2013-Towards an Approach for Evaluating the Quality of Requirements

Data · September 2015

CITATIONS
0

READS
106

5 authors, including:

Faisal Mokammel
Aalto University
17 PUBLICATIONS **132** CITATIONS
SEE PROFILE

Eric Coatanéa
Tampere University
140 PUBLICATIONS **1,517** CITATIONS
SEE PROFILE

Francois Christophe
Tampere University
6 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Mohamed Bakhouya
Université Internationale de Rabat
61 PUBLICATIONS **807** CITATIONS
SEE PROFILE

Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2013
August 4-7, 2013, Portland, Oregon, USA

DETC2013-13708

TOWARDS AN APPROACH FOR EVALUATING THE QUALITY OF REQUIREMENTS

Faisal Mokammel, Eric Coatanea, Francois Christophe, Mohamed Ba Khouya, Galina Medyna
Department of Engineering Design and Production
Aalto University School of Engineering
Espoo, Finland

ABSTRACT

In engineering design, the needs of stakeholders are often captured and expressed in natural language (NL). While this facilitates such tasks as sharing information with non-specialists, there are several associated problems including ambiguity, incompleteness, understandability, and testability. Traditionally, these issues were managed through tedious procedures such as reading requirements documents and looking for errors, but new approaches are being developed to assist designers in collecting, analysing, and clarifying requirements. The quality of the end-product is strongly related to the clarity of requirements and, thus, requirements should be managed carefully. This paper proposes to combine diverse requirements quality measures found from literature. These metrics are coherently integrated in a single software tool. This paper also proposes a new metric for clustering requirements based on their similarity to increase the quality of requirement model. The proposed methodology is tested on a case study and results show that this tool provides designers with insight on the quality of individual requirements as well as with a holistic assessment of the entire set of requirements.

1. INTRODUCTION

Requirements management for complex systems and the understanding of potential emerging impacts of requirements on the final product are two important areas of focus for many companies and institutions. A lack of requirements for describing the needs for a technical system can have negative consequences during a development project and there is ample evidence that product developments have failed due to poorly understood requirements [1]. Unclear or incomplete set of requirements can be the cause of harmful design decisions leading to the development of unreliable and non-functional systems, unnecessary increases in costs, development time, and significant quality problems.

Linguistic techniques play an important role in quality grade analysis as requirements models are often entirely or for a major part expressed in natural language. Requirements Engineering and Natural Language Processing of requirements are fields that have emerged from Software Engineering. Nevertheless, methodologies addressing the same issue have also recently emerged in other disciplines such as engineering design [1, 2]. Most methodologies reviewed in this paper tend to consider only parts of the linguistic aspects and propose metrics of requirements quality regarding to these particular aspects. Therefore, there is a need to develop an integrative approach associating the different metrics from literature in order to improve the completeness of the coverage of the quality analysis of requirements.

This paper presents a combined methodology for evaluating the quality grade of requirements, combining lexical (i.e. domain specific indexed keywords in the requirements statement), syntactical (i.e. structure of the requirements statement), and semantic (i.e. meaning of the requirements statement) approaches. This methodology combines 18 quality grade metrics, assessing the expressiveness, consistency, and completeness of requirements models [2, 3, 4, 5, 6].

Along with existing methodologies, this paper proposes a new methodology for clustering automatically requirements based on the common issues they address. This is done by first filtering out non specific words (filtering out parts of speech, such as e.g. articles) and then measuring similarity [7] between requirements represented as lists of keywords. This clustering measure helps increasing the quality of requirements models by automatically placing requirements in proper categories. A case study considering different levels of requirements abstraction and classification (i.e. operational, capability, and system level requirements) is used to illustrate the proposed methodology.

