2013/840

ISSN 1797-3457 (verkkojulkaisu)
ISBN (PDF) 978-951-25-2530-0

TIIVISTELMÄRAPORTTI (SUMMARY REPORT)

Requirements model for Engineering, Procurement and interoperability: A graph and power laws vision of requirements engineering

Prof. Eric Coatanéa (eric.coatanea@aalto.fi), MSc Faisal Mokammel, Dr. Francois Christophe, Aalto University, Department of Engineering Design and Production

Tiivistelmä / Abstract

In engineering design, natural language (NL) is often used for expressing initial needs or requirements. One of the first tasks in Requirements Engineering consists in collecting documents that potentially contain important information about such needs or requirements. Each document can be quite large and contain very specific knowledge (e.g. ISO standard or legal book). Furthermore, the variety and large number of these documents can also reveal difficult to handle, and the human cognitive abilities to recognize connections between requirements and different quality issues of individual requirements have difficulties to handle such specific terms and knowledge in a wide range of disciplines (law, marketing, business, IT, Physics ...). Another connected issue is the size of the requirements models that make simply impossible to inspect manually the entire set of requirements. The support of a new form of computer tool is required. This support can take different forms but it is important to verify the readability and understandability of individual requirements. Problems such as ambiguity, incompleteness should be detected automatically. This was the result of the work done in 2012. A second important aspect is the automatic extraction of requirements from standards or text used in requirement models and to be able to detect connections between individual requirements presented in form of sentences.

The present project is also interested by the detection of contradictions that are existing between individual requirements. Last part tackle in the project during the year 2013 has been the development of an automatic tool for searching physical contradictions existing inside architecture of systems.

In summary the project has developed 3 proofs of concept tools, to extract requirements and their dependencies, to search for contradictions in requirements and to search for contradictions in systems.

All the approaches developed in this project during the year 2013 have been verified and tested. We have been able to demonstrate the feasibility of our research program. In addition, a process, methodology for integrating these 3 tools coherently is described in this report.

1. Johdanto / Introduction (Tutkimuksen aihepiirin kuvaus ja laajempi merkitys)

Establishing and following engineering, procurement and interoperability requirements belong to the class of large scale phenomena or systems. This type of activity is crucial for Finnish defence forces and has an extremely important impact on factors such as cost, technical performances, robustness and reliability of military equipment and networks. Indeed, multiple interactions exist between the different types of requirements that have to be treated by defence forces. In addition, the requirements are evolving dynamically all along the life cycle of complex systems. Being aware of the potential interactions that might affect the requirements is necessary in order to be prepared to face the main risks that can emerge due of these cross

impacting factors. This analysis is also necessary to plan redundancy and other types of necessary measures that can be used to developed robustness and reliability of the equipment.
Requirements are presented in the form of natural language. The natural language requirements models are transmitted to external stakeholders and processed by them in order to provide systems that should fulfil the different requirements. Defects in the requirements models can take the form of ambiguity, incompleteness, and misunderstanding of the difference between requirements categories. A useful tool for requirement analysis should be able to provide a digested list of ranked information allowing the user of the approach to understand the limitation and problems of the requirements models.
Those problems are associated with the quality of individual requirements in term of understandability or readability. Those aspects have been treated in 2012. In 2013, we have tried to tackle another set of quality issues. Our goal was in particular to extract automatically from existing standards and texts; the associated requirements and the dependencies between those requirements. Another objective was also to detect contradictions between requirements.
All those steps are useful in the requirement engineering process but the system engineering process is also involving the design of concrete systems that should fulfil those requirements. Another fundamental task that we wanted to prepare in 2013 was the traceability analysis in this system engineering process. Starting after the allocation of requirements to system components or sub-systems, the goal is to be able to analyse the impact of change made at requirement level or at system level.
In simple the final idea is to develop a software tool able to answer to the two following questions:
What is impacted in the system architecture and behaviour by the change of requirement A into requirement B?
What is impacted in requirement verification by change in the system from component A to component B or sub-system C to subsystem D?
In summary the research project objectives for 2013 were mainly associated with the support of the requirement engineering stage and more specifically with the extraction of requirements from text, extraction of requirements dependencies and search for contradictions between requirements. The last goal was consisting of searching for contradictions inside system architecture. This last goal is supposed to prepare the task for the next step of research which is the traceability analysis.

2. Tutkimuksen tavoite ja suunnitelma / Research objectives and accomplishment plan (Tutkimuskysymysten tarkempi erittely ja suunnitelma tutkimuksen toteutuksesta)

2.1 Research objectives

In 2013, the proposed research plan was consisting of processing the information presented in form of natural language text and standards to fulfil three distinct objectives associated with the requirement engineering process –extracting requirements from the texts and standards, searching for potential interactions and dependencies between requirements, searching for contradictions between requirements.
An additional objective was consisting of analysing system architecture to be able to extract contradictions existing in those systems.
Why searching for contradictions at requirement and system level?
There is a long tradition of dialectical reasoning in Western and Asiatic cultures. Its key feature is the principle of integration, starting with the recognition of contradiction, followed by the reconciliation of the opposing perspectives. The non-contradiction principle is forming its rational foundation. A good solution to contradiction should be non-contradictory. This principle

can be usefully taken into consideration in system engineering. This is in particular a powerful approach to optimize systems requirements and also early conceptual solutions for system. Last but not least this is a powerful creativity method to push the engineers and future users to develop innovative solutions.
Both the research tasks associated with requirement engineering and with system analysis are supposed to prepare for a bigger goal which is in the near future to analyse the impact of changes both at requirement and system levels. The objectives of the project 2013 should already lead to proof of concepts for each of the functionalities that have to be developed. The proofs of concepts are taking the form of software tools tested on case studies provided by Finnish Defence Forces.

2.2 Accomplishment plan

The objectives described in the paragraph above can be classified in two main categories- requirements processing and system architecture processing.
In order to be able to process requirements, different aspects need to be considered. First, there is different nature of the requirements types. The different requirements are usually classified in categories. There are also relations between those categories. The combination of the categories, classification and relations is forming an ontology. This ontology needs to be defined at least with a certain level of precision. In our case with the support of FDF we have developed a profile for SysML integrating the ontology used in FDF and derived from MODAF and DODAF. MODAF is the British Ministry of Defence Architecture Framework. DODAF is its equivalent in USA. This is an Architecture Framework which defines a standardised way of conducting Enterprise Architecture, originally developed by the UK Ministry of Defence. The purpose of MODAF was to provide rigour and structure to support the definition and integration of equipment capability, particularly in support of Network Enabled Capability.
Using the profile organizing the different types of requirements and the interactions, there is then the need to analyse requirements presented in form of sentences. The use of advanced Natural Language Processing (NLP) methods is required.
At system level, a system can be described using a set of variables and associated units. Those variables are interacting with each other's and graphs representations can be used to display those relations. In addition, it is possible to use other more advance Artificial Intelligence (AI) methods to find automatically contradictions inside the networks of interactions between variables.
The research project is organized around 4 tasks.
Task 1: Extracting automatically requirements from text and standards presented in text format or pdf format and analysing dependencies between requirements. A SysML requirement diagram should be generated automatically.
Input: A text and or standards
Output- Deliverable 1: List of requirements
Task 2: Searching for interaction between the requirements
Input: List of requirements
Output- Deliverable 2: SysML requirement diagrams
Task 3: Searching for contradictions between requirements
Input: List of requirements
Output- Deliverable 3: List of contradictory requirements
Task 4: Searching for contradictions inside system architecture
Input: Variables interaction graph (i.e. SysML block definition diagram or SysML parametric diagram)
Output - Deliverable 4: a representation of the contradiction nodes inside the system interaction variables diagram

