

Analysis of Multi-Stakeholder Requirements Using Requirement Interaction Matrix

Rohayanti Hassan[#], Shahreen Kasim^{*}, Ng Kang Ning[#], Rohaizan Ramlan⁺, Mohd Aizi Salamat^{*},
Rd Rohmat Saedudin[^]

[#]*Department of Software Engineering, Faculty of Computing, Universiti Teknologi Malaysia, Malaysia*
E-mail: rohayanti@utm.my

^{*}*Soft Computing and Data Mining Centre, Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn, Johor, Malaysia*

⁺*Faculty of Technology Management and Business, Universiti Tun Hussein Onn, Johor, Malaysia*

[^]*School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia*

Abstract— Software requirements engineering is an imperative phase of the software development lifecycle in every project regardless of the project size. In a project, different people are involved in the requirements engineering process, including requirement engineers, stakeholders, end users, and system designers. Amongst them, stakeholders play an essential role. Differences in goals and priorities of multiple stakeholders would make requirements management complex and difficult, which is a huge challenge for requirement engineers. From time to time, new requirements emerge, and existing requirements need changes to fulfil stakeholders' goals. Thus, such situation leads to high requirements volatility and low stability which causes overlapping and conflicting of requirements. The correctness and validity of requirements are of paramount importance as they are the key factors toward a successful system. A deep understanding of requirement management technique that conforms to users' needs is crucial. Such technique of the concept is applied to the Labour Management System. In this study, we have discussed the implementation of analysis of multi-stakeholder requirements using requirement interaction matrix in the f. The study used real requirements to yield a solid and dependable result. We have documented the requirements using a template and assessed their respective volatility level. An algorithm is constructed to show that the technique has managed to minimize the time used when checking requirements.

Keywords— requirements; multi-stakeholder; overlap; conflict; interaction matrix

I. INTRODUCTION

Many people are involved in the requirements engineering process, amongst them are stakeholders who play an essential role. In a system development, there may be more than one stakeholder involved. According to Atkinson et al. [1], stakeholders are defined into two groups: (i) environmental for example, customers, owners, and the community and (ii) process, for example, employees, and suppliers. The organization exists to serve the objectives of the stakeholders, which become its primary objectives. The organization gives each stakeholder group to achieve its primary objectives and expects its secondary objectives from stakeholders. The organization must plan as well as negotiate explicit and implicit contracts with stakeholders and evaluate whether the plan meets the expectations of all stakeholders. Process stakeholders design, implement and

manage processes to achieve the secondary objectives, expecting the primary objectives to be fulfilled. Differences in goals and priorities of multiple stakeholders would make requirements matrix in project management complex and difficult. If multiple stakeholder requirements are not handled appropriately, it will lead to overlapping and conflicting requirements. Managing large and complicated system development [2], [3] will require multiple stakeholders who are involved in a set of requirements. All the stakeholders' needs may have potentially conflicting goals due to different viewpoints of their respective roles [4]. Therefore, all stakeholders' goal needs to be assessed accordingly. The associated requirements to their goals also need to be evaluated carefully.

The problem that becomes of our focus is to overcome the overlapping and conflicting of requirements that come from multiple stakeholders. Overlapping and conflicting of

requirements will lead to misinterpretation of functions that needs to be implemented in the system. However, when redundancy of requirements occurs, we tend to use more time to search for the requirements that have conflict. To solve the problem, we used analysis of the interaction of requirement matrix to identify and classify the types of requirements.

Besides that, requirements change from time to time during the system development. The change of requirements is considered as requirements volatility. It has a great impact on the cost, schedule, and quality of the final product delivered. Many projects fail, and some are only completed partially because of requirements volatility. This study has motivated to analyse the techniques and templates used in managing requirements from multiple stakeholders thus identify and classify the types of requirements conflict present in the requirements from the interaction matrix.

Requirements from a single stakeholder are somewhat difficult to handle, as the needs of the stakeholder will change over time. When multiple stakeholders come together, each stakeholder has his or her own goals to fulfill; in which each goal has its own requirements in order to achieve the goals, causing requirements management to be more difficult. Hence, issues including overlapping, conflicting and volatile or unstable requirements occur. Table 1 shows the summary of the issues regarding multiple stakeholder requirements that we have discussed.

A. Overlapping and Conflicting Requirements

Overlapping requirements means requirements that affect each other but not necessarily conflict with each other. Overlapping requirements share certain common sub-goals [5]. In short, overlapping requirements are redundant requirements with similar scope. On the other hand, conflicting requirements means requirements that negatively affect each other and might cause a problem later in development phase if not handled properly. Conflicts occur when a similar scope of requirements from different parties' viewpoint is contradicted [6].