2. RELATED WORK

The quality of requirements plays a major role in the development process of products and services [5] and the related domain has produced quite a large body of literature. As stated in [2], there are mainly three approaches for assessing requirements: inductive, restrictive or analytic approaches. Inductive techniques are not considered in this paper as their practical application is rather limited due to the fact that they

are mostly composed of recommendations in styles for writing requirements. The main objective of this paper is to combine restrictive [3] and analytic [2] approaches in order to obtain a broader viewpoint on the quality of requirements documents.

In relation to the present work, the work developed in QuARS project [2] on analytically measuring vagueness, subjectivity, optionality, readability, implicity, weakness, under-specification, multiplicity of requirements should be mentioned. In [2], indicators of the quality of requirements are defined as follows:

Ambiguity Indicators:

1. **Vagueness:** When parts of the requirements sentence are inherently vague. The use of vague words (e.g., easy, strong, good, bad, useful, significant, adequate, recent).
2. **Subjectivity:** When requirements sentences contain words used to express personal opinions of feelings, The use of subjective words (e.g., similar, similarly, having in mind, take into account, as [adjective] as possible).
3. **Optionality:** When the requirements sentence contains an optional part (i.e., a part that can or cannot considered) The use of words that convey an option (e.g., possibly, eventually, in case of, if possible, if appropriate, in needed).
4. **Implicity:** When the subject of object of a requirements sentence is generically expressed. The use of sentence subjects of complements expressed by demonstrative adjective (e.g., this, these, that, those) or pronouns (e.g, it, they). The use of terms that have the determiner expressed by a demonstrative adjective (e,g., this, these, that those), implicit adjective (e.g., previous, next, following, last), or preposition (e.g., above, below).
5. **Weakness:** The use of weak verbs (i.e. could, might, may) in the requirements sentence.

Specification Completion Indicators:

6. **Under Specification:** When a sentence contains a word identifying a class of objects without a modifier specifying and instance of this class. The use of words that need to be instantiated (i.e., flow [data flow, control flow], access [write access, remote access, authorized access, testing [functional testing, structural testing, unit testing]]).

Understability Indicators:

7. **Multiplicity:** When a sentence has more than one main verb or more than one subject. The use of multiple subjects, objects, or verbs, which suggests there are actually multiple requirements.
8. **Readability:** The readability of sentence is measured by the Coleman-Liau Formula of readability metrics (5.89*chars/wds-0.3*sentences/(100*wds)-15.8). The reference value of this formula for an easy-to-read technical documents is 10. If the value is greater than 15, the document is difficult to read.

Moreover, the work of Lamar [3] is also of interest as it adopts a restrictive approach based on patterns of sentences for classifying requirements into functional and non-functional categories. In [3], these patterns are defined as such:

1. **Completeness:** The linguistic structure for general requirements is as follows: <requirement>:: <subject>"modal" <verb phase>
2. **Functional Requirements:** The linguistic structure for functional requirements is as follows.

   Constraints Functions: The constraint function is linking a part of the system to develop with one and only one element of its environment.

   <Functional Requirement>::= <Subject> <Modal> <Intransitive verb>{<adjunct>}.
   Example: The airplane seat must float.

   Technical functions: The technical function is linking a part of the system to develop with two elements of its environment.

   <Functional Requirements>::= <Subject > <Modal> <Transitive verb> <Direct Object>{<adjunct>}

   <Functional Requirements>::= <Subject > <Modal> <Linking verb> <Participle Complement>{<adjunct>}

   Example: The seat must prevent injury.
3. **Non-Functional Requirements:** The linguistic structure of the sentence permits to detect non-functional requirements.
   <Non-Functional Requirements>::= <Subject> <Modal> <Linking verb> <Adjective Complement>{<adjunct>}
   Example: The seat must be easy to adjust.

Besides the different metrics found during state of the art research and integrated in the proposed approach, new quality grade metrics were also developed to cover lexical and semantic aspects, two aspects not covered by previous research works. These metrics are presented in the second part of the methodology section below.

