed for reducing the language differences between stakeholders and analysts: natural language processing of requirements documents; semi-automated identification of lexical features in requirements specifications; semi-automated optimization of feature identification; controlled English for representing knowledge; identification of knowledge domain from documents in several domains. Some progress has been made in solving this problem, but either such progress is related to other phases than the requirements elicitation or it exhibits no relation to the domain knowledge. Some of such progress is focused on techniques like information retrieval methods, identification of regular expressions, and text mining. These techniques can be applied to the requirements elicitation process, regarding the technical and methodological aspects.

Nowadays, well-known elicitation techniques require: i) high involvement of stakeholders and analysts; ii) information-specification mapping process; iii) requirements description previous to the design model conversion; and iv) a method for guiding the elicitation process. In addition to the aforementioned techniques, some other ones known as synthetic/analytical techniques non-dependent on human intervention are useful for considering diverse sources of domain knowledge. Several approaches for studying elicitation techniques have been identified in the state-of-the-art review, but they scarcely use analytical techniques focused on documents. In this research we analyze documents written by the stakeholders in their domains as a source for requirements elicitation. We propose a model, comprising methodical and structural components, to set up the elicitation process by applying analytical elicitation techniques based on technical documents (e.g. policies, regulations, and manuals).

Consequently, as a solution to deal with language differences, in this proposal we work with controlled languages—existing for specifying requirements—and the natural language from the stakeholders domain, directly translated from technical documents. We propose a formalization of mapping which comprises a variety of more linguistically-informed methods available—based on rhetorical analysis discourse and linguistics processing—which treat each document as a potential input for the natural language processing aiming requirements elicitation. The application of the model based on the analytical technique is expected to produce the following results: understanding and structured representation of the business and organizational information; comprehension of the stakeholder domain; and subsequent application of subjective techniques for specifying requirements.

This Ph.D. Thesis is concerned with natural language processing for the requirements elicitation process, looking for a model for transforming discourses on technical documents into controlled language texts. In the model we specify the transformation process based on patterns—functional, structural, and linguistic patterns—from source technical documents, for obtaining organizational domain knowledge and business information, useful for the requirements elicitation process.

© Bell Manrique-Losada, 2014. All rights reserved.

ACKNOWLEDGEMENTS

The Ph.D. Thesis presented herein was sponsored by a COLCIENCIAS studentship and supported by the National Program of Researchers Education 'Generación del Bicentenario", an initiative of the Government Colombian for promoting the high-level education. Further, this work has been funded in part by the Universidad de Medellín through Vicerrectoría de Investigaciones.

This Ph.D. Thesis compiles and corrects the results published in several articles written over the last four years in the framework of my Ph.D. Program. I am indebted for the excellent working environment

I have been offered during these years at the projects office of the Computing and Decisions Sciences Department, Faculty of Mines, Universidad Nacional de Colombia UNAL–Medellín and at the Research Group ARKADIUS of the Engineering Faculty, Universidad de Medellín; further grateful for the advice and encouragements provided by countless doctoral fellow, classmates, colleagues, friends, and family members.

I am greatly indebted to the students of all the tools, experiments, and support material covered in this thesis for kindly making them available, and in many cases for helping me with the usual teething problems—particularly César, Daniel, Jorge, and Juan Ricardo of UNAL-Medellín for their crucial debugging efforts, and Marina, Diego, Lina, Vivianas, and Tomás for supporting the state-of-the-art review and the case studies. Additionally, I am very grateful to the contributor of the ELV at PUCV for supporting the construction of the corpora and their painstaking annotation work. Without the tools and the data to test them on, none of this would have been possible; specially, I deeply appreciate the support provided by Sofia, Maria Teresa, and Juan David from the FONDECYT Project.

I would like to thank the following people:

- My doctoral advisor Carlos Mario Zapata Jaramillo for giving me the opportunity to work in his research group, write my Ph.D. Thesis with him, and for being a constant voice of reason. I sincerely appreciate his constant support and guidance, but especially becoming a life model to follow for his invaluable knowledge, teachings, and example of professional and academic performance.
- My examiners Diego A. Burgos and René A. Venegas, for all their time and advice: To Diego for suggesting I investigate linguistics patterns and algorithms for the prototype and experimentation; René for guiding my immersion in the world of discourse analysis and review my achievements and progress with sincere affection.
- The readers of my advances and/or white chapters, and the anonymous reviewers of our papers, all of whom provided critical advice and feedback, among them: Guillermo González, Roberto Manjarrés, Gloria P. Gasca, Martín Solari, Pedro Patiño, and Juan David Martínez.
- Martín Solari for guiding the development of the empirical validation methodology and for propping up the experimental orientation of this Ph.D. Thesis.
- My colleagues at Universidad de Medellín for contributing and supporting this research idea, specially Gloria P. Gasca and Lillyana M. Giraldo, for being an encouragement and motivation for my process; Jaime A. Echeverri and Liliana González for sharing the beginning of this research and the life as doctoral students;

This thesis is dedicated, with love: to my husband Juan Camilo for her affection, passion, sympathy, encouragement, and tolerance of my odd working hours; to my Mom and Sister, for all their support, trust, and push in all slack years of dedication to this work and my absence for them; and, to my angels in heaven, who accompanied me, guided, and illuminated this difficult but beautiful formation process—particularly to my Dad, who I think he would have felt especially proud of this achievement.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 1

INTRODUCTION

If the facts don't fit the theory, change the facts.
Albert Einstein

Requirements Elicitation (RE) is the first stage of requirements engineering, commonly followed by analysis and specification, integration and validation of the requirements. In broader terms RE is one the most critical stages of the software engineering process (Castro-Herrera *et al.*, 2009; Zowghi & Coulin, 2005), given the big set of activities and stakeholders involved—analysts, clients, domain experts, and final users, among others.

The goals of RE are: discovering, reviewing, documenting, and understanding the user needs, the context/environment, and the system constraints. In order to achieve such goals, an analyst should obtain and analyze the needs and constraints, and then specify them in terms of *requirements* (Nuseibeh & Easterbrook, 2000). The specifications resulted of RE are either textual or graphical descriptions (Kotonya & Sommerville, 1998; Constantine & Lockwood, 1999), reflecting the most relevant concepts and processes of the domain and the stakeholders for developing a software product.

1.1 PROBLEM STATEMENT

Owing the complexity involving the activities of RE the industry and the state of the art exhibit multiple techniques available to perform such activities (Hickey & Davis, 2003). Hundreds of techniques and approaches from different sources have been employed for RE but some of them are more widely used, which are called 'well-known techniques.'

Since the requirements elicitation process is strongly related to the context, the specific features of the project and the environment (Zowghi & Coulin, 2005), the involvement of various stakeholders, sources of information, and organizational aspects is crucial. Based on the application of well-known elicitation techniques, the intervention of stakeholders is overrated, because several methods mostly based on inspection, interviews, and dialogue activities are commonly used. Such methods—by being directly dependent on human beings, besides mainly subjective and observational—generate loss of sequence and conciseness, as well as increasing of elicitation times and costs. The obtained information and the descriptions of the application domain, captured in in the specifications and arising from the delayed and meticulous work with the stakeholders, have many problems derived from the usage of natural language (Murtaza *et al.,* 2013). Such problems are mainly related to a lot of unstructured and contradictory information, indistinct use of synonyms and ambiguities, inconsistencies, and redundant information. In this sense, we have identified the following specific problems, to be solved by this research:

- *Communication language.* One of the main difficulties of requirements elicitation—based on the well-known techniques—is the imprecise and informal nature of requirements. Such nature leads to problems in the translation from informal observations of the real world to any specification language (Yu & Liu, 2001).
- *Context understanding.* RE needs an understanding of both the organization in which the system will be located and the system operation within the organizational context. According to Mittermeir (1990), considering the organizational context from the first activities of RE helps in solving a number of subsequent restrictions related to misconceptions, management politics, technical ignorance, mistrust, established practices, and people resistance, among others. An organizational and contextual analysis allows for identifying and capturing goals, and then verifying the usability and correctness of the requirements.

© Bell Manrique-Losada, 2014. All rights reserved.

- *Scope.* The RE process should start with a broad organizational and contextual analysis in order to establish boundary conditions of the target system. Contextual issues leading to incomplete, unnecessary, or unusable requirements should be avoided (Christel & Kang, 1992).
- *Subjectivity and special skills.* During RE process, a high degree of communication, negotiation, and other interaction skills is needed from the analyst. The subjectivity is analyzed from three perspectives: *i)* the different views on what the system should do and what they have to deliver to the stakeholders; *ii)* the same words and phrases used by different stakeholders, leading to different things for them; and *iii)* the high degree of intervention from the analyst for identifying relevant information, for capturing knowledge, and for understanding/interpreting the elicited information.

1.2 RELATED WORK AND MOTIVATION

Several approaches have been developed for the sake of reducing the language differences between stakeholders and analysts. Most of them are focused on the processing of structured or semi-structured documents from requirements processes, as follows: processing of requirements specification documents (RSD, Nanduri & Rugaber, 1995; Ambriola & Gervasi, 2006); identification of lexical features in specifications and linguistics features from texts (Natt *et al.,* 2005; Natt *et al.*, 2006; Post *et al.*, 2011); and, optimization of feature identification in the former process (Dalianis, 1995; de Sousa *et al.*, 2010; Deeptimahanti & Sanyal, 2011; Popescu *et al.*, 2008).

Besides, we have found approaches from disciplines other than requirements elicitation, including controlled languages for knowledge representation; and identification of knowledge domain and business processes from documents (Reiter & Dale, 1997; Hinge *et al.*, 2009; Breaux *et al.*, 2008). The purpose of the latter approach is not related to software development.

As we describe above, the research community has a growing interest in approaching to the problem at hand within the requirements elicitation domain, but the advances are not precisely focused on the initial phase of the domain recognition and understanding. Despite this, we found useful contributions regarding technical and methodological aspects, such as retrieval techniques, regular expressions, text mining methods (Sampaio *et al.*, 2007), and other relatively lightweight approaches. However, some variety of more linguistically-informed methods are available which treat each document as a potential input for natural language processing.

1.3 OBJECTIVES AND JUSTIFICATION

The general objective of this Ph.D. Thesis is improving the requirements elicitation process by defining a natural-language-based model for transforming business-based technical documents into controlled language discourses. In this sense, technical documents play a relevant and guiding role in the requirements elicitation process, by seeking early identification and comprehension of business information and domain knowledge. The specific objectives for achieving the previous goal are:

- α Analyzing several types of business-based technical documentation and defining a knowledge representation approach, containing the main concepts and relationships belonging to the domain.
- α Defining heuristics rules for expressing the equivalence between business-based technical documentation and controlled language discourses.
- α Proposing a model to formalize the transformation of corporate technical documents—written in natural language—into a controlled language discourse, for generating domain knowledge and useful information to the requirements elicitation process.
- α Validating the proposed model by means of a case study belonging to a corporate specific domain

© Bell Manrique-Losada, 2014. All rights reserved.

This Ph.D. Thesis is justified by the following points of view: i) Significance of domain knowledge inside RE; ii) relevant role of the elicitation techniques based on documentation analysis; iii) preliminary analysis process of the current organization, before requirements specification process; and iv) benefits of linguistic processing of documents.

i) **Significance of the domain knowledge inside the RE process.** Several contributions have been developed for optimizing feature identification in the initial stages of RE process (Boutkova & Houdek, 2011). Domain knowledge is relevant to the RE process since the analyst should collect enough information for:
   - ✓ Identifying the problem boundaries under study
   - ✓ Recognizing the problem domain and the scope
   - ✓ Identifying the stakeholders involved and their goals
   - ✓ Understanding and comprehending usage scenarios and their constraints
   - ✓ Identifying feasibility and risk associated with the solution of the identified problem

ii) **Relevant role of the elicitation techniques based on documentation analysis.** We present a proactive approach to promote the relevant role of the elicitation techniques based on documents analysis, as sources of domain knowledge and business information within the RE process. This research is based on application of a data-gathering technique called 'background reading.' Rogers *et al.* (2011) describe the data-gathering technique for revising documents, as a technique for learning about quantitative procedures, regulations, and standards from a specific system. This approach has the advantage of requiring minimum involvement of the users even though some authors show a gap in the practical usage of the technique (Rogers *et al.* 2011), since the day-to-day work should differ from documented procedures on which it is based.

iii) **Preliminary analysis process of the current organization, before the requirements specification process.** A preliminary analysis of the current organization is an important source of business information and goal identification. This analysis depends on organizational documents and results in a list of problems and deficiencies, which needs to be precisely formulated. Van Lamsweerde (2004) suggests making a first list of goals to be achieved for the system-to-be, by using available documents, interview transcripts, and manuals, among others. Also, this list can be useful when looking for intentional keywords in the documents (van Lamsweerde, 2000). In frameworks such as *i** from the Tropos project (Yu & Liu, 2001), goals and tasks are identified and refined by using means-ends and task decompositions for developing flexible software systems. Besides, goals and tasks guide the development of agent-based systems from the early requirements analysis through architectural design and detailed design to the implementation.

iv) **Benefits of linguistic processing of documents**. We propose a functional and solid approach to use the rhetorical discourse analysis (RDA) and linguistic analysis for processing and studying technical documents as a source of domain knowledge within the RE process. RDA has been applied in a large number of computational applications (Taboada & Mann, 2006a) and computational linguistics applications—*i.e.*, generation, machine translation, parsing, summarization—(Hovy, 1993; Vander Linden & Martin, 1995; O'Donnell et al., 2001; Marcu, 2000). Domain knowledge is related to matters or facts from a particular domain expressing the intentions of their stakeholders. According to Taboada (2004), this kind of knowledge may be captured by information about a holistic structure from documents and expressed as knowledge about a particular genre. Our approach is based on the idea of considering the communicative purpose of the documents, their functionality according to the writer original intention, and the context in which the documents are enrolled.

1.4 THESIS FOCUS AND KEY CONTRIBUTIONS

The research activity of this Ph.D. Thesis is focused on the requirements elicitation (RE) as the core process in requirements engineering. The following activities occur in the RE general process: identification,

© Bell Manrique-Losada, 2014. All rights reserved.

capturing, analyzing, and specifying domain information, looking for obtaining a set of requirements as a result. The first two activities are related to the *specific domain* and the final activities are linked to the *solution domain*, as we show in Figure 1.1. This Ph.D. Thesis is focused on the side of *specific domain*—*i.e.,* it is a business-centered processes—mainly on the identification activity, but from the perspective of the document analysis.

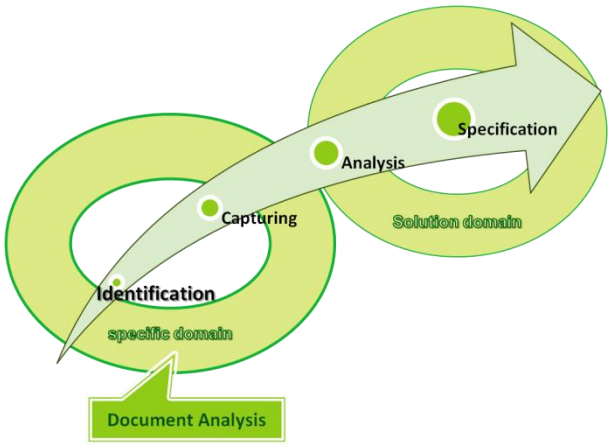


Figure 1.1 Requirements elicitation framework

The key contribution of this Ph.D. Thesis is addressing the problem of mapping business-based technical documents—included in the requirements elicitation process—into controlled language texts, for generating domain knowledge and business information to requirements elicitation. In the pursuance of this contribution, we use a framework named NAHUAL[1]. We formalize the mapping of business-based technical documents in terms of three frames, as schematically we show in Figure 1.2. The first frame is for representing the functional and structural patterns of the technical documents to be processed, organized as a *Rhetorical Organization Model* (see Figure 1.3).

Figure 1.2 General framework for mapping

[1] The name 'NAHUAL' will be justified and described on the Section 4.2.

© Bell Manrique-Losada, 2014. All rights reserved.

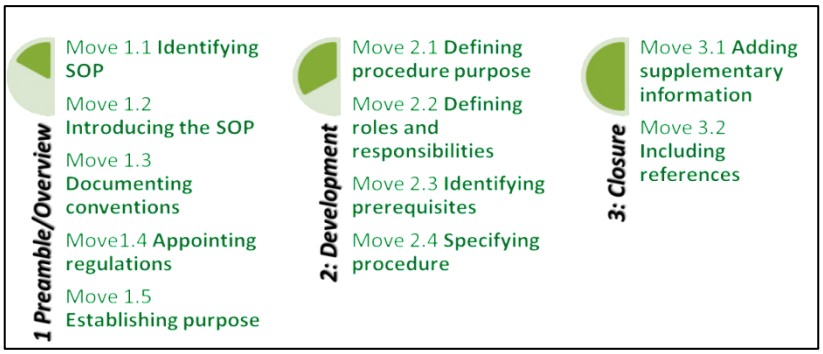


Figure 1.3 ROM for mapping

In the second frame we describe the *Behavioral Model* comprising both a 'processing model' for mapping and a 'mapping method' which should be used for developing the mapping process from scratch with other kinds of technical documents. A representation of the processing model is presented in Figure 1.4. Lastly, in the third frame we propose an *Architectural Model* with the components that should be developed for mapping and the relationships among the components, as we show in Figure 1.5.

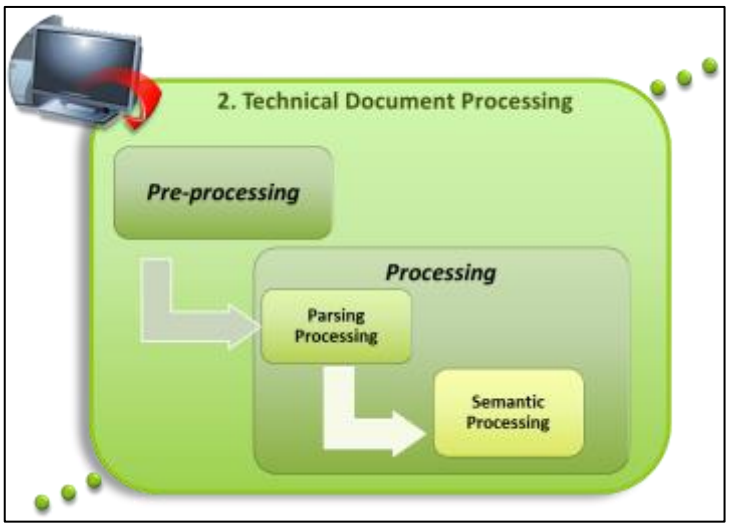


Figure 1.4 Processing model

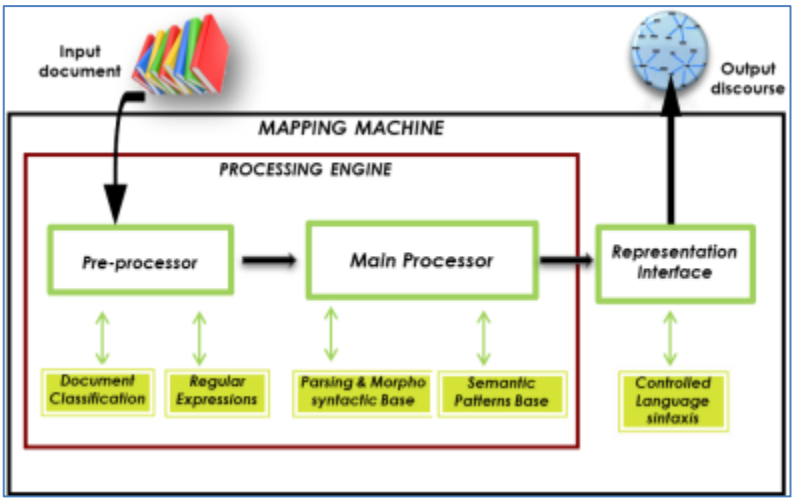


Figure 1.5 Architectural Model

© Bell Manrique-Losada, 2014. All rights reserved.

Specifically, the contribution of this research can be summarized as follows:

- ✓ A transformation model for achieving an initial understanding of the organizational context.
- ✓ A proposal for using *document analysis* as a requirements elicitation technique conductive to reduce time and costs in the initial phases of the software development process.
- ✓ A probed method for employing technical documentation (TD) as data source in RE.
- ✓ An approach to follow best practices when defining a TD structure, focused on administrative and business disciplines.
- ✓ A method for identifying 'domain knowledge' and 'business information' from the early stages of the requirements elicitation
- ✓ Innovation in early stages of the requirements engineering inside software development, by considering several sources for helping to reduce costs and facilitate subsequent processes.

1.5 THESIS OUTLINE

This Ph.D. Thesis is organized as follows.

In Chapter 2 we sketch the requirements elicitation process, its well-known techniques, and the conceptual basis from the linguistics and document processing point of view.

In Chapter 3, we describe the state-of-the-art review and we explain several approaches found for representing knowledge when mapping natural language into controlled language in the context of the requirements elicitation process.

In Chapter 4, we describe our framework in terms of the base elicitation technique, linguistic foundations and models, and the specification of technical documentation from organizations.

In Chapter 5, we propose our formalization for mapping. The formalization is described and represented by means of the following models: i) a Rhetorical Organization Model representing the functional and structural patterns of the technical documents to be processed, ii) a Behavioral Model comprising a processing model for mapping and a mapping method which should be used for developing the mapping process from scratch with other kinds of technical documents, and iii) an Architectural Model describing the components and their relationships for executing the mapping.

In Chapter 6, we present the mapping implementation by using a functional prototype. We also validate our mapping formalization by using empirical strategies.

In Chapter 7 we present the list of publications related to this Ph.D. Thesis, such as journal articles, conference proceedings, book chapters, and other divulgation activities.

In Chapter 8 we draw the conclusions and future work on the subject.

REFERENCES

A. Hickey and A. Davis, "A Tale of Two Ontologies: The Basis for Systems Analysis Technique Selection," 2003 Americas Conference on Information Systems (AMCIS 2003).

A. Lapouchnian (2005). Goal-Oriented Requirements Engineering: An Overview of the Current Research. Retrieved from http://www.cs.utoronto.ca/~alexei/pub/Lapouchnian-Depth.pdf

A. Post, I. Menzel and A. Podelski (2011). Applying restricted english grammar on automotive requirements: does it work?. A case study. Proceedings of the *REFSQ'11*. Vol. 1. pp. 166-180.

A. Sampaio, A. Rashid, R. Chitchyan, and P. Rayson (2007). EA-Miner: towards automation in aspect-oriented requirements engineering. Transactions on aspect oriented software development III. Vol. 46. pp. 4-39.

A. van Lamsweerde (2000). Requirements Engineering in the Year 00: A Research Perspective. 22nd International Conference on Software Engineering (ICSE'2000), Limerick, Ireland.

© Bell Manrique-Losada, 2014. All rights reserved.

A. van Lamsweerde (2004). Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice. Proc. International Conference on Requirements Engineering (RE'04), Kyoto, Japan.
B. Nuseibeh, S. Easterbrook (2000). Requirements Engineering: A Roadmap. Proceedings Conference on the Future of Software Engineering, Limerick, Ireland.
C. Castro-Herrera, C. Duan, J. Cleland-Huang and B. Mobasher (2011). Using data mining and recommender systems to facilitate large-scale, open, and inclusive requirements elicitation processes. Proceedings of 16th IEEE International requirements eng. conference, pp.165–168, 2008.
D. Deeptimahanti and R. Sanyal (2011). *Semi-automatic generation of UML models from natural language requirements.* Proceedings of the 4th India Software Engineering Conference (ISEC '11). Thiruvananthapuram. India, pp. 165-174.
D. Marcu (2000). The Theory and Practice of Discourse Parsing and Summarization. Cambridge, MA: MIT Press.
D. Popescu, S. Rugaber, N. Medvidovic and D. M. Berry (2008). Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models. Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs, Vol. 53, pp. 103-124.
D. Zowghi and C. Coulin (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. Springer.
E. Boutkova and F. Houdek (2011). Semi-automatic identification of features in requirement specifications. In Requirements Engineering Conference (RE), 19th IEEE International.
E. Hovy (1993). Automated Discourse Generation Using Discourse Structure Relations. Artificial Intelligence 63: 341–85.
E. Reiter and R. Dale (1997). Building applied natural language generation systems. Natural Language Engineering, Vol. 3. pp. 57-87.
E. Yu and L. Liu (2001). Modeling Trust for System Design Using the *i** Strategic Actors Framework. In R. Falcone, M. Singh, Y.H. Tan (Eds.), Trust in Cyber-Societies –Integrating the Human and Artificial Perspectives. LNAI-2246.
G. Kotonya and I. Sommerville (1998). Requirements Engineering: Processes and Techniques, John Wiley & Sons: Great Britain.
H. Dalianis (1995). *Aggregation in the NL-generator of the Visual and Natural language Specification Tool.* Proceedings of the 7*th* conference on European chapter of the Association for Computational Linguistics (EACL '95). Zaragoza, España, pp. 286-290.
H. Sharp, Y. Rogers, J. Preece (2011). Interaction Design: Beyond Human-Computer Interaction. Ed: Wiley, 2nd Edition, p214. . ISBN: 978-0-470-01866-8.
J. Kang and P. Saint-Dizier (2014). Requirement Mining in Technical Documents. In Proceedings of the First Workshop on Argumentation Mining, ACL 2014.
J. Natt, B. Regnell, V. Gervasi and S. Brinkkemper (2005). A Linguistic-Engineering Approach to Large-Scale Requirements Management. *IEEE Software*. Vol. 22, No. 1. pp. 32-39.
J. Natt, T. Thelin and B. Regnell (2006). An experiment on linguistic tool support for consolidation of requirements from multiple sources in market driven product development. Empirical Software Engineering. Vol. 11. pp. 303-329.
K. Hinge, A. Ghose and G. Koliadis (2009). *Process SEER: A Tool for Semantic Effect Annotation of Business Process Models*. Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference (EDOC '09). Auckland. New Zealand, pp. 54-63.
K. Vander Linden and J. Martin (1995). Expressing Rhetorical Relations in Instructional Text: A Case Study of the Purpose Relation, Computational Linguistics 21(1): 29–57.
L. Constantine and L.A.D. Lockwood (1999). Software for Use: A practical guide to the models and methods of usage-centered design, Addison Wesley: Reading, MA
M. Christel and K. Kang (1992). Issues in Requirements elicitation. Technical Report CMU/SEI-92-TR-012 ESC-TR-92-012. Software Engineering Institute. Available: http://www.sei.cmu.edu/library/abstracts/reports/92tr012.cfm
M. O'Donnell, C. Mellish, J. Oberlander and A. Knott (2001) 'ILEX: An architecture for a Dynamic Hypertext Generation System', Natural Language Engineering 7: 225–50.
M. Taboada (2004). Building Coherence and Cohesion: Task-Oriented Dialogue in English and Spanish. Amsterdam and Philadelphia, PA: John Benjamins.
M. Taboada and W.C. Mann (2006). Applications of Rhetorical Structure Theory. In Discourse Studies 2006, 8: 567.
M. Murtaza, J. Hussain Shah, A. Azeem, W. Nisar and Maria Masood (2013). Structured Language Requirement Elicitation Using Case Base Reasoning. In: Research Journal of Applied Sciences, Engineering and Technology 6(23). ISSN: 2040-7459.
P. Zave (1997). Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys, 29(4).
R. Mittermeir, N. Roussopoulos, R. Yeh and P. Ng (1990). An integrated approach to requirement analysis. Modern Software Engineering: Foundations and Current Perspectives. New York: Van Nostrand Reinhold, 119-164, Chapter 5, 1990.
S. Nanduri and S. Rugaber (1995). *Requirements validation via automated natural language parsing*. Proceedings of the 28*th* Hawaii International Conference on System Sciences. Hawaii, USA. pp. 362-368.
T. Breaux, A. Antón and J. Doyle (2008). Semantic parameterization: A process for modeling domain descriptions. *ACM Trans. Softw. Eng. Methodol.* Vol. 18. pp. 27.
T. de Sousa, J. Almeida, S. Viana and J. Pavón (2010). Automatic analysis of requirements consistency with the B method. SIGSOFT Softw. Eng. Notes. Vol. 35. pp. 1-4.
V. Ambriola and V. Gervasi (2006). On the Systematic Analysis of Natural Language Requirements with CIRCE. Automated Software Engg. Vol. 13. pp. 107-167.
Y. Rogers, H. Sharp and J. Preece. Interaction Design: Beyond Human-Computer Interaction. John Wiley and Sons, 3rd Edition, 2011

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 2

CONCEPTUAL FRAMEWORK

Experience without theory is blind, but theory without experience is mere intellectual play.
Immanuel Kant

The sections of this Chapter are devoted to the conceptual and theoretical foundations of this Ph.D. Thesis. We describe and develop the core knowledge areas and their related concepts about this Thesis, as follows: Requirements Elicitation, Domain Knowledge Analysis, and Language Processing. These knowledge areas are chosen particularly for their significance for understanding and developing the analytical processing from technical documents, from the requirements elicitation process point of view. We present an overview of concepts and theories supporting and providing the basis for each area.

In Section 2.1 we give an overview of the Requirements Elicitation process, techniques, and languages. In Section 2.2 we review the basics of Domain Knowledge Analysis and its challenges related to the requirements elicitation process. In Section 2.3 we include a brief introduction to well-known language processing techniques and we describe in some detail the kinds of analysis and annotations for supporting the parsing of written discourses. And finally, in Section 2.4, we present the foundations of Discourse Analysis and Corpus Linguistics.

2.1 REQUIREMENTS ENGINEERING AND REQUIREMENTS ELICITATION

One of the classic definitions of Requirements Engineering was offered by Ross and Schoman (1977) as a thorough assessment of the need to be fulfilled by a system, by considering the current or expected conditions from a context. According to Lapouchnian (2005), requirements engineering must deal with the following factors: i) the reasons why a software system is needed; ii) the functionalities for achieving the initial purposes; and iii) the constraints for designing the software. Such factors are covered by internal processes (van Lamsweerde, 2000; Nuseibeh & Easterbrook, 2000; Sommerville, 2005; Hull *et al.*, 2010), as follows:

- *Domain analysis*. The environment for the system-to-be is examined, including: relevant stakeholders, problems with the current system, opportunities for improvement, and target objectives.
- *Elicitation*. According to the identified objectives, alternative models for the target system are analyzed, and the requirements and assumptions of such models are identified.
- *Negotiation*. By evaluating the alternative requirements, assumptions, and, risks with the stakeholders, the best solution alternatives are chosen.
- *Specification*. Based on the selected solution, the requirements and assumptions are formulated and detailed. Also, such specifications are checked according to measurement and metrics.
- *Documentation*. Comprise the reporting and referencing of the decisions made during the whole process. It should eventually include the modifications or environmental changes occurring continuously.

Requirements engineering processes may vary significantly for implementing the above activities, depending on the application domain, the people involved in the process, and the context for developing the requirements. Nevertheless, the state-of-the-art exhibits a number of generic processes—called sometimes 'phases' or 'stages'—which have been adapted and defined by several authors (Kotonya & Sommerville, 1998; Jiang, 2005; Cremers & Alda, 2005; Pohl, 2010). One of the most popular models for representing such requirements engineering processes is presented in Figure 2.1, known as the spiral view. The main processes comprised in the spiral view are requirements elicitation, requirements specification, requirements validation, and systems requirements document.

© Bell Manrique-Losada, 2014. All rights reserved.

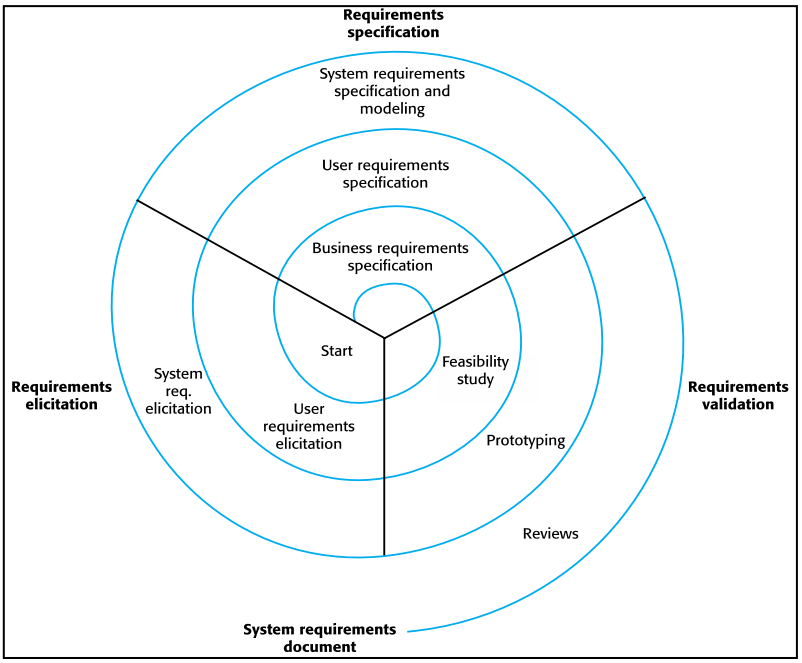


Figure 2.1 Spiral view of the Requirements Engineering processes. Source: Sommerville (2011)

In the next sections we focus on the Requirements Elicitation process, as the first one of the following common generic processes in most of the process models:

- Requirements elicitation
- Requirements analysis
- Requirements validation
- Requirements management

2.1.1 The Requirements Elicitation (RE) process

RE is the initial phase of requirements engineering. The main goal of the RE process is determining all the requirements which a future software application should meet to be considered of quality, for producing a good requirements specification (Christel & Kang, 1992). The specific goals are discovering, reviewing, documenting, and understanding the user needs and the system constraints. For achieving these goals, an analyst should increasingly and iteratively develop tasks involving the analysis of natural language (NL) and the language representation or modeling, respectively (Li *et al.*, 2003). Such tasks include: understanding the domain, capturing and classifying requirements, and negotiating requirements (Robertson & Robertson, 2008).

The RE process involves activities such as identification, capturing, analyzing, and specifying useful information, domain knowledge, and requirements from a specific domain. According to the sequence of activities for the RE process proposed by García and Monzón (2000), Arthur and Gröner (2004), and Lee (2013), we show in the Figure 2.2 our RE framework in terms of the activities and the domains involved. The structure illustrated is not necessarily linear in nature, *i.e.* RE is an iterative process constantly refining information, knowledge, and requirements.

© Bell Manrique-Losada, 2014. All rights reserved.

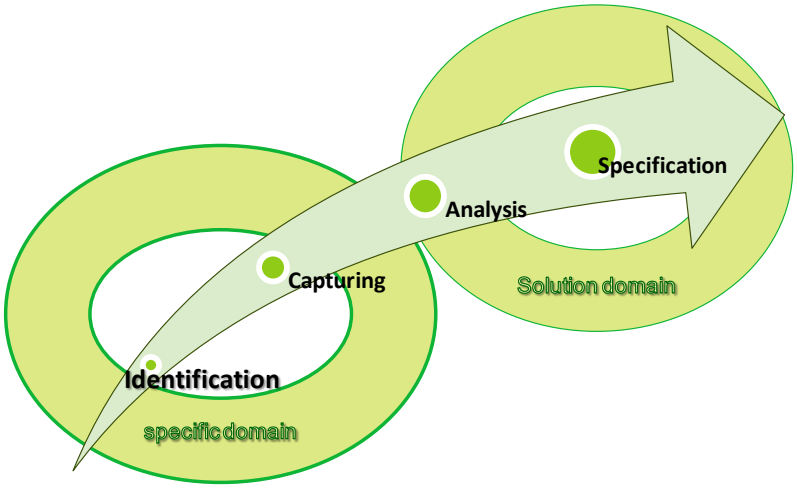


Figure 2.2 Framework of Requirements Elicitation process

The RE process is developed in two different domains: specific domain—problem domain—and solution domain. The specific domain is related to the primary context when a system or situation under study is studied. Users have a problem and a need to solve it and analysts have to understand the problem, the surrounding context, and the user expectations (García & Monzón, 2000). Such a domain exists in terms of a set of concepts, features, meanings, and relationships among them. In the RE process, an analysts tries to analyze and model the specific domain as a way to allocate the stakeholders needs and expectations in that context. Thus, such process provides the analyst an overview of the customer problem domain and needs.

The solution domain is related to the process for analyzing the stakeholder needs and expectations in terms of the solution they think suitable for the problem. Users should understand approaches supported by the analyst. Also, the user should discuss, modify, or discard such approaches, preferably before starting the following specifications activities. The importance of identifying the specific domain separated from the solution domain has been already discussed widely (Shekaran *et al.*, 1994; Jackson, 1995; García & Monzón, 2000; Barroca *et al.*, 2004; Thalheim, 2009; de Boer and & van Vliet, 2009) and identified as an essential activity of domain analysis inside the RE process.

Domain analysis is a term used to describe the systematic activities for identifying, formalizing and classifying the knowledge into a domain (Lee, 2013). In domain analysis we use a specialized domain terminology or reference to domain concepts. Due to the specialized terminology, analysts often misunderstand some concepts and relations, so we need several stakeholders involved in the process. Such stakeholders might vary in their domain-specific background (*e.g.* logistics or computing, engineering or management, services, outsourcing, supplying, etc.) and experience (e.g. trainee or expert), but they all should participate in the problem analysis for interpreting of outcomes. Additional to the analyst, other key stakeholder in the process is the domain expert. According to Neumann and Tolujew (2012), domain experts provide knowledge of the problem domain and especially problem-specific knowledge. Therefore, they are mainly involved in the problem description, the identification of input data, and the evaluation of results.

RE is primarily concerned with the communication between the analyst and the stakeholders as a way to gather the relevant domain information (Zapata & Manrique, 2012), considered the basis of requirements. Such communication is carried out in the frame of the specific domain. Likewise, RE is concerned with the domain understanding, looking for identifying and comprehending the knowledge of the general area where the system is applied. According to some authors (Christel & Kang, 1992; Cremers & Alda, 2005), the analyst should understand the domain in terms of the problem and business. The problem understanding involves the problem details regarding the specific customer of the system. The business understanding involves the recognition of how the existing systems interact and contribute for reaching the business goals, the needs and constraints of stakeholders, and the needs of people requiring system support for developing tasks.

© Bell Manrique-Losada, 2014. All rights reserved.

The products from the RE process can be summarized as follows:

- List of stakeholders involved
- Description of environment in which the system under study is enrolled—related to technical, legal, operational, and functional elements.
- Context models representing the boundaries of the problem and the solution
- Conceptual models representing the scope of the problem and the solution
- Statement of the stakeholders needs and expectations
- Statement of need and feasibility of the solution
- List of requirements organized by function and applicable domain constraints
- Set of usage scenarios providing use insight into operation of the deployed system
- Prototypes developed to better understand requirements

For the purposes of this Ph.D. Thesis, we focus on the activities to be performed into the specific domain or problem domain. Such activities are concerned with domain understanding, the relevant domain information gathering, and the knowledge identification of the general context where the system is enrolled.

2.1.2 Natural and Controlled Language in RE

In the RE process, the analyst should identify and capture most of the information related to the specific domain. The main source of such information is the stakeholder discourse, expressed in natural language (NL). Supported by several techniques, the analyst obtains such information and uses it to describe the problem domain, analyze the information, and represent the specifications typically by using models. Concerning to model building and according to Gangopadhyay (2001), the analyst identifies conceptual elements, understands their relationships, and represents them by using a controlled language (CL). This process is a mapping from a base language to another different language. The analyst should recognize and understand the NL-based symbols belonging to the domain and convert them into symbols defined in a lexicon from a CL. Thus, in RE it is common using, as a basis of the process, such kinds of language: Natural Language and Controlled Language. These are explained as follows.

Natural Language

Natural language is the language normally used by a community for transmitting or expressing something (*e.g.* information, emotions, and requests). Any language is a system intentionally used by humans for communicating and reasoning, and comprises signs which determine meanings (Castro *et al.*, 2010). When working with NL the analyst should include issues like lexical ambiguity—related to alternative parts of speech and word classes—and syntactic ambiguity, due to the complexity of the sentence structure.

According to Berry (2003), the vast majority of requirements are written in NL. The analyst needs to identify the concepts and relationships among concepts used by the stakeholders. Such concepts and relationships are the basis of the common language understood by the analyst and the stakeholder. In general sense, as suggested by Li *et al.* (2003), NL is highly informal and requires too much analyst intervention for analyzing and designing a software solution. Since NL is inherently ambiguous, and the interpretation of the NL-based text can be affected by several factors (e.g. geographical, psychological and sociological; Liu *et al.* (2014), NL requires too much human involvement for reaching an expected level of precision in the analysis.

Controlled Language

Controlled Language is a subset of NL which is restricted in a way that allows for automated translation into formal logic (Kuhn, 2010). Commonly, CLs have been proposed by humans for expressing their knowledge in a natural way and in turn, it can be understood by computers. CL is used to improve the clarity of expression in the source text and the quality of the analysis phase (Mitamura & Nyberg, 1995), in requirements engineering and other similar knowledge areas.

© Bell Manrique-Losada, 2014. All rights reserved.

The definition of a CL from a selected source language is important to establish a controlled vocabulary and a controlled grammar. The aim is restricting the translation as a way to only use a pre-defined vocabulary. With this vocabulary, we can write and standardize rule-based readable texts. According to Mitamura and Nyberg (1995), if the grammatical constraints of the source language are formally specified, then a machine translation system may take advantage of less-complex and less-ambiguous texts.

2.1.3 RE Techniques

Typically, RE is a process for collecting information and then generating articulated requirements. In fact, requirements are embedded into a big amount of information and analysts have too much work for transforming such information into explicit knowledge about requirements (Stein *et al.*, 2009).

Due to the complexity of the RE process and the problems associated (Christel & Kang, 1992), a lot of expert knowledge from several fields is needed for developing a software application (Kleppe, 2009). A simple holistic vision of the whole problem is required for supporting decision making inside the RE process. Some sources of candidate requirements seem to be a good basis for the holistic vision (Stein *et al.*, 2009; Sommerville *et al.*, 1998) in order to identify the domain knowledge (Zhang, 2007). Even if the RE process takes place in a well-known and controlled environment, some features of the process (*e.g.,* risk and uncertainty related to requirements contextualization, quality volatility of them, and ambiguity in their definition, among others) require the usage of either multiple sources or several RE techniques for properly supporting the process. The RE techniques facilitate the understanding of the users and stakeholders requirements, as well as the needs of actual users of systems (Browne & Ramesh, 2002; Hickey & Davis, 2004).

Many possible knowledge sources can be used for the RE process (Kotonya & Sommerville, 2004). Some of them are related to human beings—*e.g.* discourses of domain experts, current users, or stakeholders—and others are linked to artifacts—*e.g.* reviewing of legacy systems or document analysis.

Existing RE techniques are grouped into several categories, methods, or methodologies, based on the kind of project, the discipline in which requirements will be implemented, and the communication method involved. In this Ph.D. Thesis we recognize the categories defined by Zhang (2007), as follows: conversational methods (*e.g.* interview), observational methods (*e.g.* ethnography), analytical methods (*e.g.* background reading), and synthetic methods (*e.g.* prototyping). Byrd *et al.* (1992) provide similar categories, but they include knowledge acquisition techniques used in knowledge engineering. As we present in Figure 2.3, each Zhang's category offers a specific interaction model between analysts and stakeholders. Thus, an analyst should identify, analyze, and understand the techniques and methods by category for defining which of them will appropriately guide them through the RE process.

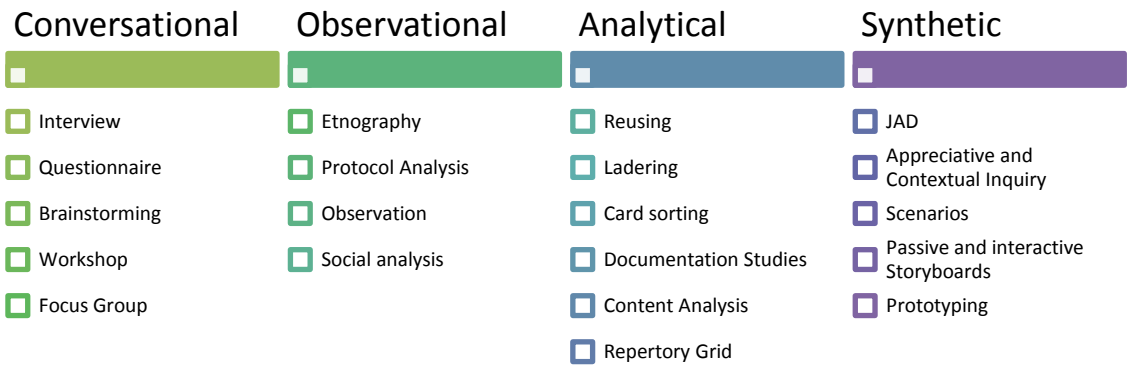


Figure 2.3 Categories for RE Methods.

Adapted from Sajjad & Qaisar (2010)

© Bell Manrique-Losada, 2014. All rights reserved.

Conversational Methods

These methods are based on a verbal communication between stakeholders and analysts (Isam *et al.* 2013). For this reason, they are known as verbal methods (Avison & Fitzgerald, 1995). The techniques grouped in this category are ones of the most commonly used, since they depend on the natural way of communication and facilitate the identification of needs and ideas. The conversational methods are used for understanding the problems and eliciting common product requirements (Avison & Fitzgerald, 1995).

The following techniques are included in this category: Interview, Questionnaire, Brainstorming, Workshop, Focus Group.

Observational Methods

The observational methods are oriented to the observation of human activities (Zhang, 2007), for understanding the application domain and specifying in a common way to the analysts. The techniques inside this category are not directly related to the verbal communication, since such communication is frequently weak for gathering tacit requirements. Therefore, an analyst can observe the stakeholders performing their work and thus, facilitates information gathering and improves the comprehension of the different perspectives of the context where the system-to-be should be used (Viller & Sommerville, 2000).

The most common used techniques in this category are as follows: social analysis, ethnographic study, protocol analysis, and observation.

Analytical Methods

In analytical methods, the domain information, knowledge, needs, and expectations are gathered by exploring the existing documents. According to Rugg *et al.* (2002), based on such exploration an analyst understands and captures the information about the problem and application domain, the work-flow, the product features. Such information is mapped into a specification.

Requirement reuse, Content Analysis, Documentation Studies, Analysis Laddering, Repertory Grid, and Card Sorting are approaches of analytical methods.

Synthetic Methods

The synthetic methods form a coherent whole by combining the above methods into a single one (Sajjad & Qaisar, 2010). The combination helps the analyst to expand the coverage of the elicitation and gain a generic knowledge of the application domain.

Synthetic methods are known as collaborative methods due to the collaboration among multiple RE methods and the different ways for exploring the requirements between stakeholders and analysts.

According to Cook (2002), Joint Application Development (JAD), Scenarios, Passive and interactive Storyboards, Appreciative and contextual Inquiry, and Prototyping are common approaches of synthetic method.

2.2 Domain Knowledge Acquisition and Representation

RE and other domains need the process of gathering information to be analyzed, organized, and modeled, in order to create a meaningful whole: a domain representation—commonly in terms of a conceptual model. The definition of a concept, derived from the information acquired, can apply for clarifying the problem solution. The previous process is based on the knowledge acquisition and analysis. Consequently, the first task in creating a knowledge representation in a specific domain is the conceptualization, defined as the usage of concepts and relationships to deal with—and probably solve—a problem (Andrade *et al.*, 2003). Accordingly, a conceptual model is an abstraction of the universe of discourse, as well as a kind of knowledge representation and a possible model of a possible solution for the problem.

© Bell Manrique-Losada, 2014. All rights reserved.

2.2.1 Knowledge Acquisition (KA)

KA is the process of achieving knowledge, from mostly a human expert or a group of experts (Kendal & Green, 2006) from books, documents, sensors, or computer files (Turban *et al.*, 2005). Such knowledge is specific to the problem domain or the solution domain; it can be general when is related to the organization/business. Byrd (1995) formally recognized the relevance of knowledge acquisition in the development; thus, much theoretical and applied research is still being conducted in this area.

KA is not a simple task, because it includes tasks for identifying, representing, structuring, and transferring knowledge to a system/machine. In this way, inside the KA process other relevant processes related to modeling and knowledge representation are performed.

Knowledge Modeling

The *knowledge modeling* process generates structures providing a framework for KA and a decomposition of the overall acquisition task into more atomic tasks. Identified parts of knowledge models—domain models or problem-solving methods—can serve in different systems. According to Turban *et al.* (2005), the knowledge modeling reflects the knowledge content of a system and makes explicit the structures where it operates the knowledge for solving several kinds of problems. Morik (1990) defines the knowledge acquisition as a transferring of knowledge from an expert source into an appropriate computer representation.

Knowledge Representation

Knowledge representation is a discipline from Artificial Intelligence oriented to the representation of human knowledge—in a specific form—facilitating a kind of automatic reasoning, *i.e.* the inference of new knowledge out of existing one. Several methods are intended to organize the acquired knowledge as a knowledge base. According to Al–Khanak (2012), some of the common used representation methods are based on logic, semantic arrays networks, multiple knowledge representation, and uncertainty representation.

Knowledge representation requires that all the information elements be identified from the analysis, and classified or grouped into levels, depending on what function they should fulfill in the specific domain. According to Andrade *et al.*, (2003), such levels are: Static information, which comprises the structural or declarative domain information about the problem and which can be used in operations—*i.e.,* concepts, properties, relationships, and constraints—and Dynamic information, needed to conform the behavior that takes place in the domain—functionality, action, etc.