The plan was to provide 3 software tool to automatize the analysis process. The first tool was integrating the outputs of task 1 and 2. The second tool was used to present the output of task 3. The last tool was used to represent the output of task 4. The intermediate SysML block diagram has been produced using an open source SysML modeller named TopCased.
The generation of a causal graph describing the interactions inside the system of variables used to represent a system architecture was in this project the result of the analysis made by a group of experts on our case studies (mainly an air bearing and a pressure regulator). We are currently developing a causal ordering algorithm to generate such type of causal graph in a semi-automated manner.

3. Aineisto ja menetelmät / Materials and methods (Tutkimuksen teoreettisen viitekehyksen kuvaus, käytetty materiaali ja menetelmät)

Extraction and modelling of requirements from documents

The tool developed for extracting requirements from NL documents is based on the linguistic analysis of documents' sentences. This analysis of NL documents is achieved at syntactic, lexical and semantic levels as advised in [7].
Initially, sentences are extracted from a document based on a basic syntactic rule: a sentence starts with a capital letter and ends with the character '.'. All sentences from a document are considered as a potential requirement. At the lexical level of analysis of each sentence, Stanford lexical parser [8] is used to find the grammatical structure of the sentence and assign a part of speech (e.g. noun, verb, modal auxiliary, adjective) to each word contained in the sentence, see Fig. 1-a.

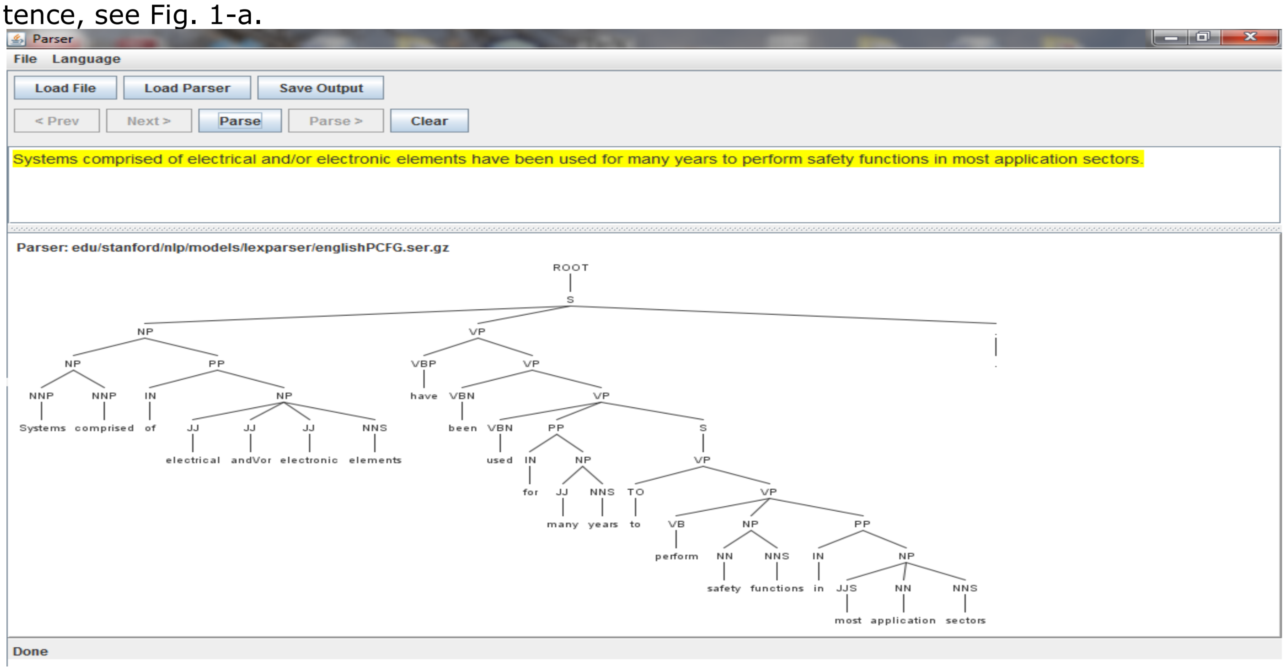

Figure 1-a - Example of representation of the grammatical structure of a sentence [8]
Based on this lexical analysis, the selection of requirement sentences from the list of document sentences is based on the hypothesis that a requirement sentence necessarily contains a modal auxiliary verb (e.g. shall, must, should, will).
In order to consider the type, i.e. the category, (e.g. operational, functional, capability) of each extracted requirement sentence, an analysis of words contained in a requirement sen-

tence is conducted at semantic level. This analysis requires the preliminary definition of a classification of requirements, see Fig. 1-b. In the case that a requirement sentence contains the name of a requirement type or a synonymous of this requirement type, then this requirement type is assigned to the corresponding requirement sentence. This approach is different from traditional clustering approaches as it allows a requirement to be of more than one type, i.e. to belong to more than one category. This is a very important aspect in Requirements Engineering in order to avoid communication mistakes when subsets of requirements are transmitted to engineering teams.