Previously we developed in our research, a methodology for clarifying requirements [6] based on syntactic, lexical and semantic aspects. Commercial software, such as a special

support module developed for Rational DOORS [8], also propose metrics, which are not ingrate the state of the art. Instead of integrating the requirement quality metrics, details of requirement language processing aspect also cover in previous research work [9].

Existing requirements quality metrics found in literature are classified into different categories that take into account the viewpoints of the related analysis as shown Table 1. The QuARS project [2] proposes a point of view centred on understandability aspects of requirements grouped under the category of expressiveness. The present research work has taken the viewpoint of analysing the redundancies between the metrics proposed in the different research works from literature in order to integrate them coherently in a prototype software tool.

Table 1 Overview of the metrics considered and implemented in the prototype software

Source of the metrics	**Matrices Developed in literature and in this work**	**Quality criteria of the language assessed**					**Element of the language assessed**		
		Expressiveness			Consistency	Completeness	Lexical	Syntactical	Semantic
		Unambiguity	Understandability	Specific Completion					
QuARS Lami, 2005]	Vagueness	X					X		
	Subjectivity	X					X		
	Optionality	X					X		
	Implicity	X						X	
	Weakness	X						X	
	Under-specification			X				X	
	Multiplicity		X					X	
	Readability		X				X		
[Lamar, 2009]	Completeness					X	X	X	
[Christiophe et al., 2012]	REfinement	X		X	X				x
	ROM	X	X					X	
	ROM Questions	X	X			X	X		X
Other existing metrics (Commercial software)	Acronyms		X				X		
	Abuse of connector							X	
	Size		X				X		
	Domain specific ambiguous Stop verb list	X	X				X		
	Speculative sentence	X					X		
	Volatility						X		
Made By Graph (Proposed in this paper)	Similarity						X		
	Domain specific ambiguous Stop verb list						**X**		

3. METHODOLOGY

In the literature [5] different levels of analysis for the quality evaluation of requirements are considered, namely syntactic, lexical, and semantic levels. Based on this viewpoint, it is possible to classify the quality grade metrics from literature according to the aspects of language they address, as represented in Table 1. Later on this table is used as a template for measuring the quality grade of requirements. Table 1 summarizes:

1- The different metrics (from literature) integrated in the software platform,
2- The research works describing these metrics (N.B. Made By Graph is the name of the project financing this work), The quality criteria (Expressiveness, Consistency,

Completeness) used to represent the understandability of the requirements by other stakeholders.

3- The elements of the part of language (Lexical, Syntactical, Semantic) assessed by the different metrics.

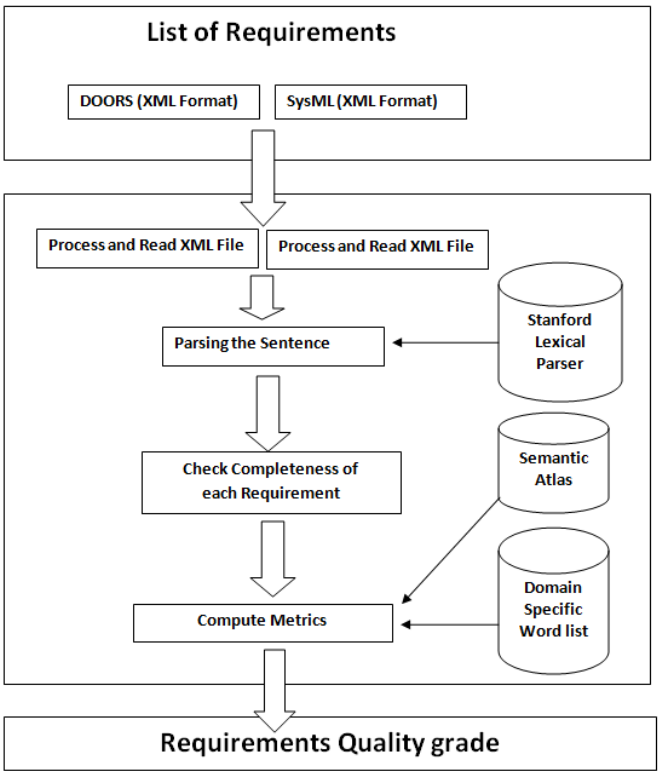

Figure 1 Architecture of the software tool developed

Description of software tools and integration of the quality metrics:

