Natural Language Processing: An Inevitable Step in Requirements Engineering

A. Egemen Yilmaz, Department of Electrical-Electronics Engineering, Ankara University, Ankara, Turkey

ABSTRACT

Requirement analysis is the very first and crucial step in the software development processes. On the other hand, as previously addressed by other researchers, it is the Achilles’ heel of the whole process since the requirements lie on the problem space, whereas other software artifacts are on the solution space. Stating the requirements in a clear manner eases the following steps in the process as well as reducing the number of potential errors. In this paper, techniques for the improvement of the requirements expressed in the natural language are revisited. These techniques try to check the requirement quality attributes via lexical and syntactic analysis methods sometimes with generic, and sometimes domain and application specific knowledge bases.

Keywords: Ambiguity Resolution, Lexical Analysis, Linguistic Ambiguity, Natural Language Requirements, Requirement Analysis, Requirement Quality Attributes, Requirements Engineering, Similarity Analysis, Syntactical Analysis

INTRODUCTION

System/software requirement quality is one of the most important drivers for the success of a final system/software related product. But ironically, as stated by Kof (2005), in practice “requirements engineering is the Achilles’ heel of the whole software development process”, because requirements documents are usually inconsistent and incomplete. Cheng and Atlee (2007) identify the main reason for this as follows: “… because requirements reside primarily in the problem space[,] whereas[,] other software artifacts reside primarily in the solution space”.

A study showed that the user involvement is the most important factor for the success of the software development projects (Standish Group, 1995). Moreover, even in the cases where the user involvement is sufficient, the success is dependent on clear statement of the requirements, which appears as the third most important factor on the list. In addition, other studies showed that the underlying reason for 60 to 85% of the software errors during a system’s life time is nothing but the requirement defects (Davis, 1990; Schach, 1992; Young, 2001). On the other hand, if such defects could not be fixed at the early phases of the projects, the cost of fixing the error would dramatically increases at each phase of the development life cycle as seen in Table 1 (Young, 2001). Defects detected during requirement analysis and design phases could reduce the rework effort between 40%
and 50% (Boehm, 1981; Boehm & Basili, 2001; Gause & Weinberg, 1989).

All the above studies support that, effective customer interaction and defect-free requirement engineering processes are the key factors for successful system/software development.

Pohl’s requirement engineering process model (Pohl, 1994), depicted in Figure 1, might be considered as a reference for the improvement of the software requirement quality. Pohl’s model suggests focusing on the three dimensions seen in Figure 1. Stakeholders of the requirement engineering process should perform a progress on the specification, representation and agreement dimensions. Hence, tools supporting progress in any of those three dimensions would increase the success rates of software development projects.

The applicability of natural language processing techniques is best understood via the following two-phase requirement engineering model. Figure 2 depicts a high level representation of these two stages.

The first phase, which can be referred to as “Perception of the domain”, is a set of activities performed in order to understand the domain components and their relationships. Engineers embark upon to this stage at the very early phases of the development. This cognitive stage is usually followed by the representation of the information learned so far, where this representation is required to guarantee the correctness and completeness.

The second phase includes the “Expression of the requirements” in terms of visual and verbal statements. Those statements are used for understanding the rest of the domain; they are also important for achieving a better understanding among the end user, the buyer (if different than the end user), and the development team(s). Both the specification and representation qualities are important in order to deliver the right ‘message(s)’ from the end user to the developers; and in order to implement the appropriate functionalities for the end user by the developers.

According to Hooks and Kristin (2001), the distribution of all requirement-related defects is as follows:

- 49% of the requirement-related defects are due to wrong assumptions and wrong requirements,
- 29% of them are due to missing requirements,
- 13% of them are due to inconsistent requirements,
- 5% of them are due to ambiguous requirements

Wrong assumptions and missing/omitting requirements can be encountered in the “Perception of the domain” phase. On the other hand, the most common mistakes made during the “Expression of requirements” phase consist of inconsistent, unclear and ambiguous statements. When proceeding in the ‘specifications’ dimension of the Pohl model, such defects can be detected while the natural language requirements are being analyzed.

In this paper, we will try to discuss and present the applicable natural language processing

<table>
<thead>
<tr>
<th>Phase in which the Error is Found</th>
<th>Relative Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Analysis</td>
<td>1</td>
</tr>
<tr>
<td>Design</td>
<td>3-6</td>
</tr>
<tr>
<td>Coding &amp; Unit Test</td>
<td>10</td>
</tr>
<tr>
<td>Development Testing</td>
<td>15-40</td>
</tr>
<tr>
<td>Acceptance Testing</td>
<td>30-70</td>
</tr>
<tr>
<td>Operation</td>
<td>40-1000</td>
</tr>
</tbody>
</table>
Fostering Analysis from Industrial Embedded Systems Modeling
www.igi-global.com/chapter/fostering-analysis-from-industrial-embedded-systems-modeling/116113?camid=4v1a

Service-Oriented Enterprise Architecture
www.igi-global.com/chapter/service-oriented-enterprise-architecture/28953?camid=4v1a