Ontology Based Requirement Interdependency Representation and Visualization

Safeeullah Soomro[1], Abdul Hafeez[2(✉)], Asadullah Shaikh[3,4], and Syed Hyder Abbas Musavi[1]

[1] Faculty of Engineering Science and Technology, Indus University, Karachi, Pakistan
ssoomro@indus.edu.pk, drhyderabbas@yahoo.com
[2] Hamdard University, Karachi, Pakistan
ahkhan@smiu.edu.pk
[3] College of Computer Science and Information Systems, Najran University, Najran, Saudi Arabia
asshaikh@nu.edu.sa
[4] Department of Computer Science and Information Technology, Institute of Business and Technology (IBT), Korangi Creek, Karachi 75190, Pakistan
ashaikh@ibt.edu.pk

Abstract. Requirement interdependency is being extensively studied with greater interest in software engineering research. However, existing tools and techniques have not properly characterized and visualized the requirement interdependency relationships between requirements. This research introduces ontology based representation of requirement interdependencies among requirements, and formal graphical notation for proper visualization of requirement interdependencies. There is evidence to point out that Ontology based approach is better technique for managing the requirement interdependency i.e., diagrammatic representation of requirement interdependency will improve the quality of software and will reduce the failure rates.

Keywords: Ontology requirements · UML ontology · Requirement traceability

1 Introduction

Software project should conclude on time, be within the specified budget and must fulfill the user requirements. All these aspects measure the quality of software. Unfortunately most of the software projects fail in at least one of these aspects. The primary factor of measuring the quality of software is fulfilling the user's requirements. Suppose that software is developed on time and within the budget but not fulfilling the user requirements then it is not considered to be quality software. Therefore, accomplishment of the user requirements is most important aspect of Quality in Software development.

F.K. Shaikh et al. (Eds.): IMTIC 2013, CCIS 414, pp. 259–270, 2014.
DOI: 10.1007/978-3-319-10987-9_24

Software requirement specification (SRS) is developed in the first phase of software development and it serves as an agreement between user and software development team. It specifies the features included in software. An important factor with SRS is that it continuously adopts change during the entire software development life cycle due to emerging changes in user's requirements. These changes affect the software development in dual perspective. Firstly Changes in requirements should be properly incorporated with release of software secondly changes can also affect the other requirements because software requirements have relationships with each other. Sometime software team overlooks these relationships among requirements due to their unclear visibility and improper representation [1]. Requirement traceability maintains complete track and trace of requirements during software development. Requirement traceability is considered more important in safety critical software [10]. Requirement interdependency being an integral part of requirement traceability is the study of understanding and managing relationships among requirements. Properly represented requirement interdependency can reduce the chances of software failure and the lack of the requirement interdependency information may lead to higher costs, wrong or unnecessary changes and wastage of time during its development. Therefore, well-managed requirement interdependency improves the quality of software.

This paper presents a formal approach in managing and visualizing requirement interdependency. Proposed methodology uses the ontology for the formal description of interdependency model and presents diagrammatic notation for representing requirement interdependency.

The remainder of the paper is structured as follows. In Sect. 2 the requirement interdependency types and current approaches of representation and visualization are explained. Ontology and web ontology language explained in Sect. 3. Section 4 presents the proposed methodology ontology based requirement interdependency management and diagrammatic notation for requirement interdependency that concludes with the results of the case study carried out thereon. Section 5 explains the tool support. In Sect. 6 related research is presented. Finally, conclusions are drawn in Sect. 7 along with future work.

2 Requirement Interdependency and Requirement Traceability

Requirement interdependency is a particular type of requirement traceability that focuses on change relationships among different requirements. Requirement traceability manages and specifies the relationship among different artifacts that are constructed during the software development. Gotel and Finkelstein [6] suggested the following common definition for the term requirements traceability: "Requirements traceability (RT) refers to the ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent development and use, and through periods of ongoing refinement and iteration in any of these phases) [8]."