The proposed approach integrating different metrics of requirements quality has been implemented in a software tool as described in Figure 1. The developed software works as follows:

1. Process and Read XML File: The developed prototype reads SysML or DOORS requirement files provided as input. This provides flexibility of use of our developed software tools with existing requirement management tools.

2. Parsing Requirement Sentences: After processing and reading XML files (from DOORS or SysML modeler), the software prototype processes each sentence to find out the syntactic structures of these sentences (i.e. the grammatical structure of each sentence). In previous research work, minipar [10] is used in order to obtain the grammatical structure of a sentence (syntactic analysis). In this research work, Stanford Parser [11] is used as it provides an updated version of minipar. Although Parsers are used only for tagging Parts-of-Speech (POS) [12], in this approach, the Stanford Parser [11] is used to analyse the structure of the requirements sentence.

3. Check Completeness of each Requirement: The result of analysis from step 2 provides first insight on the completeness of each sentence as well as its level of compliance with a pattern of requirement from [3]. Each requirement is checked for completeness as described in the work of Lamar [3]. Only requirements passing this completeness test are forwarded to the next step of quality measures.

4. Compute Metrics: In this step, the software prototype calculates the metrics described in the literature (Table-1). During this stage, the prototype finds out about specific defects regarding to expressiveness of each requirement. If a defect is found, then this requirement sentence is tagged with the corresponding defect and advice is suggested to user on how to fix it. These metrics assessing expressiveness of a requirement are addressing only individual requirements regardless of their interactions with other requirements. The additional metric proposed in this paper (similarity metric) addresses the relations between requirements as it basically compares number of words in common (words, synonyms and contexonyms) between two requirements. Based on the work of Cheong and Shu [7], a domain specific keyword list (example Table 2) is used to rule out such keywords from comparison computation. For the computation of this similarity metric, any lexical database could be used, such as WordNet [13]. However, in this research work, the similarity metric is implemented with the semantic atlas from Ploux et al. as it associates a word with its set of synonyms but also with a set of contexonyms (i.e. words often associated together in the same sentence).

5. Requirements quality grade: Based on the quality defects of each requirement, the software prototype sums up the total number of the defects found in the whole requirements file. The global result is presented to the user describing the total number of defects and related quality issues.

Proposed Similarity Metric:

In addition to existing metrics from literature, a new metric is proposed in this work, as shown at the bottom of Table 1. The proposed metric is assessing the similarity between two requirements. This metric is adapted from the similarity function from [14]. In [14], the similarity function is basically computing the number of words in common between two documents divided by the total number of words in both documents. This similarity function is adapted as in Equation1 [6]:

$$sim(A_k, B) = \sum_{j=l} \frac{1}{1+e^{-l}} \cdot \frac{\left|a_k^l \cap b_i^j\right|}{\sqrt{a_k^l || b_i^j}} \quad (1)$$

In this equation, A_k and B represent two requirements, a_k^l is a set of compound words of size l (compound words of size l meaning l ordered words) with l varying between 1 and the total number of words in the requirement sentence A_k; b_i^j is a set of compound words of B with $j = 1, \dots, m$ and m being the number of words in the requirement B. $\left|a_k^l \cap b_i^j\right|$ provides the number of identical elements in a_k^l and in b_i^j.