According to Sowa (2000), knowledge representation has constructs in other related fields: *logic* for defining the formal structure and the inference laws, *ontology* for naming and defining the core things of the knowledge, and *computation* for implementing the computer applications out of it. Several formalisms have been proposed for structuring and representing knowledge. Some of them are as follows:

- Production rules (Helbig, 2006; Turban *et al.*, 2005)
- Frames (Marvin, 1974; Fensel *et al.,* 1998; Jackson, 1999)
- Semantic networks (Woods, 1975; Russell & Norvig, 2002)
- Ontologies (Lenat, 1995; Guarino, 1998; Dittmann *et al.*, 2004)
- Description Logics (Puls, 2002; Tsarkov & Horrocks, 2006Sirin *et al.*, 2007)
- Decision trees and tables (Vadera, 2005)
- Object-oriented representation (Devedzic *et al.*, 1996; Bogarin & Ebecken, 1996)

2.2.2 Role and challenges of KA in the RE process.

One of the crucial features of RE is the process for acquiring knowledge related to organizational domain by the project team members (Loucopoulos & Karakostas, 1995). Since knowledge can be often acquired from other sources linked to—or instead of—human experts, then KA have a relevant role in the RE

© Bell Manrique-Losada, 2014. All rights reserved.

process. This approach has the major advantage of eliminating the need to use exclusively experts for identifying, acquiring, and representing domain knowledge. Turban *et al.* (2005) describe how this approach is used in knowledge-based systems where the main concern is handling a big amount of complex information rather than world-class expertise—e.g. searching through corporate policy manuals or catalogs is an example.

Otherwise, and according to Cremers and Alda (2005), the RE process can be viewed as a method where the first three stages are directly involved with KA, as we show in the Figure 2.4.

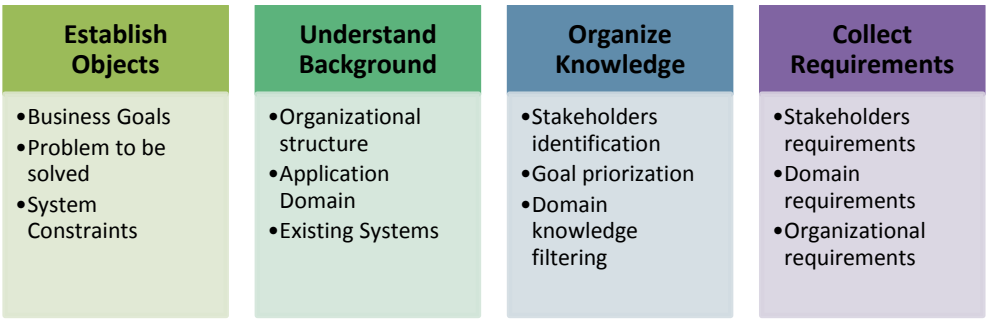


Figure 2.4 Role of KA in the RE process.

Adapted from Cremers and Alda (2005)

The first activities are focused on establishing the overall organizational objectives, identifying the reasons for developing a system, and defining a background of knowledge acquisition. Based on such preliminary business domain and knowledge, we can first gather and understand the context, and then organize and analyze such background in terms of conceptual models and requirements.

2.3 Natural Language Processing

Natural Language Processing (NLP) is a discipline for developing computer programs capable of human-like activities related to understanding, producing texts, or speech in a natural language (Gelbukh, 2010).

The NLP discipline comprehends a wide set of techniques for generating, manipulating, and analyzing natural or human languages in an automated mode. Most of the NLP techniques come from Linguistics and Artificial Intelligence, but some of them are also influenced by other newer areas such as Machine Learning, Computational Statistics, and Cognitive Sciences. In addition, NLP requires various kinds of knowledge of language, commonly grouped in linguistics analysis levels (Jurafsky & Martin, 2008) or linguistic description levels (Wilcock, 2009). Aitchison (2010) graphically represents such levels and their relationships as we show in Figure 2.5, basically in the following levels:

- Morphology —meaningful components of words
- Syntax —structural relationships between words
- Semantics —meaning of individual words and their combination
- Pragmatics —relationship of meaning to the goals and intentions of the speaker
- Discourse —linguistic units larger than a single utterance

NLP is based on the science called Computational Linguistics, which is conceived from applied linguistics and NLP in turn from linguistics. Computational Linguistics is considered as the automatic processing of NL and by its nature is interested in formal description of language relevant for automated processing. In the following sections we describe kinds of linguistic analysis which can be performed from written texts in NLP. Also, we present a conceptualization about discourse analysis, mainly the discipline of Rhetorical Analysis and Corpus Linguistics.

© Bell Manrique-Losada, 2014. All rights reserved.

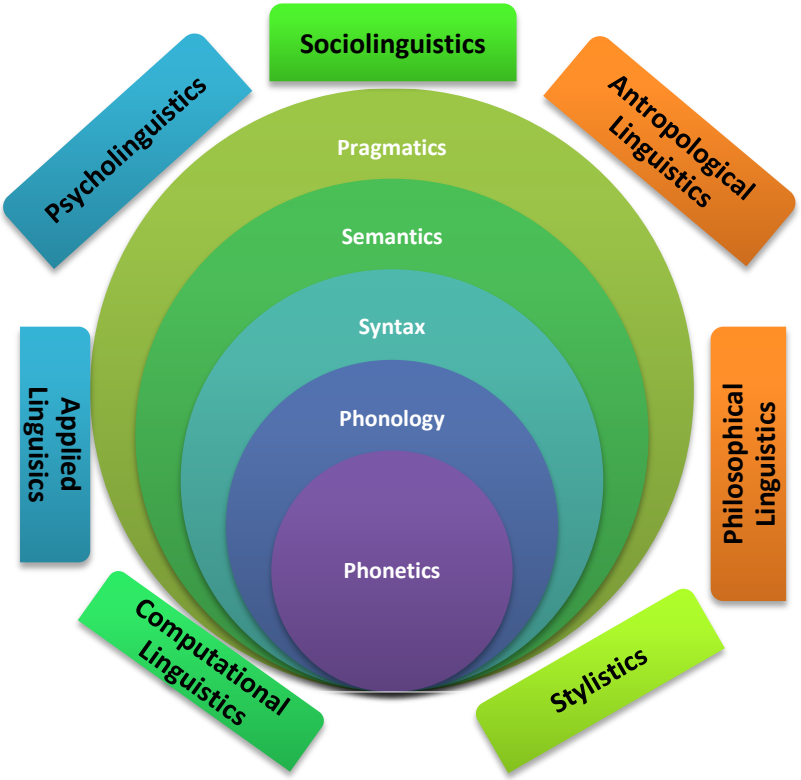


Figure 2.5 Linguistics categories.

Adapted from Aitchison (2010)

2.3.1 Linguistic Analysis

Linguistics attempts to reconstruct the organization of NLs in terms of its structural notions by following analysis levels. According to Trandabat (2010), an extended linguistics classification basically distinguishes the study of the language structure—*Syntax*—and the study of the language meaning—*Semantics*. Some other levels considered are *Pragmatics* oriented to define the informational status of linguistic expressions, as used in given contexts; and *Morphology* for describing the formation and inflection of individual words.

Thus, the language processing of a text involves several levels of linguistics analysis or linguistic annotation ranging from word recognition to lexical analysis, syntactic and semantic processing. Such analysis includes many of the following tasks: identifying words in the string of symbols (text tokenization), splitting the text into meaningful units (text segmentation), assigning part-of-speech tags by word (part-of-speech tagging), assigning the syntactic role by word and the syntactic relationships between them (parsing), and assigning semantic roles (semantic role labelling), among others. The most relevant tasks for the purposes of this Ph.D. Thesis are presented in the following sections.

Lexical Analysis

Usually the term 'lexical analysis' refers to processing an input for generating a list of tokens as an output. The purpose of lexical analysis is determining the denotations of specific words by looking for them in a lexicon. Also, in this analysis we should identify symbols, such as punctuation marks. The lexical analysis is commonly developed by the tokenization and sentence boundaries processes.

Sentence Boundaries process deals with the spelling level in the annotation tasks, which split the text into several words and sentences. Usually, such tasks are performed before the higher-level tasks are done. According to Wilcock (2009) the sentence boundary process occurs as follows: a text is splitted into sentences, by means of a sentence boundary detection method, in order to define the boundaries by sentence. Such a process is based on abbreviations (*e.g.,* "Dr." and "etc.") and end-of-sentence markers as fullstops (*e.g.,* "."), question marks (*e.g.,* "?"), or exclamations marks (*e.g.,* "!").

Meanwhile, a text is divided into words in the tokenization process. A tokenizer identifies words according to *types* of words and their occurrences as word *tokens*. The boundaries of a token are points in the text

© Bell Manrique-Losada, 2014. All rights reserved.

where the token starts and points where it ends. Commonly, the previous tasks are the two first stages in text annotation, since their results are used by later stages as the input units.

Part-of-Speech Analysis

Each word has a tag for describing a part of speech (POS) category. POS tags of a word can belong to open classes (*e.g.* noun, verb, adjective, and adverb) or closed classes (*e.g.* prepositions, determiners, conjunctions, pronouns, and open class). A POS tagging looks for the surrounding words of a specific word and to find POS tags for it.

Commonly, the POS tags for English are (Nugues, 2014; Patwardhan *et al.* 2012): Noun (N), Verb (V), Adverb (Adv), Adjective (A or Adj), Preposition (P), Determiner or Article (Det), Conjuction (Conj), and Interjection (Int). In turn, such POS tags are divided into subcategories for each one (*e.g.* Nouns are subdivided into proper nouns, common nouns, and pronouns). Also, we can distinguish different categories by using features for each tag, for example Verbs (V) can be labeled as V-past (*e.g.* in verbs like *assisted* or *gave*) or V-present (*e.g.* in verbs as *assist* or *give*). The fixed sets of tags are known as *tagsets* and are implemented with important distinctions and multiple feature combinations in different taggers (Nugues, 2014). The POS tags used in the rest of the Thesis is presented in Appendix 2.1.

Parsing—Syntactic Analysis

Parsing is the extraction of syntactic structures from the text. A set of word sequences are considered a grammar by some language. As defined by Carroll (2003), parsing is a process involving the usage of a grammar to assign a—more or less detailed—syntactic analysis to a set of words. Such a process comprises a mapping from a natural language sentence into a logical meaning representation which is domain-specific and directly interpretable by a computer (Zettlemoyer & Collins, 2005).

Parsing methods and algorithms (Sag *et al.*, 2003) usually have the following key components: a grammar—as a declarative representation for describing the syntactic structure of sentences—and a parser—as an algorithm for analyzing the input and outputs and generating a structural representation consistent with the grammar specification. A parsing method depends on the grammatical theory being used (*e.g.* dependency grammars or phrase structure grammars). Commonly, we can find limited form of parsing—different to the full syntactic parsing—for only checking noun phrases or specific phrase; such parsing is called *chunking* or *partial parsing*.

Morphological Analysis

The goal is analyzing words into their linguistic components or morphemes. The morphemes are the smallest meaningful units of the language. For example the linguistic components—morphemes—of 'rooms' are room+PLURAL. Based on the analysis of the kind of word, we can find ambiguities and generate more than one alternative linguistic component. This analysis is relatively simple in languages as English, but it is increasingly difficulty in languages like Arabic or Turkish.

According to Antworth (1994), morphological analysis can be described by means of morphological rules for splitting each word as it occurs in a text into morphemes—task of a morphological parser. Such parsing operates on single words, and retrieves the word root and its affixes from its inflected or derived form in a text. The morphological analysis in some applications is based on segmentation of each word into its different components morphemes (Aduriz *et al.,* 2000); however, some other applications such as lemmatization, tagging, phrase recognition, and determination of clause boundaries (Aduriz *et al.* 1995) refer the hierarchical structure at word level, combining morphology and syntax of the whole word.

Semantic Analysis

The semantic analysis facilitates the mapping of meanings to the structures created by the syntactic analysis. Such mapping is based on words and structures into particular domain objects. This analysis level corresponds to the higher linguistic level of semantics. Semantics is particularly focused on predicate-argument analysis, discourse connectives, and co-reference. According to Palmer *et al.* (2005), several semantic theories in linguistics give the foundation for semantic representations on different analysis levels. In the following Sections, we present the kinds of analysis we apply in this Ph.D. Thesis.

© Bell Manrique-Losada, 2014. All rights reserved.

Phrasal Expression Analysis.

A user of a language has available a large number of pre-constructed phrases conforming single choices, even though they might appear to be analyzable into segments (Sinclair, 1991). Such phrases are known as phrasal expressions (PE) or lexical phrases (LP) and may have a pragmatic function. According to Pérez (1999), the importance of PEs lies in their usage and domain, which constitute an integral part of the communicative competence.

PEs are expressions useful for writing idiomatic expressions, phrasal verbs, and collocations. According to Sugino (2008), the phrasal expressions, are categorized by sections of a text—in this research called moves from a technical document—which they are frequently used (*e.g.* expressions appearing in moves such as "introduction" or "closure"). This categorization helps an analyst or a system to find and use appropriate phrasal expressions efficiently.

For the purpose of this Ph.D. Thesis, we use the PE classification proposed by Baldwin *et al.* (2010). This and other classifications have been mostly used in NLP techniques for text-mining and information extraction. They have been also applied to the analysis of many kinds of documents, *e.g.*, technical documents, patents, and software requirement documents, as follows. Cascini *et al.* (2004) present a functional analysis of patents and their implementation in the PAT-Analyzer tool. They use techniques based on the extraction of the interactions from the entities described in the document and expressed as subject-action-object triples, by using a suitable syntactic parser. Rösner *et al.* (1997) generate multilingual documents from knowledge bases by using automated techniques. The resulting documents can be represented in an interchangeable and reusable way. For knowledge acquisition, several authors have applied NLP techniques for handling PEs: Jackendoff (1997) and Aussenac-Gilles *et al.* (2000) extract knowledge from existing documents and demonstrate its usage on the ontological engineering research domain.

Lexical-Semantic Analysis based on WordNet.

WordNet[2] is a lexical semantic resource which defines word senses by using methods for grouping senses of the same word and thus producing coarser word sense groupings (Fellbaum,1998). English words are grouped into sets of synonyms—called *synsets*—comprising a list of synonymous words or collocations for describing the relationships among *synsets*.

WordNet *synsets* are organized in 45 Lexicographer Files—called SuperSenses—based on syntactic categories—nouns, verbs, adjectives, and adverbs—and logical groupings—*e.g.* person, phenomenon, and location. Besides, WordNet have 26 basic categories for nouns, 15 for verbs, 3 for adjectives, and 1 for adverbs. For the purposes of this Ph.D. Thesis, we consider the syntactic categories for verbs, as we present in the following paragraphs.

Syntactic categories for Verbs. Verbs form language-specific structure in the WordNet ontology comprising 63 concepts and distinguishing 3 types of entities at first level. Verbs are included in the 2nd Order Entity. According to Vossen (2002), 2nd Order refers to any situation—being static or dynamic—which cannot be grasped, heard, seen, or felt as an independent physical thing. These situations can occur or take place in a time or place/space, rather than exist (*e.g.* happen, cause, occur, apply, etc.). Also, they are related to: i) verbs or events denoting nouns, and ii) events, processes, states-of-affairs, or situations located in time.

Verbs in the 2nd Order category can be further subrogated according to the physical entities involved in the following subcategories, which we present in detail in Appendix 2.2:

- *Process*. This category implies all physical entities, *i.e.* those with a location in space-time. Entities related to objects and processes are involved in it. These verbs are mostly related to processes since they are things that 'happen' and have 'temporal parts/stages'. A process can be considered as a set of denotations related to dual object process, intentional process, motion, internal change, shape, or change.

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

© Bell Manrique-Losada, 2014. All rights reserved.

- *Situation Type*. Refers to a situation—event or set of events, featured as a conceptual unit—happening over time. This kind of verbs is represented in terms of the event-structure or the predicate properties.

- *Conceptual Domain*. On the other hand, the multilingual database with WordNet for multiple language known as EuroWordNet contains 200 domain labels organized in a hierarchical structure grouping the words in categories based on a domain hierarchy. Semantic domains are knowledge areas—*e.g.* economy or politics—used to describe texts according to general subjects characterized by domain specific lexica. The domain hierarchy is represented as an ontology which comprises conceptual levels for each language. The levels of the domain hierarchy are called *basic domains*.

2.3.2 Text Analytics

According to Wilcock (2009), Text Analytics (TA) refers to a subfield of information technology dealing with applications, systems, and services for doing some kind of text analysis as a way to extract information from them. This field has received much attention due to its wide application as a multi-purpose tool, borrowing techniques from NLP. Several techniques for TA have been developed, among them: named entity recognition, co-reference resolution, information extraction, chunking, semantic role labeling, text mining, and semantic search. Such techniques are generally based on finding effective ways to exploit information extraction results, taking advantage of techniques originated in computational linguistics, statistics, and other computer science disciplines.

For the purposes of this Ph.D. Thesis, we consider useful techniques and tools for developing text analytics applications for identifying business-based information from technical documents. Some of them are described in the following paragraphs.

Information Extraction. It includes techniques for extracting any kind of information from texts. Relation extraction techniques require the identification of significant entities and relationships between entities and significant properties of entities (Grimes, 2008). The goal of IE is storing the extracted entities and relationships in a database—structured information. The information in the text is unstructured because we can express the same relation in different ways to refer the same entity, so ambiguity can arise.

Text Mining. According to Witten and Frank (2005), text mining usually refers to applications for performing information extraction from text, storing the results in a database, and then applying data mining techniques to the information in the database. The final goal of text mining is discovering knowledge from text files (Stavrianou *et al.*, 2007).

Categorization. It involves identifying the main document subjects by assigning the document into a pre-defined set of topics. When categorization occurs, a computer program should treat the document, as told by Gupta and Lehal (2009), as a "bag of words." Categorization only counts words that appear and, from the counts, identifies the main topics covering the document. Categorization tools commonly have a method for ranking the documents and looking for documents containing more on a particular topic.

Concept Linkage. According to Fan *et al.* (2005), the goal of concept linkage is connecting related documents by identifying their commonly-shared concepts. Based on such concepts, users can identify information difficult to find for them by using traditional searching methods. Browsing information is promoted rather than searching for it.

Prototypical Documents Extraction. It relies on the identification of frequent sequences of terms in the documents, and uses NLP techniques, such as POS tagging and term extraction for pre-processing the textual data (Rajman & Besancon, 1997). Such a technique can be considered as an automated, generalized indexing procedure for extracting linguistically significant structures from documents.

Text analytics can represent flexible approaches to information management, research and analysis. The major text analytics tasks are: duplicate detection, summarization, categorization, clustering, and parsing.

© Bell Manrique-Losada, 2014. All rights reserved.

2.4 Discourse Analysis and Corpus Linguistics

The linguistic analysis of discourses from documents can be performed by using methods of discourse analysis and corpus linguistics. In this Ph.D. Thesis we are focused on the following perspectives, based on the usage of language for constructing, interpreting, and exploiting technical documents.

2.4.1 Discourse Analysis

In order to clearly describe the disciplines underlying this approach, we first explain what discourse is. Two possible ways of regarding discourse (Fairclough, 1995) are: i) discourse as a social action and interaction—people interacting together in real social situations—; ii) discourse as a social construction of reality, a form of knowledge. The first way is more prominent in language studies since the focus is the language as it is used, but the second one is more concerned with people is knowledge in a historical context (Gillespie & Toynbee, 2006). Fairclough (1995) argues in favor of combining the two approaches for analyzing the language in use regarding social and cultural processes.

Language analysis in social use is studied in several areas, and *discourse analysis* is one of them. Commonly, the relationships between *form* and *function* in language (Gee, 2004) are the focus of the approaches. Discourse analysis is interpretive and explanatory and uses a systematic methodology. The study of discourse is a vast field comprising many sub-disciplines which at the same time overlap with other new one, such as sociolinguistics and semiotics, as we describe in the next subsections.

Specialized Discourse Analysis based on Genre Studies

Specialized text involves a special process in the framework of discourse analysis (Biber, 2006; Nickerson 1999). One approach for discourse analysis comes from the genre point of view. According to Swales (1990; 2004) and Yates (1989) genres are defined as variations of a language which operate by means of linguistic features present in a text. Likewise, they are linguistically confined under their communicative purposes, participants involved, production contexts, usage contexts, and discourse organization modes, among others (Parodi, 2008).

The genre theory (Nickerson, 1999; van Nus, 1999) focuses on written practices of members from specific communities and also on the design of information and business records. Specialized texts—generated from a specialized organizational discourse—are produced by specialists who have mastered the cognitive and conceptual organization of a matter. According to Cabré (1999), the specialized discourse is derived from variables related to the subject and perspective of a topic, and the intent and level of expertise of the producer. According to this genre theory, we can analyze specialized discourses, because they are written by members of specific communities. Following Cabré (1999), a discourse is derived from variables related to the subject and perspective of a topic, whose analysis is important.

Genres are described in terms of texts structures and their contexts (van Dijk, 2008). Genre analysis is an attempt to relate the text structure to the macro-social context, in particular the group of competent potential text users. Commonly, genre analysis is characterized by recognizable purposes and schematic structures, which can be as numerous as the social practices which people get involved in. Genre studies have been developed from the seminal work by Bakhtin (1986), the new rhetoric by Swales (1981; 1990), and the research within Systemic functional linguistics by Christie (1999) and Eggins (1994).

In the field of language for specific purposes and according to Hyon (1996), genres are considered as oral or written text types defined by formal properties and communicative purposes within social contexts. In this Ph.D. Thesis we take such field as foundation, since we are focused on *structural move analysis* and *steps* as developed by Swales (1990). The *structural move analysis* describes texts in function of their formal features either concerning global organization or sentence-level features. Discourse reflects the rhetorical experience of language users as they create and interpret texts, as Hyon (1996) states, by offering dense descriptions of academic and professional contexts and the text actions performing within these situations.

According to the previous foundations and the approach of Meurer (2002), we follow the genre analysis based on the definition of genres as reasonably stable types of text formal or informal—written in our work—, which can be recognized due to their rhetorical structure and function, *i.e.*, their organization

© Bell Manrique-Losada, 2014. All rights reserved.

and purpose. Relevant features of genre include the rhetorical structure and argumentative structure of the text (Fairclough, 2003). The rhetorical structure is studied in the Rhetorical Analysis field, as we will explain in the following Section.

Rhetorical Analysis

Rhetorical analysis (RA) is concerned with the discourse construction, giving priority to the communicative intent of each genre (Azaustre & Casas 1997). Rhetoric aims at discourse from its intentional—purpose-driven—and instrumental—designed to fulfil such purpose—point of view. Thus, the rhetorical organization of discourse is the way in which textual structures are employed in order to achieve a desired effect (Connor, 1996). The structure constitutes a frame where diverse discourses are articulated in a particular way and their relations are textually constituted.

From RA, the genre analysis is discussed in terms of rhetorical moves, which refer to the functional parts or sections of a genre. This approach for studying a particular genre comprises the analysis of a text and its description in terms of rhetorical structure (moves). This structure influences and restricts the contents and style (Askehave & Swales, 2001) and allows for the identification of linguistics features.

The particular configuration of the text surface is defined in terms of levels of text organization, also known as discourse rhetorical organization. The rhetorical organization considers the interaction of structures and functions at three distinct organization levels.

On a higher level of textual organization macro-units are formed by the combination of micro-units possessing coherence. As van Dijk (1980) defines, the transition of units is modeled by macro-rules, leading to macro-propositions and by the rhetorical relations identified in rhetorical structure theory (Mann *et al.*, 1992). On a most basic level of textual organization—micro-level—, a text can be seen as a set of grammatically defined units, referred as micro-units. Micro-units have a propositional sense, hence their potential for forming a coherent structure, displaying a connection of senses.

In this way, the macro units identified by genre analysis can be characterized as *moves* and *steps*, referring to the passages of the text which are larger than the largest grammatical units (*e.g.* clauses, clause complexes, sentences) and possess some unity grounded in a common function/meaning. In summary and according to Heuboeck (2009), moves are situated at a macro-level since they are represented as a sequence of micro-units and they display functional unity at that level.

Corpus Linguistics

The definition of what is generally meant by corpus and an overview of its particular aspects are included in this Section, regarding our specific approach.

Corpus Linguistics is a subfield of Linguisics[3]. According to the definition of the OLAC project[4], Corpus Linguistics is defined as the study of the linguistic properties of an extended passage, text, or corpus of texts. Such a definition includes Semiotics and Genre Analysis perspectives, as well as the computational analysis of text corpora, as we presented in the previous Sections.

According to Parodi (2008), corpus linguistics comprises a set of methodological principles for studying any language domain. Corpus linguistics allows for the description and analysis of several types of discourse by using corpus pre-processed with the assistance of information technologies. Sets of linguistic features—operated by genres—can be identified from representative corpus. In this discipline, a corpus is a comprehensive collection of texts collected as sets of linguistic data reflecting the usage of a language (Martin, 2005). A corpus-based approach is aimed to word usage, frequency, collocation, and concordance (O'Keeffe's, 2003). According to Francis (1992), such approach is the result of an empirical process of linguistic data collection combined with the author's own capacity of elicitation and introspection.

Currently, major electronic corpora are Brown Corpus, London-Lund Corpus, Lancaster-Oslo/Bergen Corpus, and the British National Corpus. The Brown Corpus was the first computer-readable corpus

[3] From the International Linguistics Community Online, in: http://linguistlist.org/LL/LingSubfields.cfm
[4] http://www.language-archives.org/REC/field.html

© Bell Manrique-Losada, 2014. All rights reserved.

including one million words of American English collected by Brown University (Kennedy, 2000), similarly to Lancaster-Oslo/Bergen Corpus with one million British English words. Currently, corpus-based approaches have been widely applied to many fields such as discourse analysis, language teaching, and stylistic analysis (Kennedy, 2000). Biber *et al.* (1998) summarize features of corpus-based studies as an empirical approach, on the basis of a sample of natural texts, computer-assisted analysis and combination of quantitative and qualitative analysis. A study made by Kennedy (2000) recognizes approaches on corpus mainly based on four aspects as lexical level, syntactic level, discourse analysis, and genre analysis.

The benefits of using corpora are related to corpus markup and annotation. Many available corpora are encoded with textual—*e.g.* register, genre and domain—and sociolinguistic—*e.g.* user gender and age—metadata to give more accurate descriptions of the usage of a lexical item. Because of the greater availability of electronic texts, studying specific purpose corpora is now made easier and more effective. Automated tools and software are available to users for processing texts (Barlow, 1996). Working with a specialized language corpus constitutes, following Sinclair (1991), a strict selection of texts for identifying common patterns. From a general sample of texts, many words and phrases are rare, but in a strict selection very frequent indeed in certain specific texts. Thus, a corpus should be representative of one or more aspects of a language. The notion of representativeness is the foremost central issue to address before building a corpus, because a corpus represents a language or some part of a language (Curado, 2000). The appropriate design of a corpus therefore depends on what it is meant to represent.

REFERENCES

A. B. Cremers and S. Alda (2005). Organizational Requirements Engineering. Chapter 5 Requirements Elicitation I. Bonn University. Available in: http://www.iai.unibonn.de/III/lehre/vorlesungen/SWT/RE05/slides/05_Requirements%20Elicitation%201-2.pdf

A. Curado (2000). A Lexical Common core in English for Information Science and Technology. PH.D. Thesis, Facultad de Politécnica, Universidad de Extremadura.

A. Gangopadhyay (2001). Conceptual Modeling from Natural Language Functional Specifications. Artificial Intelligence in Engineering, Vol. 15, Issue 2, pp. 207-218.

A. Gelbukh (Ed.) (2010). Natural Language Processing and its Applications. Special issue. Research in Computing Science, Vol. 46, pp. 3-15.

A. Heuboeck (2009). Some Aspects of Coherence, Genre and Rhetorical Structure – and their Integration in a Generic Model of Text. In: Language Studies Working Papers: O'Brien & Giannoni, Editors. Vol. 1, pp.35-45.

A. Kleppe (2009). Software Language Engineering: Creating Domain-Specific Languages Using Metamodels. 1st Ed., Addison-Wesley Professional, Pearson Education.

A. Lapouchnian (2005). Goal-Oriented Requirements Engineering: An Overview of the Current Research. Department of Computer Science, University Of Toronto. Available in: http://www.cs.utoronto.ca/~alexei/pub/Lapouchnian-Depth.pdf

A. M. Hickey and A. M. Davis (2004). A unified model of requirements elicitation. Journal Management of Information Systems 20(4):65.

A. Moschitti (2012). Natural Language Processing and Information Retrieval. Department of information and communication technology, University of Trento. Available in: http://disi.unitn.it/moschitti/Teaching-slides/NLP-IR/Lecture12-NLP-IR.pdf

A. O'Keeffe's (2003). Strangers on the Line: A Corpus-based Lexicogrammatical Analysis of Radio Phone-in. PhD Thesis, University of Limerick.

A. Stavrianou, P. Andritsos and N. Nicoloyannis (2007). Overview and semantic issues of text mining. ACM SIGMOD Record, Vol. 36 (3), pp.23–34.

A. van Lamsweerde (2000). Requirements Engineering in the Year 00: A Research Perspective. 22nd International Conference on Software Engineering (ICSE'2000), Limerick, Ireland.

Antworth (1994). Morphological Parsing with a Unification-Based Word Grammar. North Texas Natural Language Processing Workshop, Vol.1, pp.24-32.

Azaustre and J. Casas (1997). Manual de retórica española. Barcelona: Ariel.

B. Nuseibeh and S. Easterbrook (2000). Requirements Engineering: A Roadmap. Proceedings Conference on the Future of Software Engineering, Limerick, Ireland.

B. Thalheim, K. Schewe and H. Ma (2009). Conceptual Application Domain Modelling. Proceedings Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009), Victoria University. Conferences in Research and Practice in Information Technology, Vol. 96. Markus Kirchberg, Sebastian Link, Ed.

C. Fellbaum, editor. 1998. WordNet. An Electronic Lexical Database. The MIT Press.

C. Nickerson (1999). The usefulness of genre theory in the investigation of organisational communication across cultures. Document Design, Vol.1, No. 3, pp. 203–215.

C. Perez, 1999. La enseñanza del vocabulario desde una perspectiva lingufstica y pedagogica. In S. Salaberri (Ed.), Lingutstica Aplicada a las Lenguas Extranjeras, Almería: Univ. de Almería: 262-307.

© Bell Manrique-Losada, 2014. All rights reserved.

C. Shekaran, D.Garlan, M. Jackson, N.R. Mead, C. Potts and H. B. Reubenstein (1994). The role of software architecture in requirements engineering. Proceedings of the First International Conference on Requirements Engineering, Vol.1, pp. 239-245.
C.M. Zapata and B. Manrique (2012). Transforming Natural Language into Controlled Language for Requirements Elicitation: A Knowledge Representation Approach," in Knowledge Representation, Croacia: Intech Ed.
D. A. Cook (2002). Verification & Validation of Requirements. Software Technology Support Center (STSC). Available in: http://www.sstc-online.org/Proceedings/2002/SpkrPDFS/ThrTracs/p961.pdf
D. B. Lenat (1982). The Ubiquity of Discovery. Artificial Intelligence, Vol. 19, No. 2.
D. B. Lenat (1995). CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11):33-38.
D. Biber, S. Conrad and R. Reppen (1998). Corpus linguistics: Investigating language structure and use. Cambridge University Press, Cambridge.
D. Bieber (2006). University Language: A Corpus-based Study of Spoken and Written Registers. Amsterdam: John Benjamins.
D. E. Avison and G. Fitzgerald (eds.) (1995). Information Systems Development: Methodologies, Techniques and Tools. McGraw-Hill Book Company.
D. Fensel, J. Angele and R. Struder. (1998). The Knowledge Acquisition and Representation Language KARL. IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 4.
D. Jurafsky and J. Martin (2008). Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second Edition. Pearson & Prentice Hall Editorial.
D. M. Berry (2003). Natural language and requirements engineering - Nu?, In: CSD & SE Program University of Waterloo, 20-05-2011, Available from: http://www.ifi.unizh.ch/groups/req/IWRE/papers&presentations/Berry.pdf
D. Rösner, B. Grote, K. Hartmann and B. Höfling (1997). From Natural Language Documents to Shara-ble Product Knowledge: A Knowledge Engineering Approach. Journal of Universal Computer Science. 3(8): 955-987.
D. Ross and K. Schoman (1977). Structured Analysis for Requirements Definition. IEEE Transactions on Software Engineering, 3(1).
D. Trandabat (2010). Natural Language Processing Using Semantic Frames. Ph.D Thesis, University Alexandru Ioan Cuza, Romania.
D. Tsarkov and I. Horrocks (2006). FaCT++ Description Logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, Automated Reasoning | Proceedings of the Third International Joint Conference (IJCAR 2006), Lecture Notes in Artificial Intelligence, Vol. 4130, pp. 292-297.
E. García and A. Monzón (2000). A process framework for requirements analysis and specification. Proceedings 4th International Conference on Requirements Engineering. Vol.1, pp.27-35.
E. Hull, K. Jackson and J. Dick (2010). Requirements Engineering. Springer Science & Business Media (Google eBook), pp. 225.
E. Sirin, B. Parsia, B. Cuenca, A. Kalyanpur and Y. Katz (2007). Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web, 5(2):51-53.
E. Turban, J. Aronson and T. Liang (2005). Knowledge Acquisition, Representation, and Reasoning. Chapter 18. In Decision Support Systems and Intelligent Systems, 7th Edition, Prenhall Editors. Available in: http://wps.prenhall.com/wps/media/objects/3778/3869053/Turban_Online_Chapter_W18.pdf
F. Christie (1999). Genre theory and ESL teaching: A systemic functional perspective. TESOL Quarterly, Vol. 33(4), pp.759-763.
G .Rugg, M. Eva, A. Mahmood, N. Rehman, S. Andrews and S. Davies (2002). Eliciting information about organizational culture via laddering. Information Systems Journal, 12 (3).215-229.
G. Cascini, A. Fantechi and E. Spinicci (2004). Natural Language Processing of Patents and Technical Doc-umentation. Lecture Notes in Computer Science, 3163, pp.508-520.
G. J. Browne and V. Ramesh (2002). Improving information requirements determination: a cognitive perspective. Information & Management. 39(8), 625–645.
G. Kennedy (2000). An introduction to corpus linguistics. Foreign Language Teaching and Research Press.
G. Kotonya and I. Sommerville (2004). Requirements Engineering: Processes and Techniques. Ed. Wiley, Chichester, UK.
G. Kotonya and I. Sommerville (1998). Requirements Engineering - Processes and Techniques. John Wiley & Sons.
G. Neumann and J. Tolujew (2012). Requirements and Solutions for a More Knowledgeable User-Model Dialogue in Applied Simulation. Advances in Intelligent Modelling and Simulation. Studies in Computational Intelligence Vol. 416, pp 1-26
G. Parodi (2008). Lingüística de corpus: una introducción al ámbito. Revista de Lingüística Teórica y Aplicada, Vol.46 (1), pp. 93-119.
G. Wilcock (2009). Introduction of linguistics annotation and texts analytics. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Editors. Toronto, CA.
H. Helbig (2006). Knowledge Representation and the Semantics of Natural Language. New York: Springer.
I. A. Bolshakov and A. Gelbukh (2004). COMPUTATIONAL LINGUISTICS. Models, Resources, Applications. 1st Edition, Instituto Politécnico Nacional, México.
I. Aduriz, E. Agirre, I. Aldezabal, I. Alegria, X. Arregi, J. M. Arriola, X. Artola, K. Gojenola, A. Maritxalar, K. Sarasola and M.Urkia (2000). A word-grammar based morphological analyzer for agglutinative languages. Proceedings of the 18th conference on Computational linguistics, Vol. 1, pp. 1-7. Available in: http://www.aclweb.org/anthology/C00-1001
I. Aduriz, I. Alegria, J.M. Arriola, X. Artola, A. Díaz de Ilarraza, N. Eceiza, K. Gojenola and M. Maritxalar (1995). Different Issues in the Design of a lemmatizer/Tagger for Basque. From Texts to Tags: Issues in Multilingual Language Analysis. ACL SIGDAT Workshop, Dublin.
I. Askehave and J. Swales (2001). Genre identification and communicative purpose: a problem and a possible solution. AppliedLinguistics vol. 22(2), pp. 195-212.

© Bell Manrique-Losada, 2014. All rights reserved.

I. H. Witten and E. Frank (2005). Data Mining: Practial Machine Learning Tools and Techniques. Morgan Kaufmann Ed.
I. Sag, T. Wasow and E. Bender (2003). Syntactic Theory: A Formal Introduction. CSLI Publications, Standford.
I. Sommerville (2005). Integrated Requirements Engineering: A Tutorial. IEEE Software, Vol. 22 Issue 1, pp. 16-23.
I. Sommerville (2011). Software Engineering. University of St Andrews, Scotland 9th Edition, Addison-Wesley. Cloth, 792 pp
I. Sommerville, P. Sawyer and S. Viller (1998). Viewpoints for requirements elicitation: a practical approach. Computing Department, Lancaster University, Lancaster, UK, pp. 74-81.
J. Aitchison (2010). Teach Yourself Linguistics. 7th Edition, Hachette UK.
J. Andrade, J. Ares, R. García, J. Pazos, S. Rodríguez and A. Silva (2003). A methodological framework for generic conceptualization: problem-sensitivity in software engineering. Information and Software Technology, Vol. 46, No. 10, pp. 635–649.
J. D. Arthur and M. K. Gröner (2004). An operational model for structuring the requirements generation process. Requirements Engineering, Springer-Verlag London.
J. F. Sowa (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole.
J. L. Meurer (2002). Genre as diversity, and rhetorical mode as unity in language use. Revista Ilha do Desterro A Journal of English Language, Literatures in English and Cultural Studies, No. 43, Florianópolis, Brasil.
J. M. Swales (1990). Genre Analysis: English in Academic and Research Settings. Cambridge University Press, Cambridge.
J. Sinclair (1991). Corpus, Concordance, Collocation. Oxford University Press, Oxford.
J. Swales (1981). Aspects of article introductions. Language Studies Unit. The University of Aston, Birmingham.
J. Swales (1990). Genre analysis: English in academic and research settings.Cambridge University Press Cambridge.
J. Swales (2004). Research genres. Exploration and applications. Cambridge University Press, Cambridge.
J. Yates (1989). Control through communication: The rise of system in American management. Johns Hopkins Un. Press, Baltimore.
J.A.G. Bogarin and N.F.F. Ebecken (1996). Integration of Knowledge Sources for Flexible Pipe Evaluation and Design. Expert Systems with Applications,Vol. 10, No. 1.
K. Li, R.G. Dewar and R.J. Pooley (2003). Requirements capture in natural language problem statements. Heriot-Watt University, 2003. Available in http://www.macs.hw.ac.uk:8080/techreps/docs/ files/HW-MACS-TR-0023.pdf
K. Li, R.G. Dewar and R.J. Pooley (2003). *Requirements capture in natural language problem statements*, Heriot-Watt University. Retrieved from: http://www.macs.hw.ac.uk:8080/techreps/docs/files/HW-MACS-TR-0023.pdf
K. Pohl (2010). The Requirements Engineering Framework. Chapter 4 in Requirements Engineering. Springer-Verlag Berlin Heidelberg.
L. Barroca, J. L. Fiadeiro, M. Jackson, R. Laney and B. Nuseibeh (2004). Problem Frames: A Case for Coordination. Coordination Models and Languages. Lecture Notes in Computer Science Vol. 2949, pp 5-19.
L. Castro, F. Baiao and G. Guizzardi (2010). A Linguistic Approach to Conceptual Modeling with Semantic Types and OntoUML. Proceedings of 14th IEEE International in Enterprise Distributed Object Computing Conference Workshops (EDOCW). Vol. 1, 56, pp. 215-224.
L. Dittmann, T. Rademacher and S. Zelewski (2004). Performing FMEA using ontologies. In 18th International Workshop on Qualitative Reasoning. Kleer, J. &Forbus, D. (eds.): Evanston, Illinois. pp. 209-216.
L. Jiang (2005). A Framework for the Requirements Engineering Process Development. Department of Electrical and Computer Engineering Calgary, Alberta.
L. Liu, T. Li and X. Kou (2014). Eliciting Relations From Natural Language Requirements Documents based on Linguistic and Statistical Analysis. Proceedings of IEEE 38th Annual International Computers, Software and Applications Conference.
L.S. Zettlemoyer and M. Collins (2005). Learning to Map sentences to Logical Form: Structured Classification with Probabilistic Categorical Grammars. In Proceedings of Uncertainty in Artificial Intelligence (UAI05).
M. Barlow (1996). Corpora for Theory and Practice. In Computers and Applied Linguistics. Chapella (ed.), Rice University.
M. Christel, and K. Kang (1992). Issues in Requirements elicitation". Technical Report CMU/SEI-92-TR-012 ESC-TR-92-012. Software Engineering Institute. 1992. Available: http://www.sei.cmu.edu/library/abstracts/reports/92tr012.cfm Accessed: September 12, 2013.
M. Jackson (1995). Problems and requirements. Proceedings of the Second IEEE International Symposium on Requirements Engineering. Vol.1, Pp. 2-8.
M. Minsky (1974). A framework for representing knowledge. Artificial Intelligence Memo 306, Massachusetts Institute of Technology, A.I. Laboratory, Cambridge, MA, USA.
M. Palmer, D. Gildea and P. Kingsbury (2005). The Proposition Bank: An annotated corpus of semantic roles. Computational Linguistics, Vol 1, No. 31, pp.71.106.
M. Rajman and B. Besancon (1997). Text Mining: Natural Language techniques and Text Mining applications. IFIP 1997. Published by Chapman & Hall.
M. T. Cabré (1999). La Terminología: Representación y Comunicación. Elementos para una teoría de base comunicativa y otros artículos. Barcelona: IULA Universidad Pompeu Fabra.
M. van Nus (1999). Business genres and their corporate con-text. Journal of Research and Problem Solving in Organi-zational Communication: Document Design, 1(3), 187-197.
M. M. Bakhtin (1986). The Problem of Speech Genres. In Speech Genres and Other Late Essays, Emerson & Holquist (eds). University of Texas Press, Austin.
N. Aussenac-Gilles, B.Biébow and S. Szulman (2000). Revisiting Ontology Design: A Method Based on Corpus Analysis. Knowledge Engineering and Knowledge Management. Methods, Models, and Tools, 1937, pp.27–66.
N. Fairclough (2003). Analyzing Discourse and Text: Textual Analysis for Social Research. RoutledgeEditors, London.
N. Guarino (1998). Formal Ontology and Information Systems. In Nicola Guarino, editor, Formal Ontology in Information Systems, pages 3–15. IOS Press.

© Bell Manrique-Losada, 2014. All rights reserved.

O. Isam, A. Mrayat, N. M. Norwawi and N. Basir (2013). Requirements Elicitation Techniques: Comparative Study. International Journal of Recent Development in Engineering and Technology. Vol. 1, Issue 3.
P. Jackson (1999). Introduction to Expert Systems, 3rd edition. Boston, MA:Addison-Wesley.
P. Loucopoulos and L. Karakostas (1995).V. System Requirements Engineering. McGraw-Hill Ed. New York, NY.
P. M. Nugues (2014). Language Processing with Perl and Prolog. Cognitive Technologies. Springer-Verlag Berlin Heidelberg.
P. Vossen, (2002). EuroWordNet General Document. Version 3. Piek Vossen (ed.). University of Amsterdam.
R. C. de Boer and H. van Vliet (2009). On the similarity between requirements and architecture, Journal of Systems and Software, Volume 82, Issue 3, pp. 544-550.
R. Jackendoff (1997). The architecture of the language faculty. MIT Press, Cambridge, MA, USA.
R. Y. Lee (2013). Requirements Elicitation. Software Engineering: A Hands-On Approach. Atlantis Press, Central Michigan University.
S. A. K. Al–Khanak (2012). University experts' knowledge acquisition: Investigating knowledge engineering system approach. African Journal of Business Management Vol. 7(2), pp. 135-143.
S. Eggins (1994). An introduction to systemic functional linguistics. Pinter, London.
S. Grimes (2008). Text technologies in the mainstream: Text analytics solutions, applications and trends. Available in: http://altaplana.com
S. Hyon (1996). Genre in three traditions: Implications for ESL. TESOL Quarterly, Vol. 34(4), pp.693-722.
S. Kendal and M. Creen (2006). An Introduction to Knowledge Engineering, 1st Ed. Springer.
S. Patwardhan, B. Boguraev, A. Agarwal, A. Moschitti and J. Chu-Carroll (2012). Labeling by Landscaping: Classifying Tokens in Context by Pruning and Decorating Trees. Proceedings of the Conference on Information and Knowledge Management, Maui Hawaii.
S. Stein, Y. Lauer and M. El-Kharbili (2009). Using Template Analysis as Background Reading Technique for Requirements Elicitation. In Conference Software Engineering 2009. Fachtagung des GI-Fachbereichs Softwaretechnik, Kaiserslautern. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.7424.
S. Vadera (2005). Inducing Safer Oblique Trees without Cost. Expert Systems,Vol. 22, No. 4.
S. Viller and I. Sommerville (2000). Social analysis in the requirements engineering process: from ethnography to method. Proceedings of the 4th International Symposium on Requirements Engineering. (RE'99), IEEE CS press.
S.J. Russell & P. Norvig (2002). Artificial Intelligence: A Modern Approach, 2nd ed. Prentice Hall.
S.Robertson and J. Robertson (2008). Mastering the Requirements Process, 2nd ed., New Jersey: Addison Wesley.
T. A. Byrd, K. L. Cossick and R. W. Zmud (1992). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques. MIS Quarterly, Vol. 16(1), pp.117–139.
T. A. Byrd (1995). Expert Systems Implementation: Interviews with Knowledge Engineers. Industrial Management and Data Systems, Vol. 95, No. 10.
T.A. van Dijk (1980). Macrostructures: An Interdisciplinary Study of Global Structures in Discourse, Interaction, and Cognition. Hillsdale, Erlbaum, N.J.
T. A. van Dijk (2008). Discourse and Context. A Sociocognitive Approach. Cambridge University Press, Cambridge.
T. Kuhn (2010). Controlled English for Knowledge Representation. Ph.D. Thesis, Faculty of Economics, Business Administration and Information Technology, University of Zurich.
T. Mitamura and E. Nyberg (1995). Controlled English for Knowledge Based Machine Translation: Experience with the KANT System, Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI), Belgium.
U. Connor (1996). Contrastive Rhetoric: Cross-Cultural Aspects of Second-Language Writing. Cambridge University Press, Cambridge.
U. Sajjad and M. Qaisar (2010). Issues and Challenges of Requirement Elicitation in Large Web Projects. Master Thesis Computer Science, Blekinge Institute of Technology.
V. Devedzic, J. Debenham and D. Radovic. (1996). Object-Oriented Modelling of Expert System Reasoning Process. Proceedings of IEEE Conference on Systems, Man, and Cybernetics. Beijing, pp. 2716–2721.
V. Gupta and G. S. Lehal (2009). A Survey of Text Mining Techniques and Applications. Journal of Emerging Technologies in Web Intelligence, Vol. 1, No. 1, pp.60-76.
W. A. Woods (1975). What's in a link: Foundations for semantic networks. In Daniel G. Bobrow and Allan Collins, editors, Representation and Understanding: Studies in Cognitive Science, pages 35{82. Academic Press.
W. Fan, L. Wallace, S. Rich and Z. Zhang (2005). Tapping into the Power of Text Mining. Journal of ACM, Blacksburg.
W. Martin (2005). Developing Linguistic Corpora: a Guide to Good Practice. Oxbow Books, Oxford.
W.C. Mann, C.M. Matthiessen and S.A. Thompson (1992). Rhetorical structure theory and text analysis. In: Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text. Mann & Thompson (eds). Benjamins, Amsterdam.
Z. Zhang (2007). Effective Requirements Development – A Comparison of Requirements Elicitation techniques," in INSPIRE2007, Tampere, Finland.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 3

STATE-OF-THE-ART REVIEW

By necessity every book must have at least one flaw; a misprint, a missing page, one imperfection...
-- Howard Schwartz

In this Chapter we present the state-of-the-art review process related to this Ph.D. Thesis. We perform a review process comprising stages, organized in three sections: i) general review oriented to the approaches for modeling domain information in RE; ii) a systematic review about mapping NL into CL in the context of RE; and, iii) a systematic review of techniques, methods, and formalisms for knowledge representation from business processes.

In Section 3.1 we discuss the previous attempts to work on the problem of modeling domain information, from two points of view: i) the ways in which discourses can be modeled based on NL descriptions, and ii) several approaches for modeling and representing knowledge identified in the context of the RE process.

In Section 3.2 we describe and analyze the process, findings, and relationships between the contributions found in a state-of-the-art review about the mapping process between natural language and controlled language, in the context of the RE process. We define higher-level categories for differentiating and identifying certain phenomena in the mapping process above, according to the level of natural language involved in the input technical documents as well as the kind of technical document processed.

Finally, in Section 3.3 we present the process, findings and results of the state-of-the-art review about knowledge and pattern identification in the RE process.

3.1 MODELING DOMAIN INFORMATION IN THE RE PROCESS

The state of the art includes several attempts for modeling domain information in the RE process. We identify two categories: i) contributions based on the discourses modeling from natural language descriptions, and ii) contributions for modeling and representing knowledge in the context of the requirements elicitation process. We describe our findings in the following two sub-Sections.

3.1.1 Discourse Modeling based on natural language descriptions

The last three decades show a growing interest in work for making contributions to the conceptual modeling of discourse. In this section we present a review of such interest, showing the different approaches found in the state-of-the-art, contributing to the specification of a discourse expressed in NL and its conceptually modeling made by an analyst. The proposals will be grouped into four categories, but the methodology goal of this study will focus on summarizing the conceptual model approaches and the elements from the discourse which have been considered by the authors.

Approaches for generating conceptual models

Van Welie and Van der Veer (1998) propose conceptual modeling based on the discourse analysis in terms of: events, objects, tasks, roles, and agents. We show such a model in Figure 3.1. In this model the authors express how an event trigger a task, the task uses an object, a role uses the object, and in turn it is responsible of a task, developed by an agent.

© Bell Manrique-Losada, 2014. All rights reserved.

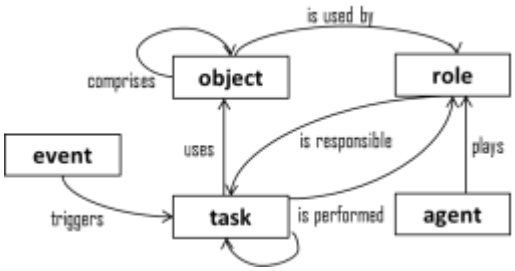


Figure 3.1. Task-based conceptual model (Van Welie & Van der Veer, 1998).