The core aspect of traceability is 'requirement to requirement traceability' that is also called requirement interdependency. It is used to discover the impact on other requirement when one or more requirements are changed.

2.1 Interdependency Types

Dahlsedt and Persoon [4] compile the different views expressed in the literature into an integrated model that is neutral as far as development of a new situation is concerned. They proposed nine interdependencies types within two categories vis-à-vis (i) Structural Interdependencies: Require, Explain, Similar To, Conflict With, and Influences (ii) Cost / Value Interdependencies: Increase/Decrease Cost, Increase/Decrease Value. We are defining the above interdependencies' types as under:

Require Interdependency describes the depending relationship among requirements. It illustrates one or many compulsory requirements for the accomplishment of other requirement. For example, calling a remote procedure requires network connection.

Explain Interdependency describes the composition relationship among requirements. It illustrates that a requirement is combination of many other requirements.
For example: A security requirement for data can be combination of authorization and authentication requirements.

Similar to interdependency describes the equivalency relationship among requirements. It illustrates that the requirement is equal to another requirement. For example, the information about students will be searched by name and the student information will be searched by roll number.

Conflict With interdependency describes the contradictory relationship among requirements. It illustrates that a requirement is inconsistent with other requirements. For example, all users can reset their account passwords and only administrator can change the password of users.

Influences interdependency illustrates that a requirement makes some effect on another requirement other than Require, Conflict, and Explain. For example, some legal requirements depend on industry and type of business for which software is developing.

Increases/Decreases cost interdependency type illustrates that a requirement can increase/decrease cost of another requirement. For example, A Copy of data should be stored on cloud platform. This will most likely increase the implementation cost of many other requirements such as security.

Increases/Decreases Value interdependency type illustrates that requirement increases/decreases the worth of another requirement. For example, online money transfer facility decreases the value of other payment mode like cheque, draft, etc.

Table 1. Traceability list [11].

Requirement	Depends-on
R1	R3,R4
R2	R5,R6
R3	R4,R5
R4	R2
R5	R6

Table 2. Interdependency types by [3,14].

Type	Meaning
R1 AND R2	R1 requires R2 and R2 requires R1
R1 REQUIRES R2	R1 requires R2, but not vice versa
R1 TEMPORAL R2	Either R1 has to be implemented before R2 or vice versa
R1 CVALUE R2	R1 affects the value of R2 for customer, either positive or negative
R1 ICOST R2	R1 affects the cost of implementing R2, either positive or negative
R1 OR R2	Only one from R1 or R2 needs to be implemented

2.2 Current Techniques of Requirement Interdependency Representation

Current practices are generally using traditional traceability techniques for representing requirement interdependency traceability. These are (1) Traceability List and (2) directed graph. Requirement traceability mainly focuses only one relationship called change. However, requirement interdependency has various relationships as discussed in Sect. 2.1 above.

Traceability list is a simple table of relationships describing dependencies among requirements as shown in Table 1. There might be several such lists, one for each type of relationship such as Require, Similar To, Conflict [11].

A directed graph (digraph) is also used for representing Requirement Interdependency. Carlshamre [3] visualized requirement interdependency through digraph for incremental software development. He also represents five interdependency relationships as shown in Table 2.

3 Ontologies

Ontology is the concept of metaphysics, and is used by philosopher from mid of sixteen century for categorization and representation of entities. Ontologies have various elements like classes, relations, individual, etc. for making clear the concept of entities. These are extensively used in knowledge Engineering, Artificial intelligence, ecommerce, natural language processing. Due to Ontology provides a great amount of expressiveness and facility of reasoning, it is widely

