Nominalizations in Requirements Engineering Natural Language Models

Claudia S. Litvak
Universidad Nacional de La Matanza, Argentina & Universidad Nacional del Oeste, Argentina

Graciela Dora Susana Hadad
Universidad Nacional del Oeste, Argentina

Jorge Horacio Doorn
Universidad Nacional del Oeste, Argentina & Universidad Nacional de La Matanza, Argentina

INTRODUCTION

The first activity of the development of a software system is to define its requirements. Requirements engineer must interact mainly with clients and users among other stakeholders (Macaulay, 1993). He or she has to understand the context in which the future system will act and he or she must carefully consider the reasons that conducted to the decision of developing such system. Requirements engineer's responsibilities include establishing a fluid communication with all stakeholders to produce reliable documents that will be used later in the software development process (Leite et al., 2004). Usually the culture, knowledge and skills of clients and users are rather different from those of the software development experts. As part of the communication with clients and users, the requirements engineer must clearly show them the characteristics of the software system that he or she is conceiving to attend clients and users' issues. Requirements Engineering process has two main activities: to understand the application domain, and to correctly define the services that the future software system will provide (Leite et al., 2004). As a general practice, both activities involve the development of models that describe such domain. It is also a common practice to develop such models in natural language to enhance the communication among stakeholders (Rolland & Ben Achour, 1998; Leite et al., 2004; Seiff et al., 2009); however, this introduces some obstacles, such as ambiguity, incompleteness and poor information structuring (Zowghi & Gervasi, 2002; Berry & Kamsties, 2004; Leite et al., 2005; Doorn & Ridao, 2009; Hadad et al., 2015). All of these inconveniences come from the natural language itself. As a consequence, Requirements Engineering has become more and more involved with linguistic considerations. Furthermore, it should be kept in mind that language conveys culture and knowledge (Nettle & Romaine, 2000; Fishman, 1999). Thereby, the terminology of clients and users holds application domain knowledge. Therefore, to document and to slightly formalize the relevant words or phrases heard from clients and users or read from documents is a valuable practice. In other words, creating a glossary of such terminology helps the Requirements Engineering process in two relevant ways: it eases the understanding of the application domain and it reduces the ambiguity of the oral communication with clients and users and the ambiguity of every produced document (Hadad, Doorn, & Kaplan, 2009).

However, the glossary construction itself introduces some drawbacks. The most important of these drawbacks, not yet treated, is the presence of nominalizations, either in the clients and users' terminology or in the produced glossary. The former is a possible source of ambiguity while the latter is a hint of an adequate or inadequate

DOI: 10.4018/978-1-5225-2255-3.ch445

creation of glossary symbols by the requirements engineer. Both may cause defects in the glossary. Nominalization refers to the construction of nouns from verbs or adjectives. Linguistic authors have largely studied nominalization in many languages such as English, French, German, Russian, Spanish, etc. (Alexiadou, 2001; Bisetto & Melloni, 2005; Grimshaw, 1990; Rothmayr, 2009; Rozwadowska, 1997). The simplest way to describe verb nominalization is by means of the phrase *action of and effect of*. In some cases, nominalization occurs only by *action of*; while in others only occurs by *effect of*.

In this chapter, the influence of nominalization on the quality of Requirements Engineering documents is analyzed. The requirements engineer should be aware of the substantial differences in meaning, that sometimes arise when using the nominal mode of a verb or its verbal mode, since the action and the effect of a verb nominalization may produce synonyms or even homonyms.

BACKGROUND

The work presented in this chapter is based on the lessons learned in several research projects where many study cases were created using Scenarios and Language Extended Lexicon (LEL) models (Leite et al., 1997; Leite et al., 2000; Leite et al., 2004). LEL is a glossary proposed by Leite and Franco (1993). Scenarios are structured natural language descriptions of situations that occur in the application domain (Leite et al., 2000).

The LEL is itself a glossary with roles and structure different from the usual ones. It is composed by a set of symbols, which are words or phrases peculiar and frequently used in the application domain. Each symbol is identified by a name or names. An acronym or an abbreviation may be also a name of a term, only if present in the application domain. In case of synonyms the more relevant name is used as the main key entry. Every symbol has two types of descriptions; this particular structure makes the difference with other glossaries. The first type, called Notion, is the usual one and describes the denotation of the word or phrase, that is, it defines what the symbol is. The second, called Behavioral Response, describes the connotation of the word or phrase, that is, it describes how the symbol acts in the application domain; this description is not usually present in other glossaries and enriches the knowledge about the symbol and the context at hand. LEL symbols contain hypertext links pointing to directly related entries.

When the denotation of the term acquires several meanings, it indicates the existence of homonyms, which forces the creation of more than one entry in the lexicon. The absence of any behavioral response indicates that the symbol does not belong to the LEL.

LEL entries are classified in four types according to its general use in the application domain. The types are: Subject, Object, Verb and State. Table 1 shows the LEL model.

Symbols of type object, verb and state may be affected by a linguistic transformation, called nominalization, which may hide the type of the symbol and may produce confusions between verbs and objects, or between verbs and states. Nominalization is a source of ambiguities, especially in behavioral responses. It has not received the necessary attention in the Requirements Engineering literature (Berry & Kamsties, 2004; Kovitz,

Table 1. Language EXTENDED LEXICON model

LEL: representation of the symbols in the application domain language. Syntax: $\{\text{Symbol}\}_1^N$
Symbol: entry of the lexicon that has a special meaning in the application domain. Syntax: $\{\text{Name}\}_1^N + \{\text{Notion}\}_1^N + \{\text{Behavioral Response}\}_1^N$
Name: identification of the symbol. More than one represents synonyms. Syntax: Word | Phrase
Notion: denotation of the symbol. Syntax: Sentence
Behavioral Response: connotation of the symbol. Syntax: Sentence

Source: (Leite et al., 2000)

Related Content