Gea *et al.* (2003) propose modeling dynamic aspects from discourse context by designing a hierarchy structure similar to those used for describing and modeling scenarios. Such structure is defined in a classes diagram by using the following elements: a *context* comprising *organization*, *law*, *artifact*, *information object*, *event,* and *group*. In turn *organization* comprises *role*, other *organizations*, *strategy*, *objectives*, and *restrictions* (e.g. *law* or *capacity*). A group is formed by *actors* developing a *role* and using *capacities*. The *roles* are related to *tasks*, and the *tasks* in turn are formed by *activities* (e.g. *actions* or related *activities*). The *actors* and *artifacts* develop *actions*, and the *events* generate *actions*.

TROPOS methodology (Bresciani *et al.*, 2004) includes the definition of intentional concepts as a fundamental base for modeling requisites and then designing an software application. Such intentional *concepts*—e.g. *actors, objectives, resources, plans, softgoasl, dependences*—support a conceptual model from an early phase in the RE process.

Ramadour and Cauvet (2002) present a method for structuring a model, based on the acquisition and representation of three kinds of knowledge: *goals*, *activities*, and *objects*. Goals represent the problems to be solved by using several alternatives domain processes. The activities are the specific process for achieving the goals and the objects are the actors and resources.

Chen (1983) proposes a set of rules for correlating certain NL-based patterns in English into a specific representation—the conceptual entity-relationship diagram (Chen, 1976). Such rules are based on the identification of the following elements from a phrase: noun, verb, adjective, and adverb. The common noun is translated into the concept of *entity*. The transitive verb is translated into a relationship. The adjective is converted into an attribute from an entity, and the adverb into an attribute of a relationship.

Antonetti and Miglio (2009) created the CASER tool for helping the automated generation of conceptual models in the form of the entity-relationship diagram. Concerning this generation, a discourse is expressed in terms of direct links to concepts with their action-connectors and/or attributes.

Linguistic Analysis from a discourse

LIDA (Linguistic Domain Analysis) Project (Overmyer *et al.,* 2009) is oriented to the classification of a domain discourse according to the kind of word—verb, noun, or adjective—and its occurrence frequency. Based on an assistant (LInguistic assistant for Domain Analysis), an analyst can generate semi-automatically classes diagrams, from the initial word classification.

The NIBA Project (Natural Language Information Requirements Analysis) was developed for analyzing information requisites written in NL (Kop & Mayr, 2002) by using a specific syntax (*Natürlichkeitstheoretische* Syntax), pre-design conceptual models and then generating UML diagrams. Kop and Mayr propose in this Project, developed in the linguistic analysis area from Klagenfurt University—a different manner for generating a kind of conceptual models called conceptual pre-design models. Such models capture the relevant domain aspects based on a NL input and its transformation into a schema closed to a glossary. The linguistic analysis is performed as follows: the words are compiled in a lexicon; a group of verbs is assigned by word; the descriptions are represented in tree expressing the constitutive relationships and dependences among words.

© Bell Manrique-Losada, 2014. All rights reserved.

The Grammalizer (Hoppenbrouwers *et al.*, 1999) performs the analysis based on the decomposition of a discourse in terms of actor/agent, associations/actions, object/patient, prepositions, and other objects. This approach is oriented to the morpho-syntactic analysis of the discourse.

Burg and Riet (1997) propose a representation language for communication systems, called Color-X. Such a language is founded on linguistics concepts and generates oriented-object code. The language is based on concept relationships (constructors) described in a lexical database in English (WordNet) for generating the constructs (*e.g.* names) and the relationships (*e.g.* verbs) among them.

Harmain and Gaizauskas (2000) propose a discourse interpreter called CM-Builder. Such an interpreter represents a discourse based on a semantic net, composed by the nouns of the phrases and verbs. The elements identified facilitate the organization of a semantic net as we show in Figure 3.2. Additionally, the interpreter automatically generates a conceptual class diagram.

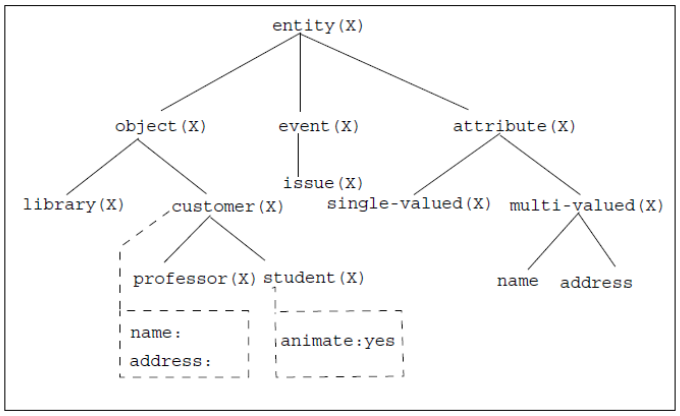


Figure 3.2. Semantic Net model (Harmain & Gaizauskas, 2000)

Analysis of dynamic and structural elements from a discourse

In the context of the ABCBesoins methodology, Urrego-Giraldo (2005) proposes a formal language for representing requisites from the discourse description. The formalism is defined by using a verb and seven parameters, as follows: *main agent* responsible for the intervention; *object*, which support the actions of verb; *initial* and *final situation* from the object; *environment* used for acting on the object; *method* for performing the actions and interviewing the object; and, *other agent* which interacts with the main one. By using these elements, an analyst specifies a stakeholder discourse written in NL which can be used for generating conceptual models.

Based on a predefined discourse, Zapata and Arango (2007) propose the identification of a set of structural and dynamic elements for including in the so-called pre-conceptual schemas. They define such schemas as an intermediate step in the transformation process from NL to a CL. The basic elements for designing the schema are: concepts, as nouns or noun phrases; structural relationships, as verbs generating permanent connections among concepts; dynamic relationships, as verbs denoting actions; implications, as cause-and-effect relationships or conditionals-dynamic relationships; conditionals, as expressions formed by a concept and operators acting as pre-condition for a dynamic relationship.

Loucopoulos and Kavakli (1995) design a technique for modeling discourses from organizational objectives, social roles, and operations. Such a technique is based on identifying the following elements: objectives as a core of the modeling process; roles and actors; processes, as the mechanisms for changing the states inside the organizational system; and, resources as the physical media or information media for producing the business processes.

In Table 3.1 we summarize the principal findings of the previous review, specifying for each approach the kind of conceptual model proposed and the elements identified from the discourse for generating such conceptual model.

© Bell Manrique-Losada, 2014. All rights reserved.

Table 3.1. Summary of findings for discourse modeling based on NL descriptions

Authors	Conceptual Model	Elements identified form the discourse
Welie and Veer (1998)	Tasks Model	Events, objects, tasks, roles, and agents
Gea *et al.* (2003)	N/A	Dynamic elements from discourse
TROPOS (2004)	Goals Diagram I*	Actor, objective (objective, resource, plan, softgoal) and dependences
Ramadour and Cauvet (2002)	N/A	Goals, activities, and objects
Chen (1983)	Entity-Relationship Diagram	Noun, verb, adjective, and adverb
Antonetti and Miglio (2009)	Entity-Relationship Diagram	Concepts, actions connectors and/or attributes
Overmyer *et al.* (2001)	Class Diagram	Noun, verb, and adjective
Kop and Mayr (2002)	Class Diagram, Activity Diagram, State-machine Diagram.	Conceptual pre-design: Nouns and verbs
Hoppenbrouwers *et al.* (1999)	Entity-Relationship Diagram	Actor (agent), association (action), object (patient), preposition and other objects
Burg and Riet (1997)	OO Code	Nouns and verbs
Harmain and Gaizauskas (2000)	Class Diagram	Nouns and verbs
Urrego-Giraldo (2005)	N/A	Verb, object, situation, environment, method, and agent
Zapata *et al.* (2007)	Class Diagram, Communication Diagram, State-machine Diagram.	Pre-conceptual schema: concepts, structural and dynamic relationships, conditional, implications
Loucopoulos and Kavakli (1995)	Goal Diagram	Objectives, roles and actors, processes and resources

3.1.2 Modeling and representing knowledge in RE process

In the RE process, analysts and stakeholders should exchange information about the context and activities from the domain. This information is useful for understanding how the process occurs, trying to obtain a CL from the NL discourse. For facilitating and improving this RE process, several approaches have been proposed. Because of this diversity, this Section of the state of the art encompasses some models from different viewpoints, but the complete review is included in Zapata and Manrique (2012). These models are categorized as meta-models, meta-ontologies, and ontologies, as we describe as follows.

Models and meta-models

The KAOS meta-model (Dardenne *et al.*, 1991) is made up of meta-concepts, meta-relations linking meta-concepts, meta-attributes characterizing meta-concepts and meta-relations, and meta-constraints upon several kinds of components. The meta-model uses the "meta" prefix referring to the meta level, where domain-independent abstractions are defined. This level is different from other levels involved: the domain level, where concepts specific to the application domain are defined, and the instance level, where particular instances of domain-specific concepts are introduced. The requirements meta-model (see Figure 3.3) is based on instances of meta-concepts, linked to instances of the meta-relations, and characterized by instances of meta-attributes. This model emphasizes both conceptual and structural aspects.

Hu *et al.*, (2010) propose a 'requirements meta-model' from the perspectives of *Role, Goal, Process*, and *Service* (RGPS). The meta-model focuses on modeling requirements from early RE processes and, then, the process can be used for dynamically capturing stakeholder needs. The relationships of the RGPS are defined in four layers (see Figure 3.4).

© Bell Manrique-Losada, 2014. All rights reserved.

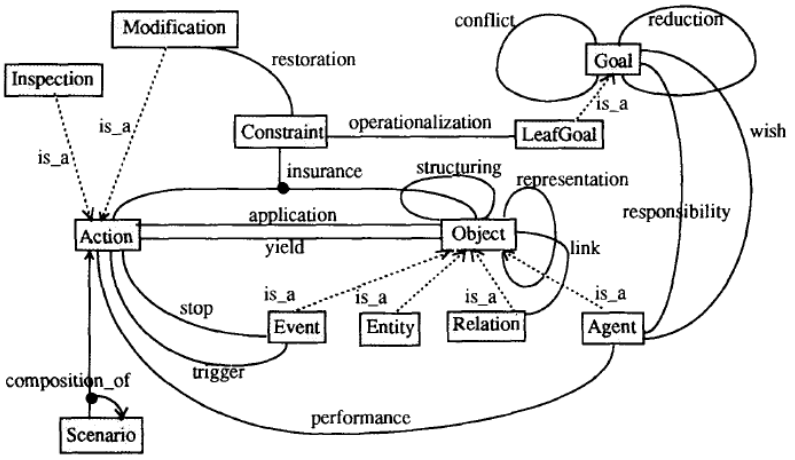


Figure 3.3. The KAOS meta-model (Dardenne *et al.*, 1991)

The *Role Layer* contains actors (*i.e.,* human or software), who are involved in the RE process, and information about the roles played respectively. The *Goal layer* depicts strategic goals (*i.e.,* Functional or Non-Functional goals) of service-oriented requirements. The *Process layer* is a multiple model, where processes can include other processes. Finally, the *Service layer* represents the service composition after the services have selected the service repository. The RPGS model refers to the roles involved in RE processes and represents the functional elements implicated.

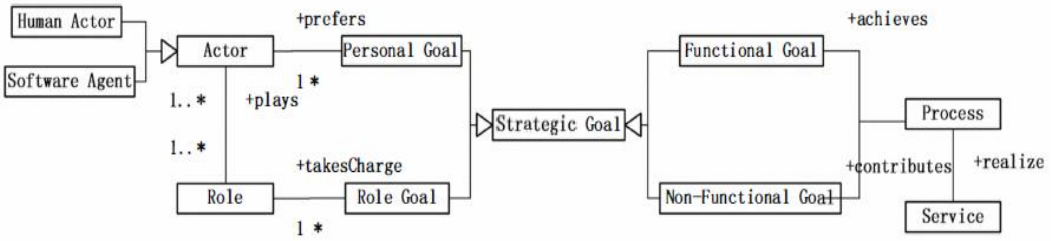


Figure 3.4. The RPGS meta-model (Hu *et al.*, 2010)

Wei *et al.*, (2008) present an approach near to the knowledge representation of requirements expressed in natural language. Also, they show how to define natural language patterns as an abstraction of the requirement statement. In this model, *patterns* (a part of the sentence) describe several requirement statements, *i.e.*, a complete sentence of various requirements. Both, we and Rubin (2009) in Figure 3.5 presents the proposal, where the *Behavior* represents a basic functional requirement, which is a kind of simple sentence. *Constraint* and *Status* are other kinds of simple sentences, where a *constraint* represents non-functional requirements (like quantitative and qualitative constrains), and the *status* represents the context requirements and services belonging to the stakeholders. An *Event* is a behavior trigger, which activates changes in either the entity state or the system state. A *Condition* is a test of the current state of an entity or a test of the current system state.

Rubin (2009) proposed a meta-model as a way to represent the domain knowledge. This model incorporates essential concepts to establish and understand the domain knowledge in the software development process. In this meta-model, Rubin proposes ontological constructs, *i.e.*, ontological components which can be combined to build a model in a specific domain. The meta-model incorporates the domain-related constructs and their relationships, link attributes, and states, as we show in Figure 3.5.

© Bell Manrique-Losada, 2014. All rights reserved.

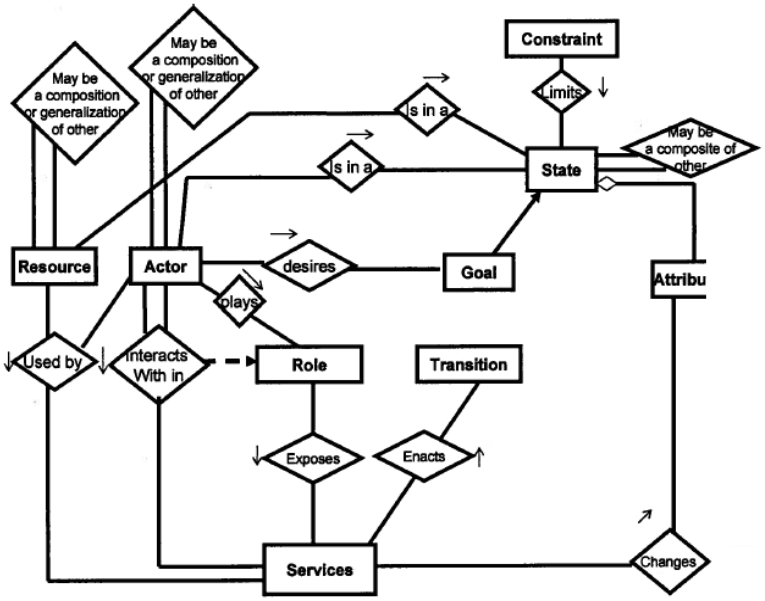


Figure 3.5. Meta-model of requirements engineering domain knowledge (Rubin, 2009)

Ontologies and meta-ontologies

On the other hand, some other approaches have been introduced oriented to ontologies or meta-ontologies. Zapata *et al.*, (2010) proposed a meta-ontology of the RE process. This meta-ontology is incremental and independent of the problem domain. The objective is using the meta-ontology to obtain stakeholder concepts from a dialogue expressed in NL. The concepts identified in the meta-ontology are: *object*, *actor*, *constrains*, *actor-function*, *activity,* and *organization*.

Kaiya and Saeki (2006) present an ontology to define semantics elements inside requirements descriptions. They define a kind of ontology called *thesaurus*. The thesaurus has concepts and relationships among the concepts. Also, it has several subclasses of concept classes and relationships, as we present in Figure 3.6. In the figure, the element "object" is a sub-class of a concept-class and the relationships "apply" and "perform" can connect two concepts, which in turn can be function, object, environment, constraint, and quality.

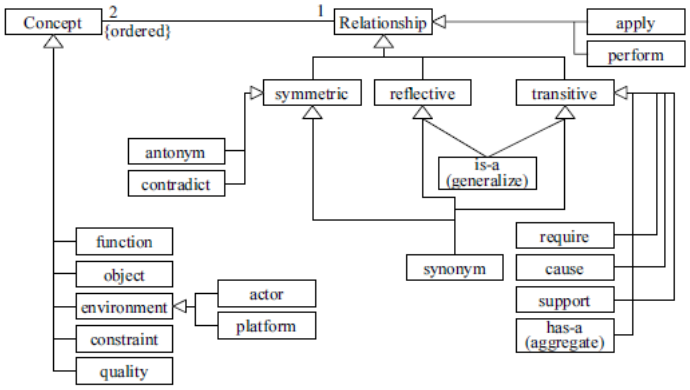


Figure 3.6. Meta-model of Ontology for requirements (Kaiya & Saeki, 2006)

A meta-model for the RE process is proposed by Viet and Ohnishi (2009). This model is based on functional requirements (FR). Each FR is modeled as a node of the ontology, including one verb and several nouns. Inheritance and aggregation relationships can be represented as functional structures belonging to systems of certain domains. FR include other relationships, as the following: agent of the function (who), location of the function (where), time (when), reason (why), and non-functional requirements.

© Bell Manrique-Losada, 2014. All rights reserved.

Finally, another relevant approach is LEL (Language Extended Lexicon) (Leite & Franco, 1993). This proposal focuses on lexical elements identified from NL. The lexicon comprises symbols like active objects or subjects (perform actions), passive objects or subjects (actions are performed on them), verbs, and significant states of the system. Demitrio (2005) proposes a framework for the automated LEL generation, based on information from specific documentation, as we present in Figure 3.7. From such a lexicon, the analyst could generate scenarios for representing knowledge in a specific context.

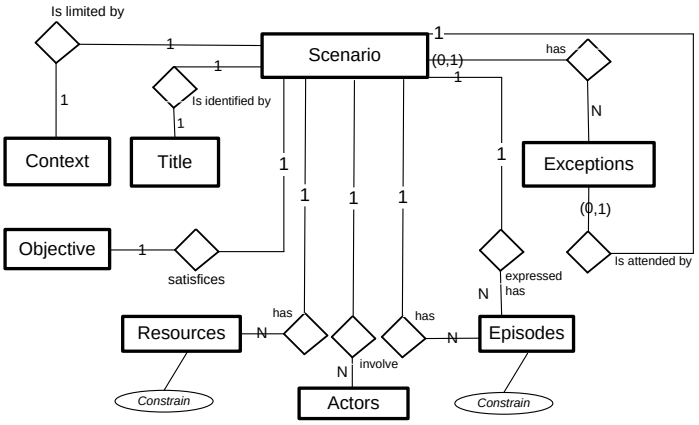


Figure 3.7 .Model to generate LEL from requirements (Demitrio, 2005)

3.2 SYSTEMATIC REVIEW FOR MAPPING NL/CL INTO THE RE PROCESS

We provide a synthesis of trends and conceptual approaches found in the state-of-the-art concerning the NL mapping into CL during the RE process. We developed a systematic state-of-the-art review process and identified higher-level groupings for the analyzed approaches, according to the level of NL involved in the discourses as well as the kind of technical document processed.

We adapted the Kitchenham methodology (2004) which integrates a set of guidelines for conducting state-of-the-art reviews in software engineering. This methodology emphasizes the need for confirming the review, by solving a research question, and following a review plan. The methodology covers the steps of basic review—review protocol—, supported by a method for structuring the search, making the data extraction, analyzing, and synthesizing. The information sources are selected, including the following:

- Databases: SCOPUS, ScienceDirect, EBSCO, IEEE
- Digital Libraries: ACM, BUCM, CitiSeer, INFOMINE
- Catalogues: Latindex, Rebiun and Biblioteca Virtual
- Directories: CCSB, DOAJ, DoCIS, OCLC, REDALYC, Worldcat

The terms and search expressions are selected, based on a set of candidates terms responding to the research question. Such terms are part of the search expressions or keywords, which are fundamentals for optimizing the searching and identification of relevant studies. The expressions are: Natural Language Transforming; Natural Language mapping; Natural Language into Controlled Language; Conversion/translation of Natural Language into Controlled Language; translation Natural Language to Controlled Language in Requirements Elicitation; Natural Language Processing in Requirements Elicitation.

Search and inclusion/exclusion criteria are defined compiling the studies found in the search phase. Then, we apply the following three filtering criteria in the title and abstract Sections of the studies: i) study in the software engineering discipline; ii) published in English or Spanish; and, iii) oriented to the requirements elicitation. Finally, the following inclusion and exclusion criteria are applied on the abstract and conclusions, for defining the relevance of the study: i) explain the context of the study; ii) describe the objectives which motivate the study; iii) propose a model, method, tool, instrument, or framework for mapping NL to CL in the RE process. As a result of the first search phase, we obtained the results organized in Table 3.2, which are classified by information source. 15,920 out of 381,833 results are published in

© Bell Manrique-Losada, 2014. All rights reserved.

Spanish. Based on the predominance of the studies publishes in English (96%), the next search iteration is exclusively developed with English expressions. Then, we developed the second iteration and obtain the results of Table 3.3.

Table 3.2 Results from the 1st search phase

Search source	# studies
ACM	283,578
SCOPUS	75,435
The Collection of Computer Science Bibliographies	13,908
ScienceDirect	5,916
CitiSeer	1,857
Rebiun: Red de bibliotecas universitarias españolas	955
Biblioteca de la Universidad Complutense de Madrid	184
Total	**381,833**

By following the review protocol, we selected the first 300 results from each source, eliminated the duplicated results, and obtained 10,504 results. After applying the first two filtering criteria on the previous results, we obtained 193 studies. From these 193 studies, we preselected 81 corresponding to which meet the 3rd criterion. Finally, we apply the inclusion and exclusion criteria and obtain 28 studies. Such studies are considered as primary studies. We define a classification category by identifying the input or source of the method used in each study and the kind of result/output of the process. The categories are as follows:

Table 3.3 Results from the 2nd search phase

Search source	# studies
Google Scholar	150,917
Worldcat	18,389
Microsoft Academic Search	5,993
The IEEE Transactions on Software Engineering (TSE)	5,366
Base	4,093
Scirus	3,289
EBSCOhost	1,011
Other sources	1,309
Total	**190,367**

A. Natural Language Processing from elicited requirements. This category is related to the studies where the processing is on requirements documents, commonly organized as a Software Requirements Specification (SRS).
B. Controlled Natural-Language Processing. Studies with their processing based on input documents with any kind of human intervention—not completely in natural language.
C. Natural Language Processing form textual requirements. This category is related to those studies whose processing is based on documents completely written in natural language, with no any modification.

Likewise, for each category we specify the following aspects: i) type of linguistic contribution, in grammatical, syntactic, semantic, or pragmatic terms; ii) language of source document for processing; contribution and description of the approach; and iii) method, model, or tool proposed. We present the synthesis of the primary studies based on the defined classification in Tables 3.4, 3.5, and 3.6.

Table 3.4 Synthesis of primary studies in category A

Primary Study	Lan guage	Linguistic contribution	Focus of Proposal	Description of Proposal	Model /Tool Proposed
Nat *et al.* (2001; 2005; 2006)	English	Unspecified	Development of customized software and market-oriented	Technique of similarity analysis. Contribution to Linguistics Engineering.	*ReqSIMILE Tool*
Post *et al.* (2011)	English	Restricted Grammar	Analysis of requirements documents	Grammar for motoring domain	*Unspecified*

© Bell Manrique-Losada, 2014. All rights reserved.

Lee (2003)	English	Two-levels grammar	Specification of formal executables	XML Grammar and structure. Contribution to NLP Contextual.	*VDM Toolkit*
Jain *et al.* (2009)	English	Unspecified	Analysis and quality evaluation of SRS in NL.	Asissted elicitation based on CASE tools. State-machine for verifying requirements.	*Requirements Analysis Tool (RAT)*
Mich (1996)	English	Unspecified	OO Models Generation	NLP System for generating OO requirements in NL.	*NL-OOPS LOLITA*
Ambriola & Gervasi (2006)	English	Morpho-syntactic analysis	Requirements Modeling, Formal validation	Automated generation of models.	*CIRCE*
Yang *et al.* (2010)	English	No specified	Ambiguity detection	Learning machine techniques for detecting ambiguities in phrases.	*Nocuous Ambiguity Identification*
Sampaio *et al.* (2007)	English	Unspecified	Aspect-Oriented Software Engineering	Early identification method for aspects, by applying mining techniques.	*EA-Miner*
Nanduri & Rugaber (1995)	English	Unspecified	Object extraction and OO-model generation.	SRS Analyzer in NL for obtaining OO models	*EDGE*
Spreeuwenberg & Anderson (2010) Bajwa *et al.* (2010)	English	Business rules semantically formulated	Formal Representation from NL to business rules	Semantic structure of statements, based on formal logic	*NL2SBVR tool RuleGenerator*
Cavada *et al.* (2009)	English	Unspecified	Formalization and validation from requirements	Automated categorization from requirements and UML formalization	*EuRailCheck*
Ormandjieva *et al.* (2007)	English	Characteristics extraction from a phrase	Quality evaluation of the SRS documents	Quality model for textual requirements and automated system for classification	*Unspecified*

Table 3.5 Synthesis of primary studies in category B

Primary Study	Lan-guage	Linguistic contribution	Focus of Proposal	Description of Proposal	Method/Model /Tool Proposed
Zapata *et al.* (2007); Zapata (2010)	Spanish	Controlled Language: UN-Lencep	Semi-automated elicitation	Intermediate model (pre-conceptual schema) and its transforming to UML models	*UNC–Diagrammer*
de Almeida & Rodrigues (2008)	English	Linguistic patterns	Model for specifying requirements	Linguistic pattern identification from SRS.	*ProjectIT-Requirements*
Dalianis (1995)	English	Surface grammar - DCG (definite-clause grammars)	Requirements specification and validation	Aggregating technique for NL (analysis and integration of redundancy), for specifying and validating.	*VINST-system*
Schwitter & Fuchs (1996)	English	Controlled Language: ACE. Knowledge representation structures	Goal verification	Tool for describing functional requirements and transforming them by using PROLOG.	*Lexical Editor and Spelling Checker Attempto (ACE)*
Cabral & Sampaio (2008)	English	Controlled Language. Special grammar in English	Formal specification of requirements	Usage of formal models for executing Use Cases automatically.	*Generador CNL - grammar CNL knowledge-bases*
de Sousa *et al.* (2010)	English	Unspecified	Consistency analysis	Consistency checking requirements, applying a formal method	*Unspecified*
Reiter & Dale (1997)	English	Unspecified	Generation LN	Building of NL generation systems	*Unspecified*
Almeida & Rodrigues (2009)	English	Unspecified	Formal Verification	Integrating requirements elicitation and model-driven techniques.. Model verification	*Wiki-based Tool: Computer Supported Cooperative Work*
Gordon, & Harel (2009)	English	Declarative Language of low level Static/ dynamic grammar	Generation of executable code	Scenarios specification from a GUI and conversion to sequence diagrams	*Play-Engine GUIEdit*
Elbendak *et al.* (2011)	English	No specified	Initial identification of class-object relationships	Automated identification of classes from requirements in Use Case descriptions.	*Class-Gen*
Videira & Rodrigues (2005)	English	Ontology for representing requirements	Proposal of patterns for representing requirements	Specification requirements language and meta-model	*ProjectIT-RSL Tool*
Hinge *et al.* (2011)	English	No specified	Formal specification from business processes	Semantic descriptions from business process models in BPMN	*ProcessSEER*

© Bell Manrique-Losada, 2014. All rights reserved.

Osborne & MacNish (1996)	English	ANLT Grammar	Formal specification from requirements, Quality of SRS	System for detecting ambiguities in SRS and generating formal logic expressions	*Unspecified*
Gervasi & Zowghi (2005)	English	Schema for mapping controlled NL into formal logic	Treatment of inconsistencies in the requirements specification	Analysis and translation techniques from controlled NL to formal logic for identifying inconsistencies in SRS.	*Framework CARL*
Popescu *et al.* (2008)	English	NL with a controlled grammar	Identifying inconsistencies and ambiguities in SRS	Semi-automated method for analyzing SRS from a restricted grammar	*Dowser*
Breaux *et al.* (2008)	English	Semantic parameterization	Goal specification without ambiguities	Definition of domain from objectives, in logic descriptions	*CERNO framework*
Deeptimahanti & Sanyal (2011)	English	Positioning technique concepts and Keyword-In-Context	Automated translation, inconsistencies identification from requirements	Technique for assisting the generation of UML models from NL normalized	*UMGAR*
Sullabi & Shukur (2008)	English	Natural Language Structured	Translation from NL specifications into a formal language	Translation technique for descriptions in NL to a formal specification	*SNL2Z*
Almeida & Rodrigues (2008)	English	Controlled Natural language	Requirements validation and specification	Tool for specifying and validating requirements from a controlled NL.	*ProjectIT-Studio/Requirements plugin*
Ko *et al.* (2007)	English – Korean	Unspecified	Requirements analysis	Method for classifying requirements in analysis categories	*Requirements analysis-system*

Table 3.6 Synthesis of primary studies in category C

Primary Study	Language	Linguistic contribution	Focus of Proposal	Description of Proposal	Method/Model /Tool Proposed
Lecoeuche *et al.* (1998)	English	Dialogue Structures	Improvement of the elicitation process from the users-oriented point of view.	Restrictions assignation technique for NL, by using a dialogue theory.	*Unspecified*
Black (1987)	English	McCord's and Slot Grammars	Semantic and syntactic structure from texts. Knowledge acquisition	Knowledge elicitation from textual phrases.	*NIAM*
Fliedl *et al.* (2007)	German	Linguistics theory. NTMS Grammar	Tagging and interpreting NL texts	Analysis and translations from textual specifications in a conceptual pre-design.	*NIBA-TAG*

The contributions of most of the analyzed studies are oriented to: improving quality, handling different levels of formalization, defining formal specifications, and approaching to automated conversion into a controlled language or formal model for the requirements. Within the studies categorized the issue of quality requirements is approached from the application of NL-processing techniques oriented to particular text classification, assisted guide from collaborative tools, conceptual understanding, and assurance of non-ambiguities based on requirement documents. Over 90% of the contributions were achieved with NL processing in English, although there are also approaches in Korean, German and Spanish. The vast minority of studies are oriented to the processing of documents in NL with no intervention, which is the orientation of this Ph.D. Thesis.

3.3 STATE-OF-THE-ART REVIEW FOR KNOWLEDGE AND PATTERN IDENTIFICATION OF LANGUAGE IN THE RE PROCESS

In this Section we present the process and findings of the state-of-the-art review for knowledge representation and Language Processing in the context of RE process. We consider the review from two perspectives: i) documents processing by applying NLP; and ii) Knowledge Identification from business processes, in terms of techniques, methods and formalisms.

© Bell Manrique-Losada, 2014. All rights reserved.

3.3.1 Document Processing by applying NLP

We developed a succinct review process based on scientific databases like Springer, ACM, IEEE, and ScienceDirect. Knowledge can be analyzed manually or with the use of artificial intelligence technologies—a combination of NLP and intelligent agents. Thus, the studies identified as relevant are related to the usage of NLP for linguistic processing, formal logic and similar formalizations, and language analysis resources, for different kinds of documents. The studies identified are summarized and discussed in the following paragraphs.

Using NLP

O'Shea and Exton (2004) use content analysis for extracting requirements from a corpus of bug reports. This technique is suitable whenever categories can be defined prior to the texts analysis and it requires a certain extent of domain understanding. Due to the large software development projects a big number of NL documents becomes available and needs to be analyzed and transformed into structured requirements, so Meth *et al.* (2012) focus on providing automated and knowledge-based support of the task elicitation sub-process.

Based on semi-structured documents we found background as follows: RARE project (Cybulski & Reed, 1998) is focused on parsing texts based on a semantic network assisted by a thesaurus; they combine NLP with faceted classification for analyzing and refining requirements. From non-functional requirements texts, Cleland-Huang *et al.* (2008) work on detection of viewpoints. Bajwa *et al.* (2011) propose mapping business rules to semantic business vocabulary. Antón (1997) shows how non-functional requirements can be implemented by means of functional requirements. These approaches are based on structured or semi-structured documents, but no experiments were developed for organizational documents written in NL.

Hahn *et al.* (1996) develop a methodology for knowledge acquisition and concept learning from texts written in German. The method relies on a quality-based model for reasoning on terminology, by using concepts from NLP. The goal of this method is being able to scan two kinds of documents: test reports on information technology products and reports of medical findings.

Related work focuses on goal refinement and delegation (Dardenne *et al.*, 1993; Darimont *et al.*, 2005; Giorgini *et al.*, 2005) identified from documents. Goals describe desired states or actions performed by actors regardless of specific consideration for normative positions (*e.g.*, permissions, recommendations, and obligations). Giorgini *et al.* (2005) describe the Tropos framework modeling ownership and delegation and defines obligation as "trust in execution." Young and Antón (2010)—based on a commitment analysis methodology—propose the identification of requirements by analyzing the commitments, privileges, and rights conveyed within online policy documents. Such policy documents have been identified as useful sources for extracting privacy requirements with a policy document and should parse it into individual statements. This parsing yields a set of elements which may be analyzed independently. In the Software Product Line domain, authors like Clements and Northrop (2002) propose processes for reviewing and studying source documents, but only oriented to define the features surrounding products, instead of including linguistics analysis.

Likewise, disciplines like spatial analysis (Kray & Blocher, 1999; Kray & Porzel, 2000) try to deal with designing agents for the deeper semantic and pragmatic understanding of the spatial concepts. Such concepts are employed by the user when he/she interacts with the system, and they are expressed in specifications. The previous approach is oriented to design NL interfaces in mobile personal assistants. In the field of Semantic Web, Lee and Bryant (2002) use contextual NLP to overcome the ambiguity and express the domain knowledge. The approach is supported by the DARPA agent markup language (DAML) which facilitates the generation of output expressions as formal representations of the informal NL requirements.

Formal logic and similar formalizations—language analysis resources

In knowledge engineering Dinesh *et al.* (2006) propose the validation of the conformance to regulations with organizational procedures texts by using formalizations; Aussenac-Gilles *et al.* (2000) apply a similar method on the ontological engineering domain; Rösner *et al.* (1997) use techniques to automatically generate multilingual documents from knowledge bases. The described techniques for knowledge

© Bell Manrique-Losada, 2014. All rights reserved.

acquisition from documents are a good base for using in particular knowledge structures. Compared with our approach, these authors identify information defining what stakeholders are permitted to do by using phrases previously created. Based on this approach, we consider documents comprising phrases in NL without modification.

Breaux *et al.* (2006) proposed semantic parameterization, a process for representing domain descriptions in first-order predicate logic. They also introduced Knowledge Transformation Language (KTL; a context-free grammar) for analyzing the most frequently expressed goals in more than 100 online policy documents to derive semantic models. In contrast, would be relevant guide to the requirements engineer in analyzing the original NL documents rather than using an intermediate representation of goals, which may not include relevant contextual cues. In contrast to organizational texts, policy documents contain statements emphasizing the commitments organizations make to their users rather than rights and obligations as prescribed by law. Legal-based approaches do not provide sufficient coverage of requirements as expressed in policy documents, because policies emphasize procedural practices rather than legal practices.

The usage of language analysis resources as corpora is a suitable means for describing and analyzing texts in a given area. The advantages of using a corpus for lexical study are many in comparison with an expert knowledge or intuition. According to Krishnamurthy (1997), some of such corpus can: i) be more comprehensive and balanced; ii) supply fairly accurate statistics; and iii) provide countless real examples. Such reasons are the basis for the extended usage of corpus for lexical research and analysis from texts.

An approach proposed by Wang (2005) uses induction of classification rules based on set theory and corpus. The capability of constructing a corpus is compared with databases, digitized books, journals, and magazine management systems. Data stored in a corpus can be electronically retrieved to either create or update a knowledge base, without the intervention of a knowledge engineer or an expert. This approach is looking for developing new methods for interpreting meanings in order to determine rules and other forms of knowledge, such as frames for case-based reasoning. A number of new methods are being developed and implemented.

Authors like Calzolari (1997) rely on computerized corpus for letting observe and analyze a set or organizational documents. As Aston (1997) asserts, corpora can play a useful role in the acquisition and restructuring of the schematic knowledge underlying communicative competence.

3.3.2 Knowledge Identification from Business Processes

The state-of-the-art review process is based on the information from scientific databases as Science Direct, IEEE, CMMI Institute, EBSCO host, Association of Computing Machinery (ACM), and search engines like Google Schoolar and CiteSeerX. The search process is based on the keyword "requirements elicitation and business processes," "requirements elicitation and knowledge engineering," "business processes identification/extraction," and "domain information extraction, knowledge engineering."

Based on such keywords, we identify the related studies and then apply filtering on title, abstract, introduction and keywords, for identifying the relevant studies according to the discipline, study object, and research questions, as follows:

- What are the techniques for helping the extraction or identification of business knowledge, as a way to eliminate the subjectivity in the requirements elicitation process?
- What approaches are useful for representing business knowledge and could be useful for supporting the requirements elicitation process?
- How can we apply analytic techniques for identifying business processes from organizational documents?
- What methods or formalisms could support the business processes identification from organizational documents written in NL?

© Bell Manrique-Losada, 2014. All rights reserved.

As inclusion criteria were defined: the publication year later than 2005, the kind of study—journal paper, Ph.D. Thesis, M.Sc. Thesis, proceedings or scientific book—and finally the online availability as a full text. The exclusion criteria were: studies whose contribution implies a high intervention of analysts and studies where the identification of business information comes after the requirements to the requirements elicitation process. After applying the review protocol, we found a set of relevant studies, as we present in Table 3.7.

Table 3.7 Synthesis of primary studies

Id.	Study	Contribution
1	Vegega *et al.* (2012)	Business Domain formalization for information exploitation projects.
2	Pytel *et al.* (2011)	Knowledge representation techniques in requirements analysis.
3	Hossian *et al.* (2007)	Knowledge-oriented methodology for requirements elicitation
5	de la Vara *et al.* (2009)	Data requirements modeling for business processes-based systems.
6	Aysolmaz & Demirors (2014)	Modeling business processes to generate artifacts for software development
9	de la Vara *et al.* (2007)	Requirements models based on processing models and goals models.
10	Zapata *et al.* (2007)	Capturing requirements based on a CASE tool.
11	da Cunha *et al.*, (2014)	Formal representation for context-aware business processes
12	Rao *et al.*, (2012)	Ontology based on knowledge maps to assist business process re-engineering
13	Hao *et al.*, (2014)	Knowledge map-based method for domain knowledge browsing
14	Stuit & Wortmann (2011)	Discovery and analysis of e-mail-driven business processes
15	Tavares *et al.* (2009)	Knowledge Management embodied in work processes

Based on the synthesis of the studies, we identify the following features of the approaches and their contributions:

- Almost the half of the proposed processes is based on techniques for representing domain knowledge. Some of the more relevant techniques for extracting knowledge are: design of scenario maps by users, objectives diagrams, and goals models. Most of them help to identify the knowledge domain from users.
- From knowledge engineering we find interesting and useful techniques to be used in the context of the RE process. Some of them are: concept-attribute-value triples, semantic nets, and knowledge diagrams. Such techniques could be applied in the RE process as a way to facilitate the domain understanding by the analysts in the process.
- Application of such techniques on technical documents—as we suggest in this Ph.D. Thesis—is not addressed in state of the art.
- In most of the studies the intervention of the analysts in the process is considered 'low' since the contributions indicate the main support from instruments or tools, helping to reduce the ambiguities in information analysis.
- Regarding the dates of the studies, we identify the timeline of such studies: 25% were presented until 2007; 50% were presented from 2009 to 2012, and the 25% from 2014.

REFERENCES

A. Dardenne, A. van Lamsweerde, S. Fickas (1993). Goal-directed Requirements Acquisition. Science of Comp. Prog., v. 20, pp. 3-50.

A. Dardenne, S. Fickas and van A. Lamsweerde (1991). Goal-directed Concept Acquisition in Requirements Elicitation. *Proceedings of the 6th international workshop on Software specification and design,* IEEE Computer Society Press ,pp. 14-21, Italia.

A. Hossian, E. Sierra, P. Britos, M. Ochoa and R. García (2007). Hacia una Metodología Orientada al Conocimiento para la Educción de Requisitos en Ingeniería del Software. Proceedings IX Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. pp. 60-70.

A. I. Antón (1997). Goal Identification and Refinement in the Specification of Software-Based Information Systems. Ph.D. Thesis, Georgia Institute of Technology.

A. Post, I. Menzel, A. Podelski (2011). Applying restricted english grammar on automotive requirements: does it work? A case study. Proceedings of the REFSQ'11. Vol. 1. pp. 166-180.

A. Sampaio, A. Rashid, R. Chitchyan, P. Rayson (2007). EA-Miner: towards automation in aspect-oriented requirements engineering. Transactions on aspect-oriented software development III. Lecture Notes in Computer Science. Vol. 46. pp. 4-39.

© Bell Manrique-Losada, 2014. All rights reserved.

B. Aysolmaz and O. Demirors (2014). Modeling business processes to generate artifacts for software development: a methodology. Proceedings of the 6th International Workshop on Modeling in Software Engineering (MiSE 2014). ACM, New York, NY, USA, 7-12.
B. Hu, K. He, P. Liang and R. Li (2010). REMO: A RGPS-based Requirements Modeling Process for Service Oriented Architecture. *Proceedings of Sixth International Conference on Networked Computing and Advanced Information Management (NCM),* IEEE Computer Society, pp. 97–102, Seoul.
B. Lee & B. R. Bryant (2002). Contextual Natural Language Processing and DAML for Understanding Software Requirements Specifications. COLING 2002: The 19th International Conference on Computational Linguistics. Taipei, Taiwan.
B. Lee (2003). Automated Conversion from a Requirements Document to an Executable Formal Specification Using Two-Level Grammar and Contextual Natural Language Processing. Ph.D. Dissertation. The University of Alabama at Birmingham. USA: Alabama. pp. 94-108.
C. Kray and A. Blocher (1999). Modeling the basic meaning of path relations. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, pp. 384–389.
C. Kray and R. Porzel (2000). Spatial cognition and natural language interfaces in mobile personal assistants. In Proceedings of the ECAI 2000 Workshop on Artificial Intelligence in Mobile Systems. Berlin, Germany.
C. M. Zapata and F. Arango (2007). Un ambiente para la obtención automática de diagramas UML a parir de un lenguaje controlado. Vol. 74, No. 153, pp. 223-236. Available in: http://redalyc.uaemex.mx/pdf/496/49615309.pdf
C. M. Zapata, G. L. Giraldo and J. E. Mesa (2010). A Proposal of Meta-Ontology for Requirements Elicitation, *Ingeniare. Revista chilena de ingeniería*, Vol. 18, No. 1, pp. 26-37.
C. M. Zapata (2010). Computational linguistics for helping requirements elicitation: a dream about automated software development. Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas (YIWCALA '10). CA: Los Angeles. pp. 117-124.
C. M. Zapata, A. Gelbukh and F. Arango (2007). A Novel CASE Tool based on Pre-Conceptual Schemas for Automatically Obtaining UML Diagrams, Proc. of Revista Avances en Sistemas e Informática. Vol. 4. pp. 117-124.
C. M. Zapata, C. Palacio and N. Olaya (2007). UNC-ANALISTA: Hacia la Captura de un Corpus de Requisitos a Partir de la Aplicación del Experimento Mago de Oz. Revista EIA, (7), 25-40.
C. Vegega, H. Amatriain, P. Pytel, F. Pollo-Cattaneo, P. Britos and R. Garcia-Martinez (2012). Formalización de Dominios de Negocio basada en Técnicas de Ingeniería del Conocimiento para Proyectos de Explotación de Información. Proceedings IX Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. pp. 79-86.
C. Videira and A. Rodrigues Da Silva (2005). Patterns and metamodel for a natural-language-based requirements specification language. Proceedings of the CaiSE'05 Forum. Portugal: Porto. pp. 189-194.
Ch. Kop and H. C. Mayr (2002). A User Centered Approach to Requirements Modeling. In: Modellierung in der Praxis - Modellierung für die Praxis. M. Glinz, G. Müller-Luschnat (Hrsg.).
D. de Almeida and A. Rodrigues (2008). The ProjectIT-Studio/Requirements Case Tool - A Practical Requirements Specification Case Study. Proceedings of the CISTI 2008 /3ª Conferencia Ibérica de Sistemas y Tecnologías de la Información. España: Ourense. pp. 12.
D. de Almeida and A. Rodrigues (2008). A requirements specification case study with ProjectIT-studio/requirements. Proceedings of the 2008 ACM symposium on Applied computing (SAC '08). Brazil: Ceará. pp. 656-657.
D. de Almeida and A. Rodrigues (2009). A Controlled Natural Language Approach for Integrating Requirements and Model-Driven Engineering. Proceedings of the 2009 Fourth International Conference on Software Engineering Advances (ICSEA '09). Portugal: Porto. pp. 518-523.
D. Deeptimahanti and R. Sanyal (2011). Semi-automatic generation of UML models from natural language requirements. Proceedings of the 4th India Software Engineering Conference (ISEC '11). India: Thiruvananthapuram. pp. 165-174.
D. Demitrio (2005). *Framework para Elicitación Automática de Conocimientos*, Tesis Magíster en Ingeniería de software, Facultad de Informática Universidad Nacional de La Plata, Retrieved from http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Daniel%20Demitrio.pdf
D. Popescu, S. Rugaber, N. Medvidovic, D. M. Berry (2008). Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models. Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs, Vol. 53. pp. 103-124.
D. Rösner, B. Grote, K. Hartmann, B. Höfling (1997). From natural language documents to sharable product knowledge: A knowledge engineering approach. Proceeding COLING '94 Proceedings of the 15th conference on Computational linguistics – Vol. 1, pp. 339-343.
D. Viet and A. Ohnishi (2009). Ontology-based Reasoning in Requirements Elicitation. *Seventh IEEE International Conference on Software Engineering and Formal Methods*, 978-0-7695-3870-9, Hanoi.
E. Reiter and R. Dale (1997). Building applied natural language generation systems. Natural Language Engineering, Vol. 3, pp. 57-87.
E. Rubin (2009). *Domain Knowledge Representation in Information Systems*, Ph.D. Thesis, University of British Columbia. Retrieved from https://circle.ubc.ca/handle/2429/15229
F. H. Wang (2005). On Acquiring Classification Knowledge from Noisy Data Based on Rough Set. Expert Systems with Applications, Vol. 29, No. 1.
G. Aston (1997). Enriching the Learning Environment: Corpora in ELT". In Teaching and Language Corpora. Wichmann *et al.* (Eds.). Longman, London.
G. Cabral, A. Sampaio (2008). Formal Specification Generation from Requirement Documents. Electron. Notes Theor. Comput. Sci. Vol. 195. pp. 171-188.
G. Fliedl, C. Kop, H. Mayr, A. Salbrechter, J. Vohringer, G. Weber and C. Winkler (2007). Deriving static and dynamic concepts from software requirements using sophisticated tagging. Data Knowl. Eng. Vol.61, No.3. pp.433-448.
G. Urrego-Giraldo (2005). ABC-Besoins: Une approche d'ingénierie de besoins fonctionnels et non-fonctionnels centrée sur les Agents, les Buts, et les Contextes. Ph.D. Thesis, Universidad Paris 1, Pantéon Sorbona, Francia.

© Bell Manrique-Losada, 2014. All rights reserved.