B. Volatile and Unstable Requirements

There are no standard definitions for requirements volatility [7]. The volatility of requirements is described by the change to requirements after the basic set of requirements has been agreed by both the clients and developers of the requirements. There are many causes for requirements volatility, both internal and external factors. External factors include government regulations and market competitors. On the other hand, internal factors include changes in organization environment, changes in technical aspects of the project, stability or requirements and others. Sudhakar [7] suggested that we use development models and phases in the system development to identify their advantages and disadvantages in managing requirements volatility.

C. Ambiguous Requirements

In general, ambiguous requirements are requirements which have unclear context or have more than one type of interpretation. Any task in requirements that can have more than one correct output that is contingent on a different

understanding of the task indicates that the requirement is ambiguous. When analyzing requirements, we must be able to deal with natural language requirements [8].

Nikora et al. [8] proposed an approach which suggests that relatively simple machine learning and natural language processing techniques are potentially useful in providing automated support to mission developers and assurance engineers for identifying ambiguous requirements within a set of specification documents. The automated support is able to help reduce the number of requirements defects in the implemented system [8].

In the proposed approach, they sampled a set of requirements as training data to be used for machine learning. After that, they read each requirement to classify it either as 89 requirement or unambiguous requirement [8].

TABLE I
ISSUES ON MULTIPLE STAKEHOLDER REQUIREMENTS

No.	Issue	Researcher	Solution / Techniques
1	Overlapping requirements	Pohl [9]	Use requirements interaction matrix to detect overlaps and conflicts in requirements
		Alshazly et al. [10]	Use combined reading technique to check software requirements specification documents
2	Conflicting requirements	Pohl [9]	Use requirements interaction matrix to detect overlaps and conflicts in requirements
		Alshazly et al. [10]	Use combined reading technique to check software requirements specification documents
3	Volatile/ Unstable requirements	Sudhakar [7]	Uses several development models and phases of the process to identify their advantages and disadvantages in managing requirements volatility.
4	Ambiguous requirements	Nikora et al. [8]	Uses relatively simple machine learning and natural language processing techniques to provide automated support in identifying ambiguous requirements.

D. Requirements Interaction Matrix

The challenges in requirements analysis stem from three sources: the sheer number of requirements and its interactions, the fact that requirements lie at the critical boundary between human understanding and machine representation, and the evolving nature of requirements as peoples' understanding improves [11].

The main objective of requirements analysis is to discover the interactions between requirements and to highlight requirements conflicts and overlaps. A requirements interaction matrix shows how requirements interact with

each other [9]. In a requirements interaction matrix, requirements are listed along the rows and columns of the matrix. Next, referring to Fig. 1 [9], the requirements are examined whether they are of which of the following types [12]:

- Requirements that use the same data (search by matching terms used)
- Overlapping requirements. According to Sommerville et al. [12], overlapping requirements means that there is some overlap between requirements in each viewpoint which should be discussed with a view to simplifying the requirements. To indicate overlapping requirements, '1000' is filled in.
- Independent requirements. For independent requirements, the viewpoints are independent to each other; a '0' is filled in the matrix.

Requirement	R1	R2	R3	R4	R5	R6
R1	0	0	1000	0	1	1
R2	0	0	0	0	0	0
R3	1000	0	0	1000	0	1000
R4	0	0	1000	0	1	1
R5	1	0	0	1	0	0
R6	1	0	1000	1	0	0

Legend:

- 0 – Independent requirement
- 1 – Conflicting requirements
- 1000 – Overlapping requirements

Fig. 1 An example of requirements interaction matrix [9]

The remainder of this paper is organised as follows. In Section II, the material and method are discussed in detail. Section III presents the result and the discussion of the experimental results. Finally, Section IV summarises the conclusion.

II. MATERIAL AND METHOD

A. Material

Labour Management System has been selected as a case study in this work. This system supports a wide range of functions that encourage the accomplishment of organization's goals. For example, it is responsible for managing payrolls and labour, reporting relationships and encouraging the accomplishment of the organization's goal. This system is deployed in many locations which are headquarters, region and branch with various roles of the user are accessed. There are 325 requirements have been captured from multiple stakeholders across locations. Thus, the interaction matrix is recommended for visualizing and documenting overlapping requirements and conflicts of requirements. It also can be used to systematically identify conflicts.