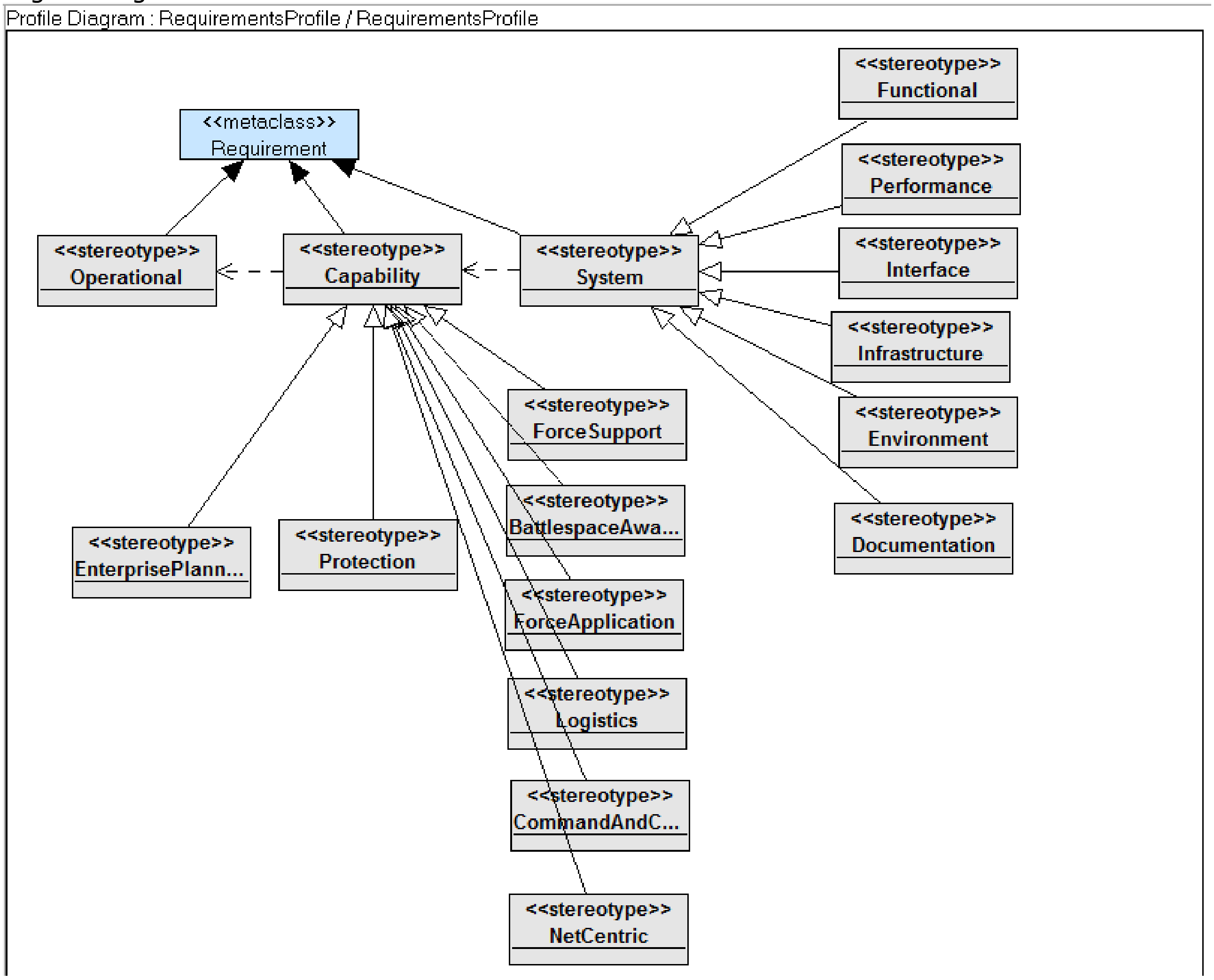

Figure 1-b – Example of classification of requirements as a SysML Profile

A semantic analysis is also conducted for establishing relations between requirements and creating a graph, i.e. a model, describing interactions between requirements. This analysis is achieved with the use of a normalized (i.e. [0,1]) metric of similarity between two documents [9]. This metric is based on the number of common words shared between two documents relatively to the size of each document as follows:

$$sim(A_k, B) = \sum_{j=\ell} \frac{1}{1+e^{-\ell}} \frac{|a_k^\ell \cap b_i^j|}{\sqrt{|a_k^\ell||b_i^j|}} \qquad (1)$$

For the entire set of requirement sentences extracted, such metric of similarity is computed between requirements pair wise. The matrix of similarity obtained from this computation is triangular superior due to the symmetry of similarity relations. From this matrix, interaction is selected from the highest similarity score in each row. This basic heuristic is proposed in this research because it ensures that each requirement will be linked through at least one interaction while avoiding the creation of interactions based on subjectivity such as establishing a threshold. In addition, such subjective selection criterion has the side effect of creating too many links between nodes of a graph, and thus generating circular references, over information or wrong information. This result of computation is shown through a simple example in Fig. 1-c. This example shows the process of selecting interactions between nodes of the graph based on similarity matrix. The graph obtained as result is a bi-directed graph.

	R0	R1	R2	R3	R4	R5
R0	x	0.29	0.27	0.06	0.30	0.26
R1	x	x	0.17	0.13	0.38	0.28
R2	x	x	x	0.31	0.24	0.25
R3	x	x	x	x	0.32	0.10
R4	x	x	x	x	x	0.20
R5	x	x	x	x	x	x

	R0	R1	R2	R3	R4	R5
R0	x	0.29	0.27	0.06	0.30	0.26
R1	0.29	x	0.17	0.13	0.38	0.28
R2	0.27	0.17	x	0.31	0.24	0.25
R3	0.06	0.13	0.31	x	0.32	0.10
R4	0.30	0.38	0.24	0.32	x	0.20
R5	0.26	0.28	0.25	0.10	0.20	x

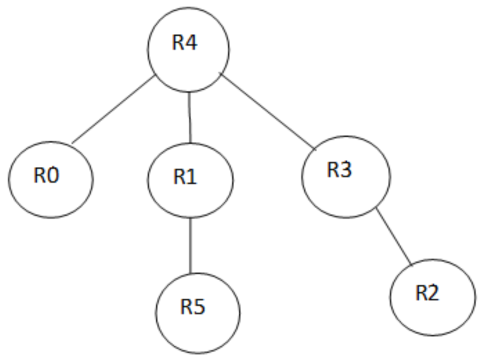

Figure 1-c – Process of selection for graph interactions based on similarity metric

However, for causality and traceability reasons, requirement graphs should be directed graphs with the direction of the interaction between nodes expressing a certain type of causal derivation. Therefore, this research suggests the use of a set of rules for transforming this graph into a directed graph. These rules are based on the topology of relations between requirement categories (see Fig. 1-b), and the type of possible relations establish by the language of representation: SysML [10]. For instance, if a capability requirement and a system requirement are in relation, this relation should be in the same direction and of the same type than in Figure #+1. Figure 1-d shows the results from applying this set of rules to this example of 6 requirements and obtaining a directed representation, i.e. SysML model, of requirements.