H. Dalianis (1995). Aggregation in the NL-generator of the Visual and Natural language Specification Tool. Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics (EACL '95). España: Zaragoza. pp. 286-290.
H. Harmain and R. Gaizauskas (2000). CM-Builder: An Automated NL-based CASE Tool. Proceedings of the fifteenth IEEE International Conference on Automated Software Engineering (ASE'00), Grenoble.
H. Kaiya and M. Saeki (2006). Using Domain Ontology as Domain Knowledge for Requirements Elicitation. *Proceedings of the 14th IEEE International Req. Engineering Conference,* pp. 189-198, Minessota USA.
H. Meth, Y. Li, A. Maedche and B. Mueller (2012). Advancing task elicitation systems – an Experimental evaluation of design Principles. Proceeding of 33 International Conference on Information Systems, Orlando Florida.
H. Yang, A. Willis, A. De Roeck, B. Nuseibeh (2010). Automatic detection of nocuous coordination ambiguities in natural language requirements. Proceedings of the IEEE/ACM international conference on Automated software engineering (ASE '10). USA: California. pp. 53-62.
I. Bajwa, M. Lee, B. Bordbar (2010). SBVR Business Rules Generation from Natural Language Specification. Proceedings of the 2011 AAAI Spring Symposium Series. USA: California. pp. 2-8.
I. S. Bajwa, M. Lee and B. Bordbar (2011). SBVR Business Rules Generation from Natural Language S pecification. Association for the Advancement of Artificial Intelligence. Artificial Intelligence for Business Agility — Papers from the AAAI 2011 Spring Symposium (SS-11-03)
J. Burg and R. Riet (1997). The Impact of Linguistics on Conceptual Models: Consistency and Understandability. Data and Knowledge Engineering, Vol. 21, No.1.
J. C. Leite and A. P. Franco (1993). A Strategy for Conceptual Model Acquisition in IEEE Computer Society Press. Proceedings of the IEEE International Symposium on Requirements Engineering, pp. 243-246
J. Cybulski and K. Reed (1998). Requirements Classification and Reuse: Crossing domains boundaries. 6th Intl. Conf. on Software Reuse (ICSR'2000). Vienna, Austria. Springer.
J. D. Young and A. I. Antón (2010). A Method for Identifying Software Requirements Based on Policy Commitments. 18th IEEE International Requirements Engineering Conference. Available in: http://ieeexplore.ieee.org.ezproxy.unal.edu.co/stamp/stamp.jsp?tp=&arnumber=5636634
J. Hao, Y. Yan, L. Gong, G. Wang and J. Lin (2014). Knowledge map-based method for domain knowledge browsing, Decision Support Systems, Volume 61, pp. 106-114.
J. Hoppenbrouwers, W. Van den Heuvel, S. Hoppenbrouwers, H. Weigand and O. De Troyer, (1999). The Grammalizer: A CASE Tool Based on Textual Analysis. Infolab. Tilburg University, Netherlands.
J. Natt, B. Regnell, P. Carlshamre, M. Andersson, J. Karlsson (2001). Evaluating Automated Support for Requirements Similarity Analysis in Market-Driven Development. Requirements Engineering. Vol. 7. pp. 20-33.
J. Natt, O. Dag, B. Regnell, V. Gervasi, S. Brinkkemper (2005). A Linguistic-Engineering Approach to Large-Scale Requirements Management. IEEE Software. Vol. 22. No. 1. pp. 32-39.
J. Natt, T. Thelin, B. Regnell (2006). An experiment on linguistic tool support for consolidation of requirements from multiple sources in market product development. Empirical Software Engineering. Vol. 11. pp. 303-329.
J.Cleland-Huang, W. Marrero and B. Berenbach (2008). Goal-Centric Traceability: Using Virtual Plumblines to Maintain Critical Systemic Qualities. IEEE Transactions On Software Engineering, VOL. 34, No. 5.
J.L. de la Vara, D. Anes and J. Sánchez (2007). Construcción de Modelos de Requisitos a partir de Modelos de Procesos y de Metas. Memorias de la X Conferencia Iberoamericana de Software Engineering, Marguerita, Venezuela.
J.L. de la Vara, M.H. Fortuna, J. Sánchez, C.M. Werner, M. Borges (2009). Modelado de Requisitos de Datos para Sistemas de Información basados en Procesos de Negocio. XII Conferencia Iberoamericana de Ingeniería de Requisitos y Ambientes de Software (CIbSE 2009), pp 43-57.
K. Hinge, A. Ghose, G. Koliadis (2009). Process SEER: A Tool for Semantic Effect Annotation of Business Process Models. Proceedings of the IEEE Int. Enterprise Distributed Object Computing. New Zealand: Auckland. pp. 54-63.
L. Mich (1996). NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA. Nat. Lang. Eng. Vol. 2. 161-187.
L. Rao, G. Mansingh and K. Osei-Bryson (2012). Building ontology based knowledge maps to assist business process re-engineering, Decision Support Systems, Volume 52, Issue 3, pp. 577-589.
L.Wei, H. Ke-Qing, Z. Kui and W. Jian (2008). Combining Domain-Driven Approach with Requirement Assets for Networked Software Requirements Elicitation. *Proceedings of IEEE International Conference on Semantic Computing*, IEEE Computer Society, pp. 354-36, Washington, DC.
Lévy et al. [34] present an environment that enables semantic annotations of the document textual units (e.g. words, phrases, and paragraphs) with ontological ones (i.e. concepts, instances).
M. Antonetti and C. Miglio (2009). CASER: Herramienta para modelado conceptual. Tesina de Licenciatura en Informática. Universidad Nacional de La Pampa –Argentina. 2009. Available in: http://revista.info.unlp.edu.ar/tesinas/tesis38.pdf (sep-2010).
M. Elbendak, P. Vickers and N. Rossiter (2011). Parsed use case descriptions as a basis for object-oriented class model generation. J. Syst. Softw. Vol. 84. pp. 1209-1223.
M. Gea, F. Gutiérrez, J. Garrido and J. Cañas (2003). Teorías y Modelos Conceptuales para un Diseño basado en Grupos. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Granada, España. Available in: http://lsi.ugr.es/~mgea/invest/articulos/gea_interacc03.pdf
M. Gordon and D. Harel (2009). Generating Executable Scenarios from Natural Language. Proceedings of the 10th International Conference on Comp. Linguistics and Intelligent Text Processing. México: DF. pp. 456-467.
M. Osborne and C. MacNish (1996). Processing Natural Language Software Requirement Specifications. Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96). pp. 229.
M. Stuit and H. Wortmann (2012). Discovery and analysis of e-mail-driven business processes, Information Systems, Volume 37, Issue 2, pp. 142-168.
M. Sullabi and Z. Shukur (2008). SNL2Z: Tool for Translating an Informal Structured Software Specification into Formal Specification. American Journal of Applied Science. Vol. 5. pp. 378-384.
M. Van Welie and G. Van der Veer (1998). An ontology for Task World Models. En: 'Design, Specification and Verification of Interactive System'98. Springer Computer Science.

© Bell Manrique-Losada, 2014. All rights reserved.

N. Aussenac-Gilles, B. Biébow and S. Szulman (2000). Revisiting Ontology Design: A Method Based on Corpus Analysis. In Knowledge Engineering and Knowledge Management. Methods, Models, and Tools: 12th International Conference (EKAW), Vol. 1937, pp. 27–66, Juan-les-Pins, France.
N. Calzolari (1997). Lexicon and Corpus: A Multi-faceted Interaction. Cicle de Conferències 95-96. Lèxic, Corpus i Dictionaris. Barcelona: Institut Universitari de Lingüística Aplicada.
N. Dinesh, A. Joshi, I. Lee and B. Webber (2006). Extracting formal specifications from natural language regulatory documents. In: ICoS-5, Buxton, England.
O. Ormandjieva, I. Hussain, L. Kosseim (2007). Toward a text classification system for the quality assessment of software requirements written in natural language. Fourth international workshop on Software quality assurance: in conjunction with the 6th ESEC/FSE joint meeting (SOQUA '07), pp. 39-45. Dubrovnik, Croatia.
P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos and A. Perini (2004). *TROPOS: An Agent-Oriented Software Development Methodology*. Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers. Vol. 8, No. 3, pp. 203-236.
P. Chen (1976). The Entity-Relationship Model: Toward a unified view of data. ACM Transactions on Database Systems, Vol. 1, No. 1, pp. 9-36.
P. Chen (1983). English sentences structures and Entity-Relationship Diagrams. Information Science. Vol. 2, No. 29, pp. 127-149.
P. Clements and L. Northrop (2002). Software Product Lines: Practices and Patterns. Boston, MA: Addison-Wesley
P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone (2005). Modeling Security Requirements through Ownership, Permission and Delegation. 13th IEEE International Requirements. Engineering Conference, pp. 167-176.
P. Jain, K. Verma, A. Kass, R. G. Vasquez (2009). Automated review of natural language requirements documents: generating useful warnings with user-extensible glossaries driving a simple state machine. Proceedings of the 2nd India software engineering conference (ISEC '09). India: Pune. pp. 37-46.
P. Loucupoulos and E. Kavakli (1995). Enterprise Modelling and Teleological Approach to Requirements Engineering. International Journal of Cooperative Information Systems, Vol. 4, No. 1, pp. 45-79.
P. O'Shea and C. Exton (2004). The Application of Content Analysis to Programmer Mailing Lists as a Requirements Method for a Software Visualization Tool. 12th Internat. Works. on Software Technology Practice, pp. 30–39.
P. Pytel, H. Ramón, M. Tomasello, P. Britos, M. F. Pollo, C. Vegega, D. Rodriguez and R. García-Martínez (2011). Propuesta de Aplicación de Técnicas de Representación de Conocimiento en el Análisis de Requisitos Software. Actas 1er Seminario Argentina-Brasil Tecnologías de Información y la Comunicación. pp. 41-50.
Ph. Ramadour and C. Cauvet (2002). Approach and Model for Business Components Specification. DEXA Database and Expert Systems Applications, 13th International Conference, pp. 628-637.
R. Cavada, A. Cimatti, A. Mariotti, C. Mattarei, A. Micheli, S. Mover, M. Pensallorto, M. Roveri, A. Susi, S. Tonetta (2009). Supporting Requirements Validation: The EuRailCheck Tool. Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering. Aukland: New Zealand. pp. 665-667.
R. Darimont, E. Delor, P. Massonet, A. van Lamsweerde (2005). GRAIL/KAOS: An Environment for Goal-driven Requirements Engineering." IEEE 19th International Conference on Software Engineering, pp. 612-613.
R. Krishnamurthy (1997). Keeping Good Company: Collocation, Corpus and Dictionaries. Cicle de Conferències 95-96. Lèxic, Corpus i Dictionaris. Barcelona: Institut Universitari de Lingüística Aplicada.
R. Lecoeuche, C. Mellish and D. Robertson (1998). A Framework for Requirements Elicitation through Mixed-Initiative Dialogue. Proceedings of the 3rd International onference on Requirements Engineering: Putting Requirements Engineering to Practice (ICRE '98). USA: Colorado Springs. pp. 190.
R. Schwitter and N. Fuchs (1996). Attempto - From Specifications in Controlled Natural Language towards Executable Specifications. Proceedings of the 3rd Knowledge Acquisition for Knowledge-Based Systems Workshop. Francia: Tolouse. Vol. 15. pp.377-394.
S. Nanduri, S. Rugaber (1995). Requirements validation via automated natural language parsing. Proceedings of the 28th Hawaii International Conference on System Sciences. USA: Hawaii. pp. 362-368.
S. Overmyer, B. Lavoie and O. Rambow (2001). Conceptual modeling through linguistic analysis using LIDA. Proceedings of ICSE, Toronto, Canada.
S. Spreeuwenberg, K. Anderson (2010). SBVR's approach to controlled natural language. Lectures Notes in Computer Science. Vol. 5972, pp. 155-169.
T. Breaux, A. Antón and J. Doyle (2008). Semantic parameterization: A process for modeling domain descriptions. ACM Trans. Softw. Eng. Methodol. Vol. 18. pp. 27.
T. D. Breaux, M.W. Vail and A.I. Antón (2006). Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations. North Carolina State University Computer Science Technical Report 2006-6. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.175
T. da Cunha, F. M. Santoro, K. Revoredo, V. Tavares (2014). A formal representation for context-aware business processes, Computers in Industry, Volume 65, Issue 8, pp. 1193-1214.
T. de Sousa, J. Almeida, S. Viana and J. Pavón (2010). Automatic analysis of requirements consistency with the B method. SIGSOFT Software Engineering Notes. Vol. 35. pp. 1-4.
U. Hahn, M. Klenner and K. Schnattinger (1996). A Quality-Based Terminological Reasoning Model for Text Knowledge Acquisition. Advances in Knowledge Acquisition. Shadbolt, O'Hara & Schreiber (Eds.). Springer-Verlag, Berlin.
V. Ambriola, V. Gervasi (2006). On the Systematic Analysis of Natural Language Requirements with CIRCE. Automated Software Engg. Vol. 13. pp. 107-167.
V. Gervasi and D. Zowghi (2005). Reasoning about inconsistencies in natural language requirements. ACM Trans. Softw. Eng. Methodol. Vol. 14. pp. 277-330.
V. Tavares, F. M. Santoro, M. R.S. Borges (2009). A context-based model for Knowledge Management embodied in work processes, Information Sciences, Volume 179, Issue 15, pp. 2538-2554.
W. Black (1987). Acquisition of conceptual data models from natural language descriptions. Proceedings of the 3th European chapter of Association for Computational Linguistics. Dinamarca: Copenhagen. pp. 241-248.
Y. Ko, S. Park, J. Seo and S. Choi (2007). Using classification techniques for informal requirements in the requirements analysis-supporting system. Inf. Software Technology. Vol. 49. pp. 1128-1140.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 4

FRAMEWORK FOR THE FORMALIZATION

It had begun with a leaf caught in the wind, and it became a tree; and the tree grew, sending innumerable branches, and thrusting out the most fantastic roots.
-- J. R. R. Tolkien, Leaf by Niggle

We present the framework for the proposal, in terms of the elicitation technique which is justified in this Ph.D. Thesis and the foundations and assumptions for the mapping formalization. Such foundations are related to the Discourse Analysis, Corpus Linguistics, and business-based technical documents.

4.1 CHARACTERIZATION OF THE ELICITATION TECHNIQUE

In Section 2.1.3 we present the existing RE techniques, and we describe how commonly the RE process is made by using a set of well-known techniques (*e.g.* interviews, scenarios analysis, and observation). Even though several researchers have increased the effort for identifying, considering, and analyzing several sources of possible requirements throughout the RE process (Islam *et al.*, 2010), eliciting requirements from sources without direct intervention of stakeholders is not as common as the other techniques.

4.1.1 Referents of the Elicitation Technique

In this Ph.D. Thesis we propose an approach for using business-based technical documents as a source for eliciting requirements, *i.e.,* an approach for using analytical techniques in the context of the RE process. This proposal is supported by the usefulness and convenience of systematically put into practice approaches of the Background Reading (BR) technique (Zhang, 2007). BR is also known as *document review* (Kotonya & Sommerville, 2004; Cremers & Alda, 2005), *document analysis* (Hossenlopp & Hass, 2007), *Content Analysis* (Krippendorf. 1980), and *documentation studies* (Stein *et al.*, 2009). '*Documentation studies'* is a superior category involving the previous one.

BR is a technique coming from qualitative research (Bowen, 2009), in which documents are interpreted from an assessment topic. This discipline is usually called *document analysis* (Guest *et al.*, 2013). As a common thread throughout most of the methods and techniques of qualitative research, which have an inductive and flexible nature, Silverman (2011) describes document analysis as a deductive technique used as a data collection method.

As specialized analytic approach, the usage of documents is called *content analysis*. This approach is more spread because it includes any form of communication (*e.g.* newspapers, emails, pictures, speeches, etc.), including written documents. Traditionally, content analysis has been considered as an objective and neutral way for obtaining a quantitative description of the content of several forms of communication and a method for describing and interpreting the artifacts of a context (Marshall & Rossman, 2006). Template analysis usually is related to content analysis (King, 2004), but it is oriented to the codification of text segments for generating quantitative data for statistical analysis.

Document review is often oriented to document analysis, but some differences between them are identified. According to Letts *et al.* (2007), document review is often used in historical research, which involves the study of past events. By using the document review technique, a set of documents considering the creation context are reviewed, following flexible methods for learning how past events are related to a topic of interest.

Generally speaking, in qualitative research document analysis incorporates coding content into subjects similar to the analysis developed in other methods such as focus group or interview analysis. The

© Bell Manrique-Losada, 2014. All rights reserved.

documents in such kind of research, as Merriam (2009) proposes, are catalogued as *public record documents*—artifacts comprising records of an organization activities; *personal documents*—artifacts describing actions, experiences or beliefs of individuals in first-person; and *physical evidence documents*—artifacts related to physical objects found within the study.

4.1.2 Background Reading

In the context of the RE process, BR is closer to the document analysis trend. BR is an analytical method which uses existing documents as a source for the RE process. BR is mostly used in business process analysis (BPA) for gathering and studying existing documentation and other relevant information. By considering such documentation, an analyst gathers details of actual processes, existing solutions, and business regulations. This technique is also used to understand the overall culture in an organization or project (Zhang, 2007). According to the BABOK® Guide[5], performing BR is one of the required techniques in the fundamental knowledge base of an effective business analyst. For conducting BR in BPA, the business analyst follows three activities: preparation, review of concrete document and wrap-up. Preparation involves locating and evaluating the relevant business documentation for a given purpose. With document review we study the documents and identify the relevant information. Wrap-up is one step for reviewing and confirming the identified and extracted information.

Several techniques are available for business requirements elicitation, depending on the size and scale of the project or analysis purpose. Reviewing existing documents (*e.g.* Procedures, requirements documents, proposals, etc.) related to a process or system and valuating such documentation can be useful for business analyst to complete BPA processes. After reading and understanding the current process or system, the analyst may continue with the following analysis steps.

In terms of Cremers and Alda (2005), by using BR an analyst can reach an understanding about: i) how the systems interact and contribute to overall business goals; ii) the needs and constraints of system stakeholders; iii) the specific needs of people requiring support in their work; and, iv) knowledge of the general area or application domain. In this way, BR is used to gather requirements during the RE process. It regards the act of reviewing the existing documentation of comparable business processes or systems in order to extract pieces of information relevant to the current project. Therefore, BR should be considered as project requirements acquisition.

Analysts can elicit requirements in many ways, and eliciting them from stakeholders—*e.g.,* by using questionnaires, interviews, and facilitating sessions, among other—is the most common way. Techniques as BR are particularly valuable for eliciting information and requirements from non-human sources, which can be relevant for supporting the current techniques. BR could be necessary when stakeholders are not available to offer insights into existing business processes or systems. Documentation can be analyzed for understanding the key domain functions and their business rules, entities, and attributes.

Some kinds of documents could be useful in the application of the BR technique. A group of prospective documents for being considered and reviewed by analysts are:

- Corporate-level documents and business documentation, *i.e.*, business process and procedure documents
- Corporative plans, *i.e.* annual reports, strategic plans, business plans
- Documents circulating on Intranet and internal communication, *i.e.*, Company memos, Requests for proposals
- Research documents, *i.e.* white papers, benchmarking studies
- Technical standards, training guides, and guidelines
- Help desk reports and logs
- Existing system specifications and vision documents for related projects

In the state of the art we can find several approaches for implementing background reading, among them:

[5] http://www.iiba.org/babok-guide.aspx

© Bell Manrique-Losada, 2014. All rights reserved.

- By using NLP techniques for extracting knowledge when applying the BR technique: Aussenac-Gilles *et al.* (2000) in the ontological engineering research domain and Fliedl *et al.* (2007) for analyzing requirements documents.
- Stein *et al.* (2009) present a technique for acquiring requirements from existing texts by analyzing texts, documenting the acquired requirements and formulating competence questions.
- O'Shea and Exton (2004) use BR for extracting requirements from a set of texts like bug reports. They use predefined categories of documents, mostly communications as interview transcripts.
- King (1998) proposes an intertwined iterative process where a code template is created based on a corpus. This technique requires high intervention of analyst, because the coding process is manually made, usually supported by qualitative data analysis tools.

Although several authors support the value of using BR, we base the approach of this Ph.D. Thesis on the fact that the state of the art there is no evidence about specific methods for applying such a technique (Rayson *et al.,* 1999; Stein *et al.,* 2009; Zapata *et al.*, 2012). We consider the BR technique—as most of the analytic methods—a supporting complementary technique to the RE process. BR can be complementary for improving the efficiency and effectiveness of the domain knowledge identification and understanding. Thus, BR is coherent and potentially combinable with other methods (Amber *et al.* 2011).

4.2 FOUNDATIONS AND ASSUMPTIONS FOR THE MAPPING FORMALIZATION

The linguistic foundations for the mapping formalization and the assumptions of our approach, are presented in this Section.

4.2.1 Linguistics Foundations and Assumptions

Based on several areas within the Linguistics discipline, we follow in this Ph.D. Thesis some knowledge perspectives as foundations for our approach and contribution. Such perspectives are related to: i) Discourse Analysis and ii) Corpus linguistics, as follows.

Related to Discourse Analysis

From discourse analysis point of view, we follow the standpoint of specialized discourse analysis proposed by Nickerson (1999), Cabré (1999), and Biber (2006). Specifically, our approach is based on the analysis stated by Cademártori *et al.* (2007) and Parodi *et al.* (2009) about professional and specialized written discourse. In this context, we continue the line of the analysis of such specialized discourses from the genre point of view (Nickerson 1999). Our approach for analyzing discourses comes from the approaches proposed by van Nus (1999) and Swales (1990; 2004), according to text structures and their contexts (van Dijk, 2008).

We implement a corpus linguistic methodology for defining a corpus for business-based technical documents, in four sub-categories, as we present in the next Section. Such methodology follows the empirical principles by Biber *et al.* (1998) and Kennedy (2000), as well as the systematic and practical principles of Parodi (2008).

Regarding the applied method of rhetorical analysis, we act according to Venegas (2013), based on the approach of Askehave and Swales (2001) for analyzing texts in terms of rhetorical structures, and the usage of moves situated at macro or micro levels (Heuboeck, 2009). Specifically, related to the business-based documents as genre, we consider the proposal of Parodi (2010) for the genre 'manual.'

Related to the Corpus

As resulted from the methodology application, we develop a domain-specific annotated corpus. Such corpus is based on the approach suggested by Parodi *et al.* (2009) and The GRIAL program[6] for working with academic and professional corpus. Likewise, we follow some trends similar to the GENIA project (Tateisi *et al.*, 2005) and BioIE project (Kulick *et al.*, 2004).

[6] www.elgrial.cl

© Bell Manrique-Losada, 2014. All rights reserved.

Based on such referents and projects, we define standard benchmarks for building the corpus and defining the featured basis of the NLP patterns for the mapping. Since the aforementioned projects have produced their own corpora, we define a set of sub-corpus for sample and experimentation with the domain-specific to business-based technical documentation. As might be expected, the corpora are smaller and more closely-tailored than those which result from dedicated corpus annotation projects. However, we make possible the identification and coding of the annotation rules and patterns, which was useful for the experimentation stage and its related documentation.

4.2.2 Foundations and assumptions of Business-based Technical Documents

Document is a term referring to an extensive range of visual, written, digital, and physical material relevant to the study at hand. While the most common use of documents is as a source of content, some researchers have been working for analyzing the function of documents in the society and its role as a social actor (Prior, 2008). Documents represent some way of communication (Merriam, 2009) and they are categorized in several forms. From qualitative research come two categories of documents: *public records* and *personal documents*. Authors like Bogdan and Biklen (2003) state the *popular culture documents* and *visual document* categories. Given the intersection among several kinds of documents in such categories, the same document can be classified in more than one way.

According to Guba (1981), *Public Records* (PR) include all kind of official record related to events happened in a society or context. Based on the context and area under study, the public documents can include different instances. For example in government, PR include police records, government documents, and association manuals, among others; in education they include educational statistics, educational program records, and individual program records, among others.

Personal Documents refer to any first-person describing experiences, actions, or beliefs from an individual point of view (Bogdan & Biklen, 2007). They include familiar records, autobiographies, travel logs, and so on. *Popular culture documents* include books, advertisements, movies, or related kinds of mass-consumed materials.

In this Ph.D. Thesis a business-based technical document is a kind of document catalogued as a public record, because it belongs to official documents produced by an organization (*e.g.* institutions, schools, government, enterprises, etc.) for using it internally or for an external dissemination. Such technical documents are artifacts comprising records of an organization activity. Documents are like a ready-made source of data easily accessible for many purposes—depend on its nature, usage, and context, as we describe in the previous paragraphs. We use the technical term 'business-based technical document' to refer a source of documentation comprising information about business information from an organization, *i.e.*, procedures, practices, functionalities, regulations, policies, rules, and statutes, among others.

Several kinds of technical documents are used in organizations. We can cite manuals (*e.g.* procedure manuals, quality standards, and user guidelines), statutes, and legacy documents. Based on the validation studies developed in our research, we emphasize in the manuals as a kind of business-based technical document. A *manual* is a set of written instructions describing how procedures are defined, developed, and managed by the members of an organization. These procedures involve technical, administrative, and operational activities. According to Parodi (2009), manuals are strongly oriented to disciplinary knowledge with a didactic character, looking for the spread or apprehension of a particular knowledge. Managing policy documents, administrative documents, or corporate technical documents has been recently studied by several researchers (Dinesh *et al.*, 2007; 2008).

Procedural information is the most important information type included in procedure manuals (Karreman & Steehouder, 2003). People read instructions because they want to know what actions they should execute in order to complete their activities. According to Ummelen (1997), procedural information comprises actions, conditions for actions, and results from actions.

The most relevant sub-corpus for the empirical studies includes instances of a kind of manual highly oriented to procedures—*Standard Operating Procedures* (SOP), named after the most used name for it. A SOP aims to explain either the structure/operation of a device (*e.g.*, engine), or the operation of a process (e.g., medical treatment). These documents contain information primarily declarative, in terms of

© Bell Manrique-Losada, 2014. All rights reserved.

definitions and descriptions of system components, their relationships, the general principles of operation, and the context in which it is used.

Several authors recognize the relevance to use SOPs in organizations, since SOP and other related technical documents have been used for supporting servicing operations (Kemeny, 1996). Also, they define the requirements of every function and department, so that management and staff can use them for achieving quality servicing. Research or studies similar to this where SOP has been used with purposes of identification of business information are not common at alone the RE discipline.

Based on such premises, we decide working with such kind of business-based technical document and contribute to the initiative of using them for identifying and understanding the context of an organization. As we present in Chapters 5 and 6, we consider significant to carry out work on the related business-based technical document—according to the genre structure of Section 6.1.3—, since it is possible find more relevant information than by using SOPs as the source document.

We devise a conceptual structure by genre, based on each sub-corpus, by considering the purposes necessary for being defined, written, and used efficiently, under the assumption that communicative purpose of the author is embodied in the analyzed texts. We are aware that the sub-categories can appear in the spectrum of the genres—specifically of the manual macro-genre, Procedure Manual genre, and Job Standard Document genre—as a result of further exploration of texts and a more inclusive and representative corpus. Also, the reader should notice that hyperlinks among sub-categories within each genre and macro-genre can occur.

REFERENCES

A. Amber, I. S. Bajwa, M. S. Naweed and T. Bashir (2011). Requirements Elicitation Methods. IEEE 2nd International Conference on Mechanical, Industrial, and Manufacturing Technologies (MIMT 2011) pp. 541-545.

A. B. Cremers and S. Alda (2005). Organizational Requirements Engineering. Chapter 5 Requirements Elicitation I. Bonn University. Available in: http://www.iai.unibonn.de/III/lehre/vorlesungen/SWT/RE05/slides/05_Requirements%20Elicitation%201-2.pdf

A. Heuboeck (2009). Some Aspects of Coherence, Genre and Rhetorical Structure – and their Integration in a Generic Model of Text. In: Language Studies Working Papers: O'Brien & Giannoni, Editors. Vol. 1, pp.35-45.

C. Marshall and G. B. Rossman (2006). Data Collection Methods. In: Designing Qualitative Research. Sage Publishers, pp.97-150, SAGE Publications, London.

C. Nickerson (1999). The usefulness of genre theory in the investigation of organizational communication across cultures. Document Design, Vol.1, No. 3, pp. 203–215.

C.M. Zapata, B. Manrique-Losada and G. González-Calderon (2012). An approach for using procedure manuals as a source for Requirements Elicitation. In Proceedings of CLEI 2012, Vol. 1, pp. 1-8.

D. Biber, S. Conrad and R. Reppen (1998). Corpus linguistics: Investigating language structure and use. Cambridge University Press, Cambridge.

D. Silverman (2011). Collecting Qualitative Data. In: Qualitative Research. Third Edition. Sage Publications, London.

E. G. Guba (1981). Educational Communication and Technology, Vol. 29, No. 2, pp. 75-91.

G. A. Bowen (2009). Document Analysis as a Qualitative Research Method. Qualitative Research Journal, Vol. 9, Issue 2, pp. 27

G. Fliedl, C. Kop, H.C. Mayr (2007). Deriving static and dynamic concepts from software requirements using sophisticated taggingstar. Open Data & Knowledge Engineering, Vol. 61, No. 3, pp. 433–448.

G. Guest, E. E. Namey and M. L. Mitchell (2013). Additional Qualitative Data Collection Methods. In: Collecting Qualitative Data. A Field Manual for Applied Research. SAGE Publications, London.

G. Kennedy (2000). An introduction to corpus linguistics. Foreign Language Teaching and Research Press.

G. Kotonya and I. Sommerville (2004). Requirements Engineering: Processes and Techniques. Wiley Editors. Chichester, UK.

G. Parodi (2008). Lingüística de corpus: una introducción al ámbito. Revista de Lingüística Teórica y Aplicada, Vol.46 (1), pp. 93-119.

G. Parodi (2009). El Corpus Académico y Profesional del Español PUCV-2006: semejanzas y diferencias entre los géneros académicos y profesionales. Estudios Filológicos, No. 44, pp. 123-147. Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

G. Parodi (2010). La organización retórica del género manual a través de cuatro disciplinas: ¿cómo se comunica y difunde la ciencia en diferentes contextos universitarios?. Boletín de Lingüística, Vol. 22, No. 33, pp. 43-69.

G. Parodi, R. Ibañez, R. Venegas and C. González (2009). Working with the Spanish Corpus PUCV-2006: Academic and professional genre identification and variations. Proceedings of the Corpus Linguistics Conference, Liverpool, UK.

I. Askehave and J. Swales (2001). Genre identification and communicative purpose: a problem and a possible solution. AppliedLinguistics vol. 22(2), pp. 195-212.

© Bell Manrique-Losada, 2014. All rights reserved.

J. Karreman and M. Steehouder (2003). Effects of declarative information in instructions for use. IEEE International Professional Communication, Vol. 1, pp. 29-33.
J. Kemeny (1996). Check the manual. Policies and procedures manuals for servicing operations. Mortgage Banking, Vol. 57, p71-74.
J. Swales (1990). Genre analysis: English in academic and research settings. Cambridge University Press Cambridge.
J. Swales (2004). Research genres. Exploration and applications. Cambridge University Press, Cambridge.
K. Krippendorf (1980). Content Analysis: An Introduction to its Methodology. Beverly Hills. Sage Publications, CA.
L. Letts, S. Wilkins, M. Law, D. Stewart, J. Bosch and M. Westmorland (2007). Guidelines for Critical Review Form: Qualitative Studies (Version 2.0). Available in: http://www.canchild.ca/en/canchildresources/resources/qualguide.pdf
L. Prior (2008). Researching Documents: Emergent Methods. In: Handbook of emergent methods. Hesse-Biber & Leavy (Eds.), pp. 111-126. Guilford Press, New York.
M. T. Cabré (1999). La Terminología: Representación y Comunicación. Elementos para una teoría de base comunicativa y otros artículos. Barcelona: IULA Universidad Pompeu Fabra.
M. van Nus (1999). Business genres and their corporate con-text. Journal of Research and Problem Solving in Organizational Communication: Document Design, 1(3), 187-197.
N. Aussenac-Gilles, B.Biébow and S. Szulman (2000). Revisiting Ontology Design: A Method Based on Corpus Analysis. Knowledge Engineering and Knowledge Management. Methods, Models, and Tools, 1937, pp.27–66.
N. Dinesh, A. Joshi, I. Lee and O. Sokolski (2007). Logic-based regulatory conformance checking. 14th Monterey Workshop, Scholarly Commons, Penn University.
N. Dinesh, A. Joshi, I. Lee and Sokolski (2008). Reasoning about conditions and exceptions to laws in regulatory conformance checking. Proceedings of DEON 2008, pp. 16.
N. King (1998).Template Analysis. In: Qualitative Methods and Analysis in Organizational Research: A Practical Guide. Symon & Cassel (Eds.), pp. 118–134. Sage Publications, London.
N. King (2004). Using templates in the thematic analysis of text. In: Essential guide to qualitative methods in organizational research. pp. 256-270, SAGE Publications, London.
N. Ummelen (1997). Declarative information in software manuals: what's the use?. Proceedings of the 15th annual international conference on Computer documentation (SIGDOC '97), ACM, pp. 283-296, New York.
P. O'Shea and C. Exton (2004). The Application of Content Analysis to Programmer Mailing Lists as a Requirements Method for a Software Visualization Tool," in 12th International Workshop on Software Technology Practice (STEP), pp. 30– 39, Los Alamitos, California.
P. Rayson, R. Garside and P. Sawyer (1999). Language engineering for the recovery of requirements from legacy documents. Technical report, REVERVE project report, Lancaster University.
R. Bogdan and S. K. Biklen (2007). Qualitative research for education: An introduction to theories and methods. Pearson A & B, Editors. Boston, MA.
R. C. Bogdan and S. K. Biklen (2003). Qualitative Research for Education: An introduction to Theories and Methods (4th ed.). New York: Pearson Education group, pp. 110-120.
R. Hossenlopp and K. B. Hass (2007). Unearthing Business Requirements: Elicitation Tools and Techniques. Management Concepts Inc., Editors.
R. Venegas (2013). Estructura y Propósitos Comunicativos en Tesis de Magíster y Licenciatura Literatura y Lingüística, Vol. 27, pp. 201-218.
S. B. Merriam (2009). Qualitative Research: A Guide to Design and Implementation. John Wiley & Sons, Editors.
E. Islam, S. Knauss, J. Jürjens and K. Schneider (2010). Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UML. Hilde Houmb Requirements Engineering, Vol. 15, No. 1, pp. 63-93.
S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein, L. Ungar, S. Winters and P. White (2004). Integrated Annotation for Biomedical Information Extraction. Proceedings of BioLINK 2004: Linking Biological Literature, Ontologies, and Databases, pp. 61-68.
S. Stein, Y. Lauer and M. El-Kharbili (2009). Using Template Analysis as Background Reading Technique for Requirements Elicitation. Software Engineering, Vol. 143, pp. 127-138.
T. A. van Dijk (2008). Discourse and Context. A Socio-cognitive Approach. Cambridge University Press, Cambridge.
Y. Cademártori, G. Parodi and R. Venegas (2007). El discurso escrito y especializado: Las nominalizaciones en manuales técnicos. In: Lingüística de corpus y discursos especializados: Puntos de mira. G. Parodi, Editor. Ediciones Universitarias de Valparaíso, Chile.
Y. Tateisi, A. Yakushiji, T. Ohta and J. Tsujii (2005). Syntax Annotation for the GENIA Corpus. Companion Volume to the Proceedings of Conference.
Z. Zhang (2007). Effective Requirements Development–A Comparison of Requirements Elicitation techniques. INSPIRE2007, Tampere, Finland.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 5

FORMALIZATION OF THE MAPPING PROCESS

A good conceptual model allows us to predict the effects of our actions. Without a good model we operate by rote, blindly; we do operations as we were told to do them; we can't fully appreciate why, what effects to expect or what to do if things go wrong. As long as things work properly, we can manage. When things go wrong, however, or when we come upon a novel situation, then we need ... a good model.
-- Don A. Norman

We present the proposed mapping formalization by exploring three dimensions, covering the methodological stages of our proposal and the main components to be considered for carrying out the discourse mapping from a technical document into controlled language texts. Such dimensions of mapping formalization are represented in models and materialized in a practical method, as follows: i) Rhetorical Organization Model; ii) Behavioral Model and Mapping Method; and iii) Architectural Model. The description and functionality of each model of our approach for formalizing the mapping process are presented in Section 5.1, Section 5.2, and Section 5.3, respectively.

5.1 RHETORICAL ORGANIZATION MODEL

We present a first approach of a rhetorical organization model (ROM), as a structural and functional pattern for business-based technical documents. By using the method of discourse rhetorical analysis (DRA) described in Section 2.1.4, we define a model as a pattern in terms of functional and structural aspects, based on macro-moves, moves, and functions (communicative purposes). The proposed pattern comprises three macro-moves, which serve an overall communicative purpose, and 19 moves shaping the organization units of the text.

The methodological procedures applied for defining the ROM in this study are divided into stages (Zapata *et al.* 2012), as follows:

1) *Definition of corpus*.
2) *Qualitative characterization of the macro-genre and genre*. This stage is not explained in this Section, since we perform the specific characterization of two case studies corresponding to the two macro-genres: '*standard operation procedure*' and '*job standard document'*. Such descriptions will be presented in the Section 6.1.3.
3) Analysis of the digital corpus.
4) *ROM Design*:
    a. Definition of a reference model for the analysis
    b. Rhetorical analysis method
5) *Evaluation of structural and functional pattern* for a specific technical document.
6) *Description of the Rhetorical Organization Model*

5.1.1 Corpus definition

The corpus definition starts by collecting possible technical documents on the macro-genre circulating on the web. We did not have many restrictions by selecting the texts for building the corpus, because the focus is getting as many samples as possible, but not the entire track rolling stock. Thus, we face the first limitation for deciding what kinds of technical documents collect for building the corpus. We explore broadly 4 types of technical documents: i) *Job Standard Document* (SUB-CORPUS 1); ii) *Corporate Policy Document* (SUB-CORPUS 2); iii) *Functions Manual* (SUB-CORPUS 3); and iv) *Standard Operating Procedure* (SUB-CORPUS 4). In Appendix 5.1 we include the specifications and detail of each text file analyzed and included in the navigation and search of business-based technical documents.

© Bell Manrique-Losada, 2014. All rights reserved.

After the analysis of collected documents from each proof sub-corpus, we select as a referent for the case studies the sub-corpus 1 and 4 for *'Job Standard Document'* and *'Standard Operating Procedures,'* supported in the following reasons:

- Diversity of sources, authors, and organizational purposes of the documents
- Quantity of found documents
- Homogeneity of support and context information in terms of: number of pages, diversity of included information (figures, maps, graphics, lists, and so on).
- Homogeneity of the contents, for facilitating the characterization and discourse analysis from the genre point of view.

The details and characterization from selected corpus are presented as follows.

Standard Operating Procedure

We obtained 50 documents constituting this corpus, as we present in the grid with its specifications in Appendix 5.1. The population selection for the corpus is based on the following criteria:

- Named as 'Standard Operating Procedure' (SOP) or 'SOP Manual'.
- Written in English
- Published online and open-accessed in the Internet
- With an author affiliated to a company/corporation. Also, we just downloaded a document per author.
- Text-based, with a low percentage of images. Not only include those containing images.

A SOP is a constitutive document of a quality system. A SOP is a document describing a set of recurring operations, which are used as a guide to implement them correctly and always in the same way. Frameworks and standards such as ITIL (Information Technology Infrastructure Library) and ISO (International Organization for Standardization) respectively configure their organizational processes by using a basic enforceable set of SOP. SOPs are vehicles used to implement organization policies. A SOP describes procedures defined as segment of business processes and describes how the policies are implemented effectively. We perform a sampling procedure after establishing the criteria for the web search program and the formation of the corpus. The documents in the corpus correspond to 100% of the population, conforming a significant sample, from a proportional estimate sample size (Fernandez, 2000), with a confidence level of 95% precision and an approximate value of 5% the measured parameter. Assuming the population is evenly distributed, we selected a sample of 32 documents, corresponding to 64% of the total population. This is the minimum percentage statistically random, calculated with the Z test of proportions. Based on this sample we conducted the rhetorical analysis.

Job Standard Document

A Job Standard Document (JSD) is a formal document for capturing and defining the work activities and deliverables, oriented to a specified job in an organizational hierarchical structure. A JSD tries to standardize the terminology, concepts, overall agreement terms, conception constraints, and a set of operations for creating, ending, and monitoring jobs, including desirable skills and resources to opt for a job.

We collect and analyze a set of documents from this corporate domain in different subject fields such as medicine, forestry, and laboratory. The corpus used as the basis for the case study comprises 39 English-written documents with independence of its variety. The documents selected are a sample of 25, assuming the population is evenly distributed and were collected following representativeness and ecological criteria, i.e., looking for the collection of documents produced, created, or promoted in the corporate or business environment.

5.1.2 Analysis of the digital corpus

The analysis carried out on the two corpus is related to tokenizing, keyword and stopword identification, characters, and creation of word frequency lists, among others, as we present in Table 5.1 and Figure 5.1. Also, a concordance analysis is carried out looking for identifying lexical forms. Such groups with a

© Bell Manrique-Losada, 2014. All rights reserved.

particular concordance may be related in several ways as a set of terms. A concrete example of these concepts is given below in Figure 5.2.

Table 5.1 Sample of results for word frequency

#Word Types: 9252		
#Word Tokens: 167905		
#Search Hits: 1		
1	12755	the
2	5683	and
3	5201	of
4	4534	to
5	3152	a
6	2598	in
7	2558	for
8	2256	be
9	1953	is
10	1849	or
11	1250	are
12	1156	by
13	1147	with
14	1111	on
15	1092	that
16	1004	project
17	997	as
18	912	will
19	806	should
20	746	this
21	731	all
22	691	at
23	672	if
24	671	an
25	619	not
26	596	must
27	554	information
28	530	staff
29	529	from
30	477	it
31	461	may
32	441	any
33	399	other
34	398	management
35	392	center
36	389	procedures
37	355	system
38	350	have
39	329	use
40	271	process
41	264	operations
42	262	data
43	247	used

I – a - about – an – are – as – at – be – by – com- de – en – for – from – how
In – is – it – la – of – on – or – that – the – this – to – was – what – when – where
Who – will – with – und – the – www – a – about – above – across – after
Afterwards – again – against – all – almost – alone – along – already - also
And – another – any – anyhow – anyone – anything – anyway - anywhere
Are – around – as – at – back – be – became – because – become - becomes
Becoming –been – before – beforehand – behind – being – below - beside
Besides – between – beyond – bill – both – bottom – but – by – call – can - cannot
Cant – co – con – could – couldn't – cry – de – describe – done – down - due
During – each – eg - e.g. – either – eleven – else – elsewhere - enough
Few – fifteen – fifty – fill – find – fire – first – five – for – former - formerly
Forty – four – from – front – full – further – had – hasn't – he - hence
Her – here – hereafter – thereby – therein – thereupon - theirs

Figure 5.1 Example of stopwords

© Bell Manrique-Losada, 2014. All rights reserved.

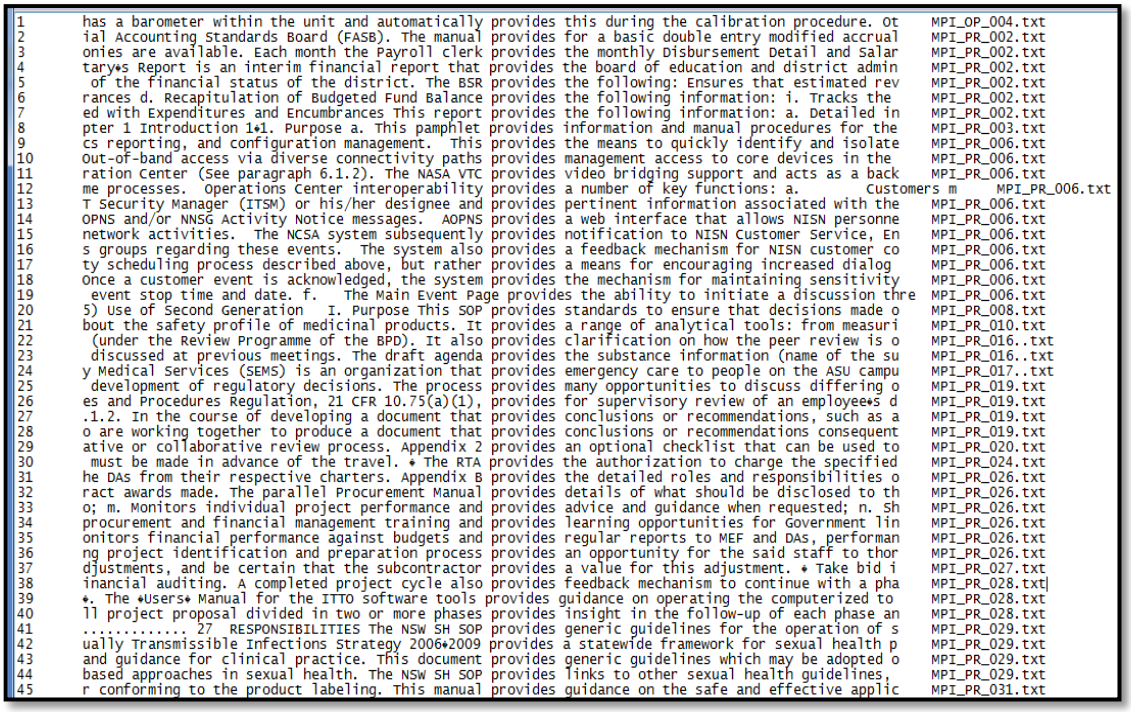


Figure 5.2 Example of concordances

All the documents from the two corpus sum 199,532 tokens and 13,091 types.

5.1.3 ROM Design

Defining a reference model

Concerning a discourse mapping, the first step is searching for the theoretical models in the state-of-the-art, which could be used as a reference for the rhetorical analysis in the genre under study. The major approaches we found are concerned to government and business documents (Trosborg 2000; Renkema 2003; McCarthy & Handford 2004; Warren, 2004) and commercial documents (Yates 1989; Jameson 2008; Freedman & Medway 1995). Directly related to the defined genres (SOP and JSD), we found no references. For this reason, we use an inductive method for defining the preliminary model. This method, according to Burdiles (2011), comprises the following steps:

- Random selection of four sample documents from the corpus;
- Incremental construction of a preliminary model from a by-hand review of the structure and superstructure;
- Peer review for analyzing the preliminary model
- Generation of reference model based on the comments of the experts.

After the sample document selection, we develop a by-hand review of the structure, superstructure, and functional analysis of each document, for identifying the common organization units in the sample (*moves*). Incrementally, and according to the genre analysis, we define *rhetorical moves* as functional sections of a genre (Swales, 1990; 2004). According to Askehave & Swales (2001), by using the *genre analysis* the formation of the schematic structure of a discourse is developed, in addition to influencing and restricting its content and style. Based on Parodi (2008), we adopt the *macro-move* concept which refers to a higher abstraction of rhetorical purpose than a *move*, namely those discursive units including a *move*. Thus, each *macro-move* serves a communicative purpose, and all macro-moves shape the overall organization of the text.

By using this type of analysis—macro-moves, moves, and steps in some cases—we also support the extension of the texts that make up the genre and the recursive functional organization of some sections.

© Bell Manrique-Losada, 2014. All rights reserved.

By analyzing the functional organization of a document, we clearly identify the organization and hierarchy levels. For this reason, we have identified purposes of a higher level hierarchy (macro-purposes), which comprise a set of more specific moves, which in turn will be composed of more detailed steps. The intention of describing and explaining the rhetorical structure of each particular genre and their associated purpose is based on the understanding of the texts under analysis and the elucidation of the communicative purpose or function from a specific discourse community.

Thus, we define a preliminary model as a set of functional and structural patterns, resulted from the identification of moves with recurrent presence (likely mandatory level). The mandatory moves are selected based on a set of categories that we define for evaluating each move in each document from the sub-corpus, related to the percentage of occurrence by move as follows:

i. 0 ; does not appear in the document (occurrence zero)
ii. 1–30 = low chance
iii. 31–70 %= average chance
iv. 71–100 % = mandatory

In the reference model we consider the moves were placed in categories iii and iv. For the SOP genre, the preliminary model comprise three macro-moves containing 19 moves—showing more specific functional units—as we show as follows:

Macro-move I: Preamble / Overview. This is a preliminary statement presenting an introduction of the document, describing the document purpose, conventions, revision schedule, approval authority, and document organization, among others.

- **Move 1:** Identifying SOP. Identify the organization authoring the SOP. This can include: author, company, location, filiation, name, and verbal or nonverbal identification.
- **Move 2:** Organizing SOP. Allude to aspects of the document body, related to content organization, lists of tables, and lists of figures, among others. This move allows for the reader to locate the document content. This section should present the entire hierarchical organization—divisions and major subdivisions—of the document, preferably with a respective list.
- **Move 3:** Introduction. Justifies and presents the document. It describes a general view of the related context and establishes what it does.
- **Move 4:** Presenting Foreword. Present a general review of the document and describes what is included in each procedure. Also, it can describe those who participated in writing the SOP, how it was organized, how to read it, the re-view process that took place, and warnings about its use and distribution.
- **Move 5:** Documenting Conventions. Give the reader the current context of the document: date of approval, version number, author, and revision number.
- **Move 6:** Appointing regulations or regulatory requirements. Name standards, contractual requirements, policy, or regulations associated with the procedures included in the SOP. It can include lists of references.
- **Move 7:** Giving acknowledgments. Present the compendium of helpers, people, or individuals acknowledged for their contributions in writing the SOP. It lists the combined effort of the human team.
- **Move 8:** Defining Intended Audience and Reading Suggestions. Define the primary audience for the SOP. It can include management team, operational team, and staff of the organization.
- **Move 9:** Establishing Purpose. Describe the general goal of the procedures included inside the SOP, in the framework of organization. This goal is oriented to contextualization and description of purpose.

Macro-move II: Development. Presents the procedures associated with each organizational process in detail. Throughout this macro-move, and their related moves and steps, a series of specific purposes, responsibilities and functions, procedural descriptions, and rules for implementation are defined.

- **Move 10:** Defining procedure purpose

© Bell Manrique-Losada, 2014. All rights reserved.

- **Move 11:** Defining roles and responsibilities
- **Move 12:** Identifying prerequisites. Identify the requisites previous to the execution of the procedure. It may include rules, cautions, warnings, or recommendations for achieving them.
- **Move 13:** Listing definitions. Includes a list of definitions, concepts, and terms of acronyms used in the context of SOP or within it.
- **Move 14:** Listing resources. List the equipment, resources, and material required for the execution of the procedure.
- **Move 15:** Establishing methods. Establish the methods used to characterize or guide the procedure.
- **Move 16:** Specifying procedure. Provide step-by-step instructions to ensure details of procedures.
- **Move 17:** Including references. List bibliographical references included within the procedure.

Macro-move III: Closure/Ending. It is related to the moves I and II. It is not mandatory, but it is intended to supplement the development macro-move.

- **Move 18:** Adding Supplementary information. (Attachments/Appendices)
- **Move 19:** Including references. (List of bibliographical references)

From the three macro-moves identified, the backbone of this genre lies in *macro-move II: development*. It acts as a unit repeated as many times as necessary for covering the detailed development of the procedures. This macro-move systematically occurs as a feature of the SOP genre. It is thus a recursive high-impact category, with a hierarchical organization stiffer. The *preamble macro-move* does not operate in the same way as the macro-move II, as its inclusion only occurs once throughout the text. Another interesting aspect among macro-moves is evident in the most flexible distribution and sometimes random distribution of the *Closure macro-move*. The latter is not mandatory, but complements the previous one in some cases, depending on the information needed to support the procedures. In the Figure 5.3 we graphically present the proposed pattern and in the Appendix 5.2 we include the complete model.

