QuARS versus QuARS Express

Gianluca Trentanni

(gianluca.trentanni@isti.cnr.it)

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo",

CNR, Pisa, Italy

Abstract. *QuARS and QuARS Express (Quality Analyzer for Requirements Specifications) are tools that make it easier to extract structured information and metrics for detecting linguistic inaccuracies and defects in software requirements expressed in Natural language. The Express edition of QuARS represent a parallel evolution of the main tool exploiting the same core engine for an increased usability and a more expressive set of reports. In this article a comparison is presented.*

Keywords: Software Engineering, Requirements Analysis, Requirements Specifications, Metrics, Natural Language Processing

Introduction

QuARS was developed to automatize NL requirements analysis systematically. The aim was to develop a modular, extensible tool with a user friendly graphical interface.

QuARS [2, 3] performs the expressiveness analysis by means of a lexical and syntactic analysis of the input file in order to identify those sentences containing defects [1] according to the specified quality model.

When the Expressiveness analysis is performed, the list of defective sentences is displayed by QuARS and a log file is created. The defective sentences can be tracked in the input requirements document and corrected, if necessary. Metrics measuring the defect rate and the readability of the requirements document under analysis are calculated and stored.

QuARS (Quality Analyzer for Requirements Specification) has been used in the ModTrain/ModControl project to analyse the project requirements documents.

The initial need has been represented by a simply customization of the major QuARS tool for the massive use on large (~ 2*10^3) sets of requisites in "a-click-away" fashion. The "*new*", namely QuARS *Express,* followed the further step that has naturally been to enlarge the data set adding some metadata fields producing highly structured and meaningful reports.

In the following a comparison amongst the two tools is presented.

Background Scenario

Figura 1 - Modtrain Web page

Within the definition of the train and vehicle architecture, the Train Control and Monitoring System (TCMS) represents a critical subsystem to achieve the required performance in terms of integration and maintenance cost. The TCMS can be considered as the "nervous backbone" of the train and is a subsystem with well-identified own characteristics, satisfying a specific set of requirements, and with clear interfaces with other on-board subsystems and ground facilities.

In the specification phase for TCMS, project partners have gathered requirements from different existing sources. These requirements had to be consolidated, harmonized and refined among the various project partners. An analysis of the natural language requirements by means

of automatic tools has been considered as an added value for guaranteeing the successful outcome of the project, due to the capability to point out potential sources of ambiguity and other weaknesses. A modified version of the QuARS tool, namely QuARS *Express,* has therefore been developed for the MODCONTROL project, to address these needs. In particular QuARS *Express* is able to handle a more complex and structured data format containing metadata and produces an analysis report richer of categorized information.