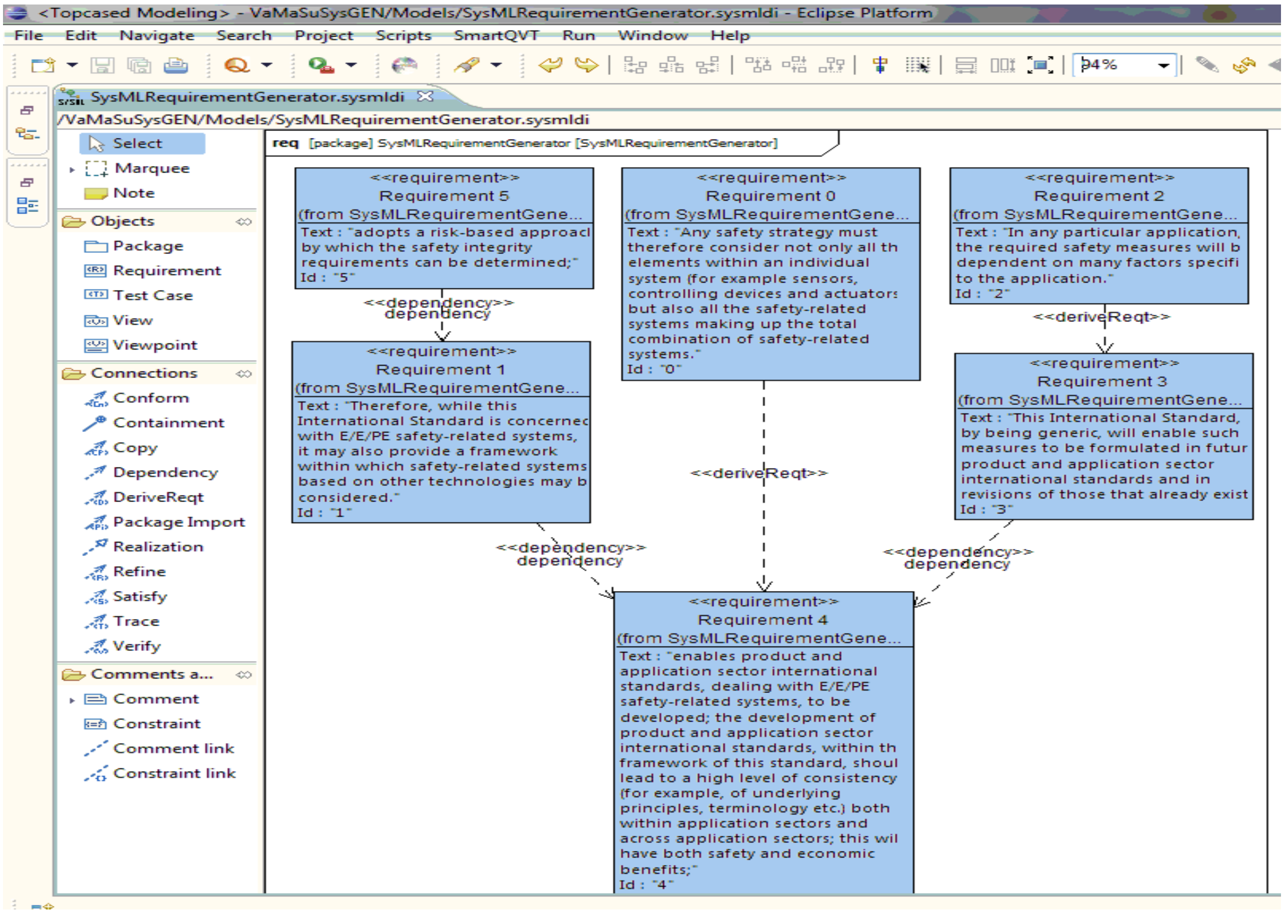

Figure 1-d – A generated requirement diagram using the approach

Contradiction search between requirements

Literally contradiction can be defined as “A combination of statements, ideas, or features which are opposed to one another”. Specifically for requirement domain contradiction can be define as “Requirement A and B are contradictory if there is no possible way in which A and B are both true simultaneously”. Finding conflicts in such type of statements impose to analyse the meaning of text. For a human it might be a rather easy task. If we want to achieve the same analysis with a computer it is a much more challenging objective. The project has developed a methodology to find some types of contradictions existing between requirements.

In literature different methodologies have been proposed to discover contradiction between statements expressed in natural language. The proposed methodologies vary in term of complexity, as well as for the required executable time [1]. In this research work we focus only on requirements contradictions, using an approach providing an acceptable executable time. In literature [1][2], contradictions are primarily categorised in two types, one is easy to detect (antonym, negation or number/date related) and other one is hard to detect (factive, previous knowledge related). Therefore, summing up together contradictions between different requirements can be categorised in 6 main forms of contradictions

1. Antonym / opposite meaning using word
   Example of antonym contradiction is following:
   Number of personnel should be increased in the factory

Number of personnel should be decrease in the factory.
Here contradiction occurs because of these two requirements refer to same system and impose different attributes.

2. Negation/ opposite meaning using negative word
   This type of contradiction related with negative word.

   Example:
   Temperature of the room should not be more than 25 degree Celsius
   Temperature of the room should be more than 25 degree Celsius.

3. Numeric / defining different numeric value for same object
   Here contradiction is not only because of referring same system, but also same unit with different numerical values.
   Example:
   Weight of handset should be less than 112 grams.
   Weight of handset should be less than 200 grams

4. Structure / Structurally impossible or contradictory
   Example:
   Internet submarine cable link should be built between Czech Republic and Finland.

   It is impossible because Czech Republic does not have any costal boarder with Finland

5. Lexical / Semantic of description made contradiction
   Example:
   All the components of the system should be manufacture locally.
   Battery and power supply should be imported from Germany.

   In case of meaning these two requirements are themselves contradictory.

6. Factive, World knowledge /Contradiction based on previous knowledge or history

   Example:
   Sand from Sahara desert should be used for constructing the building.

   This requirement refers to impossible fact because of sand of desert is not suitable for contraction work.