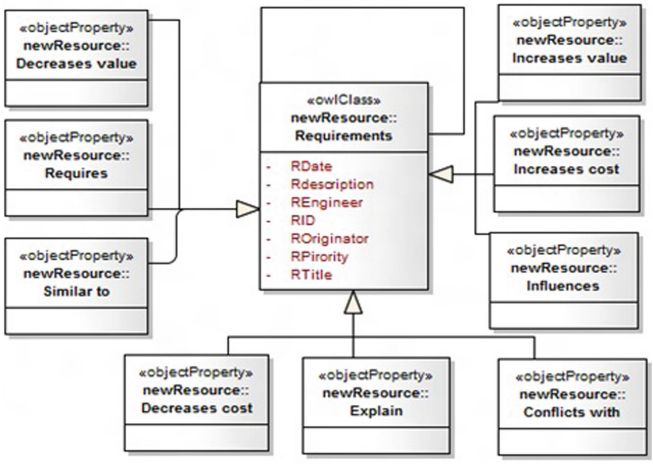

Fig. 1. Requirement interdependency ontology.

using in software engineering practices [7]. Web Ontology Language (WOL) is a declarative language for ontology development. It was developed in 2003 by W3C. It is designed to express a wide variety of knowledge as well as provide reasoning facilities to express the most important kinds of knowledge. It comes with three flavors: OWL Lite, OWL DL (description logic), and OWL Full. These three levels are in increasing order of expressivity.

4 Ontology of Requirement Interdependency and Diagrammatic Representation

In this section we are proposing requirement ontology and graphical notation for proper management and visualization of requirement interdependency. Here the requirement ontology is constructed for managing the various interdependency relationships among the requirements defined in Sect. 2.1. For managing requirement through ontology and generating requirement interdependency relationship diagram, we have built a tool that discussed the management, visualization of requirement independencies efficiently in Sect. 5 of this paper.

4.1 Requirement Ontology

The building block of present work is requirement ontology shown in Fig. 1. In this ontology a class is declared called Requirements. This class represents a concept similar to software requirements. Requirements class declared seven data types properties (RID, RTitle, RDescription, etc.) shown in Fig. 1 and complete details of these properties are listed in Table 3. Nine interdependency relationships those are explained in Sect. 2.1 are also declared in requirements class through object properties shown in Fig. 1.

Table 3. Data type properties.

S.No	Data type Properties	Description
1	RID	Identifier for requirements e.g. "U-001"
2	Rtitle	Title / Name of requirement e.g. "Add User" etc
3	ROriginator	Name of person who state the requirement
4	RPirority	Priority of requirement like (1 or 10)
5	REngineer	Name of requirement engineer
6	Rdescription	Description about Requirements
7	Rdate	Requirement gathering Date

These object properties are used to establish relationships among different requirements individual in ontology instance. A reflexive relationship (self relationship) is also specified on requirements class. For example one instance of requirements class associates with another instance of requirements class through object properties. Therefore, object properties establishing the interdependency relationship between different requirements. Finally, an external restriction is applied to specify that the object properties are 'established relationships' between only members of requirements class.

4.2 Requirement Interdependency Relationship Diagram (RIRD)

Requirement Interdependency is very important aspect of software development. Success factor of software is mostly dependent on proper management of requirement and their change relationships (interdependency). If software development team is not fully aware of requirement interdependency relationships then there is a risk that software could not fulfill the user need, exceed from estimated budget and schedule or even may be failed. Therefore, proper representation of requirement interdependency relationships for quality software development is very crucial. Earlier research shows that traceability list and diagraph have been used for representing the requirement interdependency. These methods are well suited for representing small number of interdependency relationships; however for large and complex interdependency relationships, these methods have some limitations. As an example, for the large amount of interdependency relationships, table and graph could be difficult to manage and understanding of interdependency will be difficult at a glance. Therefore, there is need of proper diagrammatic notation for representing the requirement interdependency. The RIRD is graphical representation for better representation and visualization of requirement interdependency relationships. In RIRD the requirement is represented by a rectangle and the title of requirement appears at middle of rectangle. Requirement interdependency relationships are represented by various symbols listed in Table 4 that are attached in the border of rectangle with in a circle called relationship circle. A unidirectional arrow shows the direction of interdependency relationships and optional label can be attached on arrow to show

Table 4. Symbols for requirement interdependency relationship.

