Early Analysis of Natural Language Requirements

F. Fabbrini, M. Fusani, S. Gnesi, G. Lami, G. Trentanni
Information Science and Technology Institute – C.N.R., Pisa - Italy

Abstract

While in practice Natural Language (NL) is the most used mean for expressing requirements, there is a lack of supporting tools and techniques for the analysis of this kind of requirements. The QuARS (Quality Analyzer for Requirement Specifications) tool has been designed with the aim to automatize the analysis of NL requirements "as they are", i.e. with no need to move towards another formalism. The tool is able to perform an evaluation of the Expressiveness of an NL requirements document and also provides support for completeness and consistency analysis.

1. Introduction

The achievement of the quality of software requirements is the first step toward software quality. It is well known that inaccuracies in requirement documents could determine serious problems at all the consequent phases of software development. The availability of methods and tools for the analysis of software requirements may improve the effectiveness of the requirement process and the quality of the final product. The problem of the analysis of software requirements with respect to some quality characteristics (e.g. non ambiguity, consistency, correctness) has been extensively exploited in recent years [3, 4].

For example, formal methods and tools have been used for this purpose when a formal representation of software requirements has been adopted [1]. Nevertheless, the use of formal methods for specifying and consequently validating software requirements is not widely spread in the industry: it is, in fact, usually limited to particular domains as - for example - safety critical applications. For the largest part of the software projects the mean to represent requirements is NL. Furthermore, even in the case of the application of formal methods, the formal representation of the requirements is derived starting from their initial description in NL.

For this reason, methods and tools for the analysis of requirements expressed in NL become very relevant to reduce the risk of late error detection.

In practice, there are numerous tools and techniques to manage requirements. Many are oriented to define requirements, provide configuration management and control distribution. However, there is a scarcity of automatic support for the quality analysis of NL requirements [5].I In the software industry the verification of software requirements against a quality model including ambiguity analysis, consistency and completeness is often made by humans with a clerical and tedious process that consists in reading of requirements documents looking for linguistic defects [2]. In our poster we intend describe QuARS, a tool developed at CNR-ISTI. for automatic NL requirement analysis.

2. The QuARS Tool

QuARS (Quality Analyzer for Requirement Specifications) is an innovative tool that provides the user with the capability of performing the analysis of NL requirements in an automatic way. This tool, by performing a lexical and syntactic parsing of the NL requirements, provides the following functionalities:

1) Defective Sentences Identification: Expressiveness Evaluation

Similar to any other evaluation process, the quality evaluation of NL software requirements has to be conducted against a Model. The Quality Model we defined for the Expressiveness property of NL software requirements (i.e. the capability to avoid incorrect understanding of their meaning) is aimed at providing a way to perform a quantitative (i.e. that allows the collection of metrics), corrective (i.e. that could be helpful in the detection and correction of the defects) and repeatable (i.e. that provides the same output against the same input in different contexts) evaluation. The Expressiveness quality model is composed of three quality characteristics to be evaluated by means of indicators. Indicators, in this case, are linguistic components of the requirements directly detectable and measurable on the requirement document revealing defects in requirements. Special Dictionaries contain the indicators QuARS needs for the analysis.

The Expressiveness characteristics are:

- *Non-ambiguity*: the capability of each requirement to have a unique interpretation.
- *Understandability*: the capability of each requirement to be fully understood when used for developing software and the capability of the Requirement Specification Document to be fully understood when read by the user.
- *Specification Completion*: the capability of each requirement to uniquely identify its object or subject.

QuARS, by means of lexical and syntactic analysis of the input file, is able to point out those sentences containing defects according to the quality model.

2) Requirements Clustering: View Derivation

QuARS also provides support for consistency and completeness analysis by means of the View derivation

functionality. A View is a sub-set of the input requirements document composed of those sentences dealing with a particular topic. The availability of Views makes the detection of inconsistencies and incompletenesses easier because the reviewer has to consider smaller sets of sentences where possible defects can be found with much less effort.

3. Using QuARS

The use of QuARS allows a systematic and disciplined NL requirements analysis process (Figure 1).