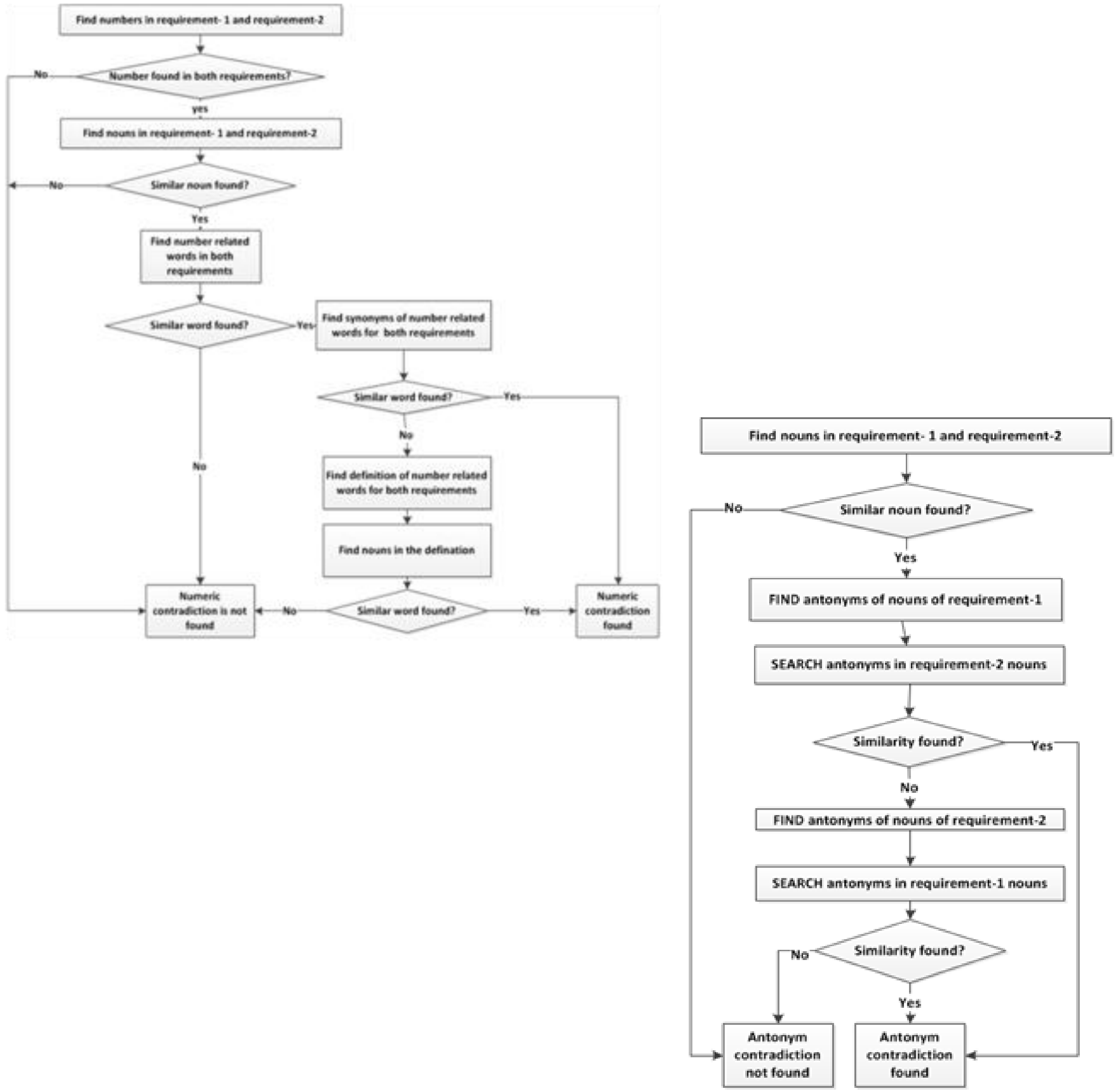

Figure 1-e: Flow charts of the algorithm used to discover numeric and antonym contradictions.

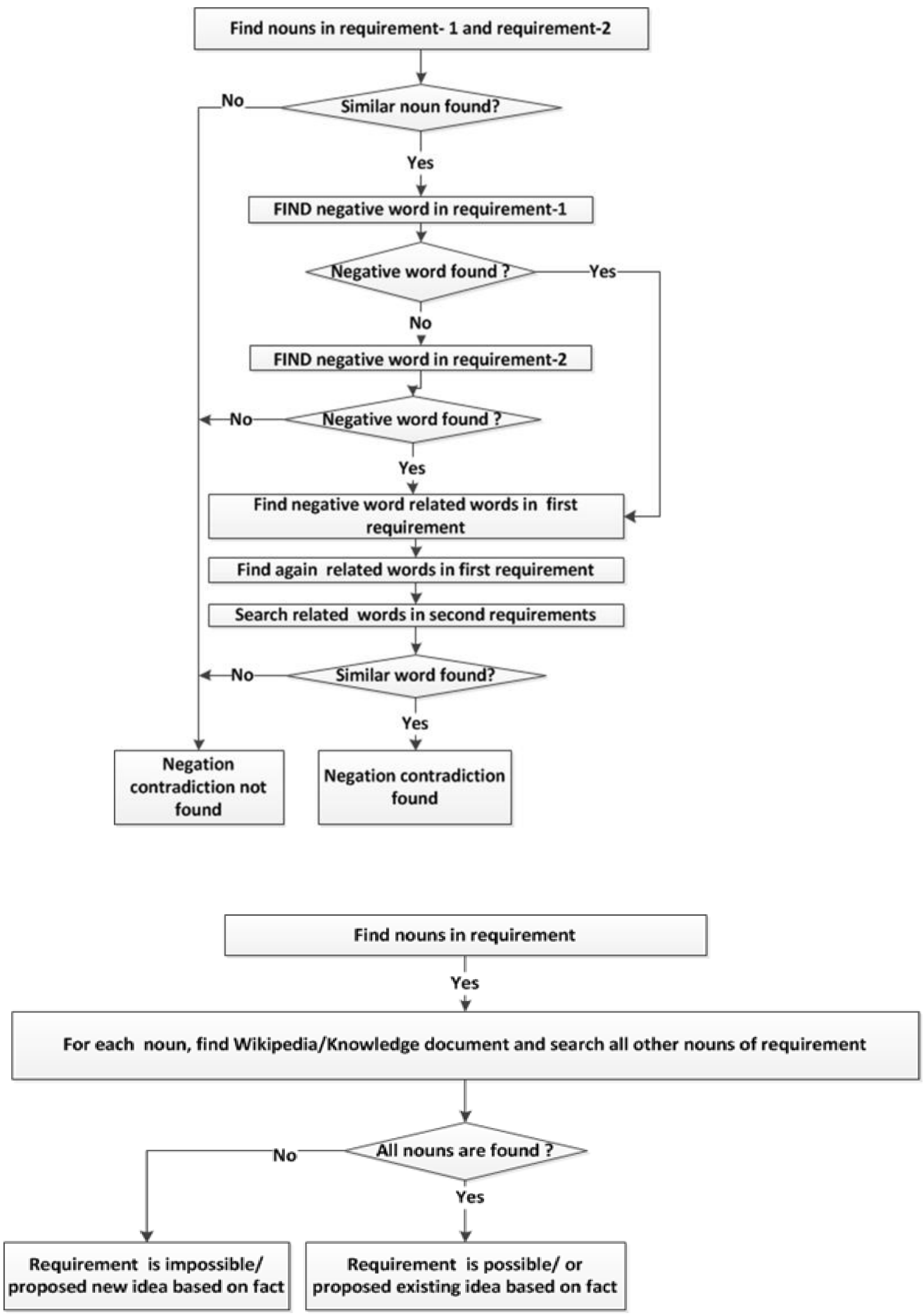

Figure 2: Flow charts of the algorithm used to discover negation contradictions and for the structural contradiction.