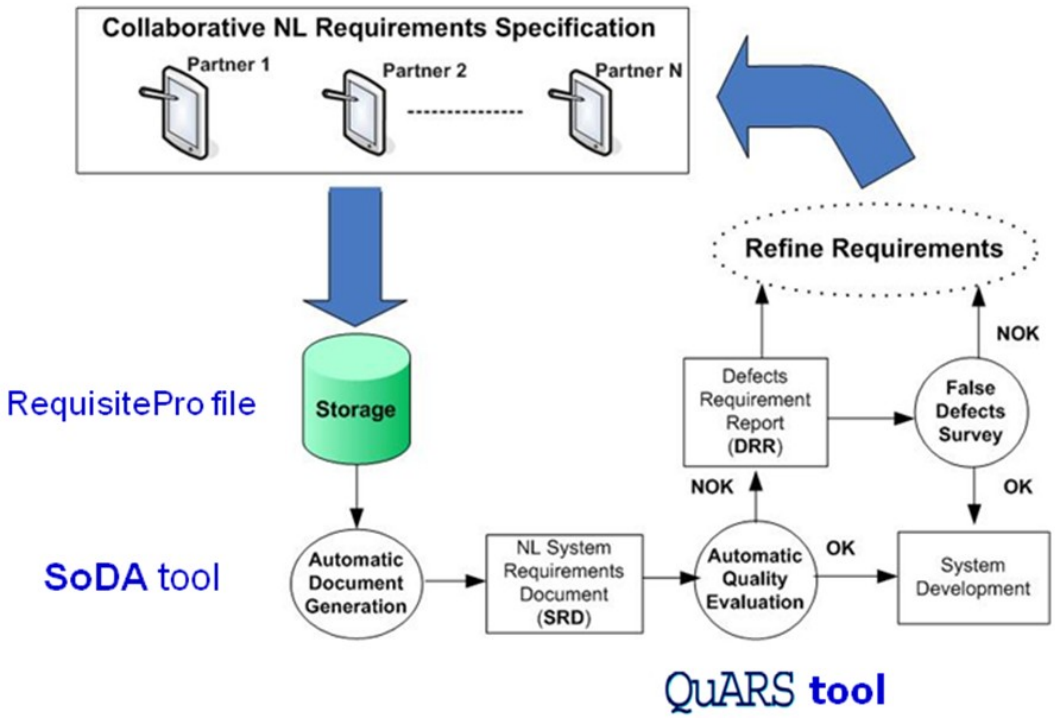

Figura 2 - Quality Analysis Process scheme in MODCONTROL

The information grows as a function of the number of metadata items available (e.g. as a function of the number of authors, the number of packages and so on) and the size of the report grows

consequently and can be composed of several pages. As an improvement of the simple text based report made by QuARS the new report exploits the HTML technology to produce structured hypertextual pages. Using QuARS *Express* the Functional and System Requirements of TCMS including more than 5700 requirements had been analyzed.

QuARS Reqs Format vs *QuARS Express* Reqs Format

It's a common strategy of several commercial product to use proprietary file formats to create a kind of "private" market.

For example the widely used (expecially in the industry environment) Microsoft products like ***MS Word*** force everybody to face their close, binary, unfriendly file formats.

On the contrary, as known, QuARS works with pure plain text and some sort of export or preprocessing is a mandatory need. This is a requiiste for QuARS *Express* too but the format is richer since the metadata fields have been introduced following the project needs.

The first analysis of the ModTrain-ModControl project requisites documents made using QuARS (version 4.1) produced good results pointing out several wording defects spread all over the documents and the sections and affecting almost all the authors This result produced the request of a more accurate analysis to point out the exact responsibility for corrections and the documents to check with a greater attention. One way to reach such results was clearly to tie up the single sentence to its belonging author, the document of origin and the related sub system referred. Hence some metadata fields have been

added to the analysed requirements as interchange data format.