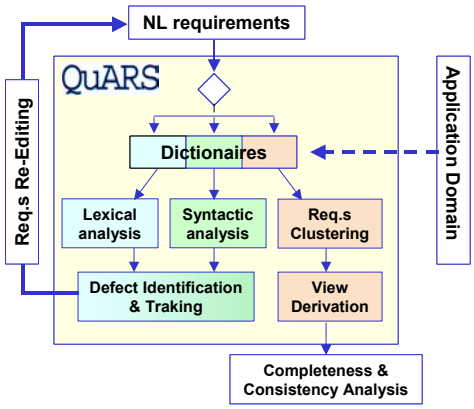

Figure 1. NL requirements analysis with QuARS

The input of the tool is composed of the requirements document to be analyzed in plain text format. The tool relies on a set of indicator-related dictionaries; they contain either the terms indicating a kind of defects according to the quality model or a domain-related corpus to be used for the View derivation. The dictionaries are in simple text format. Once the user selects the type of analysis to be performed, the correspondent dictionary is made available for being tailored according to the application domain. If the Expressiveness analysis is performed (Figure 2), depending on the type of defect to be searched, the input file is lexically and/or syntactically analyzed in order to point out the related indicators.

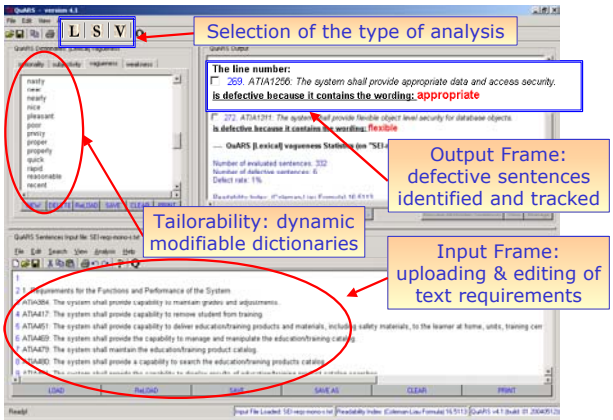

Figure 2. Expressiveness analysis

The list of defective sentences is displayed in the GUI's output frame and a log file is created. The defective sentences can be tracked back to the input requirement document and corrected, if case. Metrics measuring the defect rate and the readability of the input requirement document are calculated and stored. The output of the View derivation function is a clusters of requirements displayed in the GUI's output frame and stored in a log file ready to be used for consistency and completeness analysis purposes.
The graphical representation (as a MS Excel graph) of the number of occurrences of those sentences belonging to a View in the single sections the document is composed of is provided too (Figure 3).

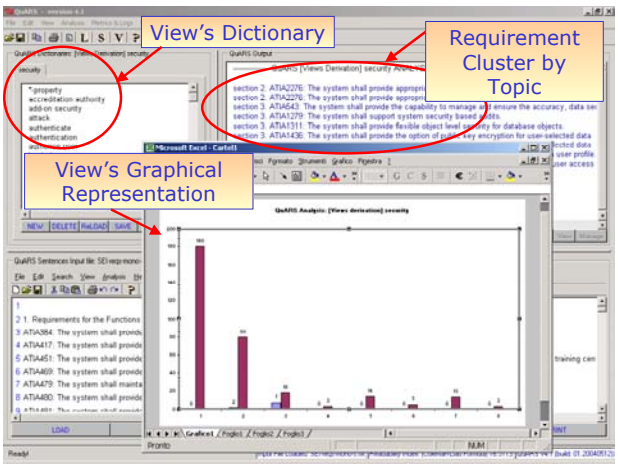

Figure 3. View derivation

4. Conclusions

The QuARS tool, because of its precision, ease of use and tailorability, can strongly improve the requirements analysis process. Experimentations aiming at achieving an empirical evaluation of the impact of the QuARS approach in the software development process and the quality of the final product are on going.

5. References

[1] J.P. Bowen, M.G. Hinchey *Industrial-Strength Formal Methods in Practice*, Springer Verlag, New York, 1999.
[2] D. Firesmith "Specifying Good Requirements" *Journal of Object Technology*, vol. 2, no. 4, July-August 2003, pp. 77-87.
[3] B. Kamsties, B. Peach "Taming Ambiguity in Natural Language Requirements", *Proc. of ICSSEA 2000*, Paris 2000.
[4] L. Mich, R. Garigliano "Ambiguity Measures in Requirement Engineering" *Proc. of the Int. Conf. On Software Theory and Practice*, Beijing, China, Aug. 2000.
[5] W.M. Wilson, L.H. Rosenberg, L.B. Hyatt "Automated Analysis of Requirement Specifications" *Proc. of ICSE 97*, Boston 1997.