The logistic function $\frac{1}{(1+e^{-l})}$ is used to foster requirements that include identical compounds to a higher level. For example, if requirement A contains "pressure regulator", then a requirement B containing "pressure regulator" will be given a higher value than an answer with two separate words "pressure" and "regulator". The last part of the metric under the square root is used for normalizing the results and allowing comparability between sets of requirements. Equation 1 also includes comparison between synonyms and contexonyms of each word used in requirements. The sets of synonyms and contexonyms of a word is found using the online semantic atlas (http://dico.isc.cnrs.fr/en/index.html) developed by Ploux et al. [15]. Additionally, before applying this similarity comparison between requirements pair-wise, it is useful to filter out irrelevant words from the requirements description. In [7], it is suggested that certain keywords can retrieve an overwhelming, and often irrelevant, number of search results. In order to limit the number of results, it is possible to filter the keywords that are useful in the domain considered; this is both useful for filtering the requirements sentence and the results found. In the context of this research, Table 2 lists the keywords that are considered as irrelevant.

Table 2 Keywords irrelevant to filter requirements descriptions

Use	Make	Minimum
Do	Help	Describe
Learn	Come	Simultaneously
Set	Need	Within
Deploy	Establish	Need
Rise	Maximum	Example
Try	Given	Propose

The results of this similarity comparison can be interpreted as follows in the case of high similarity score:

1. There is a relation between the two requirements.
2. Both requirements belong to the same category (which is defined by the user).
3. Both requirements share a common meaning.

Therefore, based on the similarity results, it is possible to detect relations between requirements unmentioned in the initial requirements document. Also, it is possible to detect if a requirement is initially classified under a wrong category. High similarity among two requirements also reflects a high quality of the requirements model as well as relevance between requirements. Whereas previous quality metrics addressed syntactic and lexical aspects of each requirement, this metric addresses the semantic links between requirements and, thus, the semantic quality of the requirements model. Furthermore, this similarity metric is used for clustering different levels of requirements classification (i.e. operational, capability, and system level requirements). Using this metric it is possible to categorize requirements on their specific category.

4. CASE STUDY

Table 3 lists the individual requirements linked to the case study.

Table 3 Case study

Requirement code	Category	Description
R0	Operational requirement	The air defence system shall be able to support joint operations with long-range capabilities
R1	Operational requirement	The air defence should prevent airspace violation.
R2	Capability requirement	The air base- 2 shall be able to engage X number of adversary fighters at the same time.
R3	Capability requirement	The air base 2 shall have air-lift capability
R4	System requirement	The five stations should be available for air Base-2.
R5	System requirement	20 F/A-18 Hornet should be operated
R6	Capability requirement	The air Base 3 shall have long-range (X km) air-to-ground capability.
R7	Capability requirement	The air Base- 3 shall provide airspace surveillance operation.
R8	System requirement	25 F/A-18 Hornet should be on the system.
R9	System requirement	Three F-15 fighters should be on the system.

It should be noted that this case study is part of an on-going collaboration with a large national agency that has an established existing requirements management approach based on three main categories of requirements (operational, capability, system). Operational requirements include the initial goal, which is what is fixed to be achieved in the beginning. Capability requirements cover such areas as networking, protection, logistics, and administrative support. System requirements cover such areas as personnel, materiel, organization, and information. Each requirement is described in English using a NL description. In this example, descriptions voluntarily include some quality defects in order to verify the ability of the prototype software to highlight such defect. The requirements listed have been anonymised prior to publication but are based on possible real-life scenarios.

5. RESULTS

The different integrated metrics (see Table 1) used for quality assessment implemented into the software tool give the results as shown in Figure 2. First, the results of quality analysis show no consistency defect between requirements. Second, the pie chart of Figure 2 shows that 3 requirements (30%) are complete syntactically (correct grammar) and lexically (no expressive defects), other requirements being defected by incompleteness (10%) or lack of expressiveness (60%). In addition to the overall pie chart visualization of defects in requirements model, the software prototype gives precision on defects for each requirement (upper right frame).