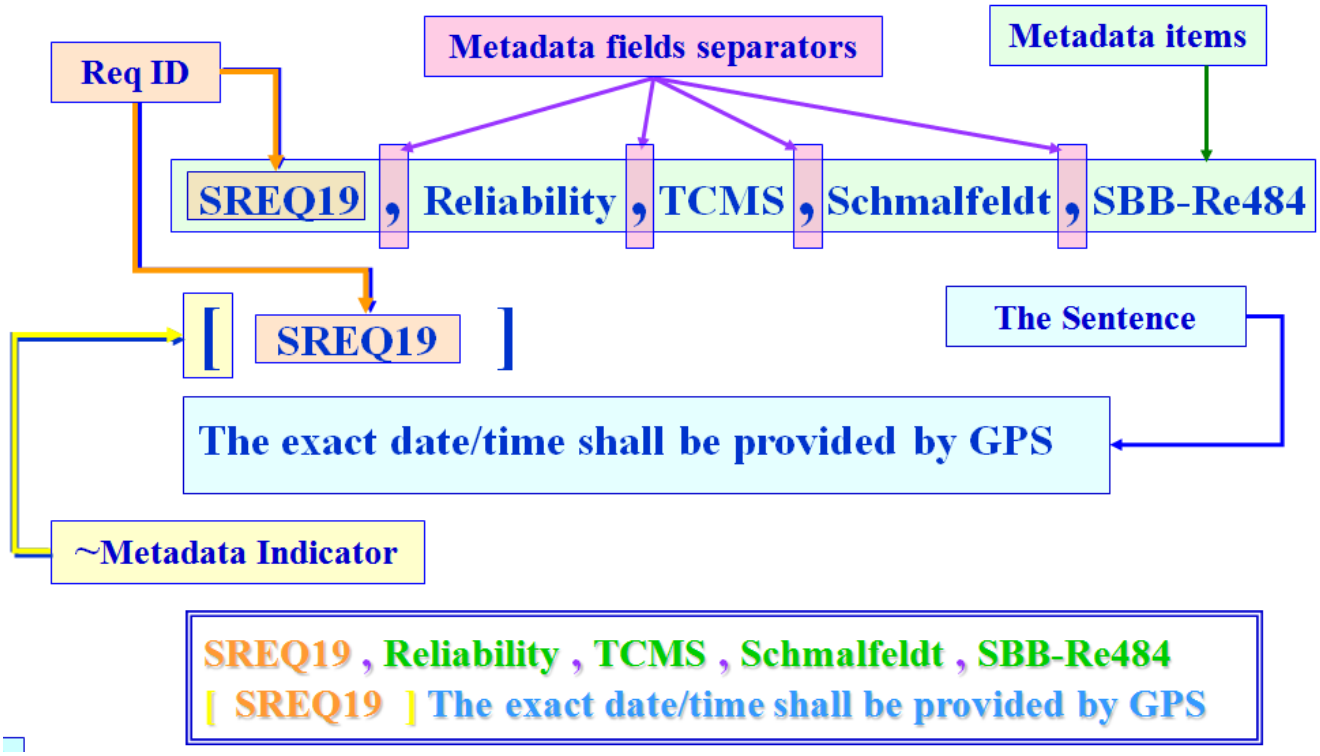

Figura 3 - QE input requisites format scheme and example

The new format is composed of (the sequence is mandatory):

1. a unique sentence **ID**
2. the first metadata field called **Type**: the kind of the requisite by means of the application field (examples: *Performance*, *Functional*, *Archit./Design Constraints*)
3. the second metadata field called **Package**: the set of requisites the sentence refers to (examples: *Detect bogie instability, Ensure passenger and crew safety, Integrating the vehicle into the complete system railway*)
4. the third metadata field called **Responsibility**: the author

of the requisite

5. the fourth metadata field called **Source**: standard documents (directives) the requisites refer to (examples: *Directive 96/48 HS,* HS-TSIrev draft 10NS_Board)
6. the requisite itself

In **Fig. 3** the new requisites structure scheme is shown

QuARS *Express* has been projected as a tiny simple application, with reduced functions, allowing to the user a quick first glance to its documents correctness by means of their wording defects.

Moreover its new feature to handle a more complex and structured data format containing metadata and to examine it extracting metrics and statistics opens the door to all the further improvements for both the old and new tool. In addition, it's highly modular implementation exploits the possibility of customizations suggesting a possible new line of activities.

Features comparison

A features comparison highlights that QuARS *Express* is not a definitive substitution of its major implementation QuARS. Simply their usage can be complementary or even with different aims. For example, QuARS can be seen more as a didactic and demonstration tool while QuARS *Express* is more effective facing the analysis, in a massive batch fashion, of large structured requirements documents.

Feature	QE	Q
Lexical Analysis	✔	✔
Syntactic Analysis	✔	✔
View Derivation Function		✔
Simple Readability Analysis	✔	✔
Complex Readability Analysis	✔	
Customizable set of Dictionaries		✔
Editable Dictionaries		✔
Metadata Management	✔	
Textual Report		✔
HTML Report	✔	

Feature	QE	Q
Simple Metrics	✔	✔
Complex Metrics	✔	
Requirements Editor		✔
Defective Requirements Highlight		✔
Requirements Traceability		✔
False-Positive Tracking/Hiding		✔
Drag & Drop facility	✔	✔
Help on line	✔	✔
Detailed status notification	✔	
Workspace preservation	✔	

Figura 4 - Q vs QE features comparison

In the following the differences listed in **Fig. 4** are described.

Lexical/Syntactic Analysis: both analysis are available with the same capability in both tools.

View Derivation Function: this kind of analysis is present in the main QuARS tool only.

Simple/Complex Readability Analysis: QuARS *Express* implements seven readability analysis (Kincaid [8, 9, 10], ARI [11], Coleman-Liau [12], Flesch [13, 14, 15, 16], FOG [17], LIX [18, 19, 20, 21], SMOG [22]) exploiting the tool Style/Diction [7]. QuARS performs the Coleman-Liau [12] readability analysis only by means of an its own internal module.

Customizable Set of Dictionaries (i.e. Set of Lexical Analysis): in QuARS it is possible the adding of custom dictionaries (Lexical analysis only) hence its analysis capability can be expanded. QuARS *Express* has the dictionaries embedded and does not perform an import mechanism for custom new domain analiysis.

Editable Dictionaries: Only QuARS GUI provides an editor for any dictionary and the belonging set of words can be customized.

Metadata Management: this feature is performed by QuARS *Express* only.

Textual/HTML Report: QuARS *Express* produces a richer report made up by several HTML pages with structured information by means of the used metadata fields. QuARS has a simple plain set of reports for any of the analysis function performed.

Simple/Complex Metrics: QuARS *Express* computes several statistics rates that lack in QuARS.

Requirements Editor: QuARS has embedded a complete text editor visualizing the loaded requiremnets and allowing to modify and store them.

Defective Requirements Highlight: after any performed analysis, QuARS has a mechanism of pointing out possible defective sentences highlighting them.

Requirement Traceability: QuARS has a mechanism of tracing sentences by means of the highlight function.