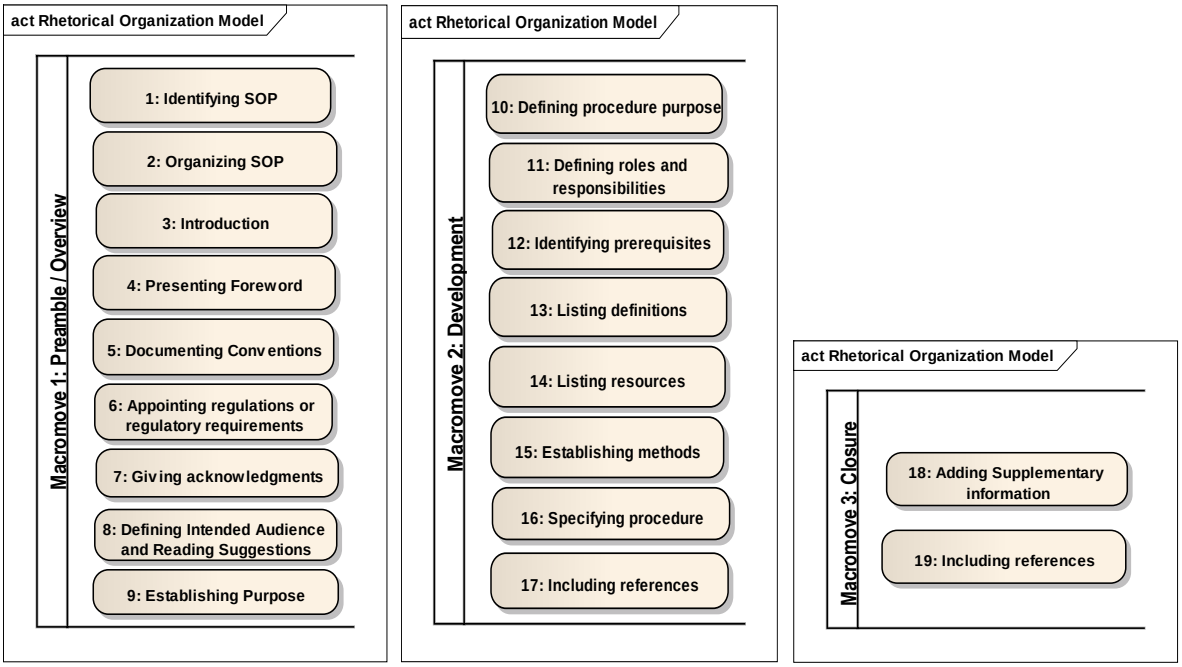


Figure 5.3 Preliminary Rhetorical Organization Model

As we present in this Section, this organization model acts as a pattern with structural and functional terms. In structural terms, depending on the macro-moves and moves, the pattern describes the organizational units of this type of technical document. In functional terms, we refer to the function that best fits the rhetorical organization units according to the communicative purpose of the author. From the

© Bell Manrique-Losada, 2014. All rights reserved.

reference model, we perform rhetorical analysis consisting in a review of each SOP of the corpus and its contrast against the reference model. We performed the recognition and registration of the findings of rhetorical units of each document from corpus and the obligation of each step, based on the recurrent presence of the move.

Rhetorical Analysis Method

The rhetorical analysis method is based on the reference model and comprises a set of by-hand activities (i, ii, iii, and iv) or meditated by computational tools (v, vi, and vii), which we present as follows:

i. *Analysis categories definition.*
In the designed grid for analyzing each rhetorical unit of the SOP, we assume as early analysis categories those defined as moves. We present such categories in Appendix 5.3.

ii. *Identification and registration of rhetorical units.*
This activity is focused on the registration of the analysis grid including the following information: identifier, summary, SOP category, *macro-move* identification and name, *move* identification and name, linguistic features identified, and an example. In addition, a process for identifying lexical and grammatical tracks from each rhetorical unit was developed.

iii. *Definition of mandatory of rhetorical units.*
Each rhetorical unit from the SOP has a presence optional or mandatory, according to the dependency with the purpose of the macro-structure and the author communicative intention. Such a recurrent presence generates the mandatory level of the move.

iv. *Corpus preprocessing.*
The collected texts in the corpus are converted into .txt format by using certain transformation actions including: format, down casing, deleting images, and complementary graphics, among others.
The non-relevant information for processing from texts (e.g. HTML tags, webpage names from downloaded information, ads, corporative information, and author information) was removed from SOPs manually and using tools like HTML Text and Multireplacer3.

v. *Identification of linguistics features by rhetorical unit.*
The identification of linguistics features is basically an observation of several variables: word frequency according to mutual information; parameters for assessing the importance of probable co-occurrence among words; lexical items that appear recurrently in given stretches of text where the conditions of the word is employed (*e.g.* from a sub-section such as a purpose or through the sub-corpus for a genre); keywords analyzed; and, frequency of the words which precede them.
Supported by several computational tools, the identification of prototypical lexical-grammatical features is performed from corpus sample. Certain of such features are essential/pertinent to current writing in specialized organizational fields. We define a set of morphosyntactic and lexical features according to those proposed by Bieber *et al.* (2004) and Venegas (2005), as we present in Tables 5.2 and 5.3 of the following Section. These features are identified in the texts from the corpus by using an automatic tagger.

vi. *Morpho-syntactic tagging.*
This analysis level aims to identify morpho-syntactic categories and then features, surrounding each rhetorical unit. Such categories involving a word or words groups, related to the morphology—depending on the word—or syntax—depending on the phrase—. We describe the features for such categories based on the feature object. The features are described by the following categories belonging to the feature object: Noun, Adjective, Verb, Subordination, Coordination, Syntactic clauses, Verbal mood, Verbal periphrasis, and person. In Table 5.2, we present the proposal of morpho-syntactic features published by Manrique *et al.* (2013). The features are proposed in terms of description and object involved, resulted from analysis and tagging.

We present morphological, syntactical, and functional set of features in Table 5.2 and Table 5.3. In the first column from Table 5.2 we specify the category—morphological or syntactical. In the second and third column we present the feature object and the feature description, based on the aforementioned object categories. The last column corresponds to the move which include the feature (according to the moves proposed in Figure 5.3).

© Bell Manrique-Losada, 2014. All rights reserved.

Table 5.2 Proposal of Morphosyntactic Features

Category	Feature Object	Feature Description	Rethorical move
Morphological	***Noun***	Extensive use of abstract nominalizations as a common feature in technical documents (e.g. *Management, establishment*)	**Move 2**: Organizing SOP **Move 5**: Documenting conventions **Move 10**: Defining procedure purpose **Move 11**: Defining roles and responsibilities
		Most common are uncountable nous (e.g. *information, process, data*)	**Move 12**: Identifying prerequisites **Move 13**: Listing definitions
	Adjective	Demonstratives most common: *this*, *that*.	**Move 1:** Identifying SOP **Move 2**: Organizing SOP **Move 10**: Defining procedure purpose
		No common usage of posesives, and scarce usage of *its* or *his/her*	-
		Undefined. They are rarely used (e.g. *any*)	**Move 12**: Identifying prerequisites **Move 13**: Listing definitions
		The qualifying most common are: *required, apropriate*, *successful, complete*. Their use is mainly subjective, rather than descriptive	**Move 11**: Defining roles and responsibilities **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
	Verb	The copulative verb are the most used mode in SOP. Among them *be, is, are*.	**Move 5**: Documenting conventions **Move 11:** Defining roles and responsibilities **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
		Can be conjugated in specific mode, commonly for describing steps (e.g. *may*).	**Move 13**: Listing definitions **Move 4**: Presenting Foreword
		Mainly conjugated in indicative mode. (e.g. *use, review, report, describe*)	**Move 11**: Defining roles and responsibilities **Move 5**: Documenting Conventions **Move 10**: Defining procedure purpose **Move 13**: Listing definitions
Syntactical	***Subordi-nation***	Common place and time subordinate (using *where*, *when, after…*)	**Move 11**: Defining roles and responsibilities **Move 13**: Listing definitions
		Adjective or Noun Subordinate Clause are used, usually begins with a relative pronoun (e.g. *that*, *how*) or using *as*, *before*, or *until*	**Move 11**: Defining roles and responsibilities **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
	Coordi-nation	The most frequent are c*opulative coordinate clauses* to express adding one stamentent to another. They are introduced by conjunctions (e.g. *and, both, as well as*)	**Move 4:** Presenting forewords **Move 5**: Documenting conventions **Move 11**: Defining roles and responsibilities **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
		The only explanatory coordinate clause found was *that is*.	**Move 4**: Presenting forewords **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
	Syntactic groups	Usage of nominal phrases (adjective+ noun) is common mainly in the roles and responsibilities section, and then, in the steps of the SOP (*e.g. appropriate project*)	**Move 11**: Defining roles and responsibilities **Move 13**: Listing definitions
	Verbal mood	Main moods: indicative and imperative, used to express either capacity or obligation.	All of the moves
		Mood in conditional steps denoting positiblity/priority (*e.g. is possible, is important,…*)	**Move 4**: Presenting forewords **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
	Verbal periphrasis	Modal Periphrasis, indicating obligation (*e.g. may+verb*, *must+verb*) and its negative form, or indicating potencial mood (*e.g. Can+verb…*)	**Move 4**: Presenting forewords **Move 12**: Identifying prerequisites **Move 13**: Listing definitions
		Aspectual Periphrasis occurs as: initiation (e.g. *take*); progression (*including performing*); habits (*used to, used for*); duration (*is/are +gerund*); Completion (*end of + infinitive*)	**Move 13**: Listing definitions
	Person	Prevails first person of plural in the sequence of procedures. In section 'roles an responsibilities' third person of singular is used. Low usage of pronouns has been detected.	All of the moves

Note: English is not a way for the subjunctive; There semantics of the subjunctive, but English is expressed by the present or the past.

Many lexical items are closely associated with particular functional patterns, since they are not isolated but exist in reference to the syntactic environments in which they typically appear (Someya, 1999). The lexical analysis is based on the observation of word behavior according to the specific context. Supported by such analysis, language variation can be determined, significantly influenced by contextual factors and grammatical functions, among others.

© Bell Manrique-Losada, 2014. All rights reserved.

Consequently, we present in Table 5.3 some *functional and lexical features* related to the syntactic ones, with the description of each feature identified.

Table 5.3 Proposal of Functional and Lexical Features

Feature Name	Feature Description
Language Function	It is informative or referential focused on the message, external reality, or referent.. The writer makes no judgments about the information or processes.
Descriptive sequence	Clear language, accurate, and direct. In most of cases clarity in the exposition of ideas can be detected. Descriptive style and usage of names, locations, and qualifying objects, people, or processes. Also, it may have actions in a temporal order looking for a specific purpose.
Terminology	Contains a large number of terms belonging to the body of knowledge and a lot of information shared between producer and reader. Such information is produced by individuals who possess specific knowledge of a subject.

vii. *Identification of occurrence frequencies for relevant features.*

By using computational tools supporting the corpus-linguistic method we try to analyze relevant features and to identify their occurrence frequencies. Thus, we have not relied on our intuition on the analysis or made-up examples. Once texts are tagged and analyzed, other programs are used for calculating the rate of occurrence of linguistics features in each text (e.g. number of verbs per a unit of words). We present sample of occurrence frequencies for some collocations, summing 68 total number of collocate types and 765 collocated tokens in Figure. 5.4. In Figure 5.5 we present the abstract noun 'management' as a feature considered relevant for the analysis, with its occurrence frequency and collocations in context.

Total No. of Collocate Types: 68 | Total No. of Collocate Tokens: 765

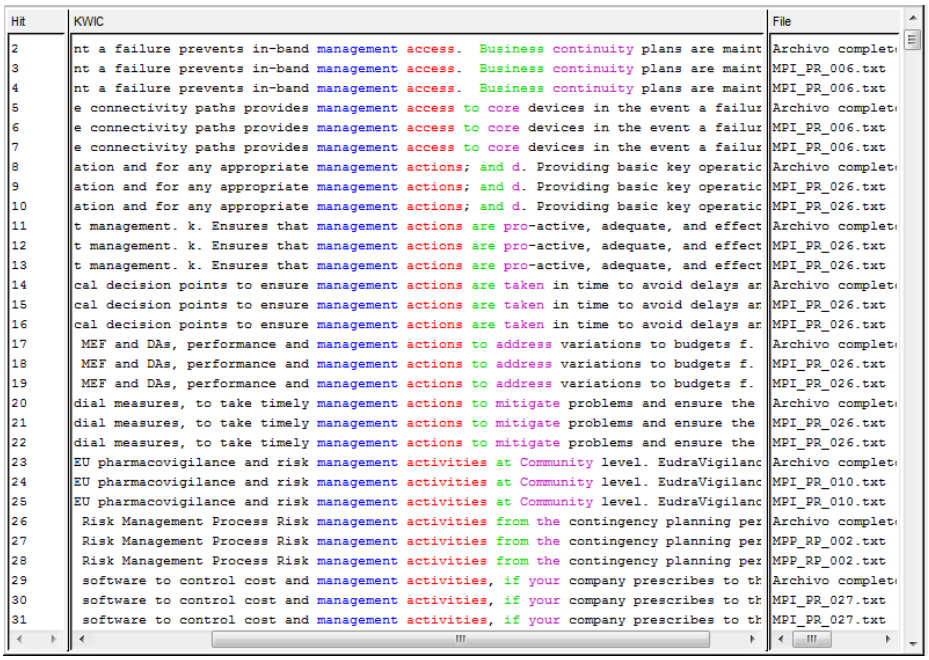
Rank	Freq	Freq(L)	Freq(R)	Stat	Collocate
1	255	0	255	4.95447	responsible
2	255	0	0	-1	is
3	27	27	0	-2	EA
4	15	15	0	-2	COMMGR
5	12	12	0	-2	JRC
6	12	12	0	-2	Center
7	7	7	0	1.60036	user
8	6	6	0	1.77310	structure
9	6	6	0	-2	MEF
10	6	6	0	0.17176	individual
11	6	6	0	-2	ENMC
12	6	6	0	1.82338	bank
13	5	5	0	-2	Contractor
14	4	4	0	-0.51078	who
15	4	4	0	-1.51382	staff
16	3	3	0	-1.89350	which
17	3	3	0	-2	Unit
18	3	3	0	-2	Training
19	3	3	0	2.53666	superintendent
20	3	3	0	-0.94396	site
21	3	3	0	-2	Services
22	3	3	0	-2	Secretariat

Figure 5.4 Sample of collocations frequency

Evaluation of structural and functional pattern

For a specific technical document we carry out the evaluation of structural and functional patterns from ROM. We choose a SOP for such evaluation. This activity is oriented to determine what rhetorical moves are nuclear or *core moves* for writing this type of technical document (SOP), as part of the methodology based on corpus linguistics. After applying the rhetorical analysis method, we follow the evaluation activities as we describe below:

© Bell Manrique-Losada, 2014. All rights reserved.



Hit	KWIC	File
2	nt a failure prevents in-band management access. Business continuity plans are maint	Archivo complet
3	nt a failure prevents in-band management access. Business continuity plans are maint	MPI_PR_006.txt
4	nt a failure prevents in-band management access. Business continuity plans are maint	MPI_PR_006.txt
5	e connectivity paths provides management access to core devices in the event a failur	Archivo complet
6	e connectivity paths provides management access to core devices in the event a failur	MPI_PR_006.txt
7	e connectivity paths provides management access to core devices in the event a failur	MPI_PR_006.txt
8	ation and for any appropriate management actions; and d. Providing basic key operatic	Archivo complet
9	ation and for any appropriate management actions; and d. Providing basic key operatic	MPI_PR_026.txt
10	ation and for any appropriate management actions; and d. Providing basic key operatic	MPI_PR_026.txt
11	t management. k. Ensures that management actions are pro-active, adequate, and effect	Archivo complet
12	t management. k. Ensures that management actions are pro-active, adequate, and effect	MPI_PR_026.txt
13	t management. k. Ensures that management actions are pro-active, adequate, and effect	MPI_PR_026.txt
14	cal decision points to ensure management actions are taken in time to avoid delays ar	Archivo complet
15	cal decision points to ensure management actions are taken in time to avoid delays ar	MPI_PR_026.txt
16	cal decision points to ensure management actions are taken in time to avoid delays ar	MPI_PR_026.txt
17	MEF and DAs, performance and management actions to address variations to budgets f.	Archivo complet
18	MEF and DAs, performance and management actions to address variations to budgets f.	MPI_PR_026.txt
19	MEF and DAs, performance and management actions to address variations to budgets f.	MPI_PR_026.txt
20	dial measures, to take timely management actions to mitigate problems and ensure the	Archivo complet
21	dial measures, to take timely management actions to mitigate problems and ensure the	MPI_PR_026.txt
22	dial measures, to take timely management actions to mitigate problems and ensure the	MPI_PR_026.txt
23	EU pharmacovigilance and risk management activities at Community level. EudraVigilanc	Archivo complet
24	EU pharmacovigilance and risk management activities at Community level. EudraVigilanc	MPI_PR_010.txt
25	EU pharmacovigilance and risk management activities at Community level. EudraVigilanc	MPI_PR_010.txt
26	Risk Management Process Risk management activities from the contingency planning per	Archivo complet
27	Risk Management Process Risk management activities from the contingency planning per	MPP_RP_002.txt
28	Risk Management Process Risk management activities from the contingency planning per	MPP_RP_002.txt
29	software to control cost and management activities, if your company prescribes to th	Archivo complet
30	software to control cost and management activities, if your company prescribes to th	MPI_PR_027.txt
31	software to control cost and management activities, if your company prescribes to th	MPI_PR_027.txt

Figure 5.5 Analysis of co-occurrences for the relevant feature 'management'

i. Selecting the experts for the evaluation of the reference model. The following criteria were considered for such selection:
   a. Relevance to the area
   b. Knowledge about discourse analysis
   c. Comprehension of the context of technical documents
   d. Availability for applying the evaluation process

   In this way, we select four experts, as follows:
   a) Jorge Gana Leay, Head of the M.Sc. program in Processing and Information Management. Computational Sciences Department, Engineering School, Pontificia Universidad Católica de Chile
   b) René Venegas Velásquez, Professor of Philosofy and Education Faculty, Pontificia Universidad Católica de Valparaíso
   c) Juan David Martínez, Coordinator of the PhD in Linguistics, Communication Faculty, Universidad de Antioquia
   d) Carlos Mario Zapata Jaramillo, Professor of the Computating and Decisions Sciences Department, Faculty of Mines, Universidad Nacional de Colombia –Medellín.

ii. Designing a template for the evaluation (Appendix 5.3), based on the reference model, and considering the following components:
   a. *Rhetorical Unit of reference*, including *code*, *type of rhetorical move* (move or macro-move), *name of rhetorical unit*, and *purpose*.
   b. *Example* extracted from a SOP
   c. *Evaluation* including a column for each element of rhetorical move to be evaluated: *name* and *purpose* or rhetorical move. Also, a column for specify the *obligatory/mandatory* character of the move.
   d. Finally, a space for adding *comments* whenever necessary.

iii. Designing a instructions guide for filling in the evaluation template (Appendix 5.4), including:
   a. Description of the model
   b. Evaluation of the following parameters: Agree / Satisfy (Valuation 1) and Valuation Parameter (Valuation 2)
   c. Step-by-step process for filling in the evaluation

© Bell Manrique-Losada, 2014. All rights reserved.

iv. Sending the request for information to the experts for evaluation purposes. We sent the requests via email, including a letter with the instructions for evaluating and filling in the form, with the evaluation template as attachment.

v. Analyzing and filtering the forms filled in.

We received the evaluation template filled by the four experts, as we present in the Appendix 5.5 and we developed an analysis *inter rater reliability*, according to the comments, evaluated parameters, and valuation of each parameter. Such analysis generates a new version of the model for the subsequent analysis, comprising only the moves considered mandatory by the experts. For this analysis, we consider as inclusion criteria only the moves which were evaluated with 3 or 4 positive responses (answer: YES). Additionally, we carried out the adjustments and changes by move in the cases where the responses disagreed or were unsatisfied (answer: 2) In Table 5.4 we show the compilation of results and the average of responses by the experts.

Table 5.4 Results and actions on the reference model

Rhetorical Unit of reference			Evaluation			
Code	Unit	Name	Name	Purpose	Obligatoriness	Action
1	Macromove	Preamble / Overview				
1.1	Move	Identifying SOP	1	-	YES	Include move
			1	1	YES	
			1	2	YES	
			1	1	YES	
1.2	Move	Organizing SOP	1	-	YES	Include move
			1	1	YES	
			1	2	YES	
			1	1	YES	
1.3	Move	Introduction	2	-	YES	Include move, correct or improve writing
			1	1	YES	
			2	2	YES	
			1	1	YES	
1.4	Move	Presenting Foreword	2	-	NO	Do not Include move
			2	2	NO	
			2	2	YES	
			1	1	YES	
1.5	Move	Documenting Conventions	2	-	YES	Include move, correct or improve writing
			1	1	YES	
			1	2	YES	
			1	1	YES	
1.6	Move	Appointing regulations or regulatory requirements	1	-	NO	Include move, correct or improve writing
			1	1	YES	
			2	2	YES	
			1	1	YES	
1.7	Move	Giving acknowledgements	2	-	NO	Do not Include move
			1	1	YES	
			1	1	NO	
			1	1	YES	
1.8	Move	Defining Intended Audience and Reading Suggestions	1	-	YES	Include move
			1	1	YES	
			1	1	NO	
			1	1	YES	
1.9	Move	Establishing Purpose	2	-	NO	Include move
			1	1	YES	
			1	1	YES	
			1	1	YES	
2	Macromove	Development				
2.1	Move	Defining procedure purpose	1	-	YES	Include move
			1	1	YES	
			1	2	YES	
			1	1	YES	
2.2	Move	Defining roles and responsibilities	1	-	YES	Include move
			1	1	YES	
			1	1	YES	
			1	1	YES	
2.3	Move	Identifying prerequisites	2	-	YES	Include move

© Bell Manrique-Losada, 2014. All rights reserved.

Rhetorical Unit of reference			Evaluation			
Code	Unit	Name	Name	Purpose	Obligatoriness	Action
			1	1	YES	
			1	2	YES	
			1	1	YES	
2.4	Move	Listing definitions	2	-	YES	
			1	1	YES	Include move
			1	1	YES	
			1	1	YES	
2.5	Move	Listing resources	1	-	YES	Include move, improve writing
			1	1	YES	
			1	2	YES	
			1	1	YES	
2.6	Move	Establishing methods	1	-	YES	Include move
			1	1	YES	
			1	1	YES	
			1	1	YES	
2.7	Move	Specifying procedure	1	-	YES	Include move
			1	1	YES	
			1	1	YES	
			1	1	YES	
2.8	Move	Representing procedure	2	-	NO	Do not Include move
			1	1	YES	
			1	1	NO	
			1	1	YES	
3.	Macromove	Closure / Ending				
3.1	Move	Adding Supplementary information	2	-	NO	Do not Include move
			1	1	YES	
			1	1	NO	
			1	1	YES	
3.2	Move	Including references	1	-	YES	Include move
			1	1	YES	
			1	1	YES	
			1	1	YES	

Based on the previous results and their analysis, we generate a new proposal of the rhetorical organization model for the SOP document, as we present in the Table 5.5. In the model we exclude the moves not mandatory, selected the appropriate nomination and a description closer to the context and intention of the SOP author according to the analysis inter rater reliability.

Table 5.5 Proposal of Rhetorical Organization Model (ROM)

Code	Rhetorical Unit Type	Name of Rhetorical Unit	Purpose Description of Rhetorical Unit
1	Macromove	Preamble / Overview	**Introducing the information of the content of the SOP by means of a preliminary statement presenting an introduction to the document, describing the document purpose, conventions, revision schedule, approval authority, and document organization, among others.**
1.1	Move	Identifying SOP	Identifying the authorship of the SOP. This can include: author, company, location, filiation, name, and verbal or nonverbal identification
1.2	Move	Organizing SOP	Listing the content of the SOP, referring to aspects of the document body: contents organization, lists of tables, and lists of figures, among others. This move allows the reader to locate the document content. This section should present the entire hierarchical organization (divisions and major subdivisions) of the document, preferably with a respective list.
1.3	Move	Introducing the SOP	Justifying the relevance of the document through a description of the related context and the functionality of the process and procedures.
1.5	Move	Documenting Conventions	Identifying the SOP in terms of coding, name, dates of publication, approval, and updating, version number, author, or revision number
1.6	Move	Appointing regulations	Contextualizing the SOP according to previous standards,

© Bell Manrique-Losada, 2014. All rights reserved.

Code	Rhetorical Unit Type	Name of Rhetorical Unit	Purpose Description of Rhetorical Unit
		or regulatory requirements	contractual requirements, policy or regulations, and/or the state of the art.
1.8	Move	Defining Intended Audience and Reading Suggestions	Defining the primary audience for SOP. It can include management team, operational team, and staff of the organization
1.9	Move	Establishing Purpose	Describing the general goal or purposes of the procedures included inside SOP, in the framework of organization.
2	Macromove	Development	**Presenting in detail the procedures associated with an organizational process. Through this macromove, and their related moves, sets forth a series of specific purposes, functions and responsibilities, procedural descriptions, and rules for implementation**
2.1	Move	Defining procedure purpose	Defining the purpose of each procedure
2.2	Move	Defining roles and responsibilities	Defining the roles of the stakeholders involved in the procedure and the responsibilities of each one
2.3	Move	Identifying prerequisites	Identifying required conditions previous to the executions of the procedure. It may include rules, cautions, warnings, or recommendations for achieving them
2.4	Move	Listing definitions	Defining concepts, terms, or acronyms used in the context of SOP or within it
2.5	Move	Listing resources	Specifying the equipment, resources, or material required for the execution of procedure
2.6	Move	Establishing methods	Establishing the methods used to characterize or guide the procedure
2.7	Move	Specifying procedure	Providing step-by-step instructions for stipulating details of the procedures
3.	Macromove	Closure / Ending	**Supplementing the information presented in macromove I and II**
3.2	Move	Including references	Listing bibliographical references

ROM for mapping

Structurally speaking, the ROM approach is the foundation of the behavioral model and the mapping method, since in such a model we define the input structure of technical documents for processing them. Based on the ROM, in the behavioral model we define how the mapping process will occur and which are the components comprised.

After processing the levels we describe in the following Section (5.2), we identify some moves from ROM which do not generate useful information when we applying the mapping rules. For such reason and seeking to lighten the assessment model, we exclude from the ROM the moves presented in Figure 5.6 as a way to ease processing. Thus, we use exclusively the useful moves for formalizing the mapping process, leading to fewer the moves that we need for characterizing the genre.

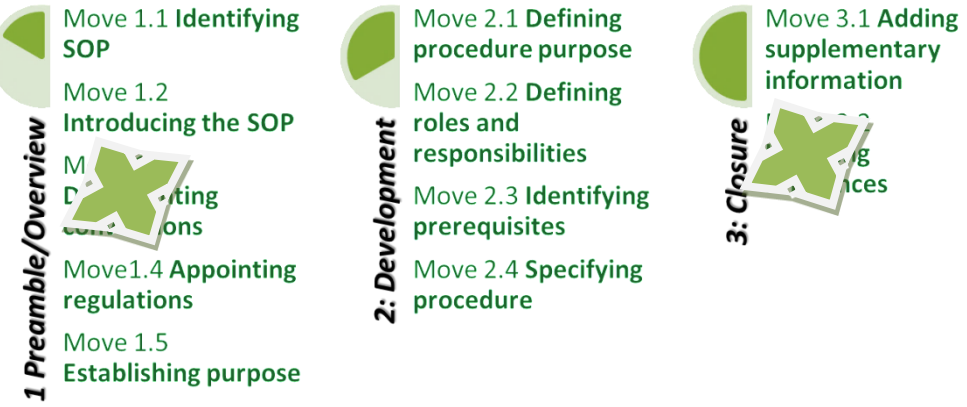


Figure 5.6 ROM for mapping

© Bell Manrique-Losada, 2014. All rights reserved.

5.2 BEHAVIORAL MODEL

Given the aforesaid functional and structural characterization of the technical documents, we define a processing model which can generate a controlled language output based on the patterns proposed in the ROM. Furthermore, we propose a mapping method which can be replicated from scratch for performing the mapping from other kinds of technical documents. We illustrate the scope of the proposal as a general framework in Figure 5.7.

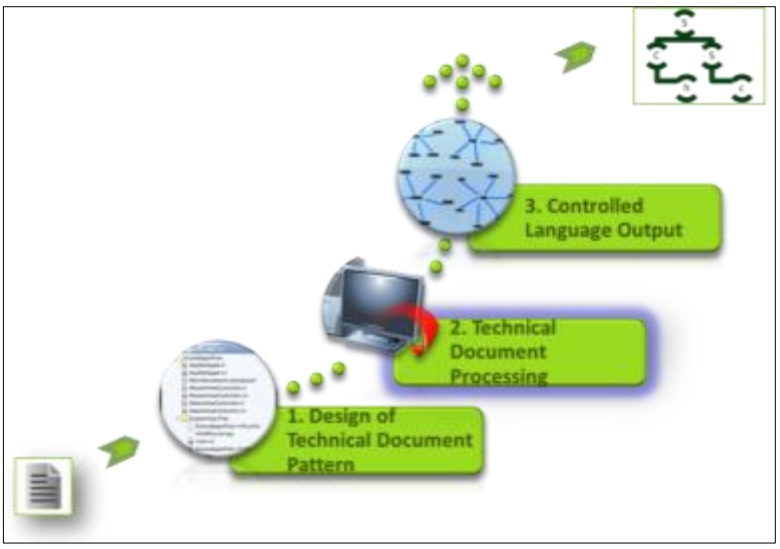


Figure 5.7 General framework for the mapping process

5.2.1 Processing Model

The step 2 'Technical Document Processing' is performed by using several processing levels, as we show in the Figure 5.8, including:

- Preprocessing: Initial filtering and ROM Filtering
- Processing: Parsing and Semantic processing (Verb-centered analysis and Phrasal-expression-centered analysis)

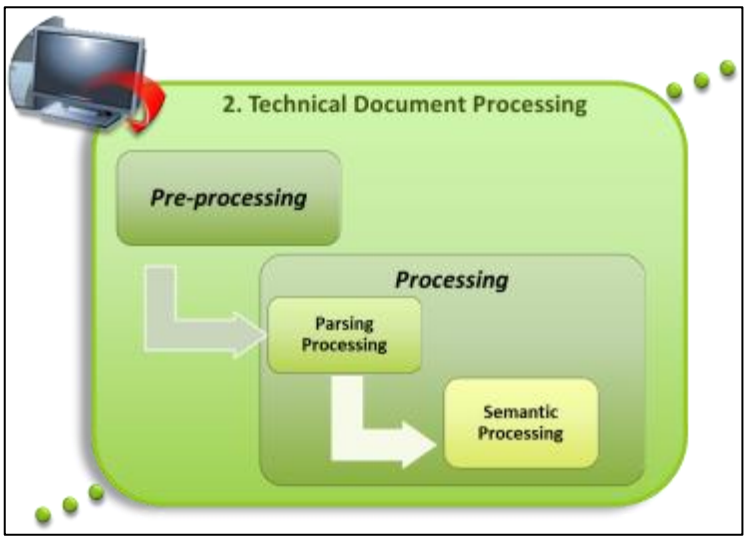


Figure 5.8 Processing model

© Bell Manrique-Losada, 2014. All rights reserved.

Pre-processing

Several document analysis operations have to be performed prior to recognizing text structure and contents. We define two moments for preprocessing:

- *Initial Filtering*
  Some of the common operations performed in pre-processing are: deletion of images; noise removal, extraction of the foreground textual matter, noise and interfering strokes; line, word, and character segmentation, separation of individual lines of text; isolation of textual words and individual characters.

  For the sake of supporting such operations, we define a script for filtering the less complex sentences to the parser. We aim to make dependence parsing and the following steps of processing to be as precise as possible. The automated pre-processing was performed based on the following routine: removing non-alphanumeric strings; replacing any punctuation mark different to comma for a period, with the exception of hyphens when no space is detected before or after it; and, inserting a period at the end of sentences when needed.

- *ROM Filtering*
  We identify the useful moves of the document (sections) by using the ROM filter and, based on patterns defined as rule-driven and combinations of synonyms. Then, we define a list of moves and synonyms of each one. Such information by move is driven as a synonym vector. The filter takes the input text file and looks in the vector the moves matching with the set of *move_name* and *move_synonyms*. The complete set of synonyms by move from ROM is shown in Appendix 5.6. For example, for the *Move 1.1: Identifying SOP*, the set of synonyms include by the vector are: SOP, SOP:, Standard Operating Procedure for the, Standard Operating Procedure for, Title:, and Title.

  Process patterns are identified as: step-by-step, enumerations, bullets, etc. They will be considered in the final document. Each move is identified, formatted, and then the information composing the move is inserted into the output document. The following steps of the processing are based on the text file resulted from this step with the most relevant information—in terms of the moves detected.

Processing

The processing is performed in two levels: parsing and semantic processing. The processing levels performed in the parsing are based on a proposed set of features related to tokenization, tagging, and parsing. The semantic processing in turn comprises other sub-levels: verb-centered and phrasal-expression-centered. We present the features and characteristics of each processing level as follows:

- ***Parsing***
  Adding annotations at multiple levels of the linguistics analysis (Wyme, 2005) by following the principles of the natural language processing (NLP). We develop the following levels of parsing:

  i. *Lexical Analysis or Tokenization*. In this analysis the most basic constituents of the syntax—*e.g.* keywords, numbers, operators, and comments—are recognized (Schmitz, 2007). In this process we split a sentence into tokens. A token can be a word, phrase, or other significant element. The tokenizer uses white spaces (*e.g.* blanks or tabs) as the main track for splitting the text into tokens.

  ii. *Part-of-Speech (POS) Tagging*. Is the process of tagging the words from the input text with a particular part of speech, based on its definition and context (Jayan & Raj, 2011). The context is related to the relationships among the word with its adjacent words in a phrase, sentence, or paragraph. Depending on the POS Tagger selected, this process brings the annotation with specific POS tags from its corresponding POS annotation scheme—in a simple list of POS adjectives, adverbs, conjunctions, nouns, prepositions, pronouns, and verbs are included.

  iii. *Syntax Analysis or Parsing*. By performing parsing we extract the syntactic structure from text, check the text against the language syntax, and build parse trees from the tokens. A parser attempts

© Bell Manrique-Losada, 2014. All rights reserved.

to extract the phrase/constituent structure, *i.e.,* to extract the structure and relationships among the words conforming phrases.

- ***Semantic***

  This level of processing comprises the analysis of the linguistic realization of content structures and functions in the input text. We adopt the semantic analysis from two points of view: i) verb-centered analysis; ii) phrasal-expression-centered analysis. We are trying to perform a meaning analysis as closely linked to the exploration of concepts and its usage scenarios.

  i. *Verb-centered analysis.*

  This analysis is based on the semantic behavior of the verb, under the premise that the analysis of meanings or senses should be closely linked to the analysis of terms used in a context. The verb-centered analysis is approached from the point of view of the possible meanings suggested by the Multilingual Central Repository (MCR)[7].

  Based on the experimental corpus (sub-corpus 4: SOP), we identify the set of most frequent verbs. Prioritized verbs are classified by categories, according to Vossen (2002). Then, we use the types of verb in order to identify patterns. Such patterns—integrated with rhetorical, morphological, and lexical patterns—will be the basis of rules for inferring and extracting organizational relationships from corporate information. In this way, we orientate the analysis to all situations concerning the verb regarding its usage in the SOPs. Based on an incremental method, we performed this analysis step-by-step as follows:

  a) Review: In this phase we look for and identify the verbs in the relevant sections of the documents (these relevant units were identified according to the DRA, in section 5.1).

  For the sake of identifying verbs, first we prioritize the most used verbs in the SOPs according to the occurrence frequency. Analysis of the occurrence frequency of verbs in corpus is supported by the corpus analysis tools AntConc 3.3.5w®[8] and TermoStatWeb[9]. We selected the first 58 verbs corresponding to the interval of hits [501–72], with 501 the highest frequency and 72 the lower occurrence, as we present in Appendix 5.7. In Table 5.6 we present the 10 most used verbs.

  Table 5.6 Sample of prioritized verbs for verb-centered analysis

No.	Verb	Category					
		facto tum	social	free-time	applied science	humanities	pure-sciences
1	assigns						
1	use	3	1				
2	include	3					
3	provide	2	1	1			
4	follow	4					
5	require	4					
6	review	2	1			1	
7	process	1	1		2		1
8	ensure	2					
9	request	3					
10	submit	4					

  b) Feature identification of verbs: This identification is addressed by identifying the semantic features by each prioritized verb, and by defining discriminant features for relevant and

[7] http://adimen.si.ehu.es/web/MCR

[8] AntConc 3.3.5w®. A freeware concordance program (Anthony, 2009)

[9] TermoStatWeb™. A term extractor that uses linguistic and statistical methods based on the potential terms' structures and relative frequencies from corpus (Drowen, 2003)

© Bell Manrique-Losada, 2014. All rights reserved.

irrelevant verbs, according to some categories—conceptual, functional, process, and situation involved. The classification of verbs is based on the meanings reported by WordNet 3.0 by using the Interlingual Index (ILI 3.0) through the EuroWordNet Interface (WEI consult mode)[10]. For each prioritized verb, we define the categories of classification according to the classes defined in section 2.3.1, by following an iterative and sequential process:

i) For each verb, check the meanings reported in WEI.
   For the first four meanings reported:
   - Assign an occurrence indicator (value) for each associated conceptual category.
   - Assign a value for each functional category the verb has.
   - Assign: values to situational category: unique value for situation type if is dynamic or static, and multiple value for the situation component

ii) Generate the sum of all values for each category

iii) Identify which categories correspond with the highest sum for the corresponding analysis.

The complete verbs classification is presented in Appendix 5.8, as a result of the previous analysis. In Table 5.7 a sample of such results is presented.

Table 5.7 Sample of prioritized verbs for verb-centered analysis

N	Verb	Conceptual classification of verb																		
		Category						Category: factotum (base concept: 1)											
		factotum	social	free-time	applied science	humanities	pure-sciences	social	Possession	change	cognition	stative	communication	creation	contact	Competition	consumption	stative	perception
1	assigns							1											
1	use	3	1					1									3		
2	include	3						1			1							1	
3	provide	2	1	1					1				1				1		
4	follow	4								2								2	
5	require	4											2				1	1	
6	review	2	1			1					3		1						
7	process	1	1		2		1	2		1	1								
8	ensure	2									1		1						
9	request	3											3						
10	submit	4							1				2			1			
11	work	3	1					3											
12	approve	2									1		1						
13	prepare	3		1						1				3					
14	identify	4									3		1						
15	involve	4																4	
16	perform	3				1		2						1					
17	describe	2				1	1				1		2		1				
18	determine	3					1				2		2						
19	need	3										1				2			
20	maintain	4							1			1	1				1		

iv) Analyze and present results. Based on the defined classification and categorization of verbs, we make and analysis and identify the following findings:

[10] http://adimen.si.ehu.es/cgi-bin/wei/public/wei.consult.perl

© Bell Manrique-Losada, 2014. All rights reserved.

Conceptual Category. The conceptual domains are based on a relation of specificity. We identified in the classification process that the verbs are not assigned to a particular conceptual category, due to their nature. Unlike nouns which somehow can be grouped by domains, the verbs can be used in similar senses by several different domains. For instance we can find that '*apply*' is used in a specific domain like '*Medicine'* for saying '*a nurse is applying medication to a patient*,' or in a general domain involving an intentional process for saying '*The operator works applying rules.'*

As we show in the results of classification, the prioritized verbs are mostly used in any domain, for example the one appearing under the label FACTOTUM, which is assigned when none of the labels were assigned. When verb is not labeled as factotum, the second mostly used conceptual category is SOCIAL label.

Functional category. According to the tracks of the analysis, most of verbs are marked as an *intentional process* (general), whose intention is no longer identified (*e.g.* attaching, comparing, substituting, and separating). Generally speaking, an *intentional process* is deliberately set in motion by a Cognitive Agent, *i.e.* it is a human action, act, or activity of a thing for accomplishing or achieving a work.

The second most frequent functional category is '*social interaction'* as a kind of intentional process involving interactions between Cognitive-Agents. This category relates a social relation, an interaction, or a socially accepted situation.

Situation Category. The *situation type* for most of the verbs is '*dynamic.'* Such verbs are related to the situations implying either a transition from one state to another or a continuous transition perceived as an unbounded process (e.g. event, act, action, become, happen, take place, process, habit, change, and activity). No change in their properties or relation is involved by the verb.

In *dynamic situations*, more than half of verbs occur with 'bounded event', when they are implied with a specific transition from one situation to another, which is bounded in time and directed to a result (e.g. *to implement*, *to remove*, *to develop, etc.*). Regarding to the *situation component*, the results show that the main semantic components characterizing the situation are the following:

- *Cause*. Component of situations involving causation. This component is coherent with the *situation type* since the *Causation* is always combined with *dynamic* and it can take several forms. Such forms depend on the grade of intervention of an agent. The form with the higher frequency is *agentive* which can be related to a controlling agent who intentionally tries to achieve some change. The agentive situations imply a controlling agent causing a dynamic change (*e.g.* to implement, to write, to record, etc.).
- *Purpose*. Abstract components reflecting the intentionality of acts and activities. Situations intended to have some effect are implied. As the previous situation component, this one reflects consistency with the context as applied to *dynamic situations*. Also, this component strongly correlates to *agentive* and *cause*, clustering mainly human acts and activities. *Situation Components* such as *usage*, *social,* and *communication* often combine with *purpose*.
- *Communication*. Component of situations involving communication (e.g. designate, request, describe, issue, etc.). Communication verbs are often speech-acts (*bounded events*) or denote more global communicative activities (*unbounded events*). Also, they include different phases of the communication referring to causation of communication effects (*e.g.,* to explain or to show) or creation of a meaningful representation (*e.g.* to write or to draw).
- *Physical*. Component of situations involving perceptual and measurable properties of objects (*e.g.* to shape, to prepare, to describe, etc.); or dynamic changes and perceptions of its physical properties (*e.g.* to monitor, to collect, to copy, to notice, etc.).

c) Dependency parsing: Processing of the evaluation corpus with a dependency parser. The goals are:

© Bell Manrique-Losada, 2014. All rights reserved.

- Defining patterns of occurrence of the identified verbs and defining a set of semantic/dependency rules for transforming each pattern to a controlled language structure.
- Defining a script for preprocessing the SOPs, trying to extract simple sentences for the parser to maximize its performance.
- Processing the evaluation corpus with a dependency parser.
- Evaluating the extracted relations and the findings.

Based on the previous review and feature identification of verbs (the above section), we propose a set of semantic rules for transforming each feature into a controlled language. We used the UN-Lencep (named by its Spanish acronym for 'Universidad Nacional de Colombia—Lenguaje Controlado para la Especificación de Esquemas Preconceptuales'), as an intermediate representation between natural language and conceptual schemas for software engineering.

In the Table 5.8 we present the rules defined for such mapping. Each mapping rule is assigned to one category (first column), expressed in terms of the pattern in the SOP [If Pattern] and the expression in UNLencep generated [then (→) Expression]. Additionally, in the second row of each some attributes (features conditioning each element of the pattern) related to the tags (*e.g.* syntactic or semantic tag—synt—, function tag—func—, etc.) are assigned by the parser.

For the parsing process we use the Freeling dependency parser[11]. Finally, in the third row of each rule, we included an example of a phrase matching the pattern and the resulting expression in UN-Lencep.

The output expression in the rules shows what generates the parser and semantic processor in the application of each rule (in the part of the sentence that matches the pattern). Thus, the parser applies multiple rules to the same sentence, such that the output will be complementary between a rule and another.

ii. *Phrasal-Expression-centered analysis.*

We propose context analysis—in which phrasal expressions (PEs) occur—for contributing to the process of adding essential information to the pattern definition. Such patterns are conceived from the structural and functional components—based on our ROM approach—inherent to corporate documents. This means that we classify PEs according to the section in the document where they prevail.

With the purpose of capturing the features of PEs we analyzed the expressions included in the corpus, according to Sugino (2008) and the categorization of PEs described in Section 2.3.1. We present the complete report in Manrique *et al.* (2013) and describe a sample of this processing level for the sub-corpus 1 '*Job Standard Document*,' as follows:

Relevant PEs are identified in the experimental sub-corpus according to the process shown in Figure 5.8. From each technical document belonging to the corpus, we carried out the PE extraction task (institutionalized expressions or lexicalized expressions) and we make the classification by analysis categories.

We classify the extracted PEs based on the document section where they prevail (see Table 5.9). Each section corresponds to a structural and functional unit of the JSD which also reflects the communicative intention of the writer.

[11] http://nlp.lsi.upc.edu/freeling/)

© Bell Manrique-Losada, 2014. All rights reserved.

Table 5.8 Proposal of rules pattern for mapping

Rule Name	Mapping rule Description	
	Pattern in the SOP	**Expression in UNLencep**
i. Transitive verbs	**If *Obj₁ + VB + Obj₂* → *Obj₁ + VB + Obj₂*** ATTRIBUTE CONSTRAINTS: *Obj₁* = func:ncsubj, tag:NN; *VB* = synt:sv, tag:VB|VBZ|VBP; *Obj₂* = func:dobj, tag:NN	
	Manager performs data analysis	*Manager+perform+analysis*
ii. Transitive verbs without inmediate subject	**If *VB + Obj₁* → *X + VB + Obj₁**** ATTRIBUTE CONSTRAINTS: Obj₁ = func:dobj, tag:NN; VB = synt:sv, ag:VB|VBZ| VBD|VBP	
	Assess the request	*X+assess+request*
iii:Passive voice	**If *Obj₁ + [to be+VBN] +by+ Obj₂* → *Obj₂ + VB + Obj₁*** **Else If *Obj₁ + [to be+VBN]* → *X + VB + Obj₁*** ATTRIBUTE CONSTRAINTS: *Obj₁,Obj₂* = func:dobj, tag:NN| NNS; [.] = tag:VBZ+VBN	
	the procedure is applied by Sector	*Sector+apply+procedure*
iv.Construction of the form 'is a'	**If *Obj₁ +[is (a|an)] + Obj₂* → *Obj₁ + is + Obj₂*** ATTRIBUTE CONSTRAINTS: *Obj₁,Obj₂* = func:dobj|ncsubj, tag:NN| NNS; [.] = tag:VBZ+VBN	
	...Internal Controls is a document...	*Internal_control+is+document*
v. Noun phrase with post-modification	**If *Obj₁ + [of the | of a] + Obj₂* → *Obj₂ + has + Obj₁*** ATTRIBUTE CONSTRAINTS: Obj₁ & Obj₂ = func:dobj; [.] = tag:IN	
	...is the responsibility of the Sector...	*Sector+has+responsibility*
vi. Noun phrase with pre-modification	**If *[JJ | VBN] + Obj* → *Obj + has + Attribute [value: JJ | VBN]*** ATTRIBUTE CONSTRAINTS: *Obj* = func:dobj|ncsubj; tag:NN|NNS, synt: sn-chunk|n-chunk; *JJ,VBN* = func:ncmod, synt:attrib, tag:JJ|VBN	
	...Agency to manage the electronic data... *...the requested product belongs...*	*data+has+Attribute [value:electronic]* *Product+has+Attribute value:requested]*
vii. Main clause + infinitive phrase	**If VB₁ + [to | for] + VB₂ + Obj → *VB₁ => VB₂ + Obj*** **Else If *VB₁ + [to] + VB₂* → *VB₁ => VB + Y*** ATTRIBUTE CONSTRAINTS: *VB₁* = synt:sv, tag:VBN| VB|VBZ; *[.]* = tag:TO|IN, func:iobj; *VB* = func:cmod, synt:inf, tag:VB|VBG|VBZ|VBP; *Obj* = func:dobj, tag:NN|NNS	
	...it is *used to classify event...*	*Use => classify+event*

* This rule corresponds to document related features (*e.g.* when the agent is stated early in the document and a list of duties for him/her is presented later with the verb in the infinitive form

Conventions: X = default subject; Y = default object; [.]= literal expression or form; Obj = NN, object or subject; + = dependency relationship; Obj_A = attribute, state, type, or characteristic from an object; => = Implication; VB = Verb base form; VBN = Verb past participle; VBZ = Verb 3rd person singular present; VBP = Verb non-3rd person singular present; VBG = verb gerund; NN = Noun singular; NNS = Noun plural; IN = preposition; JJ = Adjective; TO = particle 'to'.

© Bell Manrique-Losada, 2014. All rights reserved.

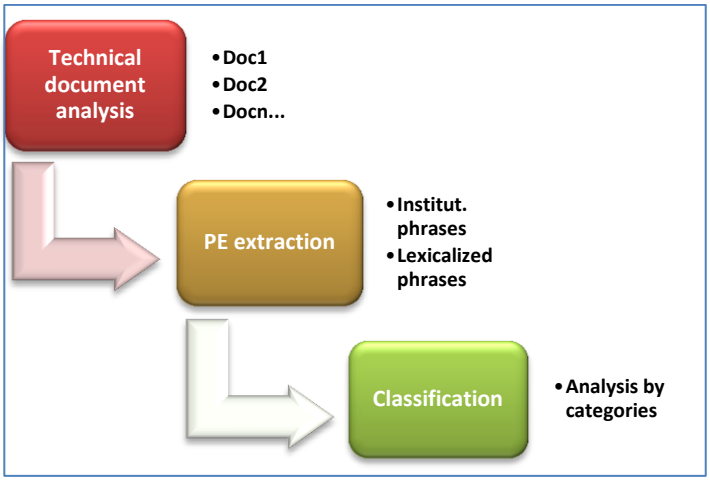


Figure 5.8. Process for identifying PEs

Table 5.9. Sections of JSD

No.	Description section
i	Job purpose / objective
ii	Main responsibilites / functions
iii	Knowledge and skills
iv	Requirements

In Table 5.10 we present the relevant PEs identified, as follows: i) the selected expressions with the corresponding PE category according to the classification proposed by Baldwin *et al.* (2010); ii) the frequency of occurrence of each expression; and, iii) the section number where the expression prevails in the JSD (from the Table 5.8).

We use brackets for indicating semi-fixed phrases or variable uses of the expression (they can take values with the same conjugation). In this way, we identify and prioritize the most frequents PEs and patterns in each category, as follows:

- *ability to, knowledge of, experience in, be able to, be required to*
- *to-V-the, the-N-and-N-of, in-N-with-the-N*
- *may be, carry out, work in, work of*

Likewise, we also found useful identifying the most frequent lexical items which could become part of PEs and alternate with the expressions and patterns presented above. For that purpose, TermoStatWeb™ was used to generate a map with the most frequent verbs, nouns, and adjectives. Some examples are shown in Figure 5.9.

The high frequency of these items in the corpus suggests they could probably be part of PEs conveying corporate information. Also, when placed in the slots of the patterns observed in Table 5.9, they increase their chance to become relevant PEs useful for detecting specific corporate knowledge.

© Bell Manrique-Losada, 2014. All rights reserved.