Following the proposed methodology, once the quality of requirements model is assessed and errors are suggested to the user of the prototype, the similarity metric is applied to also highlight potentially forgotten interactions between requirements. This similarity metric applied pair-wise to each requirement enables the clustering of requirements as shown in Figure 3. This clustering also highlights potential quality defects as for the category in which requirements belong to. The clustering is constructed by considering only the three requirements obtaining the highest scores in the similarity metric (based on numerical value) as shown in Table 4. This threshold is applied as it relates requirements with strong common meanings.

For example, Table 4 shows that for target requirement R1, high degree of similarity exists with requirement R0, due to the fact that both requirements belong to operational requirements category. Other requirements also show high similarity with requirements from the same category. As an exception, R0 and R6 do not belong to the same category but R6 obtains high similarity score with R0 as it derives from R0. Plotting requirements according to their affinity provides the results shown in Figure 3 representing the relations between requirements and the requirements at the interface between two categories. It should be noticed that this similarity metric can also be used to reproduce the requirement model as a network of requirements.

Table 4 Application of the similarity metric on the common case study.

Target Req.	Most similar	2nd most similar	3rd most similar	Comments
R0	R6	R1	R4	R6 derives from R0.
R1	R0	R4	R7	R1 and R0 Operational requirements.
R2	R3	R6	R7	All are capability level requirements.
R3	R6	R7	R2	All are capability level requirements
R4	R5	R8	R9	All are system level requirements
R5	R8	R9	R4	All are system level requirements
R6	R3	R7	R2	All are capability level requirements
R7	R3	R6	R2	All are capability level requirements
R8	R5	R9	R0	R8, R5, R9 are system level requirements, all three derive from R0.
R9	R8	R5	R4	All are system level requirements

