Quality Analysis of NL Requirements: An Industrial Case Study

A. Bucchiarone*and S. Gnesi
Istituto di Scienza e Tecnologie dell'Informazione
"A. Faedo" (ISTI - CNR)
Area della Ricerca CNR di Pisa, 56100 Pisa, Italy
{antonio.bucchiarone, stefania.gnesi}@isti.cnr.it

P. Pierini
Siemens C.N.X. S.p.A., R. & D.
Strada Statale 17, L'Aquila, Italy
Pierluigi.Pierini@siemens.com

Abstract

Nowadays common practice indicates that the Requirement Engineering (RE) process critically influences the success of the system development life cycle. Several commercial tools allow to classify, archive and manage requirements and then to print out reports and requirement documents. QuARS (Quality Analyzer for Requirements Specifications) is an automatic analyzer of such requirement documents, developed by ISTI - CNR, that can be adopted to evaluate the document quality by linguistic point of view. In this paper is presented how a requirement management tool, an automatic document generator and QuARS can be integrated to define an RE automation support. The case study investigates and highlights the efficacy and the role of such proposed support in the Siemens C.N.X. development process.

1. Introduction

Requirements engineering (RE) is the practice used to first identify and then translate stakeholder needs to system requirements. RE is the initial and the most phase of system development life cycle. RE outcomes provide inputs not only to almost all other phases of the development, but also provide information in the other phases, such as project planning, risk planning , quality planning, release planning, and configuration management planning [2]. The RE quality critically influences the success of the system development. Poor quality RE generates requirements that are incomplete, inconsistent, conflicting and unduly complex. Such requirements can cause significant defects that can be difficult to detect during system testing and are often more expensive to fix than implementation defects [2] . It is therefore important to have a good quality RE processes. System development is dependent on requirements being clear and concise and requirements definition has always been a critical phase. Recently, many commercial tools have become available to support requirements definition [3, 5, 6]. The use of these tools not only provides support in the definition and tracing of requirements, but it also opens the door to effective use of tools for a Quality evaluation. QuARS (Quality Analyzer for Requirements Specifications) is an example of these, it is able to perform an analysis of Natural Language (NL) requirements in a systematic and automatic way. QuARS allows the requirements engineers to perform an early analysis of the requirements for automatically detecting potential linguistic defects [7, 8]. This paper presents a process for the production and analysis of system requirement documents that is based on the integration of commercial tools for requirements management and of the tool QuARS. We have applied then the defined process to an industrial case studies. The case study reports the Siemens C.N.X experience using the process to produce the "System Requirement Document" (SRD), i.e. the document, defined in a project document set, expressing, among other things, the system functionalities and requirements. The relevant results of the experience are then discussed. After a preliminary introduction of requirements management commercial tools, the QuARS tool is described in section 2. The section 3 presents the proposed Quality Analysis Process, the section 4 presents an application of the process to a Siemens telecommunication systems and finally in section 5 some conclusions and future works are discussed.

2 Requirements Development and Analysis

Keeping track of requirements is vital to systems development. Clear and structured requirements management is strategic. As business processes and system requirements increase in complexity, human error in requirements management can cause recurrent and costly problems in systems development. Many tools offer management options designed to keep requirements clear, systems development

[*] IMT (Institutions Markets Technologies) Graduate School, Via San Micheletto, 3 - 55100 Lucca

Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering (RE'05)
0-7695-2425-7/05 $20.00 © 2005 IEEE

on track and project management simple. There are many commercial tools created for these purposes. We propose here to integrate them with an automatic quality evaluation tool from NL documents.

2.1 Requirements Management Commercial Tools

The ***IBM Rational RequisitePro*** [3] is a requirements management tool designed for multi-user environment. System designer teams can gather, enter and manage requirements extracted directly from a specification document or manually input in the repository. Automated traceability track requirements and changes through implementation and testing. RequisitePro includes SoDA templates to simplify production of requirements documents. ***SoDA*** (Software Documentation Automation) [4] can automatically create NL requirements documents from RequisitePro repository; it is a report generation tool that supports day-to-day reporting as well as formal documentation requirements with an easy-to-use interface for defining custom reports and documents.