B. Method

Usually, overlapping requirements are originated from the same source, which is the same stakeholder. Hence, to determine overlapping requirements, first, we determine the

keywords in each requirement. Keyword-based extraction is a technique to retrieve textual information in documents with the aim of extracting keywords with respect to their relevance in the text. Proper keywords can serve as a highly concise summary of the document and ease the process of data retrieval in the document.

By implementing referencing Ciravegna's rule induction and generalization as well as keywords-based extraction [13], [14], we determined the keywords by extracting the verbs within each requirement while ignoring the words used in the requirements template. Next, we compared the requirements by comparing the similarity of keywords in the respective requirements. With references to classification evaluation [14], we adapted and used the following formula to determine the percentage of similarity when comparing the keywords of each respective requirement:

$$z = y/x \times 100\% \quad (1)$$

where z = percentage of keyword similarity,
 y = number of similar keywords, and
 x = number of total keywords ($x1 + x2$)

After comparing the requirement, we determined the percentage of similarity between numerous sets of two requirements. Next, the degree of overlap for each requirement is determined based on the percentage of similarity, according to the grid in Table 2.

If the keyword similarity percentage is high, then the specific set is further checked for overlapping requirements. For our case, we further checked the set of requirements for overlapping cases if the degree of similarity is high (more than 60%). To simplify, the summary of the process flow to determine overlapping requirements is shown in Fig. 2 below.

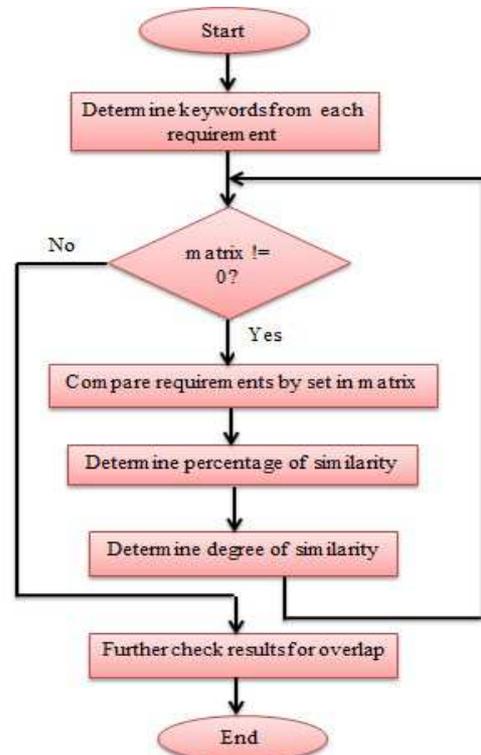


Fig. 2 Flowchart of determining overlapping requirements

TABLE II
DEGREE OF OVERLAP BASED ON SIMILARITY PERCENTAGE

Percentage of Similarity	Degree of Overlap
81% $\leq z \leq$ 100%	Very high
61% $\leq z \leq$ 80%	High
41% $\leq z \leq$ 60%	Moderate
21% $\leq z \leq$ 40%	Low

Conflicting requirements tend to occur due to the reason of (i) two requirements with a similar scope, but both were interpreted differently or (ii) it comes from two different users who have different opinions. Hence, to check for requirement conflicts, we further crosschecked every requirement to identify whether the requirements are in conflict or not. This is because every requirement has the possibility of conflicting with each other even though their similarity level is low.

According to Robertson, a requirement might potentially conflict with any other requirement in the specification [15]. To discover these problems, Robertson and Robertson suggested some clues to the situations where we most often find requirements in conflict:

- Requirements that use the same data (search by matching terms used)
- Requirements of the same type (search by matching requirement type)
- Requirements that use the same scale of measurement (search by matching requirements in which the fit criteria use the same scale of measurement)

In our algorithm, we tried to find requirements in conflict with requirements that use the same data or are in conflict in terms of function. First, the data entity and system entity for every requirement were determined. This was done using Ciravegna's rule induction and generalization [13] with keywords-based extraction by Slankas and Williams [14], where we classified the keywords and inducted them with data entity tag or system entity tag. The aim is to make it easier to determine whether they have conflict in terms of data or function.

After determining the entities for each requirement, we crosschecked those requirements to identify whether the requirements are in conflict. Then, we recorded the results of our findings. To simplify, Fig. 3 shows the flow on how we determine conflicting requirements.

III. RESULT AND DISCUSSION

After further checking for overlapping requirements, we have found from 325 requirements; there are 6 sets of requirements that overlap each other. The sets of requirements are found to have a similar scope of functions, only expressed in slightly different sentence structure. Some of them even have the same requirements. Therefore, they are considered as overlapping requirements. The results of overlapping requirements are as shown in Table 3.