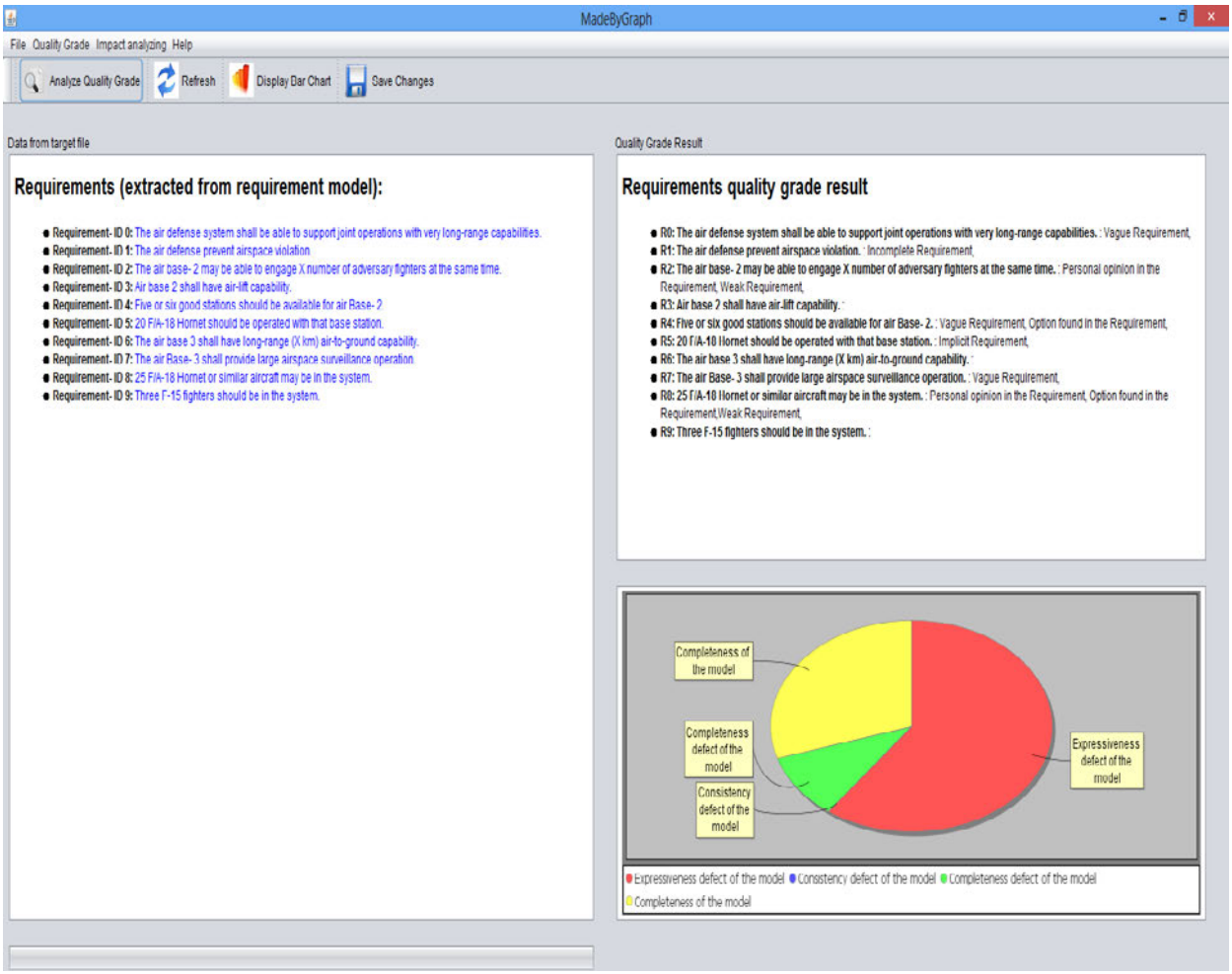

Figure 2 Snapshots of the software tool: quality grade

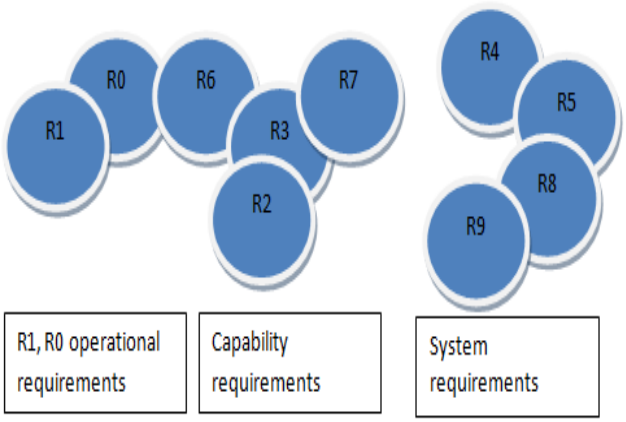

Figure 3 Result of proposed methodology

6. CONCLUSION AND FUTURE WORK

Contributions of this research work have been implemented in a prototype software tool and tested on several case studies; one of case study is presented in this research work. This work enhances existing requirements analysis approaches in several directions. The research work has integrated and developed new quality evaluation metrics both for individual requirements and list of requirements. Furthermore, the work has provided a concrete similarity metric allowing the classification of requirements into predefined categories, as well as the possibility to highlight potential links between requirements belonging to different categories. The approach proposed in this paper for clustering requirements suggests the possibility to treat requirements documents with other clustering techniques used in data mining (K-means, fuzzy C-means, Hierarchical, Mixture of Gaussians, etc.). However, the structure of data in requirements documents is specific and, requirements categories are usually predefined by users. In addition, requirements may belong to more than one category at the same time. Nevertheless, the strong connection

between the proposed approach and data mining techniques will be further investigated in future research work. The corpus is used for similarity metrics, is based on text extracted from general newspapers, making it a very general corpus that may lead to some errors if the approach is be applied to hundreds of cases. In order to improve the accuracy and avoid such errors, future work will try to develop, a corpus containing domain specific synonym and contexonym databases. Proposed metrics also can be useful for assessing the analysis of impact of changes within requirements [16, 17].