Telelogic DOORS Enterprise Requirements Suite (***DOORS/ERS***) [5] is an Information Management and Traceability (IMT) tool. Requirements handled within DOORS are discrete objects. Each requirements can be tagged with an unlimited number of attributes allowing easy selection of subsets of requirements for special purposes. DOORS includes an on-line change proposal and review system that lets users submit proposed changes to requirements, including a justification. DOORS offers unlimited links between all objects in a project for full multi-level traceability. The tool that allows DOORS to generate documentation and NL requirements documents is ***DOORSRequireIT*** [5]. It is extremely easy to use and requires a very short learning curve. No database administration is necessary and it can be used by stand-alone users without the need for administrative support. DOORSrequireIT also allows the easy distribution of data to remote, disconnected users encouraging communication from anywhere, anytime.

At the end ***AnalystPro***[6] is a tool for requirements, tracing and analysis. It uses a requirements management methodology that covers the entire life cycle including, from the initial requirements-gathering phase through the separation phase where requirements and non-requirements are set apart. AnalystPro allows users to import requirements from existing documents from various formats (doc, html and text), it allows users to share and trace requirements across project and automatically records and lists any changes to your project, when the changes were made and who made the changes. AnalystPro's documents generation capability makes requirements documents generation a snap. With a few clicks, you can generate an entire document from requirements in the database.

2.2 Quality Evaluation Tools

Several studies dealing with the evaluation and the achievement of quality in NL requirement documents can be found in the literature. In order to automatize the evaluation process some tools have been proposed [11, 12, 8, 1]. We describe here the **QuARS** (Quality Analyzer for Requirements Specifications) tool. QuARS has been developed by "ISTI - CNR" (see QuARS Interface in figure 2) [7, 8] and it has been recently used for the evaluation of NL requirements document in real world projects.

It performs an initial parsing of the requirements for automatic detection of potential linguistic defects that can determine ambiguity problems impacting the following development stages. The functionalities provided by QuARS are:

1. Defect identification: QuARS performs a linguistic analysis of a requirement document in plain text format and points out the sentences that are defective according to the expressiveness quality model described in [7, 8]. The defect identification process is split in two parts: (i) the "lexical analysis" capturing *optionality, subjectivity, vagueness, and weakness* defects; and (ii) the "syntactical analysis" capturing *implicitly, multiplicity and under-specification* defects. In table 1 we can see some examples of requirements that contain linguistic defects.

Table 1. Example of Requirements sentences containing defects

Indicators	Negative Examples
Optionality	the system shall be.., ***possibly*** without..
Subjectivity	***..in the largest extent as possible***..
Vagueness	the C code shall be ***clearly*** commented..
Weakness	the initialization checks ***may be*** reported..
Implicity	the ***above*** requirements shall be verified..
Multiplicity	the mean time..and ***restore service***..
Under-specification	..be able to run also in case of ***attack.***

2. Requirements clustering: The capability to handle collections of requirements, i.e the capability to highlight clusters of requirements holding specific properties, can facilitate the work of the requirements engineers.

3. Metrics derivation: QuARS calculates metrics (The Coleman-Liau Formula and the defect rate) during the analysis of a requirements document.