S.No	Symbols	Interdependency Types
1	+	Requires
2	*	Explain
3	=	Similar to
4	X	Conflicts with
5	>	Influences
6	> .	Increases cost
7	< .	Decreases cost
8	>	Increases value
9	<	Decreases value

Fig. 2. RIRD for require interdependency relationship visualization.

interdependency type. For example a requirement "Edit User" Require another requirement "Search User" represent a RIRD as shown in Fig. 2.

Relationship circle will be attached to only those requirements that have interdependency relationship with other requirements. Also, the unidirectional arrow starts from that requirement as shown in Fig. 2.

When changes arise in requirements, RIRD diagram also shows its impact on other requirement through different requirement rectangles and arrow styles. A change requirement is shown with double border rectangle. The direct impact of change is shown by rectangle with thick border and thick arrow. Indirect impact of change is shown by dotted rectangle and dotted arrow.

Figure 3 is the complete illustration about requirement change impact. Here, requirement 3 is presented in double border rectangle that shows change in requirement 3. A thick arrow is moving from requirement 3 to requirement 4 indicates the direct impact change. Also requirement 4 is placed in thick rectangle. Dotted arrows are moving form requirement 4 to requirement 5 and requirement 5 to requirement 6 is showing the indirect requirement change impact.

4.3 Implementation

This section illustrates the proposed work with an example as a user management system (UMS). UMS is mostly part of large software. It provides security mechanism through user authorization and authentication.

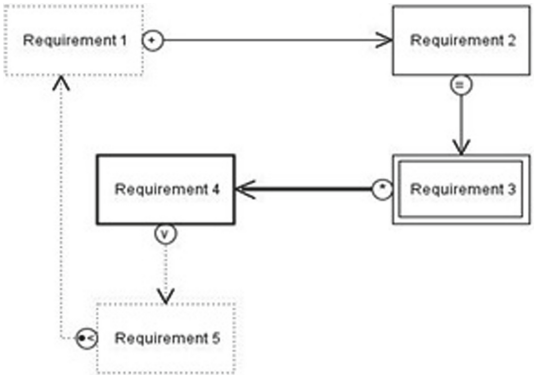

Fig. 3. RIRD of requirement change impact visualization.

1. Requirement ID: R001 Description: System shall provide the facility of edit user only by administrator like (change status, activate, deactivate etc.) and notification is sent to user
2. Requirement ID: R002 Description: Notification of 'status change' sent to user via email
3. Requirement ID: R003 Description: System shall provide the facility of search user
4. Requirement ID: R004 Description: System shall provide Search user option By ID and BY Name
5. Requirement ID: R005 Description: System shall provide the facility of Update user status
6. Requirement ID: R006 Description: Data cannot be accessed from outside the Local area Network due to security reason
7. Requirement ID: R007 Description: When the Users' status will be changed then the mobile message will also be sent to Users.
8. Requirement ID: R008 Description: User can be searched by User ID and Name from mobile

Requirement R001 has required interdependency relationship with R002 and R003 and the same with requirement R005. The value of requirement R002 is decreased by requirement R007. Requirement R003 is explained by requirement R004. Requirement R004's cost is increased by requirement R008. Requirement R008 and requirement R006 are conflicting. Figure 4 shows RFID of user management system.

Suppose User remove notification feature from the requirement R001. Figure 5 shows change impact in requirement 1 to other requirements. When the requirement R001 is changed, it will directly impact on requirement R002 and indirectly impact on requirement R007.

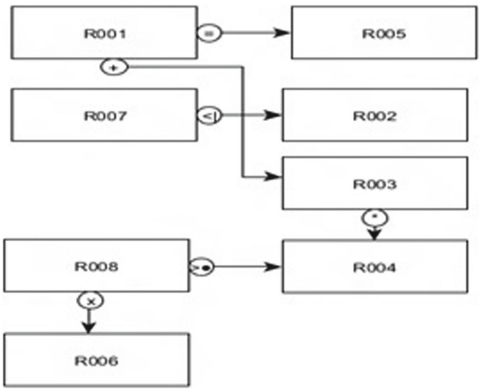

Fig. 4. RIRD of user management system.