ACKNOWLEDGMENTS

The work presented in this document is the result of a collective work. The authors of the document would like to thank the different persons cited below for the constant involvement and support that they have provided during this research work. Special thanks to Lieutenant Colonel (GS) Jyri Kosola, Director of the FDF Technical Research Centre, Master of Science Aleksi Päiväläinen from FDF, Doctor Matias Aunola from FDF, Professor Pekka Appelqvist from Finnish Ministry of Defence. Special thanks also to Kati Vuorenvirta from Finnish Ministry of Defence for her practical helps in dealing with the administrative aspects of the project.

REFERENCES

[1] CHAOS chronicles Standish group “The Analysis of IT Project Success: Understanding Previous Performance to Create Future Prosperity,” URL: www.mgmiller.co.uk/ files/report.pdf

[2] G. Lami: Quars: A tool for analyzing requirement (cmu/ sei-2005-tr-014). Tech. rep., Software Engineering Institute, Carnegie Mellon University (2005). URL: http:// www.sei.cmu.edu/library/abstracts/reports/05tr014.cfm

[3] C. Lamar,.: Linguistic analysis of natural language engineering requirements. Master’s thesis, Clemson University (2009).

[4] C. Lamar and G. M. Mocko "Linguistic Analysis of Natural Language Engineering Requirement Statements," in TMCE 2010, Ancona, Italy, 2010.

[5] V. Berzins, C. Martell, Luqi and P. Adams, "Innovations in Natural Language document Processing for Requirements Engineering," Monterey Workshop 2007, pp. 125-146, extraction 2008.

[6] F. Christophe, F. Mokammel, T. Nguyen, E. Coatana, M. BaKhouya, A. Bernard: “A methodology sup-porting syntactic, lexical and semantic clarification of requirements in systems engineering”. International Journal of Product Development (2013). Accepted for publication.

[7] Cheong, H., Shu, L.: “Automatic of causally related functions from natural language text for biomimetic design.” In Proceedings of the Computers and Information in Engineering conference, IDETC/CIE 2012. Chicago, USA (2012).

[8] IBM Rational DOORS Managing Rational DOORS Release 9.2 URL: http://publib.boulder.ibm.com/infocenter /rsdp/v1r0m0/topic/com.ibm.help.download.doors.doc/pdf 92/managing_doors.pdf

[9] Lash A., Murray K., Mocko G., “Natural language processing applications in requirements engineering.” ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.

[10] Lin, D. (1998) Dependency-based Evaluation of MINIPAR. Workshop on the Evaluation of Parsing Systems, Granada, Spain

[11] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in 41st Meeting of the Association for Computational Linguistics, 2003.

[12] M. G. Georgiades, A. S. Andreou and C. S. Pattichis, "A requirements engineering methodology based on natural language syntax and semantics," in 13th IEEE Internation Conference on Requirements Engineering, 2005.(POS Tagging).

[13] G. A. Miller, "WordNet: A Lexical Database for English," Communications of the ACM, vol. 38, no. 11, pp. 39-41,1995.

[14] Johan Natt och Daga, Björn Regnell, Pär Carlshamre, Michael Andersson, and Joachim Karlsson. A feasibility study of automated natural language requirements analysis in market-driven development. Requirements Engineering, 7:20 – 33, 2002.

[15] S. Ploux, A. Boussidan, Hyungsuk Ji.: “The semantic atlas: an interactive model of lexical representation”. In Proceedings of the ELRA Conference 2010. Valletta, Malta (2010)

[16] B. Abma : “Evaluation of requirements management tools with support for traceability based change impact analysis.” Master’s thesis, University of Twente (2009).

[17] F. Mokammel, E. Coatanea, M. Bakhouya, S. Nonsiri “Impact Analysis of Graph-based Requirements Models using PageRank Algorithm” Submitted and accepted in IEEE International Systems Conference is Engineering of Complex Systems, 2013.