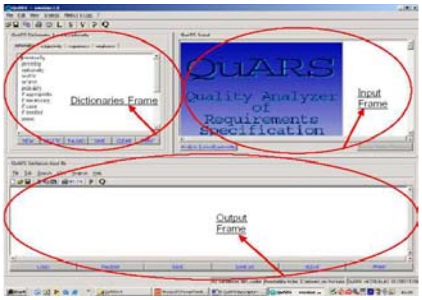

Figure 1. QuARS Graphical User Interface

3 Quality Analysis Process

In this section we present the Quality Analysis Process for generating the requirements document and his quality evaluation (see figure 2). The process is composed of the following steps:

1. Writing of requirements with commercial tools e.g. RequisitePro, Doors, AnalystPro, etc.
2. Generation of the Natural Language (NL) Requirements Document in automatic way (Doc File or Text File, for example with commercial tools such as SoDA, DOORSRequireIT, etc.).
3. Quality analysis of Requirements Document with tools for quality analysis of NL requirements document e.g. QuARS, ARM etc.

The requirements engineer creates a new file project (in RequisitePro, Doors, AnalystPro, etc.) and inserts the requirements with the name and description; at this point, in automatic way, a tool (SoDA, DOORSRequireIT, etc.) generates a text document and saves it with ***txt*** format (alternative formats are *doc*, *html* and *xml*). The txt file is input to QuARS that analyzes the sentences (Requirements) and giving in output:

- The log files listing the indications of requirements containing defects. Two logs file are produced respectively for lexical and syntactical analysis.
- The calculation of metrics about the defect rates of the analyzed document.

In the case QuARS points to some defect lowering the quality of the requirements document, a refinement activity is needed, followed by another quality analysis step. Otherwise, the development process can start on the basis of the approved requirement document.

On the basis of the described process, the developer is required to provide a NL system requirement description while all the other steps are automatically supported by tools. In the next section we will present the application of the Quality Analysis Process to a System under design in Siemens CNX Telecommunication.

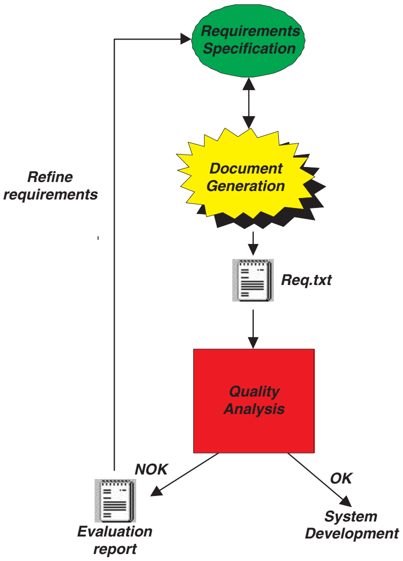

Figure 2. Quality Analysis Process

4. An Application of the process to a Siemens Telecommunication System

Siemens CNX is a R&D labs specialized in design and development of telecommunication equipments, mainly based on Synchronous Digital Hierarchy (SDH) technology.

Siemens CNX uses industrial design and production processes, which comply with consolidated factory standards, in order to guarantee good quality levels in resulting products. Nevertheless, the emerging technologies are continuously under study. Researches, cooperating with Universities, are in progress with the aim to enhance the efficiency of the whole production process optimizing costs, efforts, design consistency, etc...

Talking about SDH products, one of the main issues to be highlighted in advance is that they strictly obey to international standards (mainly ITU-T [10] and ETSI [9] for telecommunication aspects, in addition to many others like IEEE, IEC, ... etc.). Standards collect and describe the functions a SDH equipment must conform to. Customers (i.e. telecom companies) concern with standards compliance to guarantee several needs, like the ones related on their internal organization and procedures, multi-vendor equipments, network interworking and so on. As a consequence, many functional aspects do not require deep detailed description but a reference to the relevant standards.