In this research work, we have been able to implement right now1, 2, and 3 of these 6 existing contradictions that can potentially exist between requirements. In the proposed methodologies first we process each requirement sentence individually to detect potential contradiction. We used Stanford parser for POS tagging and dependency analyser (find out the relation-

ship between requirements.) Wordnet (database of version 3.1) dictionary/thesaurus used for finding out synonyms and antonyms of the related noun. In Figure 1-e and Figure 2 flows chart of the proposed methodologies are depicted in details. The last flow chart of Figure2 shows factive/ knowledge based contradiction, where we use Wikipedia as a corpus, we measure the frequency of relevant word to detect the contradiction These proposed methodologies have been implemented in the computer tool.
Methodologies for the other types of contradictions (i.e. lexical, world knowledge and structure) will be developed next year. The development of this search tool is demanding and is requiring time.

Contradiction analysis at system architecture level

In the case of early systems design, being able to systematically extract the contradictions is a real challenge when systems are becoming more and more complex and when an overall understanding of the system behaviours and interactions is still difficult to define. Nevertheless, the potential benefits of systematically extracting contradictions in early systems designs are potentially very important. This would in particular provide a powerful approach to systematically analyse potential innovative design strategies. This will also support the development of approaches to validate early in the development process system architecture. The approach developed in this research is briefly described hereafter.
The approach developed is first taking into account the possibility to derive behavioural characteristics of a system using the set of elementary variables used to describe the system, the existing interactions between those variables (i.e. the expected interactions) and the units associated with the individual variables. The basic mechanism used to propagate changes in a network of interacting variables is based on the qualitative physics machinery developed by researchers such as Bashkar and Nigam [4].
In this research we have associated the qualitative physics machinery with the concept of graph. The method consists of analysing each of the nodes of the graph pointed out by arrows using the following approach.
Let $y=\sum_i a_i x_i$ be a law. Then all $a_i x_i$ must have the same dimensions as y. If a_i are dimensionless constants, then x_i must have the same dimension than y. This is the principle of dimensional homogeneity. If the system of fundamental quantities needed in the law is in the form of 3 basic quantities namely the length *L*, the mass *M* and the time *T* and if $[y]$ the dimension of the variables is a combination of the 3 basic dimensions then $[y]$ has the form:

Eq. 1 $$[y]=C_1.L^{\alpha_1}M^{\alpha_2}T^{\alpha_3}$$

This form is called the product theorem in which the constant C1 and the exponents α_1, α_2 and α_3 are dimensionless numbers.

When the dimensional validity of the graph has been verified, the next step consists of defining the objectives that are targeted on the performance variables. Traditionally three types of objectives are targeted. A performance variable should be maximized, minimized or a target value is expected.
Using the exact same basic principle described above, it is possible to infer the impact of those performance variables on the other categories of variables implied in the same graph.
The principle used to do this is the following.
It follows from the product theorem described in Equation 1, that every law which takes the form $y_0 = f(x_1, x_2, ..., x_n)$ can take the alternative form:

Eq. 2 $$\Pi_0 = f(\Pi_1, \Pi_2, ..., \Pi_n)$$

Π_i are dimensionless products. This alternative form is the final result of the dimensional analysis and is the consequence of the *Vashy-Buckingham theorem.* A dimensionless number is then a product which takes the following form:

Eq. 3 $$\Pi_i = y_i.(x_1^{\alpha_{i1}} x_2^{\alpha_{i2}} x_3^{\alpha_{i3}})$$

where $\{x_1, x_2, x_3\}$ are called the *influencing variables*, $\{y_1, y_2, y_3\}$are called the *influenced variables* in our work and $\{\alpha_{ij} | 1 \leq i \leq n-r, x_2, x_3\}$ are the exponents. The denomination influenced variables has replaced the denomination performance variables used initially by several authors such as Bashkar and Nigam or Butterfield [5]. This choice made by the authors of this article reflects better the graph considerations included in this work. The graph representation associated with Equation 3 is the following.

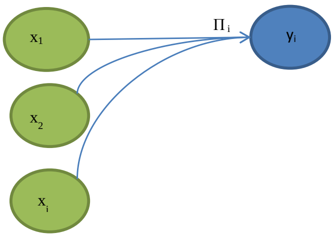

Figure 3: Representation of the Vashy-Buckingham theorem

The dimensional homogeneity should be valid for each node *y* of the generated graph otherwise this is the sign that a problem exists in the generated graph. It can mainly be the sign of two potential problems; a variable might have been forgotten or a causal link is missing or wrong.

This mathematic algorithm has been implemented in a prototype proof of concept presented hereafter. In a first step, the parametric diagram (see Figure 4 below) used to represent the interactions between the variables representing the behaviour of the air bearing example (see Figure 4 below) is downloaded in the proof of concept tool presented in Figure 5.

In Figure 5, we see that an automatic clustering phase is done by analysing the degree of the nodes. This approach is developed using approaches presented the book from Newman about network theory [6]. When the clustering is realized it is possible to define objectives on the performance nodes presented in red in Figure 5. Three types of objectives can be defined – maximizing a node, minimizing a node, getting a target value on a node-

The Figure 6 is presenting the propagation made in the graph after selection of the objectives on the red nodes. The yellow nodes represent the nodes where a contradiction has been found. In the case of the air bearing the Figure 6 shows that 3 contradictions have been found. The diameter *d* of the injector that needs to be small and big at the same time. The chamber diameter *Di* needs to be small and big and the gap *h* needs also to be small and big.

Defining the causes of the contradictions and finding ways to overcome them is an efficient source of innovation in system design. The present work needs now to be completed by a thorough analysis of the different design strategies. This will be the goal of a future research work.

The tool presented in this research is the outcome of the task 4 previously presented in section 2.2.

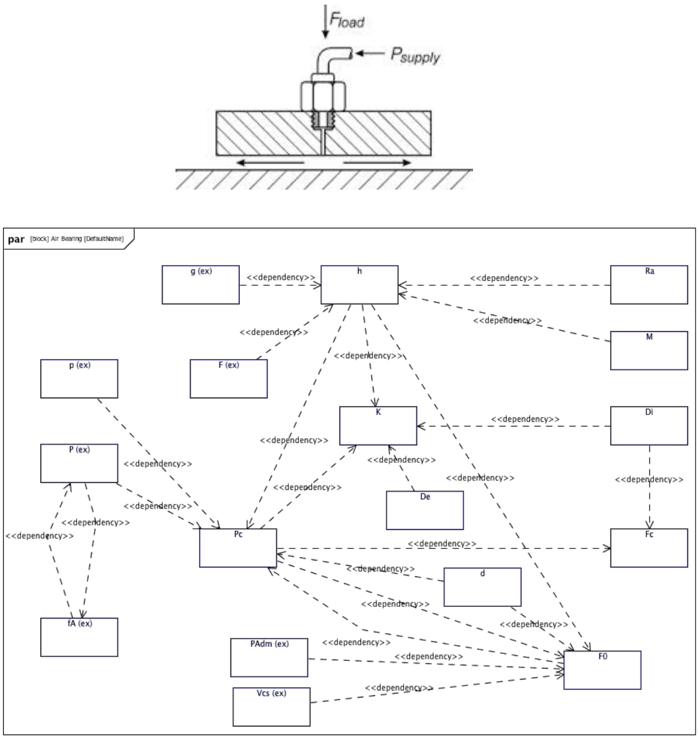

Figure 4: The example of an air bearing and the SysML parametric diagram used to represent the interactions between the variables.