Table 5.10. Extracted PEs

Category	Phrasal Expressions			Frequency	JSD Section number
1. Statistically-idiomatic phrases	be	Able	to	13	iii
	be	required	to	13	ii
	are	required	to	7	iv
	be	responsible	for	5	ii
	-	knowledge	of	49	iii
	-	experience	in	15	iv
	-	ability	to	61	iii
	related	duties	as	11	Ii
	the	duties	of	6	ii
	skills	and	abilities	11	iii
	level	experience	-	12	iv
	job	code	-	4	i
	job	description	-	9	i
	job	specification	-	7	i
	office	equipment	-	5	ii,iii
	working	relationships	with	12	ii,iii
	at	all	times	10	ii
	as	well	as	11	ii
2. Syntactically-flexible phrases	be	*[adquired]*	on	5	iv
	to	*[support]*	the	29	ii
	the	*[priority]* and *[schedule]*	of	24	ii,iii
	the	*[work]* of	*[others]*	12	iii,iv
	by	*[giv]ing*	*[time]*	11	iii,iv
	in	*[contacts]* with the	*[public]*	13	ii
3. Semi-fixed phrases	-	work	in	7	ii,iii
	-	work	of	6	ii
	-	work	with	5	iii
	-	may	be	30	ii
	-	may	have	5	iv
	-	follow	up	4	i,ii
	-	carry	out	9	i,

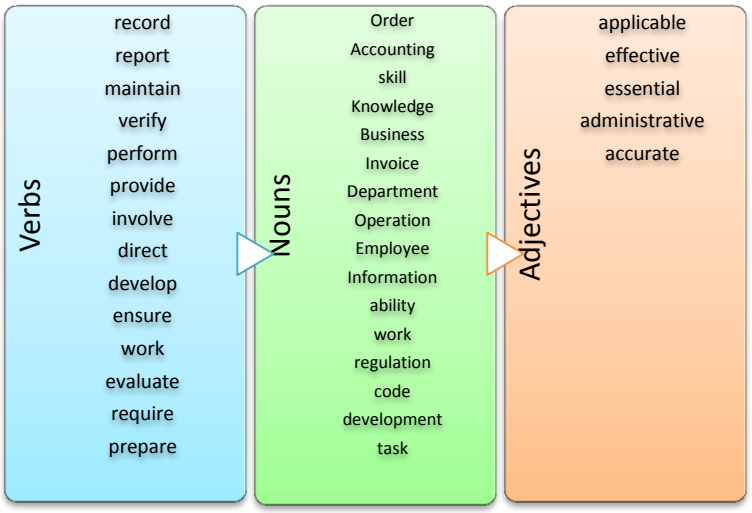


Figure 5.9. Some frequent verbs, nouns, and adjectives.

© Bell Manrique-Losada, 2014. All rights reserved.

5.2.2 Method for Mapping

In this section, we present an approach to a method for identifying useful information in the requirements elicitation process from corporate technical documents. This approach is supported by techniques and tools used in NLP. The general structure of this approach is presented in Figure 5.10.

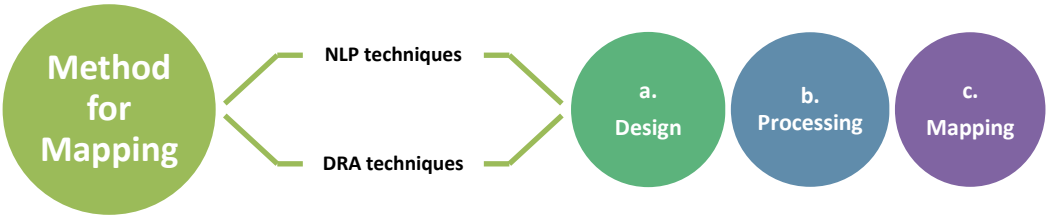


Figure 5.10. Method for mapping.

After the source document has been selected, the analyst should format it by using a template. As soon as the document has been formatted, he/she begins the analysis and processing. Finally, the output is produced as a set of phrases in a controlled language.

Before performing the phases of the approach, the analyst needs to define both the input and output languages of the analysis process applied to the source document. The first one is the natural language used in the source document, which is normally expressed by using procedural—technical—terms, which are related to the corresponding system domain (*e.g.*, English and Spanish). This input language is defined in the first phase of the approach, represented in Figure 5.11. The output language is the CL.

After that, we describe each activity of the approach. The analyst should be aware that the proposal is intended to identify and extract domain knowledge and business information. In order to reach this goal, he/she should define whether the document analysis will be supported by an automated process or not.

The current proposal is intended to be automated and the analyst should take the following steps for phases, with the support of techniques and tools from NLP and DRA. We present the representation of each phase in Figures 5.11, 5.12, and 5.13.

a) Design phase.

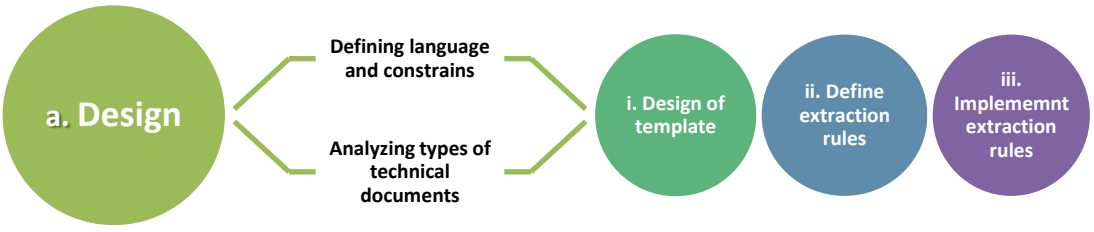


Figure 5.11. Design phase.

i. Design of the document template.
   a. A document template is used in order to assist the analyst in structuring a technical document as a product of a well-defined procedure. We can repeat the procedure whenever the same kind of document is required. The goal is classifying each document type into a template created by an analyst.
   b. The template comprises predefined fields to hierarchically organize the information of the technical documents belonging to each organizational process. We defined it in the extensible markup language (XML) to describe a set of rules for encoding such documents in a human and machine-readable format.

© Bell Manrique-Losada, 2014. All rights reserved.

ii. Define extraction rules.
   a. We aim to create a set of standard rules for identifying useful information from each grammatical construction of the technical document. The set of rules are helpful to look for useful information from each piece of the information which can be processed, analyzed, and finally written in a controlled language.
   b. The proposed rules should consist of a grammatical structure describing the behavioral aspects and static aspects from events, facts, orders, situations, or actions expressed inside the description of each section and their corresponding output elements.
   c. The analyst can define rules for each section. This task could improve the speed and liability of processing and implementing a specific software application.

iii. Implement extraction rules in a software application. We need to select the set of rules from the standard output and then implement them in a tool, application, software program, or software utility. This tool can be aimed at programming languages, where the syntaxes are less flexible than natural language, allowing us to have the flexibility required to analyze the natural language extracted from pieces of technical documents.

b) Document processing phase.

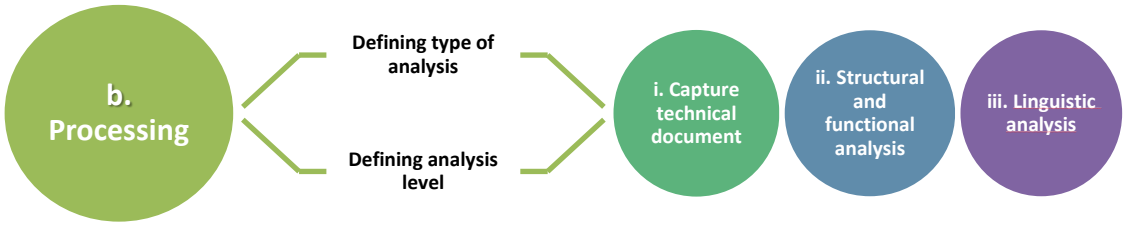


Figure 5.12. Processing phase.

An analyst can perform the document processing by-hand or supported by an automated tool. By implementing document processing based on a software application, analysts can noticeably realize many benefits for improving the elicitation process efficiency. The analyst should be guided for determining the actual needs and identifying the best method for implementing the document processing done automatically.

Document processing systems are software applications for capturing technical documents and providing storage to these documents, searching for processing and transforming them into a specific output format. Document processing begins with the capture of the electronic technical document. Then, an internal analysis (structural and functional) based on the proposed ROM is applied, for generating the formatted document. Finally, a linguistic analysis is performed in the selected levels such as morphological, lexical, or semantic.

A document processing system should have the following basic components:

- Capture tools for uploading documents into the system
- Methods for analyzing documents, from the organizational, structural, and grammatical point of view.
- Processing and retrieval tools for saving analyzed documents.
- Generating tools for exporting formatted documents from the system.

c) Mapping to a controlled language phase.

© Bell Manrique-Losada, 2014. All rights reserved.

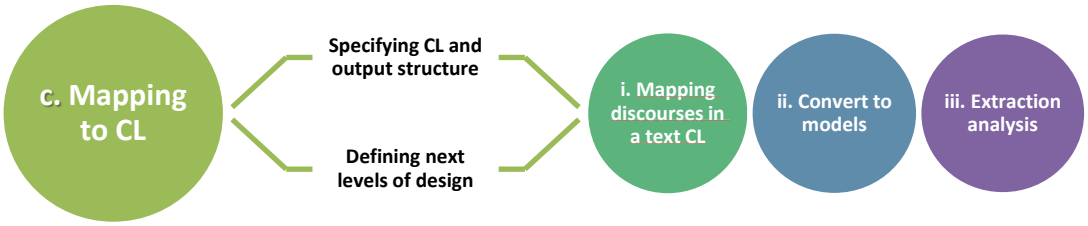


Figure 5.13. Mapping phase.

The mapping process of a technical document, as expressed in a specific CL, is the target of this activity. The CL simplifies the automated generation of a particular representation from a discourse—initially oriented to the software application—to generate and represent ideas of a specific domain. Also, the representation is intended to perform automated translation and subsequent conversion to several design diagrams of a software product.

5.3 ARCHITECTURAL MODEL

Architectural Model approach describes how the structural components of the mapping process and the relationships among them are integrated. The representation of the general architectural components of the model is displayed in the Figure 5.14.

The proposed architectural components represent an approach for implementing the mapping method—based on the previous rhetorical organization model and process model—and should be design according to a modeling software language like UML 2.3, and frameworks for developing. The proposed architecture can be described as follows: an input document is entered into a *mapping machine* comprising a *processing engine* and a *representation interface*. The *processing engine* comprises a *pre-processor* and *main processor* component, which are supported by databases regarding: document classification, regular expressions, parsing and morpho-syntactic features, and semantic patterns. The *representation interface* is based on a controlled language syntax which allows the final generation of an output text.

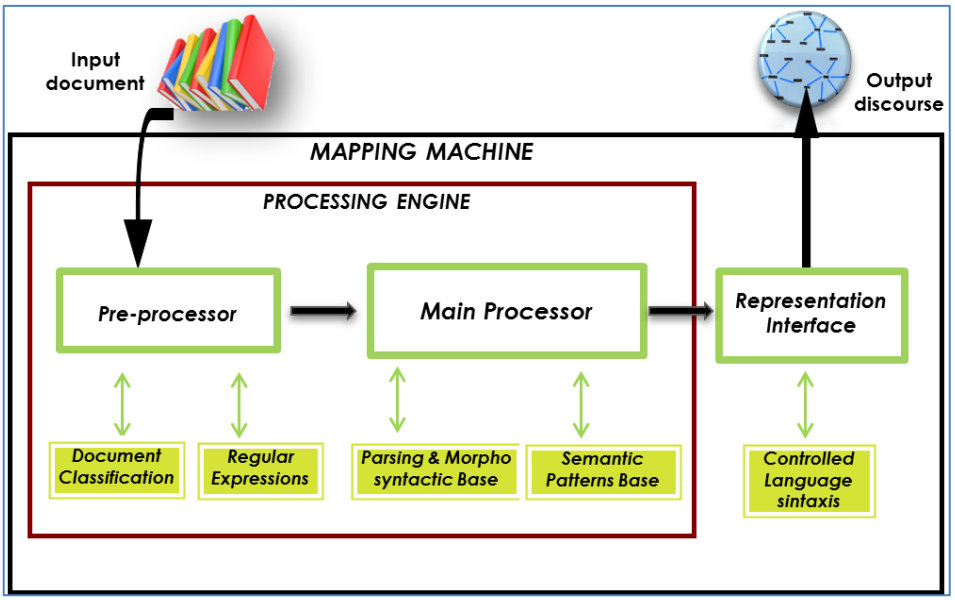


Figure 5.14. Architectural components.

© Bell Manrique-Losada, 2014. All rights reserved.

REFERENCES

A. A. Holzbacher (1996). A software environment for concurrent coordinated programming. Coordination Languages and Models, Lecture Notes in Computer Science, 1061, Springer.

A. Freedman and P. Medway (1994). Locating genre studies: Antecedents and prospects. In: Genre and the New Rhetoric, Freedman & Medway (Eds.), pp. 79-101.

A. Trosborg, (Ed.). (2000). Analyzing professional genres. Amsterdam: Benjamins.

B. Manrique, C.M. Zapata and D. Burgos (2013). Exploring MWEs for Knowledge Acquisition from Corporate Technical Documents. In: 9th Workshop on Multiword Expressions MWE 2013.

C. M. Zapata Jaramillo, B. Manrique-Losada and G. González-Calderon (2012). An approach for using procedure manuals as a source for Requirements Elicitation. In Proceedings of CLEI 2012. Vol.1, pp. 1-8.

D. A. Jameson (2008). The Making of Positive Meaning in Thai Annual Reports during the 1997 Economic Crisis: A Discourse Analysis of the "Message from the Chairman". Available in: http://rc.nida.ac.th/en/attachments/article/75/OraOng_Research_51.pdf

D. Biber, S.M. Conrad, R. Reppen, P. Byrd, M. Helt, V. Clark, V. Cortes, E. Csomay and A. Urzua (2004). Representing Language Use in the University: Analysis of the TOEFL 2000 Spoken and Written Academic Language Corpus. In: Monograph Series. Educational Testing Service, NJ.

D. Rosner, B. Grote, K. Hartmann, B. Hofling (1997). From Natural Language Documents to Sharable Product Knowledge: A Knowledge Engineering Approach. Journal of Universal Computer Science, Vol. 3, No. 8, pp. 955-987.

E. Boutkova and F. Houdek (2011). Semi-Automatic Identification of Features in Requirement Specifications. 2011 IEEE 19th International Requirements Engineering Conference.

G. A. Burdiles (2011). Descripción de la organización retórica del género caso clínico de la medicina a partir del corpus CCM-2009. Ph.D. Thesis in Applied Linguistics. Pontificia Universidad Católica de Valparaíso, Chile.

G. Parodi (2008). Lingüística de corpus: una introducción al ámbito. RLA. Revista de Lingüística Teórica y Aplicada. Concepción (Chile), Vol. 46, No. 1, pp. 93-119. Retrieved from: http://www.scielo.cl/pdf/rla/v46n1/art06.pdf

I. Askehave and J. M. Swales (2001). Genre identification and communicative purpose: a problem and a possible solution. Applied Linguistics Vol. 22, No. 2, pp. 195-212.

J. Cole, J. Derrick, Z. Milosevic and K. Raymond (2001). Author Obliged to Submit Paper before 4 July: Policies in an enterprise specification, in: Proceedings of Policy Workshop, Bristol, UK.

J. M. Swales (1990). Genre Analysis: English in Academic and Research Settings. University Press, Cambridge.

J. M. Swales (2004). Research Genres: Explorations and Applications. Univ. Press, Cambridge.

J. P. Jayan and R. V. Raj (2011). Parts of Speech Tagger and Chunker for Malayalam –Statistical Approach. Computer Engineering and Intelligent Systems, Vol 2, No.3.

J. Renkema (2003). En torno a la solución del problema de Orwell en la comunicación gubernamental: Investigación experimental de la estructura de la información en los sitios Web. Revista Signos, 36(53), 103-120.

J. Yates (1989). Control through communication: The rise of system in American management. Baltimore, MD: Johns Hopkins University Press.

M. McCarthy and M. Handford (2004). Invisible to us: A preliminary corpus-based study of spoken business English. In: U. Connor & T. Upton (Eds.), Discourse in the professions. Perspectives from corpus linguistics, pp. 167-202. Benjamins, Amsterdam.

M. Warren (2004). So what have you been working on recently? Compiling a specialized corpus of spoken business English. In: U. Connor & T. Upton (Eds.), Discourse in the professions. Perspectives from corpus linguistics, pp. 115-140. Benjamins, Amsterdam.

M. Wyme (2007). Developing linguistics corpora: a guide to good practice. Oxbow Books, 2005.

P. F. Linington, Z. Milosevic, J. Cole, S. Gibson, S. Kulkarni and S. Neal (2004). A unified behavioral model and a contract language for extended enterprise, Data & Knowledge Engineering, Vol. 51, Issue 1, pp. 5-29, ISSN 0169-023X, http://dx.doi.org/10.1016/j.datak.2004.03.005.

P. Vossen (2002). EuroWordNet General Document. Version 3. Piek Vossen (ed.). University of Amsterdam.

R. Venegas (2005). Hacia una identificación automatizada de rasgos argumentativos en corpus. En G. Parodi (Ed.), Discurso especializado e instituciones formadoras. Ediciones Universitarias de Valparaíso, Valparaíso.

S. Neal, J. Cole, P.F. Linington, Z. Milosevic, S. Gibson and S. Kulkarni (2003). Identifying requirements for business contract language: a monitoring perspective. Proceedings of 7th International Conference on Enterprise Distributed Object Computing (EDOC'03), Brisbane, Australia.

S. Schmitz . Approximating Context-Free Grammars for Parsing and Verification. Ph.D. Thesis, Universit´e de Nice, Sophia, Antipolis.

T. Baldwin and K. S. Nam (2010). Multiword Expressions. In: Handbook of Natural Language Processing Nitin, Indurkhya and Fred J. Damerau (Eds.), Second Ed., CRC Press, USA, pp. 267-292.

T. Kuhn (2010). Controlled English for Knowledge Representation. Ph.D. Thesis, University of Zurich.

T. Sugino and F. Ito (2008). How to write a better English thesis. Natsume-sha.

Y. Someya (1999). A Corpus-based Study of Lexical and Grammatical Features of Written Business English. PhD. Thesis, Department of Language and Information Sciences, University of Tokio.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 6

VALIDATION

Results! Why, man, I have gotten a lot of results. I know several thousand things that won't work.
-- Thomas A. Edison

We present the implementation of the formalization in a functional prototype, and its validation by focusing on empirical methods. The goal of the validation process guiding this Chapter is demonstrating: i) the approach we propose can be applied on business-based technical documents regardless of the organizational process involved; ii) the feasibility of identifying organizational domain knowledge from the documents is feasible; and iii) the activities related to the domain understanding can be executed in a low-costs RE process.

To this extent, here we report our experience in the analysis of three case studies whose performance issues have been boarded with the usage of our proposed model. We discuss the advantages and the disadvantages both in the prototype usage and in the implementation of the model. Also, we compare the cases running out on the prototype in order to identify the suitable features the software systems should have.

6.1 EMPIRICAL VALIDATION

According to research methods in the field of software engineering, we define a validation design supported on the principles of scientific and engineering method (Glass, 1994), since the proposed model was built by observing the world and following the applied engineering, grounded on several current solutions to the problem. The design is based on the empirical method, considering the proposed model is generated from empirical strategies. The design type is fixed as qualitative/quantitative in nature, given that case studies and proofs of empirical observation were used on each case.

6.1.1 Hypotheses for the validation

The following hypotheses were formulated for the empirical validation:

- Hypothesis 1. An automated transformation process based on technical documents introduces fewer and less severe errors related to the ambiguity and subjectivity of the information than the ones analysts identify in the tasks of natural language processing.
- Hypothesis 2. Software Engineers spend less rework effort due to incorrect or incomplete information when using a technique based on technical documents for identifying, specifying, and understanding the knowledge within the process elicitation, than the same Engineers when using subjective and synthetic techniques of process elicitation.
- Hypothesis 3. A technique based on technical documents can complement the well-known elicitation techniques during the requirements elicitation process and can compensate the weakness of such process in terms of the implicit knowledge identification activities.

For the sake of operationalizing the validation goals, we addressed the problem according to the empirical method. Specifically, we follow the case study strategy and the application of validation studies taking place in the context of a class-room and academic projects, as we show in the next subsections.

6.1.2 Empirical validation strategies

The **empirical method** is based on **quantitative research**, as a way to quantify the cause-and-effect relationship in the analysis of the transformation process. The research is conducted by collecting data

© Bell Manrique-Losada, 2014. All rights reserved.

based on case studies and setting up controlled validation studies. This kind of research method is appropriate for the analysis, since we are attempting to assess the effect of the transformation model of different technical documents on the results, in terms of the controlled language text generated. Besides, the data from the real world are analyzed in order to understand the connections and interactions expressed in the data measured from the model. According to Wohlin *et al.* (2000), the strategies carried out in this validation are focused on **case studies**, their validation studies, and the consequent empirical observation, as follows.

A **case study** is an observational method used for monitoring projects. In this method the data are collected from a body of elements—universe of discourse—into corpora—formed by corpus of document categories—which aimed to identify linguistic patterns characterizing each document type or corpus category. We defined a case study because it is useful as an observational study, other than the controlled study supported by an experiment (Zelkowitz, 1998). By using the case-study strategy the empirical validation was structured for defining the qualitative features, thereby facilitating the design and implementation of the validation studies as an internal strategy for each case. As usual, the case study is aimed at tracking a specific attribute—elements of the structure from the input text written in natural language—based on applying several patterns and features and establishing relationships among different attributes. Then, based on a data collection and the results of the validation studies, statistical analysis can be carried out.

In this Ph.D. Thesis, we define the **validation study** according to Lilienblum *et al.* (2008), as a small-scale inter-laboratory study which is designed to obtain the preliminary assessment of the reliability and relevance of the study within specific conditions. In this kind of studies different measurements are made on the same situation searching for some level of agreement among them. A validation study generates evidence of a situation/scenario for supporting future performance. In this way, when a situation has been validated by a study, the results ensure the future performance in order to support the decision making process.

According to Armbrust (2003), a validation activity is oriented to gather empirical findings with the purpose of confirming a hypothesis, *i.e.*, comparing the results of one extraction—automated transformation by using a software application—with the baseline results—by-hand transformation made by an expert. The validation studies enable the evaluation of assigned variables in different case studies. Such validation takes place in an environment similar to a laboratory by manipulating controlled variables and fixed level variables. Further, they are measured for performing a statistical analysis.

The validation studies are intended to distinguish between two situations: a controlled situation (baseline) and the situation under investigation. In our case, such situation is the method for inspecting, extracting, and transforming discourses from a technical document and to generate controlled language. In Figure 6.1 we show the empirical strategies used in the experimental design. The general process is developed by following the case study strategy. Two cases are included in this study for the business-based technical documents under study: *Standard Operating Procedures* and *Job Specification Document*. The same design was used in the two cases, as shown in Figure 6.2.

Figure 6.1. Design to the validation study

© Bell Manrique-Losada, 2014. All rights reserved.

We selected the case study strategy and then, we implemented a set of validation studies as a way to compare the transformation method in all technical documents. Some facts have been considered in such selection: the dependence on the scale of the evaluation; the ability to isolate factors; and the feasibility for randomization (Wohlin *et al.,* 2012) needed for the variables analyzed by studies. The case studies were executed from a point of view with high level of abstraction, while the validation studies were executed with low level of abstraction, *i.e.*, they were used to evaluate how the transforming process occur, and what the differences between the two methods—baseline and under investigation—are.

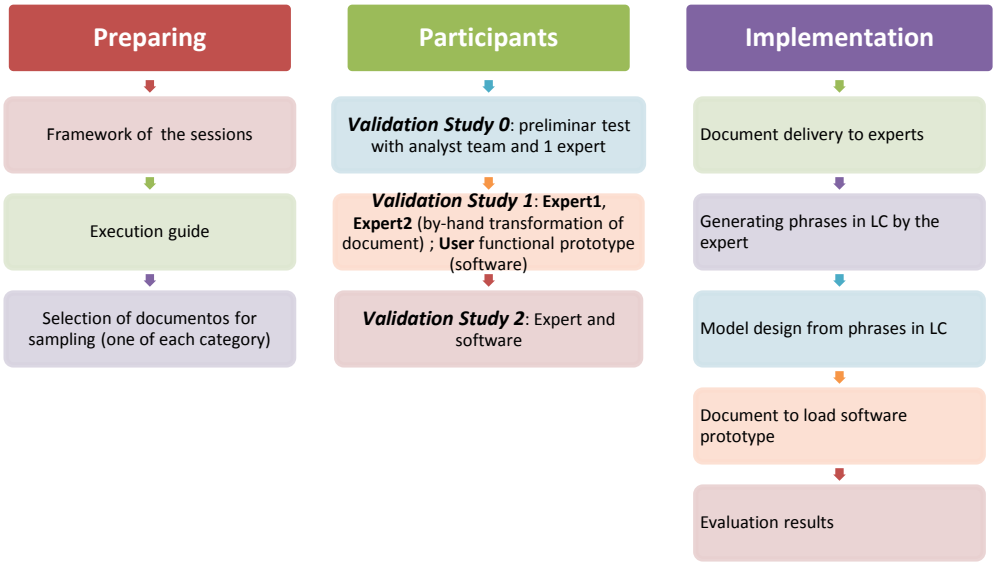


Figure 6.2. Design process of validation studies

The validation study was conducted in three separate and consecutive phases. The first phase (validation study 0) was conducted as part of an undergraduate student room project, where the acceptability, reproducibility, and validity of the transforming process of the documents were evaluated in a sample of 20 students. By means of a preliminary pilot, we defined whether a set of analysts for identifying the baseline for comparing the results or one analyst—acting as an expert—should be used. Additionally, based on this study we identify the categories or types of rules to be implemented in the prototype. In the second phase (validation study 1) we equate the CL identified by each of two experts against the CL previously identified and assumed as a baseline. We were looking for reducing the subjectivity implied in the by-hand transformation process of the document, codifying patterns and transformation rules. And, in the third phase (validation study 2), we compare the CL identified by the expert and by the software, after we obtained the feedback conducted in the previous validation study.

6.1.3 Organizing and conducting the case study

According to the foundations stated by Stake (1995) and the guidelines defined by Runeson & Host (2008), we followed the next procedure for case study research for organizing and conducting the research successfully.

i) **Defining the research questions**

Text analysis or documents review techniques are methods for identifying the useful business-based information of the RE process (Keller, 2011; Stein *et al.,* 2009). We carried out this method by using the analysis of the discourses involved and the discursive genres used in technical documents containing such information.

We were primarily interested in this research on determining whether or not the technical documents are valuable in some way for identifying business-based information. The research process began with the state-of-the-art review for determining the prior studies about this issue. Then, we used such a

© Bell Manrique-Losada, 2014. All rights reserved.

review for defining several questions related to the study of mapping NL discourses contained in business-based technical documents into a CL expressing domain information and knowledge:

- How do the writers of business-based technical documents determine what to organize in the structure and contents of such documents?
- What is the internal organizational structure of a business-based technical document enabling the automated processing for the identification of useful domain information?
- How do the constitutive features of the discourse genres followed by the analyzed technical documents are presented?

ii) Selecting the cases and techniques for data gathering and analysis

Following the case study strategy, we defined the empirical validation design comprising two cases for the business-based technical documents under study: *Standard Operating Procedures* and *Job Specification Document*. In the structural and analytical level the case-study strategy eases the characterization of each kind of technical document by following a set of specific and unique attributes/elements of the document structure. Such characterization was possible based on the identification of several pattern and features and the establishment of relationships among them.

For analyzing the documents, we use a methodology based on corpus linguistics (Simpson & Swales 2001; Tognini-Bonelli, 2001), according to the proposal developed by Parodi (2005a). We aimed to perform a descriptive analysis of linguistic texts from business-based technical documents, but focused on procedures. Specifically, we use one called SOP as a linguistic genre. From the processes of searching, reviewing and analyzing documents from the Web, we define—for each case—a hierarchy composing the macro-genre, based on the genre analysis theory (Swales, 2004). In addition, we use several tools and techniques for characterizing each case as a macro-genre and the genres included in each one from the genre point of view. Most of them are oriented to the exploration of word usage, frequency, collocation, and concordance (O'Keeffe, 2003) on documents, as follows:

- AntConc 3.5w—a freeware concordance program (Anthony, 2009).
- TermoStatWeb[12](Drouin, 2003)—as a term extraction program for automatically extracting single-word units and multi-word units which have high frequency in the text.
- ElGrial[13]—a computational interface for tagging and parsing texts from corpora collected and stored online (Parodi, 2005b).
- NLTK-Demos[14] with different word tokenizers, stemming text, part of speech tagging, and chunk extraction modules.

We choose a qualitative method for text analysis, the discourse analysis. This choice seemed obvious since this method allows for the researcher: i) to interpret the findings; ii) to reach conclusions undetectable to a quantitative method; and iii) to achieve depth knowledge about a specific area. According to Molin (2007), since the researchers are focused on a smaller amount of data, then they are able to conclude a lot about little information by using a qualitative method. Moreover, such a method have an advantageous aspect for analyzing text, since facilitate a deeper approach, with higher validity and with the ability to measure what you set out to measure (Østbye *et al.,* 2003).

iii) Planning and collecting the data

We collected and analyzed a set of documents from the corporate domain in different subject fields such as medicine, forestry, and chemistry/physics laboratory. We built a corpora with two sub-corpus consisting of the two different groups of texts collected, corresponding to the case studies. The resulting corpora comprise 75 texts, equivalent to a total of 199,532 words. This general corpus is divided into two sub-corpora as follows:

[12] Available in: http://termostat.ling.umontreal.ca/index.php?lang=en_CA
[13] Available in: www.elgrial.cl
[14] Available in: http://text-processing.com/demo/

© Bell Manrique-Losada, 2014. All rights reserved.

- **Sub-corpus1.** 25 texts of English-written documents with independence of its variety, and which sum 31,627 tokens and 3,839 word types under the category *Job Specification Document* –JSD.
- **Sub-corpus 2**. 50 texts under the category *Standard Operating Procedures* –SOP, which sum 167,905 Tokens and 9,252 word types.

The SOP corpus was collected first than the JSD sub-corpus, with the purpose of developing comparative procedures among several logs and thus providing an accurate description of the collecting and organizational method. According to Parodi (2007), the unequal number of texts per corpus is not relevant for comparison purposes and for considering the standards currently employed in corpus linguistics of English texts.

iv) Evaluating and analyzing the data, and preparing the report

Each business-based technical document under study is reviewed, analyzed, and characterized, as we describe below:

– **SOP: *Standard Operating Procedures***

The present qualitative characterization following the genre point of view (Swales, 2004) has the *manual* macro-genre and the *procedures manual* genre. The term *manual* applies to academic manuals, instructional and teaching textbooks, and technical procedure manuals (Parodi, 2008). In this work, we use the meaning of procedures manual referring to the latter category. Also, we can identify several sub-types, using the idea of genre.

The *manual* macro-genre or genre grouping corresponds to the name bringing genres of a particular area, a usage domain, or a basic purpose. In this Chapter, we accept the concept of macro-genre defined by García (2009) as "*an abstract class that groups texts with a function and audience shared*". After the analysis process on the compiled documents performed according to Parodi (2008), we defined the following genres included in the *manual* macro-genre: *quality manuals* (e.g., function and procedure manuals) and *usage manuals* (*e.g.* user manuals).

We defined the following hierarchy of the *manual* macro-genre by searching, reviewing and analyzing Web documents and based on the theory of genre analysis (Swales, 2004), as we show in Figure 6.3. A ***Manual*** is a specialized technical document in the business-based context—corporative or organizational—specified in: *Usage Manual* and *Quality Manual*. The *Usage Manual* is a technical document of a particular system, device, application, or equipment which tries to provide assistance to a single user or a group of users. Commonly, this manual is associated with an electronic device, physical component (*e.g.*, computer hardware), or software application and includes User Manuals and Technical Manuals. The latter are primarily distinguished by the type of audience whom are addressed: the first addresses a wider audience, without specific knowledge base in the interest area, while the second one is aimed to an audience with some expertise on the issue.

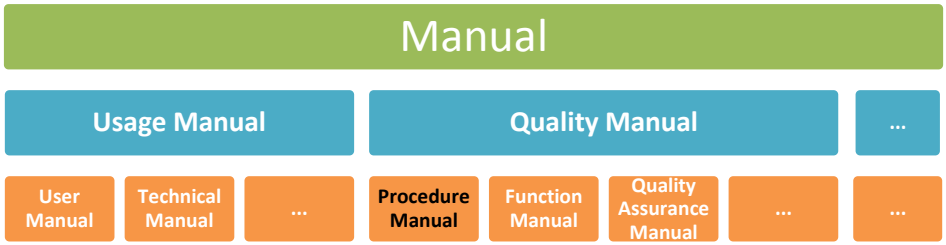


Figure 6.3. Hierarchy of *manual* macro-genre

On the other hand, the *quality manual* is a document specifying the quality policies, along with their features (*e.g.*, mission, vision, principles, goals, and processes), aiming to implement those

© Bell Manrique-Losada, 2014. All rights reserved.

policies. Commonly, this document is public and establishes how to apply quality assurance standards in organizations, the main components of a 'Quality Management System.' A *quality manual* contains the Function Manual, the Procedure Manual, and the Quality Assurance Manual. A Function Manual describes the job location and functions, tasks and responsibilities. A Function Manual is intended to define and divide the work of an organization, to know and reformulate the responsibilities of positions and, therefore, to facilitate the task of functional control. A *Procedure Manual* is a document describing the systematic operational procedures associated with administrative units. This document contains the rules, activities, and steps required for the process compliance. Finally, the Quality Assurance Manual contains the set of actions that are necessary to provide adequate confidence that a product or service will satisfy quality requirements.

Figure 6.4 shows the realization of the Procedures Manual in the way of Protocols, Instructional Manual, and Analytical Manual.

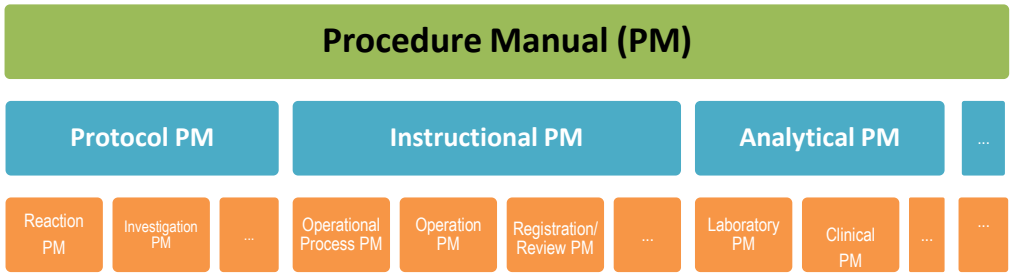


Figure 6.4. Classification of the procedure manual genre

Protocol category establishes a set of standards, methodologies, and/or formulas to react in situations, achieving a unique style in the company. These protocols can be Reaction Prevention Protocols (*e.g.*, Disaster Prevention Manuals) or investigation protocols (*e.g.*, Research Methodology Manuals).

Instructional category comprises the procedure manuals attempting to describe, in terms of activities and steps, sequences to follow instructions for performing certain tasks. Three kinds of documents belong to this category: operational process (*e.g.*, Operating Procedure Manual of Tropical Forest Management), Operations (*e.g.*, Operation Manuals for a recycling machine), and registration/review (*e.g.*, Registration Manuals of laboratory samples).

Finally, *Analytical* category refers to those methods involving analysis and systematic review activities, methodological and/or scientific phenomena. These include Laboratory (*e.g.*, Procedures Manual for the analysis of organic carbon), or clinical processes (*e.g.*, Procedures Manual for labeling laboratory samples).

According to Parodi *et al.* (2008), we can characterize a particular genre based on the constitutive features of the discourse genres and the conjugation of their criteria and their operationalization into specific variables. Such characterization facilitates the accurate distinction and classification of the texts inside the corpus. The set of criteria and genre variables characterizing the operating procedures are the following:

- *Communicative macro-purpose*. To standardize behaviors and/or procedures.
- *Type of discourse organization*. Descriptive, due to names, locations, and qualifying objects, people, or processes. Narrative, it may have actions in a temporal order looking for a specific purpose.
- *Relationship among participants*. Expert writer vs. expert reader, and/or Expert writer vs. semi-layman reader.
- *Ideal context of circulation*. Whether it is running in a technical / professional environment.

© Bell Manrique-Losada, 2014. All rights reserved.

- *Mode*. Verbal (written) and nonverbal. In some cases, it includes scientific formulas, images, and drawings, among others. It is mono-modal, whenever occurs predominantly as a way (verbal), despite including nonverbal information.
- *Terminology*. Contains a large number of specific terms belonging to the body of knowledge and a lot of information shared between sender and receiver. Such information is produced by individuals who possess specific knowledge of a subject.
- *Scope*. Intra-professional, due to circulation within a given professional field and among members of that community (Gunnarsson, 2004).

– JSD: *Job Standard Document*

In this section we present the qualitative characterization of the *Job Standard Document* macro-genre and the *Job Specification Document* genre.

A *Job Standard Document* (JSD) is the term we used in this Ph.D. Thesis as a kind of a specialized technical document in the business-based domain. JSD is a document similar to the *Statement of Work* (SOW) as a formal document for capturing and defining the work activities and deliverables, but JSD is not oriented to a specified work for a client, but for a specific job in an organizational hierarchical structure. In a JSD several elements are standardized terminology—concepts; overall agreement terms—conception constraints; and a set of operations for creating, ending, and monitoring jobs, including desirable skills and resources to opt for a job.

A JSD comprises different documents written and specially designed as a summary of significant features from job analysis to characterization. Based on our analysis and review, such documents are classified as we shown in Figure 6.5: *Job Description*, *Job Specification*, and *Job Evaluation*. These are the names given to the by-products of job analysis process and the job characterization process. Thus, these documents are different in nature but complementary. All of them help in the proper selection, training, and development program, as well as in job evaluation.

Figure 6.5. Classification of *Job Standard Document macro-genre*

Job Description –JD is a document for describing in general terms a job such as is the standard of a job. This description outlines the main responsibilities and expectations of the job (position), as well as any other significant activities that the candidate opting for such position may be asked. According to Swanson (2007), the process to make a job description starts with a strong reflection on the available sources of expertise. In this sense, the target audience belongs to the organization (the employers).

Job Specification –JS is a structured technical document of factual statement of functions and objectives from a job. JS describes person's ability and qualifications required to perform the job. JS is the reflection of the ideal person required to do the job. JS is the result of the job analysis (Sackett & Laczo, 2003), which helps to determine the qualifications needed for each vacant job (Motwani & Mehrotra, 2013). In this sense, the target audience is external to the organization.

Job Evaluation –JE is the document for guiding the implementation of evaluation activities of a job. JE is aimed to an audience with some expertise on the job and on the organization. JE provides information and recommendations about procedures to support the job evaluation (NHS, 2013). Commonly, a JE defines the job evaluation processes in terms of: the objectives and parameters of the evaluation, the requirements for the evaluation process, an optional guidance for planning and preparing evaluation projects, and for reviewing the evaluation outcomes.

© Bell Manrique-Losada, 2014. All rights reserved.

While job description contains the main features of the job, job specification is the outline of personnel qualification regarded as essential for performing a job. A job evaluation is intended to support the evaluation process of a specific job.

In this genre the texts inside the specific corpus can be classified by using a set of criteria and genre variables characterizing JS documents are as follows:

- *Communicative macro-purpose*. To standardize the job activities and the operations for creating, ending, and monitoring jobs.
- *Type of discourse organization*. Descriptive, due to names, locations, and qualifying objects, people, or processes. Narrative, it may have non-temporal options in order to relate the requirements or constrains related.
- *Relationship among participants*. Expert writer vs. expert reader, and/or Expert writer vs. semi-layman reader.
- *Ideal context of circulation*. Organizational/business/professional environment.
- *Mode*. Verbal (written).
- *Terminology*. Contains a large number of specific terms belonging to the business body of knowledge and a lot of information shared among stakeholders. Such information is produced by individuals who possess specific knowledge of a subject.
- *Scope*. Intra-professional for the *Job Description*, due to circulation within a given professional field and among members of that community (Gunnarsson, 2004). In the case of *Job Specification* the scope is broader because of the target audience is external to the organization/business.

In the criteria and variables, we considered both the internal features of the texts, and the external context in which they are produced and used. This classification has been proposed from the empirical point of view, and it is conceived under the idea that the *manual* macro-genre and the *job standard* macro-genre like other discourse genres, are a highly complex and dynamic units essentially belonging to the cognitive, social, and linguistic dimensions (Parodi *et al.* 2008).

6.1.4 Description and Results of the Validation Studies

i. Validation Study 0 (preliminary)

The ***validation study 0*** is designed by following a data collection method. Such method is applied for defining whether an analyst team for identifying the baseline for comparing the results or just one analyst—acting as an expert— should be used. Both of them—analyst team and expert—have the knowledge, experience, and previous contextualization of the natural language transforming process from technical documents.

Students were instructed in the application process of the natural language processing, along the undergraduate course. Moreover, the students developed several workshops by applying and using the controlled language, as an output language used to specify the transformation results.

After providing informed consent, a pilot study involving the first 20 students with the best indicators following instructions and practicing the learning of the training was conducted. The method is used in two different environments to study two technical documents developed by organizations with different backgrounds and business vision, by using very different writing strategies. In both environments it yielded answers to most questions of interest and some insight into its use.

The pilot study has a limited number of participants who brought valuable information for the analysis. In this pilot, we use complete sessions with the analyst team as units of analysis and 5 teams (corresponding to 20 students) were analyzed. Details on the methodology are given in the following paragraphs with the findings of the study and we present the complete results of this pilot study in Wopatec 2012 (Zapata & Manrique, 2012). We follow the next methodological procedure

© Bell Manrique-Losada, 2014. All rights reserved.

in the pilot study, which we exemplify in this chapter just with a small piece of text of the original technical document:

a. *Set of goals to be satisfied*
The goal of the pilot is the analysis and identification of phrases written in UNLencep from the document to be assigned.

b. *Conforming work teams*
The work teams are formed by students belonging to a course of Requirements Engineering from the Universidad Nacional de Colombia–Sede Medellín—from 4th to 7th semesters of the Systems Engineering and Informatics program.
The students had previous experience in the analysis process of technical documents and in the representation of extracted information into pre-conceptual schemas.

c. *Assigning and reading of technical document*
The technical document assigned is "*One Hundred Rules for NASA Project Managers*" (Madden, 2003). This document is considered technical since it is commonly used by NASA directives for training their projects managers.

The reading and analysis of the document is developed by each work team. The following piece of text of the cited technical document is used as a small sample of the pilot and the analysis process developed by the teams:

> *"A project manager should visit everyone who is building anything for his project at least once, should know all the managers on his project (both government and contractor), and know the integration team members. People like to know that the project manager is interested in their work and the best proof is for the manager to visit them and see first-hand what they are doing".* (Madden, 2003, 2pp).

d. *Step-by-step analysis and identification of a set of phrases in a controlled language*
The analysis of the technical document carried out in the pilot answers some questions of interest. The procedure relies heavily on an interactive analysis process; some students identify interesting data and the phrases matching the mapping rules and then—concurrently with the peer analyst—review the document and refine the phrases identified. We present in Table 6.1 a sample of the extracted phrases by two work teams.

Table 6.1 Sample of extracted phrases by work teams

Work Team 1:	Work Team 2:
✓ *PROJECT MANAGER* **visit** *TEAM.*	✓ *PROJECT MANAGER* **has** *PROJECT.*
✓ *PROJECT MANAGER* **has** *TEAM.*	✓ *PROJECT MANAGER* **visit** *TEAM.*
✓ *PROJECT MANAGER* **has** *PROJECT DIRECTOR.* *PROJECT DIRECTOR* **is** *GOVERNMENT.*	✓ *PROJECT MANAGER* **visit** *MANAGER.*
✓ *PROJECT DIRECTOR* **is** *CONTRACTOR.*	✓ *PROJECT* **has** *MANAGER.*
✓ *TEAM* **build** *PROJECT.*	✓ *PROJECT* **has** *TEAM.*

e. *Phrase representation in a conceptual model*
Each team represented the extracted phrases into a pre-conceptual schema. The representation of the previous phrases for the work team 1 is presented in Figure 6.6 and for the work team 2 in Figure 6.7.

© Bell Manrique-Losada, 2014. All rights reserved.

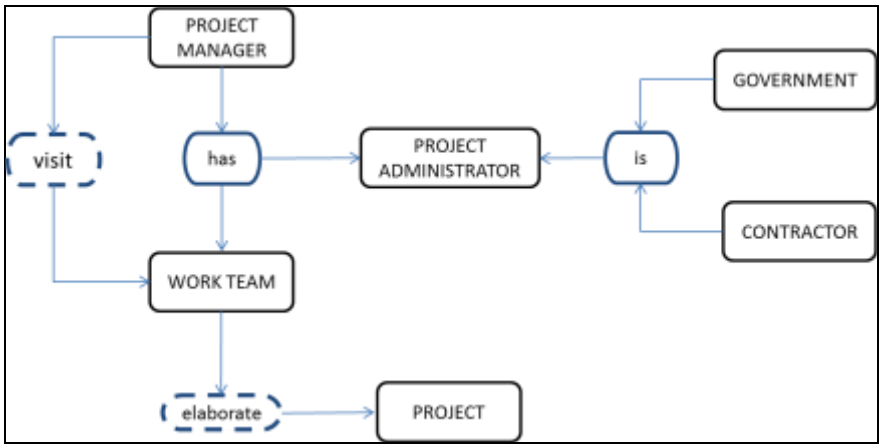


Figure 6.6 Phrases representation from Work Team 1.

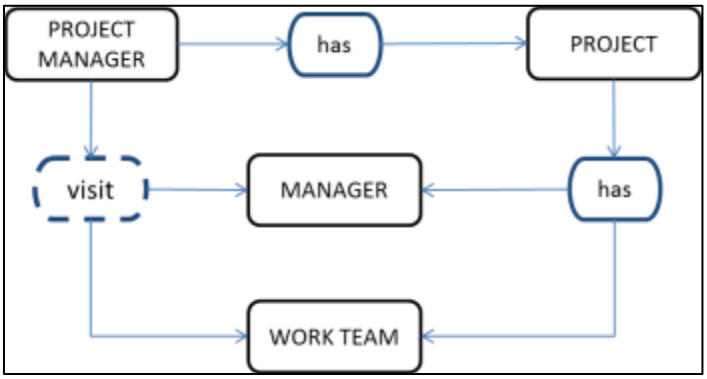


Figure 6.7 Phrases representation from Work Team 2.

f. Validation mechanism

The validation mechanism comprises the comparison of both pre-conceptual schemas made by the work teams against the same schema made by the expert.

The pre-conceptual schema from Figure 6.8 was drawn by the expert. As we show, the quantity of concepts and relationships considered by the expert is bigger than the work teams discovered.

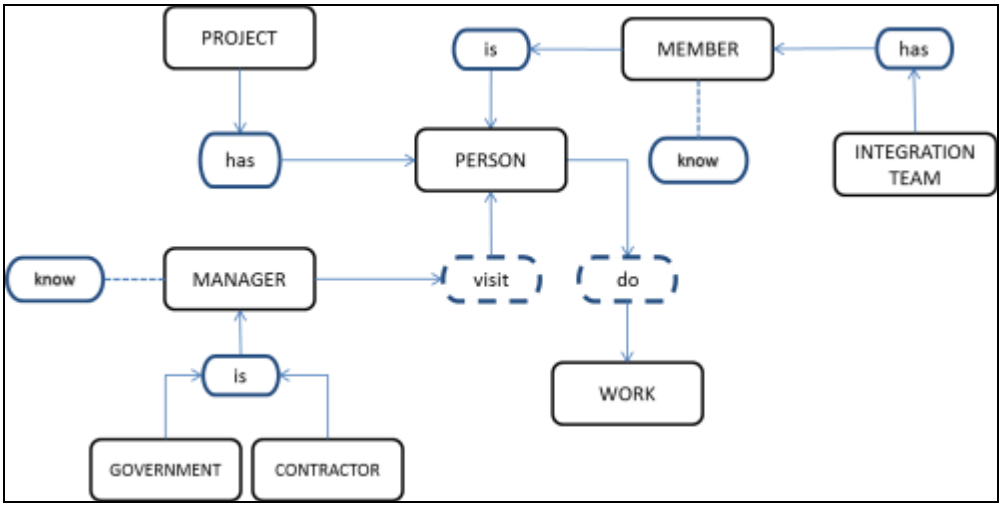


Figure 6.8 Phrases representation by the Expert.

© Bell Manrique-Losada, 2014. All rights reserved.

The schemas generated by work teams have significant differences among them. Further, the differences are bigger against the schema drawn by the expert. Some of the features from the by-hand analysis of the technical document and the information extraction process are:

- Identifying concepts non-existent in the original text and including them in the phrases and schemas generated
- Assigning dynamic relationships nonexistent in the text
- Defining structural relationships inconsistent with the text

The pre-conceptual schema drawn by the expert exhibits the following features:

- A direct relation among the concepts, verbs, and relationships from the schema and some grammatical element from the text is discovered
- The expert schema comprises more elements than the schemas designed by work teams.
- With the exception of the achievement relationship 'know,' all of elements identified by expert are represented in different ways in the student schemas.

Based on the previous features, we decide to support the next validation studies only with the intervention of an expert. Other reasons for such decision are as follows:

- The pre-conceptual schema is a completely unambiguous controlled language in which the elements require interpretation process from the analysts (*e.g.* the analyst should even though some other times its used in the original text). Besides, for drawing the schema, the analyst should solve all the conjunctions and relative pronouns used in the text, because of the triad structure of the pre-conceptual schema.
- The interpretation process is still subjective, due to the identification of the elements within the text carried out by the analyst. Also, the additional information from the analyst could influence the generation of additional elements implicit in the text.