False-Positive Tracking Hiding: via the embedded text editor QuARS has a mechanism of sentences tracing.

Drag & Drop Facility: several areas of the tools GUIs provides drag-&-drop facility.

Help On Line: both tools have a specific help on line.

Detailed Computation Status Notification: QuARS *Express* reports meaningfully computation advancing while QuARS doesn't.

Workspace Lock/Preservation: QuARS *Express* checks wether there is some session aborted and preserve data from being overwritten.

References

[1] D. Berry, E. Kamsties, "*Ambiguity in requirements specification*", in Perspectives on Requirements Engineering, Kluwer 7-44, 2004

[2] Lami G., Gnesi S., Trentanni G., Fabbrini F., Fusani M. "*An Automatic Tool for the Analysis of Natural Language Requirements"* in Computer Systems Science and Engineering, vol. 20 (Vol.20, N. 1) pp. 53 - 62. CRL Publishing, 2005.

[3] QuARS (Quality Analyzer for Requirements Specification) http://www.quars.isti.cnr.it

[4] Bucchiarone A., Fantechi A., Gnesi S., Lami G., Trentanni G. "*QuARS Express, an automatic analyzer of natural language requirements" i*n ASE 2008 - Automated Software Engineering, 2008. ASE 2008. 23rd IEEE/ACM International Conference on Automated Software Engineering (L'Aquila, Italy, 15-19 September 2008). Abstract, pp. 473 - 474. IEEE/ACM, 2008.

[5] Bucchiarone A., Fantechi A., Gnesi S., Trentanni G. "*An experience in using a tool for evaluating a large set of natural language requirements*" in SAC '10 - 2010 ACM

Symposium on Applied Computing (Sierre, Svizzera, 22-26 marzo 2010). Proceedings, pp. 281 - 286. Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Michael Schumacher, Chih-Cheng Hung (eds.). ACM, 2010.

[6] MODTRAIN: Innovative Modular Vehicle Concepts for an Integrated European Railway System. See: http://www.modtrain.com/.

[7] http://www.gnu.org/software/diction/diction.html

[8] Kincaid, J.P., Fishburne, R.P, Rogers, R.L. & Chissom, B.S. "*Derivation of new readability formulas (automated readability index, Fog count and Flesch reading ease formula) for navy enlisted personnel*" in Research Branch Report 8-75. Naval Air Station, Memphis, TN, 1975

[9] Farr, J. N., Jenkins, J. J., and Paterson, D. G. "*Simplification of Flesch Reading Ease Formula*" in Journal of Applied Psychology, Volume 35, Number 5, (October), pp. 333-337, 1951

[10] Kincaid, J. P. & McDaniel, W.C. "*An inexpensive automated way of calculating Flesch Reading Ease scores*" in Patient Disclosure Document 031350, US Patent Office,Washington, DC, 1974

[11] Smith, E. A. and R. J. Senter "*Automated readability index*" in AMRL-TR, 66-22 - Wright-Patterson AFB, OH: Aerospace Medical Division, 1967

[12] Coleman, M. and Liau, T.L. "*A computer readability formula designed for machine scoring*" in Journal of Applied Psychology, 60, No.2, 283-284, 1975

[13] Flesch R. "*A new readability yardstick*" in Journal of Applied Psychology, Vol. 32, pp. 221-233, 1948

[14] Flesch R. "*How To Test Readability*" in New York, Harper and Brothers, 1951

[15] Flesch R. "*The Art of Readable Writing*" in New York, Harper and Row., 1974

[16] Flesch, R. "*How to Write Plain English*", HarperCollins - 1st ed edition, August 1979

[17] Gunning, R. "*The Technique of Clear Writing*", McGraw-Hill New York, 1952

[18] H. Björnsson and Birgit Hård af Segerstad "*Lix på franska ochtio andra språk*", Stockholm,Pedagogiskt centrum, Stockholms skolförvaltning, 1979

[19] Anderson, J. "*The Challenges of the Multicultural Classroom*", Paper presented at the United Kingdom Reading Association Conference, Edinburgh, July 1981 (ED 207 023)

[20] Anderson, J. "*Analysing the Readability of English and Non-English Texts in the Classroom with Lix*", Paper presented at the Australian Reading Association Conference, Darwin, August 1981 (ED 207 022).

[21] Anderson J. "*A little-known readability index that is quick to use, reliable and easy to interpret*" in Journal of Reading, 1983

[22] McLaughlin, H "*SMOG grading - a new readability formula*" in Journal of Reading issue 22, pp639-646, 1969