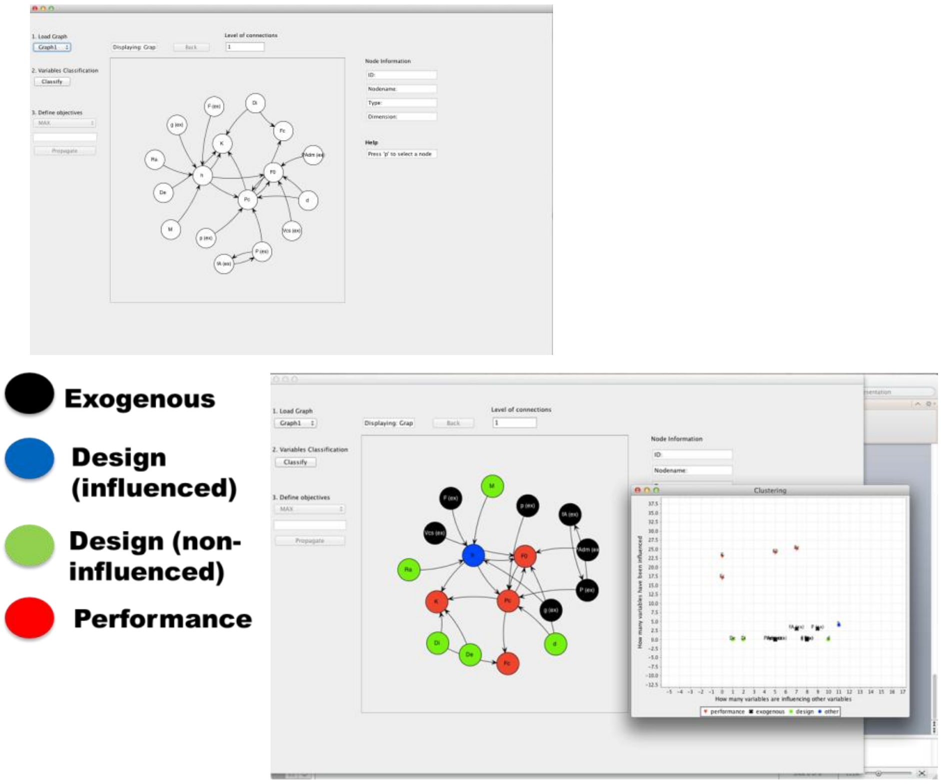

Figure 5: Graphical interface of the proof of concept tool (top: Graph before clustering, down: graph after clustering)

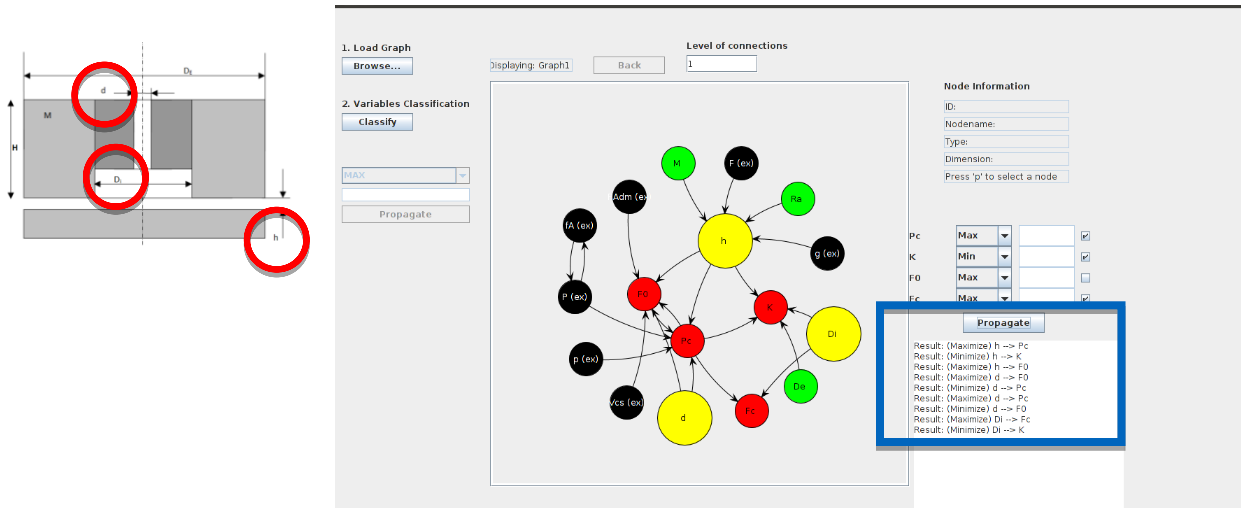

Figure 6: Graphical interface of the proof of concept tool (left: Location of the contradictions on the air bearing concept, right: graph representation of the contradictions after propagation of the objectives with rational for the contradictions)

4. Tulokset ja pohdinta / Results and discussion

(Saavutetut tulokset ja arvio tulosten oikeellisuudesta, kattavuudesta, merkityksestä, hyödynnettävyydestä)

The different tools presented above have been tested on different case studies. For the task 1 to 3 the main objectives have been to check the capability of the algorithms developed in this research to extract requirements, to find dependencies, to search for contradictions.

Another aspect that has been tested was the scalability of the algorithms developed in this research.

Tasks 1 and 2:

Capability of the tool: Extract requirements from text and generate requirements model in forrm of SysML requirements diagrams.

Actual limitations of the tool:

- Still semi-automatic process,
- The evaluation process is still going-on,
- Integration in a process required in the future.

The algorithms have been able to extract requirements and an analysis has been made by professionals to compare the automatic extraction with a manual one. Different scalability tests have been done using 2 test cases:

– 1 page: page 17 of ISO61508-3

– Full document: from p.17 to p.90

The results where the following:

– Case 1: 24 sentences, 6 requirements extracted and organised as SysML diagram

– Case 2: 848 sentences, 259 requirements extracted and 227 relations between them...

These results show that this tool enables a systematic (i.e. error-free) method for extracting requirement sentences from documents. The use of such tool by engineering teams assists them to visualize requirements and their interactions easier than in documents. This tools drastically helps reducing document reading time as it helps focusing on relevant parts of documents.

However, it was noticed that the graph produced is composed of several isolated independent graphs (composed of 2 or 3 nodes). Such graph represents redundancy within the set of requirements. In addition, at high scale (case 848 sentences), the visualiziation of the entire set of requirements as one hole network becomes unreadable for users. It is now thought of representing the requirements model at different levels of visualization in order to provide a better overall visibility on the model. Such limitations are the ground for our research project during the coming year.

