

An Automatic Tool for Improving the Quality of Software Requirements

by Giuseppe Lami and Gianluca Trentanni

The Quality Analyzer for Requirements Specifications (QuARS) is an innovative tool that enables the user to analyse natural language requirements automatically.

Natural Language (NL) is still the most common way to express software requirements. In practice, there are numerous tools and techniques for managing such requirements; many are oriented towards the definition and provision of configuration management, and control the distribution of requirements. However, there is a scarcity of automatic support for the quality analysis of requirements expressed in NL. In the software industry, ambiguity analysis and consistency and completeness verification of software requirements are performed by humans, generally through a tedious clerical process that consists in reading requirements documents and looking for linguistic errors. For these reasons, we have developed QuARS (Quality Analyser for Requirements Specifications), an innovative tool that enables the user to analyse requirements expressed in NL automatically.

QuARS performs a lexical and syntactical parsing of software requirements expressed in NL, and provides the following functionalities.

Defective Sentence Identification (Expressiveness Evaluation)

Similar to any other evaluation process, the quality evaluation of NL software requirements must be conducted against a model. The Quality Model we have defined for the Expressiveness property of the requirements produces an evaluation that is quantitative (ie that allows the collection of metrics), corrective (ie that could be helpful in the detection and correction of defects) and reproducible (ie that provides the same output against the same input in every domain).

The Expressiveness quality model consists of three features, to be evaluated by means of indicators. Indicators are linguistic components, directly detectable and measurable in the requirements document, which reveal a defect in requirements. Special dictionaries contain the indicators QuARS needs for its analysis.

The Expressiveness features are:

- non-ambiguity: the property of having a unique interpretation
- understandability: the property of being fully comprehensible both when used by the software developer and when read by the user in the Requirements Specification Document
- specification completion: the ability of each requirement to uniquely identify its object or subject.

QuARS performs a lexical and syntactic analysis of the input file in order to identify those sentences containing defects according to the quality model.

Requirements Clustering (View Derivation)

QuARS also provides support for consistency and completeness analysis by means of the View derivation functionality. A View is a subset of the input requirements document, composed of those sentences dealing with a particular argument. The availability of Views makes the detection of inconsistencies and incompleteness easier because the reviewer need consider only smaller sets of sentences, in which possible defects can be found with much less effort.

Using QuARS

QuARS provides a systematic and disciplined NL-requirements analysis process (see Figure 1). Input to the tool is composed of the requirements document which is to be analysed. The permitted format of this input file is plain text. This file is given to the syntax parser component, which produces a new file containing the parsed version of its sentences. The tool relies on a set of indicator-related dictionaries; these may contain either terms indicating a defect type according to the quality model, or the domain dictionaries used for the View derivation. The dictionaries are in simple text format. Once the user selects the type of analysis to be performed, the corresponding dictionary is made available and can be tailored, if desired, according to the application domain.

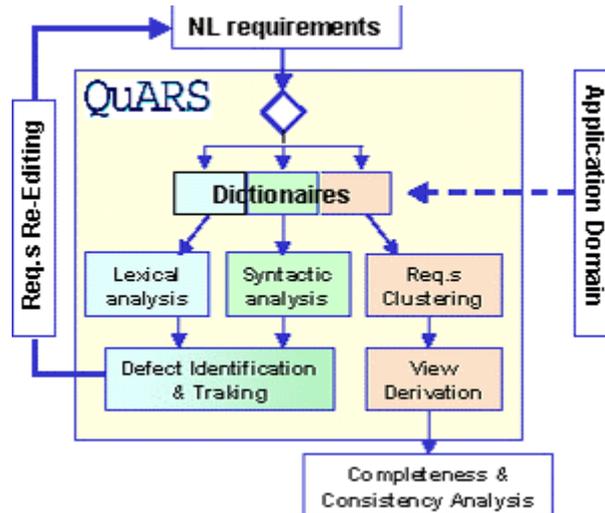


Figure 1: The Natural Language requirements analysis process with QuARS.

When the Expressiveness analysis is performed, 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 in the input requirements document and corrected if necessary (see Figure 2). Metrics measuring the defect rate and the readability of the requirements document under analysis are calculated and stored.