4.1 System Requirement Document and Supporting Tools in Siemens C.N.X.

Large experience has been consolidated by several years in using the IBM-Rational toolset. In detail: RequisitePro has been used to archive, classify and structure requirements. It allows to better manage the requirement traceability and to highlight relationships between requirements. RoseRT, integrated with RequisitePro, supports the use case analysis and the subsequent design steps down to the implementation phases. SoDA, applied on RequisitePro project, allows to easily generate SRD formatted documents. The SRD results in a big but well structured document, written in natural language, describing and covering all the relevant functions a product should implement. Each requirement is uniquely identified and it is completed by additional notes, external references and back trace to relevant feature list items (i.e. classified stakeholder needs). An example of a resulting requirement is given below:

SR686: OS1 LOS detection The optical STM-1 Loss Of Signal defect (dLOS) is detected when the received signal has degenerated to a level where SDH frame alignment would be interrupted, and the cause is evidently a drop of incoming power level below operational level.
NOTES: Suitable values for threshold, of drop of incoming power detection, are between 2 and 3 dB below the receiver sensitivity.
REF: ETS 300 417 1 1 subclause 8.2.1.6 - ed.1/96
FEAT: SF503350

The SRD is the output of one of the first design steps and is the base for all the subsequent design choices. Thus, the SRD is a strategic document, completeness and correctness must characterize such a document. RequisitePro helps in maintain a clear view of the whole set of requirements and their internal relationships. This results in a less error prone designer activity, but, most of all, it allows to better control requirement changes during the project life cycle, supporting the identification of the affected requirements and the changes propagation across requirement relationships. But supporting tools are not enough and the so called ”Walk Through Review” (WTR) is the protocol adopted to guarantee the SRD quality. WTR, increasing the designers cooperation, remains the key point of the requirement analysis and validation steps in Siemens CNX approach. As soon a draft is ready, it is submitted for comments to a set of reviewers; a meeting is then scheduled to discuss the resulting comments and to define the modifications required to accept and officially release the SRD.

4.2 Using QuARS on SRD

QuARS has been experimented with the SRD related to the SXA project with the intent to enhance the WTR job. The main issues that QuARS should address are: *(i)* writing style uniformity across all the SRD parts (it has to be noted that the authors are mainly german and italian speaking with an average good but different skill in writing and structuring english sentences) *(ii)* rising the reviewers by linguistics checks and avoiding any misinterpretation due to linguistic artifacts; *(iii)* additional quality indicators availability useful to enhance the global product quality.

The following example shows a set of defects captured by QuARS on a draft SRD document; the bold words are the indicators by QuARS to point out the sentence as defective:

1. 4 ECCs [DCCm **or** DCCr **or** HCOC3 **or** F2] **can** be diverted to MCF (*weak and multiple sentence*).

2. The VLAN Concentrator **can** be used with all basic network topologies. (*weak sentence*).

3. The Hold-off Times are **useful** for inter-working of protection schemes (*vague sentence*).

4. A blocked port receives a more **useful** (path cost) BPDU than the one it would send out on its segment (*vague sentence*).

5. Terminal-to-terminal links are supported by SXA network elements, **optionally** with (1+1)-MSP for STM-1, STM-4 and STM-16 (only SURPASS hiT 7050 CC) interfaces (*optional sentence*).

6. **Depending on** The maximum capacity required to cross-connect (FP1and CC) the implementation of the matrix is realized with one or two **similar** ASICs respectively (*subjective and vague sentence*).

7. A blocked port receives a more **useful** (path cost) BPDU than the one it would send out on its segment (*vague sentence*).

QuARS lexical and syntactic analysis reports that the sentences are defective because they contain some wording as ”can”, ”useful”, ”optionally”, etc.. that should degrade or affect the document consistency. For example, the first sentence expresses an upper limit of the MCF function and should be rephrased as: ”*up to* 4 ECCs ... *must be* diverted to MCF”. The syntactical multiplicity analysis complains too due to square bracket’s item list, suggesting a preliminary definition of the abbreviations. It is curious to note that it seem a practice fussiness by several authors to list all the instances specializing a general concept every time it has to be referred in the text, also if it has been already introduced. This can create problems during the system life cycle, as in the true story: HCOC3 was added in a subsequent product release. Missing to update all of the ECC's lists resulted in an incomplete release delivery disclosed and fixed during the relevant system test phase.