We have checked the requirements by matching the data and system entities to determine whether the requirements have conflict in terms of data and function respectively.

After checking the sets of requirements, the results are recorded as shown in Table 4.

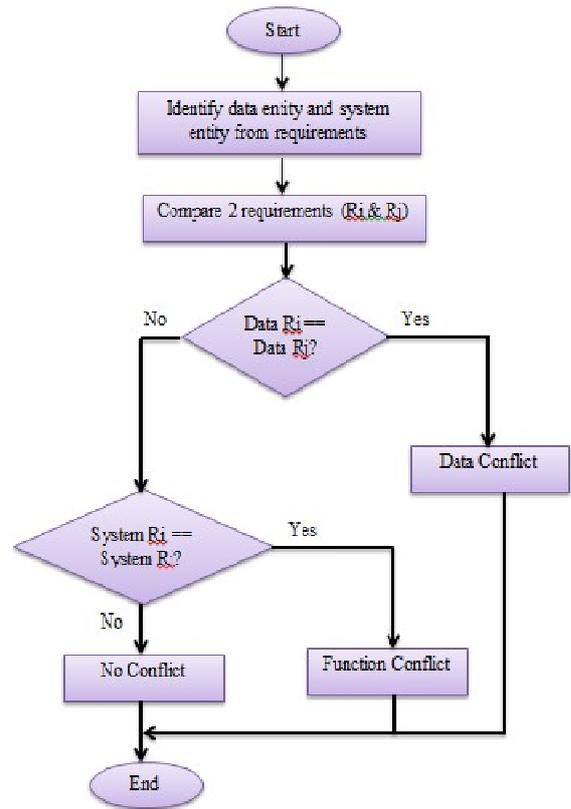


Fig. 3 Flowchart of determining conflicting requirements

Overall, from the case study data, we have found that out of all requirements that we crosschecked, there are four (4) sets of requirements in conflict, which includes two (2) data conflicts and two (2) function conflicts. For the two (2) sets of data conflict, it is because the data entities of requirement 302 are “view list of labour application” and “display details of selected user from list”, while data entities of requirement 602 are “view list of labour application” and “display details of selected user from list”. They have the same scope in terms of data and come from different users. Therefore, they are considered as conflicting in terms of data.

On the other hand, there are two (2) function conflicts identified. To exemplify, in our case, requirement 101 and requirement 601 that have been checked for function conflicts. Requirement 101 states that “the system shall provide the user with the ability to perform labour management functionality”. The system entities identified for requirement 101 is “perform labour management functionality”. On the other hand, requirement 601 states that “the system shall provide the user with the ability to perform labour administrative functionality”. The system entity identified for requirement 601 is “perform labour administrative functionality”. The terms administrative and management have a similar meaning. The requirements are interpreted as a similar function but expressed in different terms. Therefore, they are considered as conflicting in terms of function.

TABLE III
OVERLAPS IN REQUIREMENTS

ID	R _x	ID	R _y	Keyword Similarity Score	Reason
302	The system shall provide the user with the ability to view a list of labour application. The system shall be able to display application information details of the selected user to select from the list.	402	The system shall provide the user with the ability to view labour application details. The system should be able to select specific labour from the list and display the information details.	82.35	The similar scope of function. Different sentence structure.
402	The system shall provide the user with the ability to view labour application details. The system should be able to select specific labour from the list and display the information details.	302	The system shall provide the user with the ability to view a list of labour application. The system shall be able to display application information details of the selected user to select from the list.	82.35	The similar scope of function. Different sentence structure.
501	The system shall provide the user with the ability to monitor updated labour information.	701	The system shall provide the user with the ability to monitor updated labour information.	100	Same requirements
603	The system should be able to display labour information report. The system shall provide the user with the ability to choose a list of the report to be verified.	803	The system should be able to display labour information report. The system shall provide the user with the ability to choose a list of the report to be verified.	100	Same requirements
701	The system shall provide the user with the ability to monitor updated labour information.	501	The system shall provide the user with the ability to monitor updated labour information.	100	Same requirements
803	The system should be able to display labour information report. The system shall provide the user with the ability to choose a list of the report to be verified.	603	The system should be able to display labour information report. The system shall provide the user with the ability to choose a list of the report to be verified.	100	Same requirements