Tasks 3:

Capability of the tool: Find contradictions within requirements

<table><tr><td>Actual limitations of the tool:
- Search for 3 types of contradictions out of the 6 that should be implemented at the end,
- The evaluation process is still going-on,
- Integration in a process required in the future.</td></tr></table>

The contradiction analysis in the case of requirements has been done in the following manner. The case study used to test the methodologies has been adopted from a literature [1] in requirement perspective which composed of different types of contradiction.

Case study for contradiction:

1. Number of combat personnel should be increased in coastal area.
2. An investigation into the strike in Qana found 28 confirmed dead thus far.
3. The Supreme Court decided that only judges can impose the death sentence.
4. Number of combat personnel should be decreased in coastal area.
5. A closely divided Supreme Court said that juries and not judges must impose a death sentence.
6. The tragedy of the explosion in Qana that killed more than 50 civilians has presented Israel with a dilemma.
7. The air defence system should able to support joint operations with long range capabilities.
8. Construction work of air base-3 should not start before 2015.
9. Weight of F-16 fighter should be less than 8,575 kg.
10. The air base-2 shall have airlift capability.
11. The air defence system should support operations in short range distance.
12. Empty weight of F-16 fighter should be less than 8,570 kg.
13. Construction work of air base-3 should start before 2015.

Result:

Contradiction type	Expected	Detected
Antonym	2	2
Negation	2	2
Numeric	2	2
Factive / Knowledge based	under development	

From the above result it shows that 3 type of contradictions are detected successfully. First Antonym contradiction was between 1, 4 and 7, 11 . Negation contradiction was between 3 5 and 8, 10. 2, 6 and 9, 12 shows the Numeric contradiction.

For the case study the expected result are similar except for the knowledge based contradictions where the implementation is still going on.

Tasks 4:

Capability of the tool: Find contradictions within system architecture

Actual limitations of the tool:

- Require the development of causal ordering graph implying the work of group of specialists,
- An automatic causal ordering algorithm and platform is under development,
- The evaluation process is still going-on,
- Integration in a process required in the future.

The evaluation process is still going on, on bigger case studies. This work is currently taking place in our group with a cooperation with VTT colleagues.

5. Loppupäätelmät / Conclusions
(Kriittinen tarkastelu tutkimuksen tuloksista ja aikaansaannoksista tavoitteisiin nähden, keskeiset havainnot riittävän yleistajuisesti kiteytettynä, näkymät ja suositukset tutkimuksen jatkosta)

The research results obtained during the year 2013 have been significant and for each of them a proof of concept have been realized and tested. The results obtained are extremely interesting and can form a completely new set of tools not yet existing at commercial level.
Several other developments will continue next year based on the cumulative progresses made during the last two years. The research has generated also numerous scientific insights that are currently under finalization in form of several journal articles. Those journal articles are promoting the scientific results obtained in this project and they are supporting the scientific development of the staff that has been financed by the project.

We would to thank FDF for the support provided during the two last years that have permitted us to develop this good piece of research. We expect that those results are bringing novel contributions to the research communities in Requirements Engineering and Systems Engineering. In addition, the software prototypes that have been developed and tested during the project are being developed further in this project.
We argue that the tools developed in this project can provide a very useful support to system and requirements engineers. They help them to expend the analysis capability and limit the risks for mistakes and poorly written requirements. They are also supporting the creativity and innovation aspects by highlighting problems that are not visible to engineers in requirement models and in system architectures.

6. Tutkimuksen tuottamat tieteelliset julkaisut ja muut mahdolliset raportit / Scientific publishing and other reports produced by the research project (Lyhyt kuvaus julkaisun keskeisestä sisällöstä sekä täydelliset bibliografiset tunnistetiedot (kirjoittajat; julkaisun nimi; sarjan, julkaisun tai journaalin nimi ja numero; julkaisija; paikka; vuosi))

This year the research has currently resulted in 4 scientific publications published to high level conferences and journals:
1. Christophe, F., Mokammel, F., Nguyen, T.A, Coatanéa, E., Bakhouya, M., Bernard, A.:"A Methodology supporting Syntactic, Lexical and Semantic Clarification of Requirements in Systems Engineering". Submit-ted and accepted in the International Journal of Product Development, 2013.
2. Mokammel, F., Coatanéa, E., Bakhouya, M., Nonsiri S.: "Impact Analysis of Graph-based Requirements Models using PageRank Algorithm." Submitted and accepted to the IEEE International Systems Conference on Complex Systems, 2013, Orlando, Florida, USA.
3. Mokammel, F., Coatanéa E., Christophe, F., Bakhouya M.: "Towards an Approach for Evaluating the Quality of Requirements." Submitted to ASME 2013 International Design Engineering Technical Conferences (IDETC), 2013. Portland, USA.
4.Model-based approach for change propagation analysis in requirements, S Nonsiri, E Coatanea, M Bakhouya, F Mokammel, Systems Conference (SysCon), 2013 IEEE International, 497-503.
We are currently preparing **6 more journal articles** based on this year results. We are expecting to submit them by December 2013 and January 2014. The journals in which the articles will be submitted are Requirements engineering, Systems journal, and CIRP journal.

Scientific references used in this report:

Marie-Catherine de Marneffe, Anna N. Rafferty, Christopher D. Manning: Finding Contradictions in Text. In: Proceedings of ACL-08: HLT, Association for Computational Linguistics, Ohio, USA, 2008.
Sanda Harabagiu, Andrew Hickl and Finley Lacatusu: Negation, Contrast and Contradiction in Text Processing.
Christiane Fellbaum. 1998. WordNet: an electronic lexi- cal database. MIT Press.
Bhashkar R., Nigam A., *Qualitative physics using dimensional analysis,* Artificial Intelligence, vol. 45, pp. 73-111, 1990.
Butterfield R., Dimensional analysis revisited, Proc Instn Mech Engrs Vol 215 Part C, ImechE, pp. 1365-1375, 2001.
M. E. J. Newman (2010). Networks: An Introduction. Oxford: Oxford University Press. ISBN 0-19-920665-1.
Lami G., "QuARS: A Tool for Analyzing Requirement," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2005-TR-014, 2005.

Klein D., Manning C. D., Accurate Unlexicalized Parsing, Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430, 2003.
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.
SysML specifications, http://www.sysml.org/sysml-specifications/, Object Management Group, OMG SysML v. 1.3, June 2012.

7. Hankkeen seuraajan lausunto raportista

(Tutkimuksen hyödyntäjän tai seuranneen tahon esim. jaoston tai puolustushaaran lausunto projektin onnistumisesta ja tulosten hyödynnettävyydestä. Lausunnon pyytämisestä vastaa hankkeen johtaja. Vapaaehtoisesti täytettävä kenttä.)