The QuARS analysis joined with the RequisitePro/SoDA support aids in providing document less defective both in lexical and syntactic, sacrificing a little the general readability of the document. Considering lexical analysis, applied to the different sections of the SRD, the defect rate is closed to 0-1% but the weakness indicator varies in a range of 0-5%. Similarly, the syntactical analysis provides values, for implicitly and under-specification indicators, less than 4%; the multiplicity defect rate is worst with value between 6-28%. Readability indicator was in a range of 5-10.

Another important indicator is the count of the "modification request" (MR) arose on the documents during the project life cycle: only 15 MRs arose, 3 of them was of high priority (i.e. a wrong definition of requirements with impact to project), all the other was related to the definition of new product releases and customer stakeholder changes.

5. Conclusions and Future Work

In the industry practice, the analysis of system requirements is made by humans with a clerical and tedious process that consists of reading of requirements documents looking, among other things, also for linguistic defect. In this paper an automatic process, for requirements analysis and validation has been presented. The advantages in introducing support tools and automatic verification was highlighted by the case study. Future work and enhancements using QuARS in association with a requirement management tool, like RequisitePro, DOOR, etc., are related to a better integration between the two tools. It should be useful, for instance, to evaluate each single requirement as soon as it is added to the requirement archive. Further experience in this direction are in progress in Siemens CNX. The process presented in this paper is currently used in MODCONTROL FP6 project for the analysis of the developed Requirements specifications. In this project IBM RequisitePro is used for supporting the definition of the requirements. The automatic generation of the NL requirements document is performed by the IBM tool SoDA. Then QuARS takes in input this NL document and performs the analyze of the requirements (approximately 1000 among functional and non functional requirements).

6 Acknowledgments

The Authors would like to acknowledge the European project MODTRAIN (FP6-PLT-506652/TIP3-CT-2003-506652) subproject ModControl which partially supported this work.

References

[1] L. Mich and R. Garigliano. *"Ambiguity Measures in Requirement Engineering"*, International Conference on Software Theory and Practice. ICS 2000, Beijing, China, Aug.2000.

[2] I. Sommerville and G. Kotonya (1998), *"Requirements Engineering: Processes and Techniques"*, John Wiley and Sons, New York.

[3] IBM Rational RequisitePro,
See: http://www-306.ibm.com/software/awdtools/reqpro/.

[4] IBM Rational SoDA,
See: http://www-306.ibm.com/software/awdtools/soda/.

[5] Telelogic DOORS/ERS,
See: http://www.telelogic.com/products/.

[6] Goda Software: AnalystPro,
See: http://www.analysttool.com/.

[7] S.Gnesi, G.Lami, G. Trentanni, F. Fabbrini, M. Fusani: *An automatic tool for the analysis of application of Natural Language Requirements*, International Journal on Computer Systems Science and Engineering on Tools for Requirements Engineering Vol. 20, N. 1, pp 53-62 CRL Publishing 2005.

[8] F. Fabbrini, M. Fusani, S. Gnesi, G. Lami. *The Linguistic Approach to the Natural Language Requirements Quality: Benefits of the use of an Automatic Tool,* 26th Annual IEEE Computer Society - NASA Goddard Space Flight Center Software Engineering Workshop, IEEE November 2001.

[9] ETSI EN 300 417 - Generic requirements of transport functionality of equipment.

[10] Int. Telecommunication Union. ITU-T G.783 et other. Principal characteristics of multiplexing equipment for SDH

[11] Automated Requirement Measurement (ARM) Tool
See: http://satc.gsfc.nasa.gov/tools/arm/.

[12] Wilson WM, Rosemberg LH., Hyatt LE. *"Automated Analysys of Requirement Specifications"*, Proceedings of the 19th International Conference on Software Engineering (ICSE-97), Boston, MA, May 1997.