* R_x – requirement in row position, R_y – requirement in column position

TABLE III
CONFLICT IN REQUIREMENTS

x	R _x	ID	R _y	Keyword Similarity Score	Conflict	
					Data	Function
101	The system shall provide the user with the ability to perform labour management functionality.	601	The system shall provide the user with the ability to perform labour administrative functionality.	75	N/A	√
302	The system shall provide the user with the ability to view a list of labour application. The system shall be able to display application information details of the selected user to select from the list.	602	The system shall provide the user with the ability to view a list of labour information. The system should be able to select specific labour from the list and display the information details.	75	√	N/A
601	The system shall provide the user with the ability to perform labour administrative functionality.	101	The system shall provide the user with the ability to perform labour management functionality.	75	N/A	√
602	The system shall provide the user with the ability to view a list of labour information. The system should be able to select specific labour from the list and display the information details.	302	The system shall provide the user with the ability to view a list of labour application. The system shall be able to display application information details of the selected user to select from the list.	75	√	N/A

* R_x – requirement in row position, R_y – requirement in column position

IV. CONCLUSIONS

This project is believed to potentially improve the understanding between requirement engineers and stakeholders of an organization or system during requirements engineering process. In addition to that, it could also ensure a certain level of completeness and correctness of requirements by documenting requirements using requirements template. The study also proposes that the technique can be tested on the requirements of other large-scale systems in the future. The results and observations acquired are documented as a reference for future work. This is to encourage improvement when applying the technique in the future. It is hoped that in the future, we will be able to come up with a better version of the algorithm to determine conflicting and overlapping requirements automatically in order to save more effort and time.

ACKNOWLEDGMENT

This research was funded by research contract grant, Vot No: 4C097. Also many thanks to collaborative sponsor GATES Scholars Foundation of GATES IT Solution Sdn. Bhd. Company and RMC, UTM.

REFERENCES

- [1] A. A. Atkinson, J. H. Waterhouse, and R. B. Wells, "A Stakeholder Approach to Strategic Performance Measurement," *Massachusetts Institute of Technology*, 1997. [Online]. Available: <http://search.proquest.com/openview/6938632d1a8049e1bb07bc5534ffde20/1?pq-origsite=gscholar&cbl=1817083>. [Accessed: 06-Aug-2017].
- [2] A. A. Hamid and Z. Mansor, "Client's Readiness Assessment Success Factors for Outsourcing Software Projects," *International Journal on Advanced Science, Engineering and Information Technology*, vol 6, no. 6, p.p 820-827, 2016.
- [3] M. Alqudah and R. Razali, "A Review of Scaling Agile Methods in Large Software Development," *International Journal on Advanced Science, Engineering and Information Technology*, vol 6, no. 6, p.p 828-837, 2016.
- [4] N. Abdul Rahman, R. Hassan, R. M.Othman, and H. Asmuni, "The Role-based Goal Modeling," *Journal of Software*, pp. vol 11, no 12, 1242-1249, 2016.
- [5] H. Liu, W. Shao, L. Zhang, and Z. Ma, "Detecting overlapping use cases," *IET Softw.*, vol. 1, no. 1, p. 29, 2007.
- [6] G. Kotonya and I. Sommerville, "Requirements engineering with viewpoints," *Softw. Eng. J.*, vol. 11, no. 1, p. 5, 1996.
- [7] M. Sudhakar, "Managing the Impact of Requirements Volatility," Umeå University, Sweden, 2005.
- [8] A. Nikora, J. Hayes, and E. Holbrook, "Experiments in Automated Identification of Ambiguous Natural-Language Requirements," *To Appear. Proc. 21St ...*, 2010.
- [9] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*. 2010.
- [10] A. A. Alshazly, A. M. Elfatraty, and M. S. Abougabal, "Original Article: Detecting defects in software requirements specification," *Alexandria Eng. J.*, vol. 53, pp. 513-527, 2014.
- [11] M. S., Feather, S. L., Cornford and M. Gibbel, "Scalable Mechanisms for Requirements Interaction Management", 2000.
- [12] I. Sommerville, and P. Sawyer, "Viewpoints: principles, problems and a practical approach to requirements engineering", *Annals of Software Engineering*, vol 3, no 1, p.p 101-130, 1997.
- [13] F. Ciravegna, "Adaptive Information Extraction from Text by Rule Induction and Generalization," *Proc. 17th IJCAI*, pp. 1251-1256, 2001.
- [14] J. Slankas and L. Williams, "Automated extraction of non-functional requirements in available documentation," in *2013 1st International Workshop on Natural Language Analysis in Software Engineering, NaturaLiSE 2013 - Proceedings*, 2013, pp. 9-16.
- [15] S. Robertson and J. Robertson, "Mastering the Requirements Process Getting Requirements Right," *Work*, vol. 44, no. February, p. 15, 2013.