Results from the first study show that an additional performing of the experts could support the accuracy of transforming. In addition, we should validate just one of the intervention in the formal validation.

ii. Validation Study 1:

This training experiment had two goals: i) comparing the CL identified by each one of several experts, and ii) identifying the preliminary rules for transforming a business-based technical document. We were looking for reducing the subjectivity implied in the by-hand transformation process of document, by means of the patterns and transformation rules to be codified. This study is very much related to the first, as the attributes somehow correspond to the specific observed variables: number and type of rules for interpreting/understanding a document, quantity of identified phrases in a CL, and distinguished features of the transformation process. The terms of the study are as follows:

With this study, we aim to illustrate, describe, and discuss the practice of applying by-hand document analysis and transforming a text by using a processing method, based on the variables observed. For reasons of space, in this section a sample of the analysis carried out is presented, as a practical example. The complete analysis and study results were published by Zapata, Manrique, and Gasca (2012). We assume the following aspects: i) the document is written in a natural language (NL) discourse; ii) the experts acting as analysts in charge of the document analysis have knowledge about the domain, terminology, language used on the document, and controlled language of output; and iii) the experts have the sufficient experience in the identification of patterns matching with the rules for transforming NL into a CL text. The process takes place when each expert:

- Defines the source document.
- Analyzes the assigned section and the contents of each one.

© Bell Manrique-Losada, 2014. All rights reserved.

- Identifies fundamental concepts and the interaction among fundamental concepts by means of actions/verbs.
- Expresses the information extracted in phrases from a specific controlled language (CL).
- Defines rules allowing for the transformation process from NL phrases into CL phrases.

The step-by-step development of the process, is the following:

a. Description of the source document

We take as source document a specific procedure manual. This manual is considered a technical document, because it defines and deploys a set of instructions and rules that a particular organization is intended to comply. The document is MORE IEEE (IEEE, 2012), by its Spanish name "*Manual de Operaciones de Ramas Estudiantiles IEEE*." IEEE is the Institute of Electrical and Electronics Engineers, Inc.

MORE is the IEEE student branch operation guide, which is designed to guide universities in their activities in the student branch level. MORE covers branch administration, branch operations, membership, activities, and detailed description of the awards and realizations of the students. The branch operation guide is a technical document categorized as manual, because it has significant material from the leadership training manual and supply. Such a guide is a reference to motivate and conduct student volunteers of regions for performing effectively as branch officers.

For this validation study with the experts, we defined five sections of the document. We choose the following sections because we can analyze the findings and the rules generated by the analyst groups:

- *Section 1: Introduction*. It is focused on the general objective of the student branch. Also, the common organizational aspects like mission, vision, growth, and globalization are included.
- *Section 2: IEEE organization*. This section describes the organization as sectors of the geographic regions as well as the governing bodies of high-level rules directing the activities of the student branches.
- *Section 3: Branch composition*. It includes the definition of the student branch, the levels of formation activities, and the steps for creating a student branch.
- *Section 4: Branch direction*. It describes responsibilities, clues, and composition of the administration board.
- *Section 5: Student branch affinity groups*. This section includes several kinds of affinity groups that a student branch could follow and the administration of those groups.

b. Extraction from Document

Manual analysis of the document was completed from a list of paragraphs which contains many of the main concepts. All the normative paragraphs identified (a set of phrases with a defined structure) were used for the extraction process. For the purpose of illustrating the manual extraction process as a whole, we show a portion of text from the Section 4 delivered to an analyst group:

Portion 1: "*The key for successfully driving a student branch is a good administration. The administration of a branch has several key positions. It is composed by the Board of Directors (President, Vicepresident, Treasurer, and Secretary), the Counselor and the Mentor. All directors should promote the benefits of the IEEE membership for the students.*"

After analyzing each section and its contents, the expert should identify the fundamental concepts and the interaction among them by means of actions/verbs. The preliminary results are summarized in Table 6.2.

© Bell Manrique-Losada, 2014. All rights reserved.

Table 6.2. List of concept and concepts interaction

Concept	Interaction	Concept
A student branch	Has	An administration
A student branch	Has	A successfully driving
A good administration	Is	key for a branch
Branch administration	has	Key positions
Administration	Is composed by	Board of Directors, Counselor, Mentor
▪ Administration	▪ has	▪ Board of Directors
▪ Administration	▪ has	▪ Counselor
▪ Administration	▪ has	▪ Mentor
Board of Directors	Is composed by	President, Vicepresident, Treasurer, Secretary
▪ Board of Directors	▪ has	▪ President
▪ Board of Directors	▪ has	▪ Vicepresident
▪ Board of Directors	▪ has	▪ Treasurer
▪ Board of Directors	▪ has	▪ Secretary
Directors	Promote	Benefits
IEEE membership	Has	Benefits
IEEE	has	Membership
IEEE membership	Is for	students

c. Identification of rules for generating CL and Representation in CL

In this step, the experts should express the information extracted into specific CL phrases. The language selected was UN-LENCEP (*Universidad Nacional de Colombia—Lenguaje Controlado para la Especificación de Esquemas Preconceptuales*, a controlled language for specifying pre-conceptual schemas). UN-LENCEP is a subset of the NL that has been created for writing specifications translatable into pre-conceptual schemas.

Table 6.3 has the preliminary analysis of the resulting annotations for sections, by categorizing the UN-Lencep extracted phrases and the rules reported of the process made by-hand by the analysts.

Table 6.3. List of phrases with concept interaction

Controlled natural language expression	Formal construction	Transformation Rule
*rama estudiantil***<tiene>***administración*	A <TIENE> B	A posee B
rama estudiantil **<tiene>** *manejo exitoso*	A <TIENE> B	
*Administración***<es>***clave*	A <ES> B	
*rama***<tiene>***administración*	A <TIENE> B	B de una A
*Administración***<tiene>***Puestos clave*	A <TIENE> B	Hay B [Hay es igual a *pertenencia*]
*Administración***<tiene>***Junta Directiva*	A <TIENE> B	A está compuesto por B
*Administración***<tiene>***consejero*	A <TIENE> B	A (B, C, D) [Los paréntesis indican *pertenencia*]
*Administración***<tiene>***mentor*	A <TIENE> B	
*junta directiva***<tiene>***presidente*		
*junta directiva***<tiene>***vicepresidente*		
*junta directiva***<tiene>***tesorero*		
*junta directiva***<tiene>***secretario*		
*directivos***<promover>***beneficios*	A <ACCIÓN> B	
*membresía***<tiene>***beneficios*		
IEEE **<tiene>***membresía*		
*estudiante***<tiene>***membresía*		B es para A

For listing the bugs in the analysis of the experiment, we performed empiric verification of the findings, and according to Juristo and Moreno (2001), by attempting to prioritize the most important points to be documented or answered at each phase.

Thereby, we try to define the accuracy of the extracted phrases and utility of the rules generated by the analysts. The preliminary findings are the following:

© Bell Manrique-Losada, 2014. All rights reserved.

- Most information is extracted in terms of roles, responsibilities, and restrictions to the responsibilities.
- All found rule types can be structural or behavioral and in turn, simple (concept-concept) or complex rules (concept0 (concept1, concept2…)).
- The extraction by-hand from the experts should be evaluated regarding the standard used in the source document. The quality of the CL extracted is limited and directly related to the structure of the sections in the document.
- The experts found a big challenge to identify the subject for sentence fragments that appear in sub-paragraphs. In linearly written texts of procedures, the sentence fragments semantically relate to the last proposition that appears in the text. In this sense, for the expert is important to link the element used in the last sentence, with the sentence immediately following, to give sense to the phrases extracted. Such a task has a high grade of subjectivity and correlation in the process analysis.
- In the sections that we analyzed in this sample, we found several terms that we could generalize in this kind of technical documents, because they contain common type, events, roles, and information. An expert or analyst could improve the document analysis, based on an input like that, including a set of common terms belonging to the specific domain and similar terms.

In this second study we suggest and allow for making decisions related to:

- Distinguishing between different patterns of individual transforming process (spending a long time on the analysis, identification/specification of the controlled language) can be irrelevant for our purposes. Since our final goal is demonstrating the differences between the automated transformation and the by-hand transformation—made by an expert—so the patterns of the both transformation process are the same, regardless of the subject acting as an expert.
- Certain subjective aspects are implicit inside a transformation or analysis process from documents. Such aspects should be considered as less relevant for trying to automate the process; however we think to define a posterior process for validating the first results of an automated mapping process, where the intervention of the analyst imply less costs and time associated.
- We became aware about a possible bias by observing the participants and their behavior in the current level of validation. Despite some preliminary activities of the study aim to eliminate such bias, we are aware that there are preconceived notions implicit in human actions. When the experts first analyze the document, they exhibit a somewhat "unorganized" and "non-sequential" behavior as they probably are exploring the document before starting to identify the phrases matching with the mapping rules. This exploratory behavior is quite different from the prototype behavior and may have added a negative bias to our predictions.
- The observation of these variables inspired the introduction of the mapping speed attributes and thus, we assume that the usage of these attributes would improve the detection of the patterns of the mapping process made by each expert.

iii. Validation Study 2:

Comparing the CL generated by the expert with the one by the software, after the feedback conducted in study 1, we developed this final validation study.

In this study and according to Prechelt (2008) we consider as experimental variables the controlled variables (**input**), the observed variables (**output**) and fixed levels variables (**extraneous**), as follows:

a. **Input** variables:
   i. The inspection method: **by-hand method** or **automated method**
   ii. **Size** of the text moves

© Bell Manrique-Losada, 2014. All rights reserved.

b. **Output** variables:
   i. **Quantity of identified phrases**
   ii. **Mapping time**
c. **Extraneous** variables, as additional factors influencing the outputs
   i. Subjectivity of the discourses
   ii. Processor capacities for automated mapping process
   iii. Expert abilities for by-hand mapping process
   iv. Errors in the input text

As empirical study, for carrying out this validation study we assume an existing systematic relationship among inputs and outputs. The hypothesis was presented in the Section 6.1.1 which presented certain expectations about such relationship. The threats to validity to third study are related to extraneous variables describing the aspects they could get to invalidate the results, or could significantly affect these results.

This third study demonstrates the influence of contextual knowledge by an expert, for identifying, comprehending, and mapping domain knowledge before starting to elicit requirements. In order to assess the validity of the proposed design (Figure 6.2 in Section 6.1.2), different runs of mapping proofs were tested on the implemented application. The results and performance of the application were compared with the baseline—data and performance taken from the by-hand mapping by the expert. The adopted experimental application for the present validation study corresponds to the software prototype NAHUAL—building details will be described in Section 6.2.2.

Experimental Results of the Validation Study 2

The technical documents used for the validation study correspond to the SOP sub-corpus and the MPI-PR corpus category from 'Instructional Procedure Manual' (see Figure 6.4 from Section 6.1.3). Specifically, we selected the 'Operational Process' documents MPI-PR-004, MPI-PR-006, MPI-PR-020, and MPI-PR-006. Regarding to rhetorical moves from MPI-PR, for mapping rules design, implementing pre-processing, and validating the performance of mapping on NAHUAL App, we follow the model presented in Figure 5.6 (see Section 5.1.3). Such model was preliminary tested with experts, then validated in the previous validation studies—validation study 0 and validation study 1—and generated it internal structure change from 19 moves to 12, and finally to 8 moves. The final rhetorical moves of such model, used for pre-processing the input texts and documenting this validation study in NAHUAL, are summarized as follows:

- Macromove 1: Preamble/Overview
  - Move 1.1 Identifying SOP
  - Move 1.2 Introducing the SOP
  - Move1.4 Appointing regulations
  - Move 1.5 Establishing purpose
- Macromove 2: Development
  - Move 2.1 Defining procedure purpose
  - Move 2.2 Defining roles and responsibilities
  - Move 2.3 Identifying prerequisites
  - Move 2.4 Specifying procedure

The pre-processing module receives an original input text and returns a txt file containing only the sections corresponding with the previous moves. The module applies a kind of vector algorithm for finding the sections matching with any variable of name-vector—comprising a set of possible synonyms that can take a move—with the input text. For explaining this and the following processes, we document the validation study with the MPI-PR-020 document from the sub-corpus validation sample. In Figure 6.9 we represent the validation setup for pre-processing, based on the MPI-PR-020 document—testing-text.

© Bell Manrique-Losada, 2014. All rights reserved.

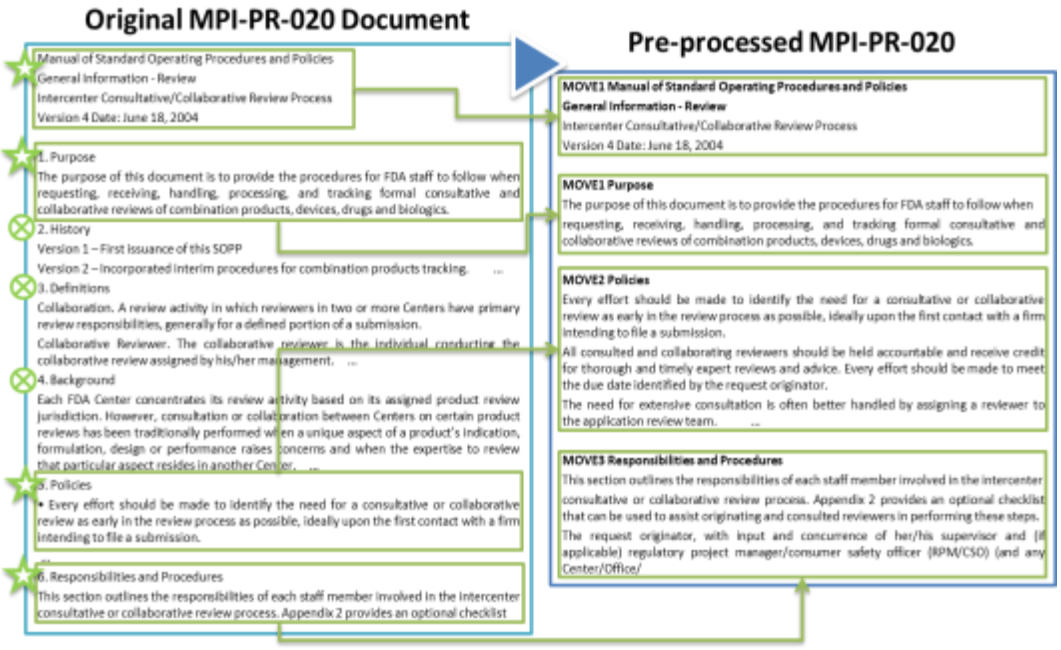


Figure 6.9 Setup for pre-processing of testing-text.

The pre-processed MPI-PR-020 is uploaded in NAHUAL and then processed. In this validation study, the parser was trained on the corpus with the reference grammar, which included the punctuation tokens, the conditions for matching, and reserved phrases identified. Additionally, the same pre-processed text is given to the expert, for performing the mapping by-hand. We show in Figure 6.10 some mapping steps, based on the components implemented in NAHUAL modules.

Manual of Standard Operating Procedures and Policies

[('**MOVE1**', 'NP', '<unknown>'), ('.', 'SENT', '.'), ('Purpose', 'NN', 'purpose'), ('The', 'DT', 'the'), ('purpose', 'NN', 'purpose'), ('of', 'IN', 'of'), ('this', 'DT', 'this'), ('document', 'NN', 'document'), ('is', 'VBZ', 'be'), ('to', 'TO', 'to'), ('provide', 'VV', 'provide'), ('the', 'DT', 'the'), ('procedures', 'NNS', 'procedure'), ('for', 'IN', 'for'), ('FDA', 'NP', 'FDA'), ('staff', 'NN', 'staff'), ('to', 'TO', 'to'), ('follow', 'VV', 'follow'), ('when', 'WRB', 'when'), ('requesting', 'VVG', 'request'), (',', ',', ','), ('receiving', 'VVG', 'receive'), (',', ',', ','), ('handling', 'NN', 'handling'), (',', ',', ','), ('processing', 'NN', 'processing'), (',', ',', ','), ('and', 'CC', 'and'), ('tracking', 'VVG', 'track'), ('formal', 'JJ', 'formal'), ('consultative', 'JJ', 'consultative'), ('and', 'CC', 'and'), ('collaborative', 'JJ', 'collaborative'), ('reviews', 'NNS', 'review'), ('of', 'IN', 'of'), ('combination', 'NN', 'combination'), ('products', 'NNS', 'product'), (',', ',', ','), ('devices', 'NNS', 'device'), (',', ',', ','), ('drugs', 'NNS', 'drug'), ('and', 'CC', 'and'), ('biologics', 'NNS', 'biologic'), ('.', 'SENT', '.')]
[('**MOVE2**', 'NN', '<unknown>'), (':', ':', ':'), ('Policies', 'NNS', 'policy'), ('Every', 'DT', 'every'), ('effort', 'NN', 'effort'), ('should', 'MD', 'should'), ('be', 'VB', 'be'), ('made', 'VVN', 'make'), ('to', 'TO', 'to'), ('identify', 'VV', 'identify'), ('the', 'DT', 'the'), ('need', 'NN', 'need'), ('for', 'IN', 'for'), ('a', 'DT', 'a'), ('consultative', 'JJ', 'consultative'), ('or', 'CC', 'or'), ('collaborative', 'JJ', 'collaborative'), ('review', 'NN', 'review'), ('as', 'IN', 'as'), ('early', 'JJ', 'early'), ('in', 'IN', 'in'), ('the', 'DT', 'the'), ('review', 'NN', 'review'), ('process', 'NN', 'process'), ('as', 'IN', 'as'), ('possible', 'JJ', 'possible'), (',', ',', ','), …

Figure 6.10 Sample of tagging of testing-text inside processing.

The testing-text was split by the parser and comprises 58 lines/phrases, 4307 characters, 644 words, and 26 sentences[15]. Based on by-hand analysis of an expert, we identified 127 potential relations—potential phrases for mapping into a CL (see Appendix 6.7). From such potential phrases, we compare the relations identified in the mapping by expert and by NAHUAL, for evaluating the output variable '*quantity of identified relations*'.

The complete results are included in Appendix 6.8, but a results sample of the mapping is compared in Table 6.4 and the summary of the results is in Table 6.5. The results comprise: in column 3 the potential relations as a reference, based on the rules and pattern proposed in this Ph.D. Thesis, implemented in NAHUAL, and given to the expert; in column 4 we state the number of relations identified by the expert, and in Column 5 the generated by NAHUAL; finally, in

[15] According to the information generated by http://textmechanic.com/Count-Text.html

© Bell Manrique-Losada, 2014. All rights reserved.

column 6 we include the relations generated by NAHUAL, but not based on the rules. As we show in some relation values, we count as a half relation (0.5) the cases when the returned relation is not complete regarding the rule-based relation, *i.e.*, if the rule-based relation is 'project manager file submission', the generated relation is 'project file submission', and such a result count as 0.5. In the same line, we identify as 'Out' relations (column 6) as the generated relations, but irrelevant or erroneous. The list of the relations identified is included in Appendix 6.9, and a sample is in Table 6.6.

Table 6.4. Sample of first 15 results of the mapping

#	Phrases	Number of Relations			
		Based on rules			
		Potencial	Expert	NAHUAL	Out
1	Manual of Standard Operating Procedures and Policies	1	0	0	
2	General Information - Review	1	0	0	
3	Intercenter Consultative/Collaborative Review Process	1	0	0	
4	Version 4 Date: June 18, 2004	0	0	0	
5	MOVE1 Purpose	0	0	0	
6	The purpose of this document is to provide the procedures for FDA staff to follow	3	3	3	
7	when requesting, receiving, handling, processing, and tracking formal consultative and collaborative reviews	2	2	1.5	1
8	of combination products, devices, drugs and biologics.	1	0	0	
9	MOVE2 Policies	0	0	0	
10	Every effort should be made to identify the need for a consultative or collaborative review	4	4	4	
11	as early in the review process as possible, ideally upon the first contact with a firm intending to file a submission.	3	2	2	
12	All consulted and collaborating reviewers should be held accountable	2	2	2	
13	and receive credit for thorough and timely expert reviews and advice.	2	2	1.5	
14	Every effort should be made to meet the due date identified by the request originator.	4	4	3.5	
15	The need for extensive consultation is often better handled by assigning a reviewer to the application review team.	3	0	0	

Table 6.5. Summary of relations comparison of the mapping

	Number of Relations		
Phrases	Potencial	Expert	NAHUAL
From 1 to 10	13	9	8,5
From 11 to 20	28	18	16
From 21 to 30	21	12	11,5
From 31 to 40	23	18	16,5
From 41 to 50	20	15	14,5
From 51 to 58	22	13	10,5
TOTAL	**127**	**85**	**77,5**

As we show in previous Tables, all the potential relations from the texting-file are not processed by NAHUAL, since the rules pattern are not covering all possible combinations. For that reason, the following evaluation is focused on the results of the relations generated by expert and NAHUAL. Based on them, we performed a preliminary evaluation in terms of the potential/useful relations being extracted (127), the number of relevant extracted relations by expert and by NAHUAL (85 and 77.5), and the number of irrelevant extracted relations (10); that is, the necessary components for measuring precision and recall. According to Chaowicharat and Naruedomkul (2012), we evaluated the accuracy of results via percentage of correctness measurements: precision (P) and recall (R).

© Bell Manrique-Losada, 2014. All rights reserved.

Table 6.6. Sample list of the identified relations

#	Identified Relations	Number of Relations		
		Potencial	Expert	NAHUAL
1	-	1	0	0
2	-	1	0	0
3	-	1	0	0
4	-	0	0	0
5	-	0	0	0
6	'document has purpose' - 'x provide procedures' - 'FDA staff follow procedure'	3	3	3
7	'request then receive then handle then process then track' - 'reviews has attribute: collaborative | consultative'	2	2	1.5
8	-	1	0	0
9	-	0	0	0
10	'x made effort' - 'made then identify' - 'x identify need' - 'review has attribute: collaborative | consultative'	4	4	4
11	'intend then file' - 'x file submission'	3	2	2
12	'consult then collaborate' - 'x held reviewers'	2	2	2
13	'x receive credit' - 'expert reviews has attribute: thorough | timely'	2	2	1.5
14	'x made effort' - 'made then meet' - 'date has attribute: due' - 'request originator identify date'	4	4	3.5
15	'consultation has attribute: extensive'	3	1	1

$$R = \frac{\text{Number of correct relations in the solution}}{\text{Number of relations in the reference text}}$$

$$P = \frac{\text{Number of correct relations in the solution}}{\text{Number of relations in the solution}}$$

Gelbukh and Sidorov (2006) define P as the relationship between the correct results and the results obtained in total, while the R is the ratio between the correct results and the results that should have been obtained, *i.e.* P indicates how accurate or precise was the mapping, while R disclosed if the mapping generated all results that should be generated. For comparing the general results between expert and NAHUAL, the value for P in this validation study based on the information generated from Table 6.5, is as follows: R= 0.9118, such the Recall and Precision are superior to 90%.

$$R = \frac{77{,}5}{85} \quad P = \frac{77{,}5}{83.5}$$

The harmonic mean of P and R is called F-measure, for obtaining a value of efficacy. Usually beta is 0.5, by valuing equally important precision and recall and by using the following equation. In this way, the F-measure obtained by NAHUAL performance—regarding to the ideal/potential relations—is 0.739, but regarding the expert mapping is 0.925. The evaluation results in terms of precision, recall, and F-measure are shown in Table 6.7 and the graphic for detailing the values for precision and recall by generated relations is shown in Figure 6.11. We can observe that when the number of retrieved documents grows, recall is increasing since this is sorted according to the relevance.

$$F = \frac{2\,\mathrm{P}\,\mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

Table 6.7. Sample list of the identified relations

	Expert	NAHUAL base:expert	NAHUAL base:potential
RECALL	0,66929134	0,91176471	0,61023622
PRECISION	0,94444444	0,93939394	0,93939394
F-measure	0,78341014	0,92537313	0,7398568

© Bell Manrique-Losada, 2014. All rights reserved.

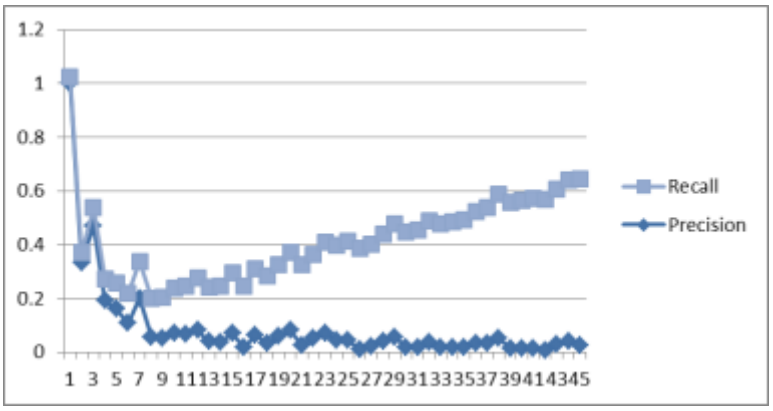


Figure 6.11 Precision and Recall

The output variable 'Mapping Time' can be variable since directly depends on the expertise level of the expert and familiarity with the mapping rules. The first running of the mapping was about 45 minutes. The second one was about 30 minutes. We think after a periodically training mapping time should decrease, but nothing close to the NAHUAL mapping. The mapping time of NAHUAL is also dependent of the length of the text. For the testing-text the time was between 40 and 100 seconds.

Findings of Validation Study 2

By analyzing the previous results of the pre-processing and processing in NAHUAL, we can present the most important findings as follows:

- We should refine the regular expressions belonging to the pre-processing for trying to decrease the following deficiencies:
  - The pre-processor takes in the last found move all paragraphs scanned until found another move or synonyms, even when they have no relation.
  - When pre-processor does not find a move synonym in the searching function through the text, all of paragraphs scanned are not included in the output file.
  - The inclusion of lists, bullets, and numbers for step-by-step in pre-processing have not been developed.
  - We decide to delete complete phrases comprising long text elements different to text—*e.g.* phone numbers, emails, web pages links, among others.
  - We plan to consider the Move 2.4 'Definitions", since it can be useful for identifying information about terms and domain referents.
- The linguistic quality of the source document affects the performance of the pre-processing and then, processing. For example, very long sentences are widely used, but they are not typical of the structure of the English sentence.
- The main mistakes generated by the processing are caused by:
  - Lack of pattern for identifying others combinations, as follows: Verb gerund (VBG or VVG), composed adjectives, complex noun phrases
  - The following typical structures are not yet implemented:
    - Past participle with gerund (VBN + by + VVG)
    - Modal structure (NN + MD + be)
    - Determiners between transitive verbs (DT + VB + NN)
  - Phrases with very long structured predicates are not included
  - The inappropriate performance of NAHUAL occurs when:
    - A phrase has the subject separated from verb by either an adjective, a preposition, or a subordinating conjunction
    - sentence has a sequence of more than one verb in past participle without punctuation

© Bell Manrique-Losada, 2014. All rights reserved.

6.2 DEVELOPMENT PROCESS OF THE SOFTWARE PROTOTYPE

We implement the formalization of the mapping process in a computational application, by developing two approaches: the first approach is the implementation of the mapping rules by using a software app for pattern recognition, as we present in the Section 6.2.1. The second one is the development of a functional prototype called NAHUAL, with some modules for processing and mapping NL discourses into CL texts. This last prototype is developed following a software development process, as we show in Section 6.2.2.

6.2.1 Approach 1: Software app for pattern recognition of language

We designed and built a software app for generating CL from discourses written in NL text. As we described in the Section 5.2, the rules to be implemented in the software app should allow for the generation of a UN-Lencep text. The process followed for obtaining such controlled language is the following:

- ***Analysis and selection of the pattern recognizer.***
  Several recent approaches using automated techniques and processing machines have been proposed in the state-of-the-art for language pattern recognition applications from speech, voice, and discourses. According to Campbell *et al.* (2006), the goal of language pattern recognition is trying to extract information about an entire expression rather than specific word content. Several successful approaches to pattern recognition of language have been proposed, as follows:
  - PPRLM system of Zissman (1996) is a classic approach using tokenization of discourses combined with an output analysis to classify the language.
  - Methodologies combining recognition features with language recognition by using math models (*e.g.* Gaussian mixture models) to improve performance (Torres-Carrasquillo *et al.*, 2002; Campbell *et al.,* 2006).
  - Sonic is a discourse recognition system for large vocabulary from the University of Colorado (Pellom, 2001). The recognizer toolkit can be used for recognizing speech, keyword and phrase spotting, and phoneme of large vocabulary. Sonic incorporates adaptation and normalization methods for regression, combination, adaptation, and normalization. Advanced language-modeling strategies such as concept language models (Hacioglu & Ward, 2001) are also incorporated into sonic.

  Specific tools found in the state-of-the-art review, implementing pattern recognition of language or language transformation process, are:
  - **ReqSIMILE** [16] . Analyze and measure the similarity among requirements, based on a classification of them. The output is a report of measures (*e.g.* statistics from a set of requirements) and the comparing rates among them.
  - **GULP** (Covington, 1994). Tool using a Graph Unification Logic Programming and a Definite Clause Grammar as a special grammar. The input is a written text and the output is a syntactic and semantic tree on Prolog.
  - **NLForSpec** (Leitiao, 2006). Provides a NL processing unit in java, for verifying whether a set of paragraphs is written following defined rules. The final purpose is obtaining formal test case specifications.
  - **RAT** (Jain *et al.*, 2009). RAT is a tool for reviewing software requirements documents, by following a two-phased approach: i) Lexical Analysis and ii) Syntactic Analysis. The lexical analysis converts a requirement statement into a set of tokens and also classifies it as a particular type of a requirement. The syntactic analyzer uses state machines for analysis of syntaxes.
  - **Class-Gen** (Elbendak *et al.,* 2011). Class-Gen employs use-case descriptions as the starting point for identifying classes and relationships, and uses an existing parsing tool to identify noun phrases, verb phrases, adjectives, and adverbs. Then, a class model is produced by identifying objects/classes, attributes, operations, associations, aggregations and inheritance so as to produce a class model.

[16] http://reqsimile.sourceforge.net

© Bell Manrique-Losada, 2014. All rights reserved.

- **ANLT** (Carrol *et al.*, 1991). ANLT is the Alvey Natural Language Toolkit from the University of Cambridge. ANLT provides facilities for defining and editing syntactic rules written in a formalism for developing a semantic component, and for building a lexicon compatible with the grammar. The goal is helping the user for testing and debugging the grammar, based on a parser, a generator, and tools for inspecting the grammar.
- **ANTLR** (Parr, 2007). ANTLR is a language tool for constructing recognizers, compilers, and translators from grammatical descriptions. It is free, open-source, and generates several languages like Java, C++, C#, and Python.

We developed a first prototype by using an application programming interface (API) and a standard-based solution like resource control protocol. Based on the skills, requirements, and features (*e.g.* availability, reliability, and support), ANLTR was selected as the language for the prototype development. The main reason for such a selection is that ANTLR is useful for creating grammars more accessible to the average programmer, by accepting a larger class of grammars and generating recursive parsers very similar to what a programmer would build by hand (Bovet & Parr, 2007).

- ***Configuration and description of an app for pattern recognition***

After downloading and decompressing ANTLR, we installed a Java JVM. Next, we set PATH and CLASSPATH according to the java ANTLR. Tool. ANTLR requires a grammatical description which is received and translated by a target program. Internally, the translator is supported by the EBNF (Extended Backus-Naur Form) and in the Abstract Syntax Tree, for the sake of being able to describe the lexer and parser.

The language recognizer compares the input text vs. a list of phrases to be recognized, called a grammar. The grammar is used to constrain the search, enabling the ASR to return the text best fitted. This text is then used to drive the next steps of the speech-enabled application. By using a C++ API, we integrate the interface for entering the input texts into an application. Thus, the software solution converts texts into a controlled language, providing users with a more efficient means of input and processing paragraphs from documents.

ANTLR is actually a code generator, since it takes a **grammar file** as an input and generates two classes: **lexer** and **parser**. The **lexer** breaks up an input stream of characters into vocabulary symbols for the **parser**, which applies a grammatical structure to that symbols stream.

The **Grammar file** contains the specifications needed by ANTLR for generating the right **lexer** and **parser**. The grammar file describes how to split the input into tokens (lexer rules) and how to build trees from tokens (parser rules). **Lexer** runs first and splits input into pieces called tokens representing meaningful pieces of input. The stream of tokens is passed to the **parser** which does the rest of the work. The parser builds an abstract syntax tree, interprets the code, and translates it into the controlled language text.

The designed app, as a small software system, uses the following process to transform an input document:

i. The translator loads a list of words from a text to be recognized. The text is presented as a small set of paragraphs. This list of words is called a grammar.

ii. This text is turned into a term string, as a syntactical representation of the texts.

iii. The engine looks at features—characteristics of texts—derived from the aforementioned representation and compares them with its own rule set. The translator searches its base space, by using the grammar for guiding this search.

iv. Then, it determines which words best fit the grammar and returns a result. The translator compares identified terms and terms construction with the loaded grammars in favor of producing the recognized text.

© Bell Manrique-Losada, 2014. All rights reserved.

- ***Classification of rules for implementing them in the app***

The rule-based approach was implemented by formulating general and reliable rules and then, by automating the generation of these rules from a large database of characters and words. Since 90's, this approach has incorporated fuzzy rules and grammars that use statistical information on the occurrence frequency of particular features (Parizeau & Plamondon, 1995). Recently, some language recognizers software applications that have been supported on the rule-based approach have been released, exploiting the properties of this paradigm mainly grounded on the fact they not require a large amount of training data and that the number of features used for describing the patterns may vary from one class to another (Plamondon & Srihari, 2000).

The analysis of NL text for understanding policy or procedure rules is a difficult task. We needed to create a set of grammars for identifying key elements in each rule with high reliability. The rules defined within a lexer grammar in ANTLR implicitly match characters on the input stream instead of tokens on the token stream. Referenced grammar elements include token references, characters, and strings. Lexer rules are processed in the exact same manner as parser rules and, hence, may specify arguments and return values. In ANTLR the rules have the following form:

```
rulename [args] returns [retval]
options { … }
{ optional initiation code } :
        alternative_1 |
        alternative_2 ... |
        alternative_n ;
```

We used UN-Lencep as output CL since ANTLR takes as input a grammar defined and then generates a CL. We were looking for useful information extracted from each piece of a procedure document which can be processed, analyzed, and finally written in a CL text.

The proposed rules comprise a grammatical structure for describing the behavior from situations and actions expressed in the description of procedure documents. Situations are conditions requiring an action to be taken. Actions are commands or instructions. The first approach to grammatical rules for this Ph.D. Thesis is designed and implemented in this app by using categories, as we describe below:

i. **Structural relationships**. Refer to structural verbs or permanent relationships between concepts ("to be," "to have," and "to uniquely have")

Rule.1. *noun + HAVE + noun or noun + IS + noun*

```
static_rule returns [String value] : exp = ruleIS {$value = $exp.value;} | exp1 = ruleHAVE
{$value = $exp1.value;}

ruleIS returns [String value] : exp1=noun {$value = $exp1.value;}
'IS ' exp2=noun { $value = $exp1.value + " is " + $exp2.value;}
```

ii. **Dynamic relationships**. Refer to dynamic verbs or temporal relationships between concepts (verbs such as "to assess," "to apply," "to analyze," and so forth). This category recognizes the dynamic relationship in several ways. As a verb is the realization of an action, we need to identify the action even when the actor who performs the action or the object on which it rests is not explicit on the text. Concerning this, we use the following rules for identifying possible dynamic relationship:

Rule.2. *noun + VERB + noun or noun + VERB or VERB + noun*

```
dynamic_rule returns [String value] : exp1=substantive {$value = $exp1.value;}
(exp2 = verbs_function exp3=noun {$value = $exp1.value + $exp2.value + $exp3.value;})
| exp2 = verbs_function {$value = $exp2.value;}
( exp6=noun { $value = $exp2.value + $exp6.value;})
| exp7=substantive {$value = $exp7.value;}
( exp8= ws exp9 = verbs_function {$value = $exp7.value + $exp8.text + $exp9.value;} )
| exp10= verbs_function
```

© Bell Manrique-Losada, 2014. All rights reserved.

iii. **Implications**. Represent cause-and-effect relationships between dynamic relationships. Two uses of implications are: linking a dynamic relationship to another one and linking a conditional to a dynamic relationship. In favor of identifying the start of an implication, we have the following reserved words: if and when; the means of an implication is represented by the following reserved words: then, now, and before.

Rule.3. *If + relation + then + relation*

```
implication_rule returns [String value]
: exp= condition  exp1 = relations   {  $value =  "if " + $exp1.value;}
(' THEN ' exp2 = relations {$value = "if " + $exp1.value  + " then " + $exp2.value;}
        |' NOW ' exp3 = relations {$value = "if " + $exp1.value  + " then " + $exp3.value;}
        |' BEFORE' exp4 =relations {$value = "if" + $exp1.value +"then" + $exp4.value;})?
```

iv. **Composed Rules**. Indicate the join of different basic rules in the same phrase. This rule comprises structural, dynamic, or adjective rule.

Rule.4. *Relation Rule:*

```
relations returns [String value] : exp =  static_rule        {$value =  $exp.value;}        | exp1 =
dynamic_rule      {$value =  $exp1.value;}
| exp2 = ruleadjetive        {$value =  $exp2.value;}
```

Rule.5. *Unlencep Rule: Grouping regla_relations and implication rule, such that rule_unlencep can take any of the two values.*

```
rule_unlencep returns [String value] : exp  = relations {$value =$exp.value; }
| exp2 = implication_rule    {$value =  $exp2.value;}
```

Rule.6. *Conjunction Rule: a conjunction is the union of two relations, returning the two relationships have been identified with the Unlencep rule.*

```
relation_conj  returns [String value]  : (exp= rule_unlencep) +  {$value =  $exp.value;}
((exp2= conj)? (exp3= rule_unlencep)+)? {$value =  $exp.value + '\n' +  $exp3.value;}
```

- ***Definition and design of the app components***

In this Section, we describe the general architecture of the app which is common to all software systems. Each one has two main components: Front-end and Back-end component, related to the view layer and logic layer respectively, as we show in Figure 6.12.

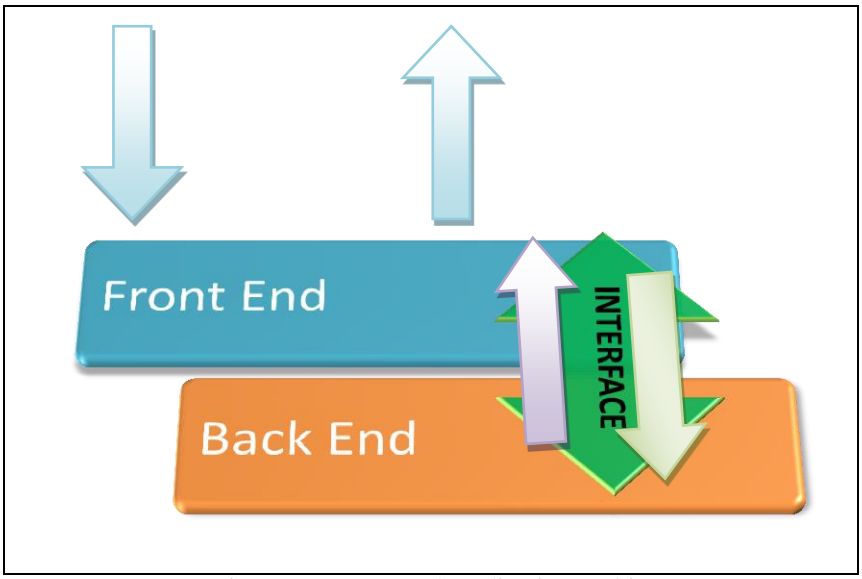


Figure 6.12. General application architecture

Front-end Component mapping the input document segment to a vector by means of the lexer. We denote these front-end outputs as amorphous scores.

© Bell Manrique-Losada, 2014. All rights reserved.

The front-end component contains the user interface and selects a utility for text files, allowing for users to enter technical documents held on their computer. This component is also referred as Feature extraction, the initial stage of the system that converts inputs into feature vectors for analyzing and testing the language recognizer. A feature extraction technique is applied to extract features from the set of sentences entered by a user. A feature vector represents the features of each recorded sentence, which is considered as an input to the next component.

Back-end component performs the logic process. The back-end gathers the results from the front-end and outputs a controlled language text.

This component is divided into the following sub-components: the first one called *file reader* contains the necessary logic for the file chooser to perform its function, by receiving a user input file and then reads it. The second sub-component is the *file processor* which gets the file sent by the file reader, applies a set of predefined rules, and makes the pre-processing required before sending the file to the language recognizer. This task is essential, because the recognizer—when receives simple sentences—ensures better information processing.

Finally, after the file is transformed, it is transferred to the third sub-component where the *language recognizer* is found. This recognizer is formed by the *Lexer* which is responsible for analyzing the input received from the file processor and creating tokens. Such tokens are subsequently read by the *parser* which builds a data structure from the UN-Lencep rules implemented in the parser, achieving the generation of controlled language texts. The sub-components of back-end are further described in the Figure 6.13.

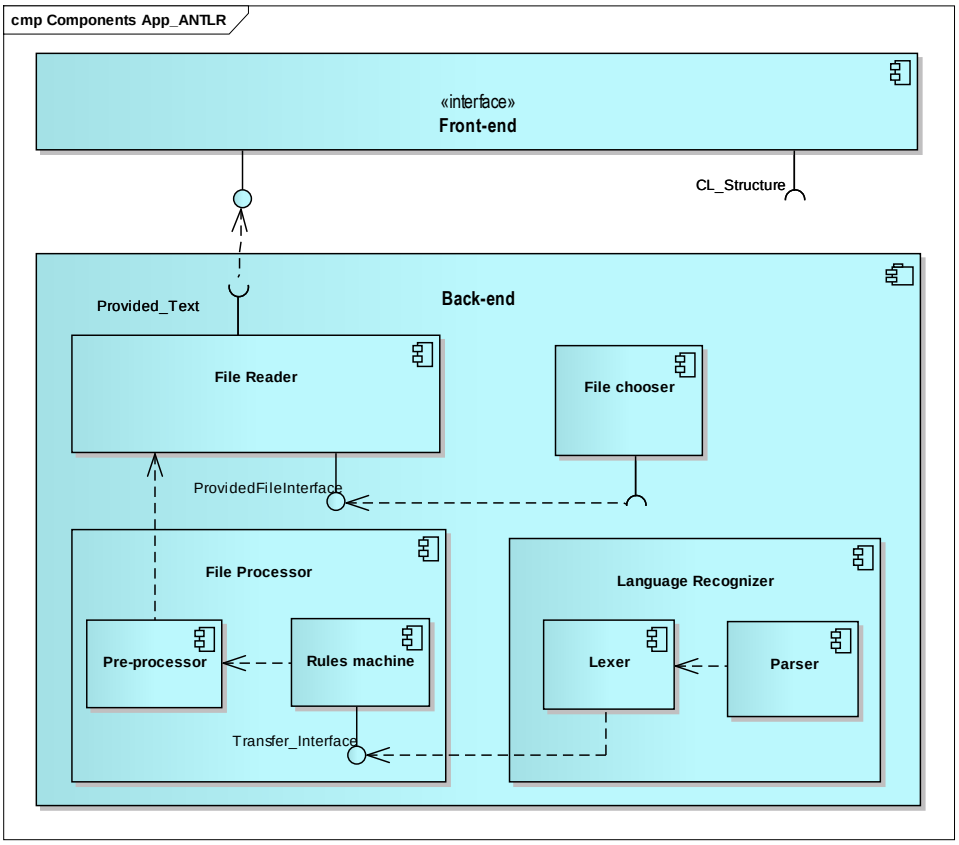


Figure 6.13. Back-end sub-components for the App ANTLR

- ***Implementation and testing***

  In order to show the performance of the designed and built application, an evaluation approach was applied (Zapata & Manrique, 2013). In this Section we present a part of such evaluation, based on an example extracted from the original document used.

© Bell Manrique-Losada, 2014. All rights reserved.

We started by analyzing procedure rules we collected from portions of the text extracted from the SPEED model (De Brito, 2002). SPEED (called by its French acronym to Written Procedure Following in Dynamic Environments) is a model for explaining the use of operational procedures in dynamic situations. The model comprises of 9 stages, among them:

1. Detecting triggering conditions. In this stage the pilot develops a diagnosis of the current situation.
2. Elaborating the diagnosis. The pilot develops a diagnosis of the current situation, to determine the type or the nature of the procedure to be used.
3. Determining whether a procedure is needed. In this stage, as a way to use a procedure, the pilot assesses whether the information is useful and estimates if he needs it.
4. Accessing and searching procedural information. If the pilot needs assistance, he looks for a procedure and should be able to find it.

We use the following short text extracted from SPEED, due to the extension, because it is consistent with the sections defined previously from procedure manual:

> ***Procedure name**: 'Elaboration of diagnosis'*
> ***Operating rules**: To use a procedure, the pilot assesses whether the information is useful and estimates if he needs to use the procedure.*
> ***Narrative description**: If the pilot has time, he will try to understand the dysfunction before applying the procedure. If the pilot is in an emergency situation, he first applies the procedure, but while applying it, tries to understand the dysfunction by comparing his own action plan with that presented to him.*

Therefore, the discourse—for three preview Sections from '*Elaboration of diagnosis'* procedure—results from the analysis of whole procedure and its use context, based on the model description by De Brito (2002). The generated discourse in UN-Lencep from the original text can be seen in Table 6.8.

Table 6.8. Generated discourse in UN-Lencep from original procedure

Input	Output
- to use a procedure - the pilot assesses whether the information is useful	a. to use procedure b. pilot assesses c. information have state d. state is useful
- estimates if he needs to use the procedure	e. if he needs procedure
- if the pilot has time - he try to understand the dysfunction before applying the procedure	f. if pilot has time g. he try dysfunction h. applying procedure
- if the pilot is an emergency - he applies the procedure	i. if pilot is emergency j. He applies procedure
- but while applying it - tries to understand the dysfunction	k. l. tries dysfunction
- by comparing his own action plan with that presented to him	m.

According to the output of the ANTLR prototype, the major gaps in the output discourse without rewriting are:

- Absence of the actor who directs the verbs (a). Consequently, it is not clear who is using a procedure.
- Using the term 'if', without the consequence action 'then' (e). The output phrase is incomplete.
- When two or more verbs are together within the phrase (*e.g.* 'try to understand'), the prototype only considers the first one (e, f, k, l). We are working in this case.
- The expression 'the pilot has time' (f) is an affirmation without sense in this context, because it is not a property that can be measurable practically.
- When the term 'if' (g, h) is not included inside a conditional phrase, the output phrase is not taken as an implication, because it lacks either if or when.

© Bell Manrique-Losada, 2014. All rights reserved.

- The phrase 'pilot is an emergency' (i) if a conditional phrase is not included is not appropriated, because it is interpreted as if an emergency were a property of the pilot, not as an environmental situation that is perceived by the pilot. For this reason, the output is not clear and precise.
- The term *but* before an expression of condition as while (k), is not processed by prototype.
- We have not included the prepositions inside the grammar (m). For this reason, the prototype does not recognize this character and no results are generated.

After the previous gaps and mistakes identified based on the application results, we consider building a preliminary component for pre-processing the input text. Due to the limitations of the ANTLR, we decide to carry out a by-hand rewriting of the input text, based on the following considerations for each expression:

- '*If the pilot has time'* is related to the '*criticality'* state of the emergency situation. Based on this scenario, if the emergency situation is too critical, the pilot would not have the information for reviewing and will have to apply the procedure immediately.
- If the '*emergency situation'* is not too critical, the pilot first *analyzes* the 'emergency situation,' '*compares'* the plan and proceeds with the next steps.

From this analysis, the 3th part of the proposed discourse—narrative description, could be defined as follows: "*If the emergency situation is not critical—or status is equal to not critical, the pilot analyzes the emergency situation.*" Thereby, we rewrite the discourse for facilitating the processing, and the input text for the final processing is the following:

> *The pilot 'assesses' whether the status of information is useful. The pilot 'applies' the procedure when the information is needed. If the emergency situation is not critical, the pilot analyzes the emergency situation. If the pilot 'recognizes' an emergency situation, then the pilot applies the procedure. When the pilot applies the procedure, then the pilot 'analyzes' the* situation and 'compares' the action_plan.

The generated discourse in UN-Lencep from the proposed discourse (with re-writing), is shown in Table 6.9.

Table 6.9. Generated discourse in UN-Lencep from the proposed procedure

Input		Output
the pilot assesses whether the information have status is useful	- -	pilot assesses information have status
The pilot applies the procedure when the information is needed	- -	information have state if state is needed then pilot applies procedure
If the emergency situation is not critical, the pilot analyzes the emergency situation	- -	emergency_situation have state if state is not-critical then pilot analyzes emergency_situation
if the pilot recognize an emergency then the pilot applies the procedure	-	if pilot recognize emergency then pilot applies procedure
When the pilot applies the procedure then the pilot analyzes the situation and compares the action_plan	- -	if pilot applies procedure then pilot analyzes situation compares action_plan

Findings and Common assumptions

As we have mentioned before, this approach deals with typical issues occurring in writing documents. In principle, we identify the following findings:

α *Regarding conceptual issues:*
- Actions with missing agent/actor
- Actions are written in passive voice
- Use of non-standard terms
- Actions using different names for the same entity
- Use of two continuous action expressions without much relevance in a sentence. In most cases exist a better term to express the same.

© Bell Manrique-Losada, 2014. All rights reserved.

α *Regarding syntax:*
The idea is enforcing "*Syntax Best Practice*" to write procedures contained into procedures manual. Enforcing the use of this syntax serves two purposes. First, it directly promotes the usage of best practice syntax, and second it makes the problem of analyzing the natural language from this documentation, to promote its processing and comprehension practices tractable. According to De Carvalho (2006), a good procedure guarantees at least three major advantages:
(1) Reducing workload,
(2) Reducing the possibility of human error, and
(3) Standardizing human performance.