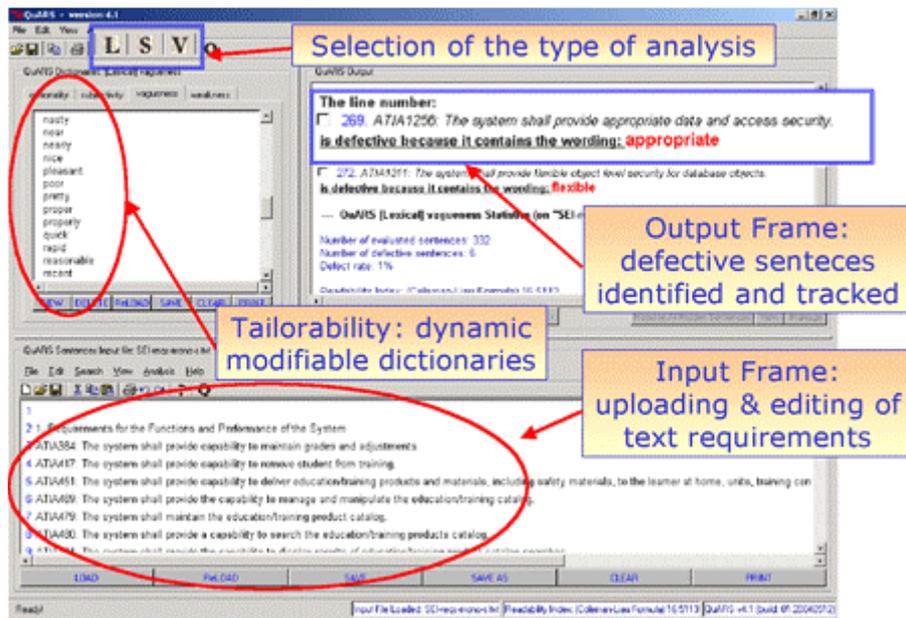


Figure 2: QuARS expressiveness analysis.

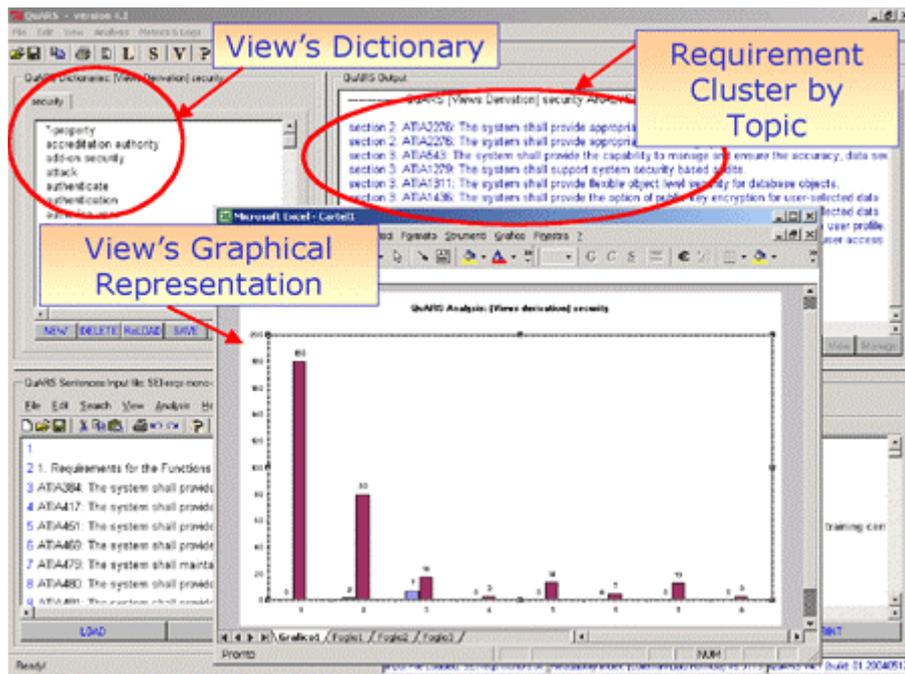


Figure 3: View Derivation.

The output of the View derivation function is a log file containing the necessary clusters of requirements, ready to be used for consistency and completeness analysis purposes. The graphical representation (in an MS Excel graph) of the number of occurrences of these sentences belonging to a View in single sections of the document is also provided (see Figure 3).

Because of its precision and ease of use, the QuARS tool can greatly improve the analysis of NL requirements. Experiments aimed at evaluating the impact of QuARS in the software development process and on the quality of the final product are currently in progress.

Link:

<http://fmt.isti.cnr.it/quars>

Please contact:

Giuseppe Lami, ISTI-CNR, Italy
 Tel: +39 050 315 3493
 E-mail: giuseppe.lami@isti.cnr.it

Gianluca Trentanni, ISTI-CNR, Italy
 Tel: +39 050 315 3471
 E-mail: gianluca.trentanni@isti.cnr.it