Another advantage is related to align each procedure-type with a conceptual category of actions. For example, we have a different procedure-type for solution—actions defining the capability of a system—and enablement—actions defining a capability an actor should provide for reaching a goal. Aligning a procedure-type with a conceptual category makes it easier for the reader to quickly understand the intent of each step of procedure. On the other hand, we need to design each procedure-section in such a way that it can be easily distinguished from others. This is done by using a unique set of keywords as part of each procedure-section. For example, the usage of "shall" or "must" denotes a mandatory action, whereas "shall be able to" denotes a conditional action.

α *Regarding general process and the application:*
The evaluation of the app prototype reports our first approach towards developing a transforming system grounded on business-based technical documents corpus. This report generates the following conclusions, useful for the development of improved software systems:

- Creating the corpus with full or less marks and noise than current one from preprocessed documents
- Building a dictionary and a robust grammar
- The components of the systems should be supported on a processing-levels model.
- The system could be highly usable as support for analyzing documents in the elicitation requirements process.

6.2.2 Approach 2: Prototype of the Software System NAHUAL

Following the Agile Unified Process (AUP) method, as a simplified version of the Rational Unified Process (RUP), we developed **NAHUAL**, a functional prototype of a software system implementing the proposed formalization of this PhD Thesis. Such a method was selected because we need: i) to describe the context and domain of a software application in a simple way; ii) to easily understand one business application software; iii) to document the followed development process by using a method less rigorous than RUP, but using agile techniques, principles, and concepts; and iv) to plan what teams typically deliver—development releases—at the end of each iteration into pre-production staging area(s).

We tried to keep the AUP as simple as possible, in terms of its approach, application, and description, as we present in the four phases proposed by Ambler (2005): inception, elaboration, construction, and transition. We address iteratively by phase the disciplines shown in Table 6.10[17], defining the activities described by Ambler:

[17] In the implementation of the software system for validating the model, we ignore the 'Configuration Management' and 'Project Management', due to the fact that the goal of such disciplines is managing the access to the project artifacts (including tracking artifact versions and managing changes), and the activities executed on the project (including managing risks, directing people, and coordinating estimations and costs), respectively. Such disciplines are oriented to people and budget managment which are outside the scope of this Ph.D.Thesis.

© Bell Manrique-Losada, 2014. All rights reserved.

Table 6.10. Generated discourse in UN-Lencep from proposed procedure

Discipline	Goal
Model	Understanding the business and problem domain and identifying a viable solution to address the identified problem
Implementation	Transforming the designed models into executable code and performing the preliminary testing
Test	Performing an objective evaluation to ensure quality, including finding defects, validating the system, and verifying the requirements are met
Deployment	Planning the system deliveries and making the system available to the end users
Environment	Discipline to support the rest of the effort by ensuring the availability of process, standards, and tools

- **Inception.**

The goal of this phase is identify the initial scope of the project and the probable architecture for the system. We follow the UNC-Method (Zapata & Arango, 2009) mainly for developing the initial activities of the inception and elaboration, as we will present in the following Sections and related Appendices. UNC-Method is a problem-based software development method including a mixture of well-known artifacts.

The initial scope of the project and the software context are the first steps for achieving agreement between analysts and stakeholders, based on a common vocabulary. The UNC-Method employs pre-conceptual schemas and domain models (Zapata *et al.*, 2006) to establish such vocabulary.

In the inception phase and the preliminary activities of the model discipline, we define the scope of the project and the software context in the pre-conceptual schema presented in the Appendix 6.1. The domain model is presented in Figure 6.14. Also, the goal diagram and the cause-effect diagram, as models resulted of problem analysis and identification of scope are attached in Appendix 6.2 and 6.3.

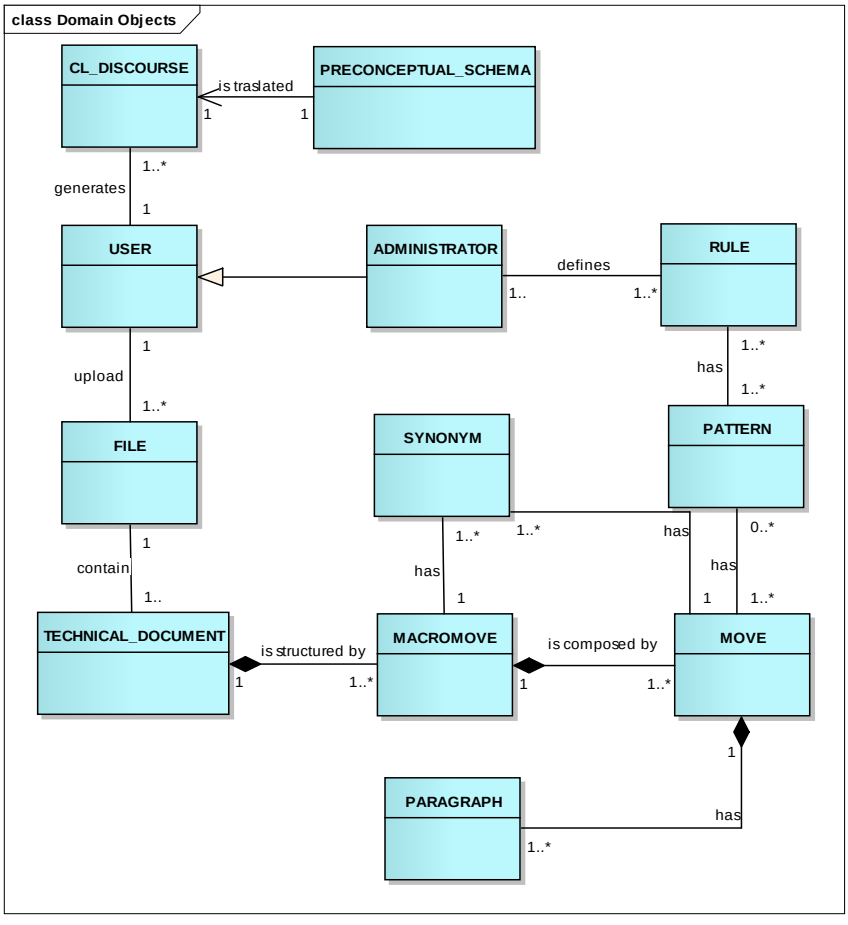


Figure 6.14. Domain Model

© Bell Manrique-Losada, 2014. All rights reserved.

The detailed explanation of the process to be implemented in the software system is shown in Figures 6.15 and 6.16 in terms of process models. Additionally, the workflow of the process is described by-using use-case specification templates, as it is included in the Appendix 6.4.

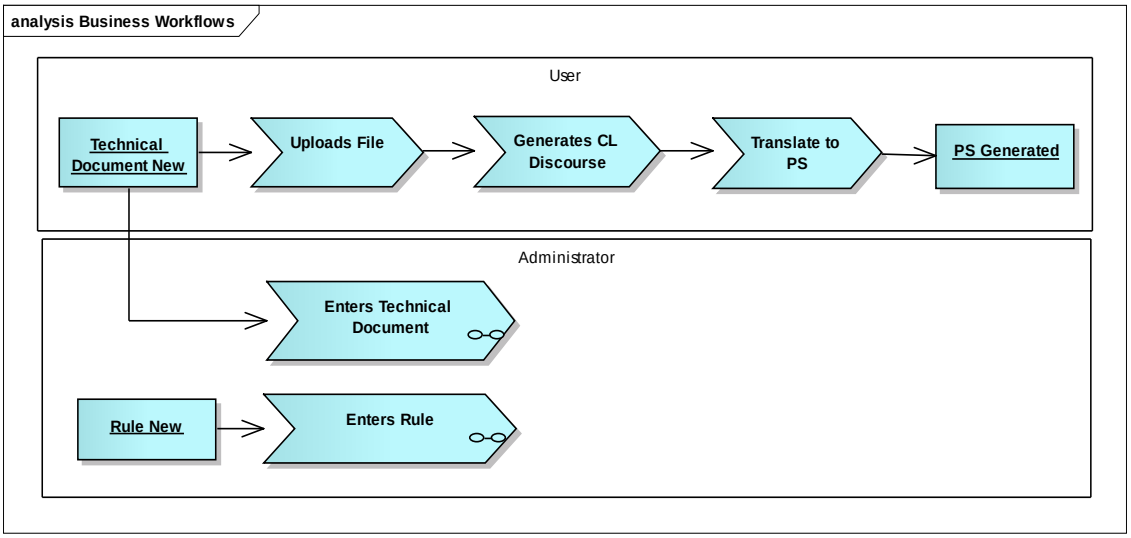


Figure 6.15. Process Model

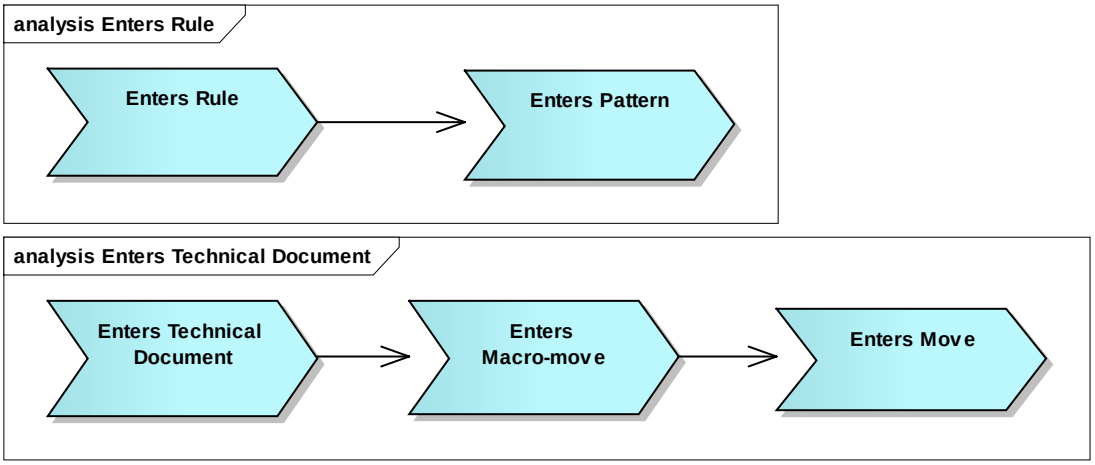


Figure 6.16. Process Model for specific activities

- **Elaboration.**

  In this phase, the system architecture is designed and proved. We describe such architecture by means of a class model (by using UML 2.3 on a tool CASE), as we show in Figure 6.17.

© Bell Manrique-Losada, 2014. All rights reserved.

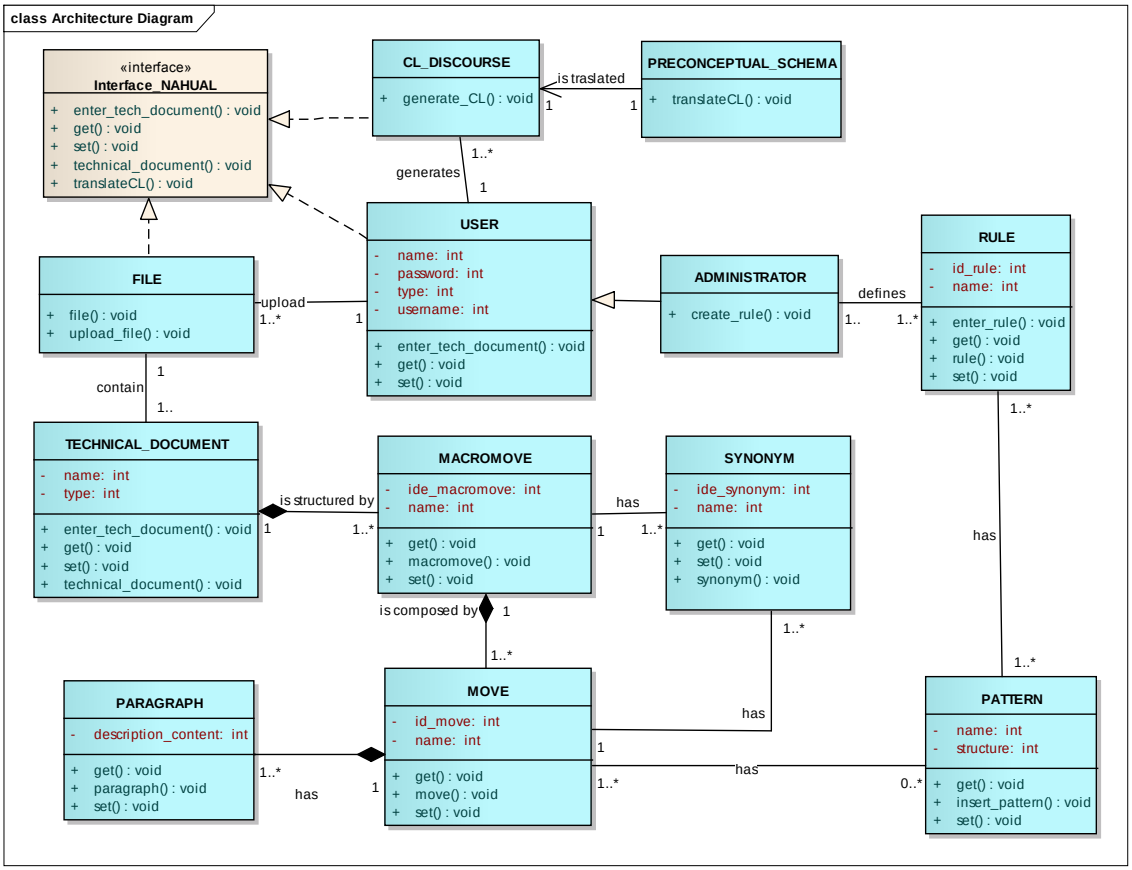


Figure 6.17 Classes Model for NAHUAL

- **Construction.**

The goal in this phase was building the NAHUAL app by following the implementation, deployment, and architectural decisions, based on the previous system designs.

The implementation of the solution was performed using a set of Python programs receiving as input TXT files. We chose Python since is a language for natively and easily handling regular expressions. It is a standard technology for language processing applications. In this way, we develop a Python web app with Django[18]. Django is a Web framework for developing high-level Python applications.

Such implementation was structured in two separated components, related to the pre-processing and processing. Pre-processing is an independent component receiving a text file—in several formats like html or pdf—and its conversion into a txt ready for processing—based on the conditions from Section 5.2.1.1.

Processing was developed by means of the following step-by-step process for developing NAHUAL:

- Auto-generating code for establishing a Django project.
- Defining a collection of settings for such an instance of Django, including database configuration, Django-specific options, and application-specific settings.
- Building modules respectively for the previous components.
- Building the core module containing the WSGI application used by Django development server, based on the module-level variables (*e.g.* 'application,' 'runserver,' and 'runfcgi') and the development server configuration.
- Creating models for each Python package related to the main functions for NAHUAL—called modules. Such packages contain the essential fields and behaviors of the app. A sample of the code implemented in Python for NAHUAL app is included in Appendix 6.5.

[18] https://www.djangoproject.com/

© Bell Manrique-Losada, 2014. All rights reserved.

- Creating the database schema and Python database-access for this app.
- Running the tests on the app.

The modules are related to the components shown in Figure 5.14 from Section 5.3. The NAHUAL modules are related to the Main Processor and representation interface. Specifically, they are defined as follows and are graphically represented by using a component model in Figure 6.18.

Lexical Module. This module executes the lexical analysis, by determining signs of specific words by looking for in the methods from a lexicon. The analysis comprised a tokenization and sentence boundaries processes. The input text is divided into words in the tokenization process and then deals with the spelling level by splitting the text into several words and sentences.

Tagging Module. The fixed sets of tags or *tagsets* are implemented, by means of the inclusion of the TreeTagger[19]. After corpus training and lexicon definition, the text is annotated with POS tags and lemma information—by means of such POS Tagger.

Parsing Module. This module involves the definition and usage of a grammar. The tagged text is parsed and its syntactic structure is extracted. We implemented an algorithm for analyzing the input and generated a structural representation by using a parsing tree.

Semantic Module. This module facilitates the mapping of structures created by the parsing module. First, we defined the rules and the input and output patterns by rule. The set of mapping rules implemented was derived from the proposal of rules pattern (see Table 5.8 from Section 5.2.1.2), and we show them in Table 6.11. Then, we implement the Boyer-Moore algorithm for searching all occurrences of strings on a text. The strings correspond with the constitutive elements of the rule pattern and are a way for processing them by determining the optimal amount to shift the string and skip comparison.

Representation Module. In this module we defined the controlled language syntax, as format of output. The output text is presented as a list of phrases written in the controlled language defined.

Table 6.11. Mapping rules implemented in NAHUAL

#	Name	Description	Input Base	Input Options	Output
1	TRANS-VB	*i. Transitive verbs*	**1.1 NN + VV + NN**	*NN | NP | NPS | NNS* *VV | VVZ*	**NN + VV + NN**
2	TRANS-VB-subj	*ii. Transitive verbs without inmediate subject*	**2.1 VV+ NN** **2.2 VV + [the] + NN** **2.3 To + VV + NN** **2.4 To + VV +a+ NN_2+ NN_1** **2.5 To + VV + [the] + NN**	*NN | NP | NPS*	**X + VV + NN_1**
3	PASS-VOICE	*iii:Passive voice*	**3.1 NN_1 + VBZ + VVN +by+ NN_2** **3.2 NN_1 + [to be + VVN]** **3.3 NN + VVN +[by] +NN** **3.4 NN + VVN +[by] +DT +NN**	*VBZ (to be) |IS | ARE* *NN: | PP | NP | NPS*	**NN_2 + VV + NN_1** **X + VV + NN_1**
4	IS-A	*iv. Construction of the form 'is a'*	**4.1 NN_1 + [is a] + NN_2** **4.2 NN_1 + [is an] + NN_2**	*NN | NP | NPS |NNS*	**NN_1 + is + NN_2**
5	NN-POST-MOD	*v. Noun phrase with post-modification*	**5.1 NN_1 + [of the] + NN_2**	*of the | of a | of this | of these*	**NN_2 + has + NN_1**
6	JJ-NN-MOD	*vi. Noun phrase with pre-modification*	**6.1 JJ + NN** **6.2 DT + JJ + NN** **6.3 VVN + NN** **6.4 JJ_1 + CC + JJ_2 + NN** **6.5 VVN_1 + and + VVN_2 + NN**	*NN | NP | NPS | NNS* *CC | AND | OR*	**NN + has + Attribute [value: JJ]** **NN + has + Attribute [value: VVN]** **NN + has + Attribute [value: JJ_1 + JJ_2]** **NN + has + Attribute [value: VVN_1 + VVN_2]**

[19] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

© Bell Manrique-Losada, 2014. All rights reserved.

#	Name	Description	Input Base	Input Options	Output
7	IMPLI~ INFINITIVE	*vii. Main clause + infinitive phrase*	**7.1 VV_1 + to + VV_2 + NN** **7.2 VV_1 + for + VV_2 + NN** **7.3 VV_1 + to + VV_2** **7.4 VVG +[TO] +VV +DT +NN**	*NN | NP | NPS | NNS*	**VV_1 then VV_2 + NN** **VV_1 then VV_2 + Y**
8	IMPLI	*viii. implication*	**8.1 VV_1 + VV_2 + NN** **8.2 VVG +[CC] +VVG** **8.3 VVG +, +VVB +, +VVG +[CC] +VVG**	*VV | VVN | VVG NN | NP | NPS | NNS CC | AND | OR*	**VV_1 then VV_2 + NN** **VV then VV** **VV then VV then VV then VV**
9	NN~for~NN	*ix. composed verb*	**9.1 NN1+[for]+NN2+[TO]+VV**		**NN1 + VV + VV**
10	MODAL~VB	*x. Modal*	**10.1 DT+NN+MD+VV+VVN** **10.2 NN+MD+VV+VVN**	*NN | NP| NPS | NNS MD | should be | could be | might be | would be |may be | must be | ought be | will be)*	**X+VVN+NN**

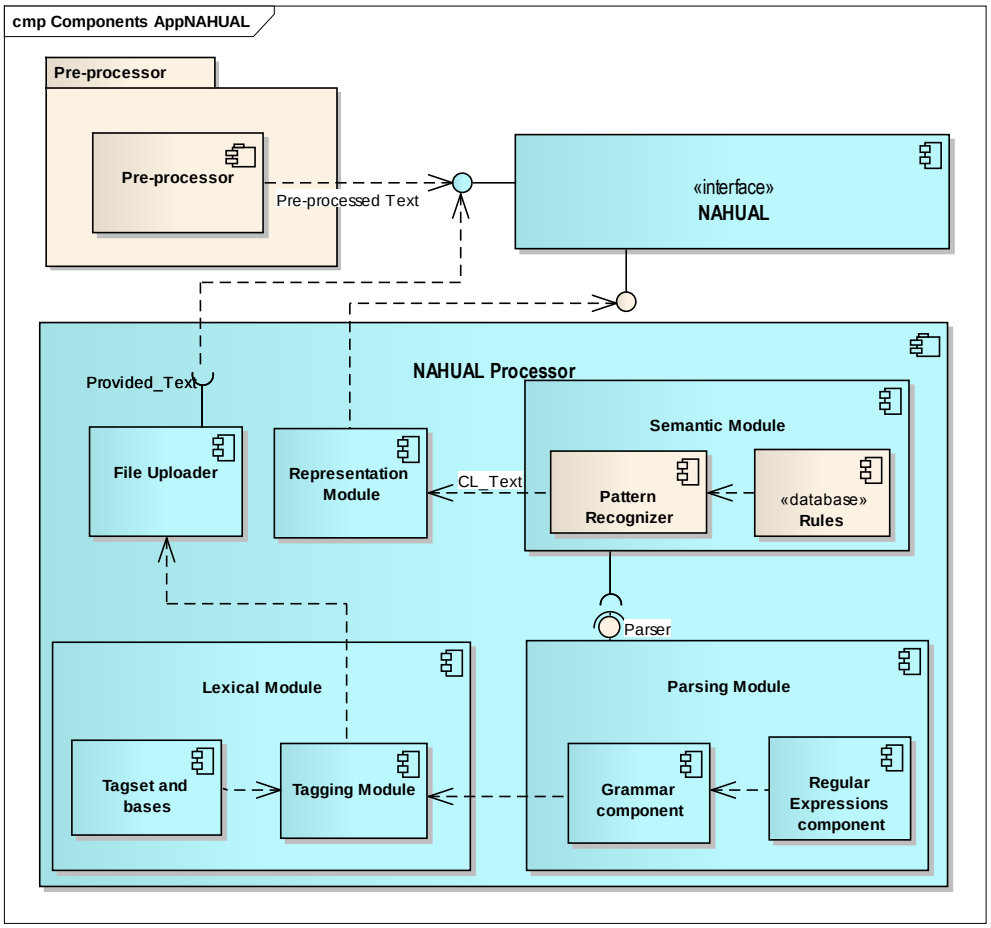


Figure 6.18 NAHUAL Component Model

Not all processing modules are worked out in detail in this prototype and hence they are not fully integrated. Until now, the pre-processing module is isolated from NAHUAL, since some aspects need further verification, and are subject to current research. We present the GUI and snapshots of the NAHUAL in Appendix 6.6.

- **Transition.**

The goal is to validate and deploy the software system into the production environment. This validation was presented in the final Section

© Bell Manrique-Losada, 2014. All rights reserved.

REFERENCES

A. Gelbukh and G. Sidorov (2006). Procesamiento automático del Español con enfoque en recursos léxicos grandes. Instituto Politécnico Nacional, México.

A. O'Keeffe's (2003). Strangers on the Line: A Corpus-based Lexico-grammatical Analysis of Radio Phone-in. PhD Thesis, University of Limerick.

B. Gunnarsson (2004). The multilayered structured of enterprise discourse. Information Design Journal + Document Design, Vol. 12, No. 1, pp. 36-48.

B. Pellom (2001). Sonic: The University of Colorado Continuous Speech Recognizer. Technical Report TR-CSLR-2001-01, CSLR, University of Colorado.

C. M. Zapata, A. Gelbukh and F. Arango (2006). Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation. MICAI 2006, LNAI 4293, pp. 27-37, Springer-Verlag.

C.M. Zapata and F. Arango (2009). The UNC-method: a problem-based software development method. Revista Ingeniería e Investigación, Vol. 29 No. 1, pp.69-75.

C. M. Zapata and B. Manrique (2013). Diseño de Reglas Gramaticales para Transformar Documentos Técnicos Corporativos Escritos en Lenguaje Natural en Discursos Controlados. Revista Latinoamericana de Ingenieria De Software. Vol 1, No 2.

C. M. Zapata and B. Manrique (2012). Desde documentos técnicos hasta esquemas preconceptuales: un caso de estudio. Proceedings WOPATEC 2012, Vol 1, Valparaiso.

C. M. Zapata, B. Manrique and G. P. Gasca (2012). An Analysis of the Requirements Elicitation Process from Technical Documents. In: Innovative ways of knowledge representation and management. Ed. Sello Editorial Universidad de Medellín. Zapata & González-Calderón (Eds.), 1st Edition, pp.11-24, Medellín.

C. Wohlin, P. Suneson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén (2000). Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers.

C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén (2012). Experimentation in Software Engineering, Springer-Verlag, Berlin Heidelberg.

D. Leitiao (2006). NLForSpec: Translating natural language descriptions into formal test case specifications. Master's thesis, Universidade Federal de Pernambuco (UFPE),

D. Molin (2007). A Study of Gender Representation in Californication. PhD Thesis Malmö University. Available in: http://dspace.mah.se/bitstream/handle/2043/7913/GenderrepresentationinCalifornication.pdf?sequence=1

D. Motwani and R. Mehrotra (2013). Recruitment Practices Adopted by IT ITES companies in Bhopal, Madhya Pradesh. Search and Research, Vol. 4, No. 3, pp. 17-20. Available in: http://journalsar.in/may2013/Deepak.pdf

D. Tufis, O. Mason, "Tagging Romanian Texts: a Case Study for QTag, a Language Independent Probabilistic Tagger", Proc. of the Intn'l Conf. on Language Resources & Evaluation (LREC), Granada, Spain, May 1998, pp. 589-596.).

E. Chaowicharat and K. Naruedomkul (2012). Co-ocurrence-based error correction approach to word segmentation. In: Cross-Disciplinary Adavances in Applied Natural Language Processing. Issues and Approaches. Boonthum-Denecke, McCarthy & Lamkin (Eds.). Information Sciences Reference Publishers, pp. 354-364, United States.

E. Tognini-Bonelli (2001). Corpus linguistics at work: Studies in corpus linguistics, v. 6. Amsterdam: John Benjamins

G. De Brito (2002). Towards a model for the study of written procedure following in dynamic environments. In: Reliability Engineering and System Safety 75, pp. 233-244. Elsevier Publisher.

G. Parodi (2005a). Discurso especializado e instituciones formadoras. Valparaíso: Ediciones Universitarias de Valparaíso.

G. Parodi (2005b). El Grial: interfaz computacional para anotación e interrogación de corpus en español. RLA. Revista de lingüística teórica y aplicada, Vol. 44, No. 2.

IEEE (2012). Manual de Operaciones de Ramas Estudiantiles IEEE. Comité de Actividades Estudiantiles Región 9 (Latinoamérica y El Caribe). Available in: http://sites.ieee.org/r9-sac/files/2012/02/MORE_IEEE.pdf

I. García (2009). Divulgación médica y traducción: el género información para pacientes. Peter Lang Publisher, Bern.

J. Bovet and T. Parr (2007). ANTLRWorks: An ANTLR Grammar Development Environment. Unpublished Draft. Available in: http://www.antlr.org/papers/antlrworks-draft.pdf

J. Carrol, T. Briscoe and C. Grover (1991). A development environment for large natural language grammars. Technical Report 233. University of Cambridge. Available in: http://www.sussex.ac.uk/Users/johnca/papers/ucam-cl-tr-233.pdf

J. Madden (2003). 100 rules for NASA project managers. ASK Magazine, Issue 14.

J. M. Swales (2004). Research genres. Exploration and applications. Cambridge: Cambridge University Press

J. M. Swales (2004). Research genres. Exploration and applications. Cambridge University Press, Cambridge,

K. Hacioglu and W. Ward (2001). A Word Graph Interface for a Flexible Concept Based Speech Understanding Framework", Proc. Eurospeech-2001, Denmark.

L. Anthony (2009). Issues in the design and development of software tools for corpus studies: The case for collaboration. In Contempurary corpus linguistics, P. Baker (Ed.), pp.87-104, London.

L. Prechelt (2008). Methodology and Study Design in Empirical Software Engineering: Two Case Studies. Freie Universität Berlin, Institut für Informatik. Available in: http://www.inf.fu-berlin.de/inst/ag-se/

H. Østbye, K. Knapskog, K. Helland and L. O. Larsen (2003). Metodbok för medievetenskap. Liber AB, Malmö.

M. A. Covington. GULP 3.1: An Extension of Prolog for Unification-Based Grammar. Artificial Intelligence Center, University of Georgia, 1994

M. Elbendak, Paul Vickers, Nick Rossiter, Parsed use case descriptions as a basis for object-oriented class model generation, Journal of Systems and Software, Vol. 84, Issue 7, July 2011, Pages 1209-1223, ISSN 0164-1212.

M. Parizeau and R. Plamondon, "A Fuzzy Syntactical Approach to Allograph Modelling for Cursive Script Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 7, pp. 702-712, July 1995.

M. V. Zelkowitz and D.R. Wallace (1998). Experimental Models for Validating Technology. IEEE Computer, Vol. 31, No. 5, pp.23-31.

© Bell Manrique-Losada, 2014. All rights reserved.

M. Zissman (1996). Comparison of four approaches to automatic language identification of telephone speech. IEEE Transaction on Speech and Audio Processing, Vol. 4, No. 1, pp. 31-44.
NHS Staff Council (2013). NHS Job evaluation handbook. Fourth Edition. Available in: http://www.nhsemployers.org/~/media/Employers/Publications/NHS_Job_Evaluation_Handbook.pdf
N. Juristo and A. Moreno (2001). Basics of Software Engineering Experimentation, Kluwer Academic Publishers, Boston, MA.
O. Armbrust (2003). Using Empirical Knowledge for Software Process Simulation: A Practical Example." PhD Thesis, University of Kaiserslautern, Kaiserslautern, Germany.
P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, J. R. Deller, Jr., , 2002. Approaches to language identification using Gaussian mixture models and shifted delta cepstral features, in: Proceedings of the International Conference on Spoken Language Processing, pp. 89-92.
P. De Carvalho (2006). Ergonomic field studies in a nuclear power plant control room. Prog. Nuclear Energy. Vol. 48, pp. 51–69.
P. Drouin (2003). TermoStatWeb 3.0 n'est désormais utilisable qu'après enregistrement. Available in: termostat.ling.umontreal.ca/
P. Jain, K. Verma, A. Kass and R. G. Vasquez (2009). Automated review of natural language requirements documents: generating useful warnings with user-extensible glossaries driving a simple state machine. Proceedings of the 2nd India software engineering conference, pp. 37-46, New York.
P. R. Sackett and R. M. Laczo (2003). Job and Work Analysis. In: Handbook of Psychology Industrial and organizational psychology, Vol. 12. Borman, Walter C. (Ed.), John Wiley & Sons Inc, NJ.
P. Runeson and M. Höst (2008). Guidelines for conducting and reporting case study research in software engineering. In: Empiric Software Engineering, Sjoberg (Ed.).
P. Runeson, M. Höst, A.W. Rainer and B. Regnell (2012). Case Study Research in Software Engineering. Guidelines and Examples. Wiley, Hoboken.
R. E. Stake (1995). The art of case study research. Thousand Oaks, Sage, CA.
R. K. Yin (2009). Case Study Research Design and Methods, 4th Edn. Sage Publications, Beverly Hills.
R. Keller (2011). The Sociology of Knowledge Approach to Discourse (SKAD). Human Studies, Vol. 34, Issue 1, pp. 43-65.
R. L. Glass (1994). The software research crisis. IEEE Software, Vol. 11, pp. 42–47.
R. Plamondon and S. N.Srihari (2000). Online and off-line handwriting recognition: a comprehensive survey. Transactions on Pattern Analysis and Machine Intelligence, IEEE, Vol.22, No.1, pp. 63-84.
R. Simpson and J. Swales (2001). Corpus Linguistics in North America. Selections from the 1999 Symposium Ann Arbor: the University of Michigan Press.
R. A. Swanson (2007). Analysis for Improving Performance: Tools for Diagnosing Organizations and Documenting Workplace Expertise. 317p. Available in: http://theseus17-kk.lib.helsinki.fi/bitstream/handle/10024/65583/Students_Julien%20Boulanger.pdf?sequence=1
S. Ambler (2005). The agile unified process (AUP). Available in: http://www.ambysoft.com/unifiedprocess/agileUP.html
S. Stein, Y. Lauer and M. El-Kharbili (2009). Using Template Analysis as Background Reading Technique for Requirements Elicitation. Available in: http://citeseerx.ist.psu.edu/viewdoc/summary?
T. J. Parr (2007). The Definitive ANTLR Reference: Building Domain-Specific Languages. The Pragmatic Programmers, United States.
V. R Basili (1993). The experimental paradigm in software engineering. In: H.D. Rombach, V.R. Basili, R.W. Selby (eds.) Experimental Software Engineering Issues: Critical Assessment and Future Directives. Lecture Notes in Computer Science, vol. 706. Springer, Berlin Heidelberg
W. Lilienblum, W. Dekant, H. Foth, T. Gebel, J. G. Hengstler (2008). Alternative methods to safety studies in experimental animals: role in the risk assessment of chemicals under the new European Chemicals Legislation (REACH). In: Archives of Toxicology, Springer-Verlag.
W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, P.A. Torres-Carrasquillo (2006). Support vector machines for speaker and language recognition, Computer Speech & Language, Vol. 20, Issues 2–3, pp. 210-229.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 7

PUBLICATIONS

Publication is to thinking as childbirth is to the first kiss.
Karl Wilhelm Friedrich Schlegel

Book Chapters

C. M. Zapata, B. Manrique and G. P. Gasca (2012). A proposal for Software Engineering Education based on a Joint Project. Latin American Software Engineering Symposium- LASES 2011. Ed. Sello Editorial Universidad de Medellín. Zapata & González-Calderón (Eds.), 1st Edition, pp.11-24, Medellín.

C. M. Zapata and B. Manrique (2012). Designing a structural a functional pattern of a corporate technical document for requirements elicitation: a first approach from corpus linguistics. In: Software Engineering: Methods, Modeling, and Teaching. Pontificia Universidad Católica del Perú (Eds.), Vol.2. ISBN: 978-612-4057-84-7.

C. M. Zapata, B. Manrique and G. P. Gasca (2012). An Analysis of the Requirements Elicitation Process from Technical Documents. In: Innovative ways of knowledge representation and management. Ed. Sello Editorial Universidad de Medellín. Zapata & González-Calderón (Eds.), 1st Edition, pp.11-24, Medellín.

C. M. Zapata and B. Manrique (2012). Transforming Natural Language into Controlled Language for Requirements Elicitation: A Knowledge Representation Approach. In "Advances in Knowledge Representation". Carlos Ramírez Gutiérrez (Ed.), InTech.

B. Manrique, C. M. Zapata and R. Manjarres (2013). Hacia una base de conocimiento terminológico para identificar conocimiento del dominio desde documentos técnicos corporativos. Memorias de las Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento.

B. Manrique, C.M. Zapata and D. Burgos (2013). Applying linguistic methods for a semiautomatic analysis of corporative documents: Standard Operating Procedure. In: LSP in Colombia: advances and challenges. Pedro Patiño & Gabriel Quiroz (Eds.). Peter Lang, Inc. ISBN: 978-3-0343-1434-3, pp.16-32.

Journal Publications

C. M. Zapata and B. Manrique (2011).Una revisión al modelado conceptual de un discurso a partir de descripciones en lenguaje natural. Conferencia Iberoamericana en Sistemas, Cibernética e Informática -CISCI 2011.

C. M. Zapata and B. Manrique (2011).Transformación de Lenguaje Natural a Lenguaje Controlado para la Educción de Requisitos, a Partir de Documentación Técnica. Cuaderno Activa, Institución Universitaria Tecnológico de Antioquia. ISSN: 2027-8101.

B. Manrique and G. Gasca (2011). Method to construct a taxonomy: basic structure to risks in software. Revista Facultad de Ingeniería, Universidad de Antioquia. ISSN 0120-6230, No. 60, pp. 92-101.

F.J. Moreno, B. Manrique and G. Orrego (2012). Smart PV: A Software Application For Managing English Phrasal Verbs. Revista Íkala, Vol. 17, No 3 ISSN: 0123-3432. www.udea.edu.co/ikala

G. Gasca and B. Manrique (2013). Taxonomía de riesgos de outsourcing. Revista chilena de ingeniería, Vol. 20 No. 1, ISSN 0718-3305.

C. M. Zapata and B. Manrique (2013). Diseño de Reglas Gramaticales para Transformar Documentos Técnicos Corporativos Escritos en Lenguaje Natural en Discursos Controlados. Revista Latinoamericana de Ingeniería De Software. Vol. 1, No 2.

B. Manrique and C. M. Zapata (2014). Transforming Natural Language into Controlled Language in Requirements Elicitation: A Preconceptual-Schema-Based Conceptual Synthesis. Revista Facultad de Ingeniería –Universidad de Antioquia, No. 70 pp.9-22, ISSN 0120-6230.

© Bell Manrique-Losada, 2014. All rights reserved.

Conference Proceeding

G. Gasca (2011). Similarity Study: A case study on software outsourcing. CERMA 2011: The Electronics, Robotics and Automotive Mechanics Conference. ISBN 13978076954568.

C. M. Zapata and B. Manrique (2012). Desde documentos técnicos hasta esquemas preconceptuales: un caso de estudio. Proceedings Workshop en procesamiento automático de textos y corpus (WOPATEC) 2012, Vol 1, Valparaiso.

C.M. Zapata, B. Manrique-Losada and G. González-Calderon (2012). An approach for using procedure manuals as a source for Requirements Elicitation. In Proceedings of Conferencia Latinoamericana en Informática -CLEI 2012, Vol. 1, pp. 1-8.

B. Manrique and C.M. Zapata (2012). Requirements Elicitation Process From Technical Documents. Seminario Internacional en Ciencias de la Computación, Universidad de Medellín.

C. M. Zapata and B. Manrique (2012). Designing and Solving Crossword Puzzles: Examining Efficacy in a Classroom Exercise. -39th ANNUAL CONFERENCE ABSEL 2012. In: Developments in Business Simulation & Experiential Exercises, Vol. 39 -Bernie Keys Library.

C. M. Zapata and B. Manrique (2012). Desde documentos técnicos hasta esquemas preconceptuales: un caso de estudio. Workshop en Procesamiento Automatizado de Textos y Corpora 2012, Valparaíso Chile.

B. Manrique and C. M. Zapata (2013). Processing rhetorical, morphosyntactic, and semantic features from corporate technical documents for identifying organizational domain knowledge. 25th International Conference on Software Engineering and Knowledge Engineering -SEKE 2013. Boston, USA.

B. Manrique, C.M. Zapata and D. Burgos (2013). Exploring MWEs for Knowledge Acquisition from Corporate Technical Documents. In: 9th Workshop on Multiword Expressions MWE 2013.

B. Manrique (2013). Modelo de transformación de documentación técnica corporativa en lenguaje natural en discursos en lenguaje controlado para la educción de requisitos. Simposio de Maestría y Doctorado en Universidad del Quindío.

Research Stays

Escuela de Lingüística de Valparaíso, Instituto de Literatura y Ciencias del Lenguaje –Pontificia Universidad Católica de Valparaíso. From May 15 to August 13, 2012. www.elv.cl/

Department of Romance Languages, Wake Forest University, Winston-Salem North Carolina. From April 15 to July 15, 2013. http://romancelanguages.wfu.edu/

Tutoring

Tomás Sarrazola. Implementación de un método para la generación de esquemas preconceptuales, a partir de documentos técnicos del área de negocios, en el dominio de Satrack Inc. Servisat S.A. Tutoring to Master Degree Project. Maestría en Ingeniería de Software, Universidad de Medellín.

Juan Ricardo Cogollo. An approach for defining indicators oriented to organizational control and technical Document Structures. Contributor and co-tutoring to Master Degree Project.Maestría en Ingeniería, Universidad Nacional de Colombia Sede Medellín.

Edwin Humberto Hincapié. Asistente Web para la generación y personalización de CRUDEL en el Framework PHP CORE. Tutoring to Master Degree Project.

Lina María Marín Vargas, Diego Alejandro Ruiz Gutierrez and Marina Graciano Velásquez. Aplicación de un método de revisión sistemática para el análisis y síntesis de literatura en transformación de lenguaje natural a lenguaje controlado en la educción de requisitos. Project to Specialization Degree. Especializacion En Ingenieria De Software, Universidad de Medellín, 2010.

Derived Research Projects

© Bell Manrique-Losada, 2014. All rights reserved.

'Revisión de Literatura en Transformación de Lenguaje Natural a Lenguaje Controlado en la Educción de Requisitos, a partir de documentación técnica'. Universidad de Medellín – Universidad Nacional de Colombia –Sede Medellín.

"Método de Transformación de Lenguaje Natural a Lenguaje Controlado en la Educción de Requisitos, a partir de documentación técnica'. Universidad de Medellín – Universidad Nacional de Colombia –Sede Medellín.

"Especificación de un Lenguaje Controlado de Dominio Específico: Fundamentos Lingüísticos y bases de Transformación desde Documentos Técnicos Corporativos en Lenguaje Natural". Proyecto aprobado en "Programa Nacional De Proyectos Para El Fortalecimiento De La Investigación, La Creación Y La Innovación En Posgrados De La Universidad Nacional De Colombia 2013-2015" y Convocatoria No. 33 de la Universidad de Medellín. Universidad Nacional de Colombia, Universidad de Medellín y Wake Forest University.

Conference Organization

IV SEMINARIO INTERNACIONAL DE CIENCIAS DE LA COMPUTACIÓN, October 27 and 28, 2011, Universidad de Medellín.

LATIN AMERICAN SOFTWARE ENGINEERING SYMPOSIUM- LASES 2011. Medellín, Antioquia.

V SEMINARIO INTERNACIONAL DE CIENCIAS DE LA COMPUTACIÓN, November 4 and 5, 2012, Universidad de Medellín.

VI SEMINARIO INTERNACIONAL DE CIENCIAS DE LA COMPUTACIÓN, October 15 and 16, 2013, Universidad de Medellín.

© Bell Manrique-Losada, 2014. All rights reserved.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

Let's not pretend that things will change if we keep doing the same things
-- Albert Einstein

Knowledge is power.
-- Francis Bacon

The previous chapters have shown how a formalization for mapping discourses from business-based technical documents into controlled language texts for RE process can occur. This final part of this Ph.D. Thesis deals with the conclusions in Section 8.1; and a brief look into the future work in this research field in Section 8.2.

8.1 CONCLUSIONS

We organize the conclusions in terms of the several points of view, by answering the research questions from the introduction on the basis of the results presented in the previous chapters.

Related to the RE process

We support the value of using Background Reading technique as a relevant elicitation technique in the context of the RE process, by proposing a specific method for applying it. We consider the BR technique a supporting complementary technique, for improving the efficiency and effectiveness of the domain knowledge identification and understanding. Thus, BR is coherent and potentially combinable with other methods. Thus, the usage of business-based technical documents in the RE process has an incremental growth.

In this sense, we describe a set of business-based technical documents potentially useful for identifying domain knowledge by applying BR. One of them is SOP, for which we define a model comprising structural, functional, and linguistic features of a SOP. Such features can be processed and analyzed for generating a controlled language.

We found that, despite its maturity, NLP is a lively and flourishing discipline, with many challenges to overcome in the context of RE process, given the underlying ambiguity of NL related in this discipline.

Related to the discourse analysis approach from the RE process

We paid special attention to the distinction among the discourse analysis methods and NLP for business-based technical documents. Although scientific communities that have approached the processing of such documents usually work in isolation, NLP applications achieve a combined use of techniques from various approaches. Our experience in this field makes us think that it is possible to say that the usage of one mixed approach is very appropriate and that the inclusion of methods and theories of discourse analysis for disciplines such as RE generates very good results.

Likewise, we state the discourse analysis approach as efficient as others applied for the analysis of documents, such as data mining, knowledge engineering techniques and formal methods. This approach can be really useful for similar purposes to the Ph.D. Thesis purpose. Our proposal lies in the differentiation in the preliminary document analysis by considering each kind of business-based technical document as a genre. Based on such a consideration, we look for the functional, structural, and subsequently linguistic patterns—morphological, lexical, syntactic, and semantic—actually obey and are aligned with the writer communicative purpose. In this way, inside the information and knowledge generated from the document analysis—following our approach for mapping—the original function of the text will remain according to the communicative intention of the author.

© Bell Manrique-Losada, 2014. All rights reserved.

Related to the validation

The validation result show measures for frequencies, precision, recall, and time of mapping. The results achieved with the validation of the model in terms of comparing the production of expert with NAHUAL are excellent. Based on the maturity level of the processor built is a major achievement that is important to emphasize and represents the spirit of this proposal.

In the analysis of the generated information with the validation studies, most of the extracted information was in terms of roles, responsibilities, and restrictions to the responsibilities. We might think that it is a great approach to a domain model, information requirements, and non-functional requirements.

The identification of organizational knowledge is achieved in terms of roles, actors and responsibilities of those who are involved in procedures, objectives of organization, procedures or functions of agents, and conditions/requirements for developing procedures.

The performed analysis is a part of the initial knowledge acquisition and preliminary processing for understanding of organization domain. The knowledge will then be readily available for use by future applications in a specific domains, to validate the findings and then to automate the process.

Related to the mapping formalization

Based on our work for discourse processing from business-based technical documents, we propose a formalization in terms of a rhetorical organization model, a processing model, and an architectural model, and demonstrate its performance by means of a document mapping system, as follows:

- We present a first approach of a pattern for analyzing and writing SOP, under the concept of a rhetorical organization model. By using the rhetorical analysis method, we show features in terms of functional and structural aspects, based on macro-moves, moves, and functions—communicative purpose. The proposed features comprise three macro-moves, which serve an overall communicative purpose, and 11 moves shaping the organization units of SOP.
- This study aims at characterizing SOPs and JSDs by revealing key Multiword expressions (MWEs) used in an English corpus. We proposed a set of rules—morpho-syntactic and semantic—for mapping NL into CL, MWEs as lexical phrases which can be processed as input for further knowledge engineering processes.
- We propose a processing model for executing the business-based document analysis by following levels of text processing—pre-processing, lexical, parsing, and semantic.
- We present an architectural model as an approach for implementing the proposed formalization in a development environment.

Related to the general contributions

The main contributions of this research can be summarized as follows:

- A proposal for using *background reading* as a RE technique, conductive to reduce time and costs in the initial phases of the software development process.
- A justification for employing business-based technical documents as data source in the RE process.
- An approach to follow best practices when defining, reviewing, and writing business-based technical documents, focused on administrative and business disciplines.
- A mapping formalization for achieving an initial understanding of the organizational context.
- A method for identifying domain knowledge and business information from the early stages of the RE process.
- Innovation in the requirements engineering, regarding:
    - Considering several sources for helping to reduce costs and facilitate subsequent task in the context of RE process.
    - Applying of discourse analysis methods and techniques in the processing of technical documents as sources for RE techniques
    - Approaching to an automated process for eliciting information with purposes of RE.

© Bell Manrique-Losada, 2014. All rights reserved.

- This research has presented work towards new corpora and tools for business-based technical documents research. Moreover, a new balanced corpus of documents has been presented along with a text-corpus collected from websites.

8.2 FUTURE WORK AND OPEN ISSUES

In the short term many directions can be identified for future work. A list of the most important ones is given below:

- The approach has to be more extensively validated in order to determine the extent to which it can offer support to mapping. Such validation includes two aspects: i) it has to be exposed to a set of expert users, such as software developers and software analysts, in order to analyze its scope and usability; ii) it has to be applied to complex case studies by involving industry partners, in order to analyze its scalability. Such experimentation is of worth interest because the final purpose is to integrate our proposal in the daily practices of the software analysts.

- Detection and the solution of anti-patterns for the proposed mapping rules require some pending issues that give rise to short term goals.

- As additional future work, we expect:
  - To increase the number of technical documents in the corpus and refine the study of lexical and semantic features.
  - To consider statistical association measures for reinforcing term identification and pattern extraction in the frame of knowledge acquisition from corporate documents.

- Potential applications of the Ph.D. Thesis contributions

  1) Business Views from specification in CL. The views from the business to architecture are visualized from the RE process. We define as a potential research line the generation of such views from the RE, based on the proposed mapping. Some of the views where tried to establish a set of categories in which the information's role in the enterprises is treated and that could be generated, are:

  - Scope: Defines the enterprise direction and business purpose in order to establish the context for a specific development software project. This view includes definitions of the boundaries of system and processes involved.
  - Enterprise: defines the business terms expressing the nature of the business, e.g. structure, processes, organization, and so forth.
  - Fundamental concepts (architect's view): defines the business based on the fundamental structures in more rigorous terms.

  2) Discourse processing in the organizational adaptation of information systems, looking for emphasizing on the role of professional discursive practices in shaping the process of organizational adaptation of information systems.

  3) Rhetorical characterization of management discourses. By using our proposal of rhetorical analysis for business-based technical documents, we consider a challenging application for processing management of written discourses.

  4) Monitoring an exploiting of enterprise behavior. The area of models and languages of business has been subject of research over the last decade as part of disciplines such as normative systems and multi-agent systems. The autonomous nature of stakeholders and the implicit dynamic of the process involved are an important way of specifying behavioral constraints and describing basic behavior by using proposal as we present in this Ph.D. Thesis. In broader terms, we refer to exploiting constraints sets as enterprise policies, useful for understanding and identifying the business domain.

© Bell Manrique-Losada, 2014. All rights reserved.

5) Authoring Tool for supporting the creation of technical documents. The creation of documentation should be supported by an authoring tool (which could be used, for instance, by technical editors, technical stakeholder, or even a business director), where they specify the content of a particular document and then can be selected and combined into a representation of such document—controlled language. A prototypical authoring tool for generating technical documents can designed which provides interactive support in managing knowledge bases and creating documents.

© Bell Manrique-Losada, 2014. All rights reserved.

APPENDICES

(ATTACHMENT)

© Bell Manrique-Losada, 2014. All rights reserved.