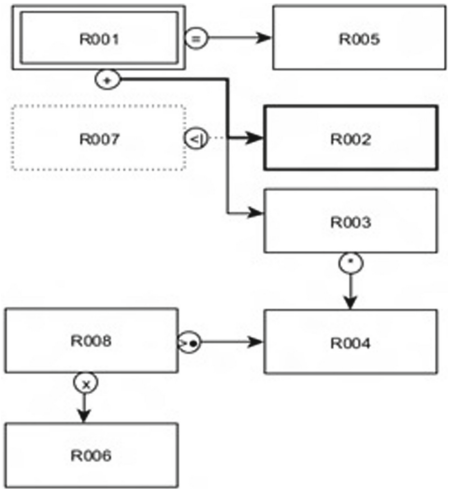

Fig. 5. RIRD after change of user management system.

5 Tool Support

This section will give the details about a tool that has been designed for managing requirement through ontology and generate requirement interdependency relationship diagram. Figure 6 is showing the structure of the mentioned tool. The software tool depicted in Fig. 6 has been built using java language with jena API for ontology processing. Jena provides management of OWL and RDFS ontology. It also provides rule based inference engine for reasoning on RDF and OWL. For diagram publishing, this software uses Graph visualization (Graphviz). Graphviz is an open source tool for creating the graph and is developed by AT&T labs for graph drawing that uses the script language DOT for specifying the graph.

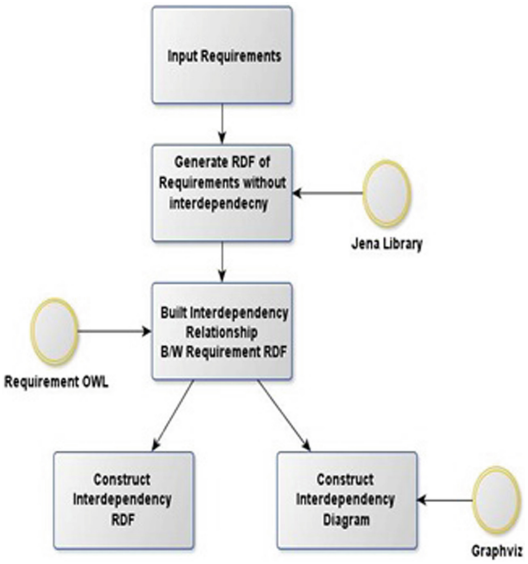

Fig. 6. Software tool structure and flow.

The proposed tool receives the user requirement from a user through a Form that captures the seven data type properties of requirement. The 'Requirements' are saved in RDF format without interdependency relationship among the requirement. After providing input as the "requirement", the user specifies the relationship among requirement through a form. When user finished the relationship specification then system creates interdependency relationship among requirements through requirements ontology and inserts them between previously generated RDF. Finally Requirement Interdependency Relationship diagram (RIRD) and RDF are generated by the software.

6 Related Research

In the early research of requirement traceability, Pohl [11] proposed 18 different types of dependency relationships between various types of trace objects used in software engineering. However, many dependency type presented in Phol [11] model cannot exist among requirements. Carlshamre and Regnell [2] proposed different types of interdependencies. In the extended work of [9] and [3], Carlshamre et al. [3] conducted an industrial survey on requirements interdependencies within release planning. He identified six different types of interdependencies, partially based on the types presented in [9], and relatively analyzed 20 high priority requirements within five different companies. Ramesh and Jarke [12] presented the reference models for requirements traceability. They did not focus on requirements interdependencies, but identified the 'requirements interdependencies' as a "traceability problem". The majority of the dependency

types discussed by Ramesh and jarke [12] are related to requirements management and requirements evolution. Robinson et al. [13] reported on a field called requirements interaction management. This field focuses on managing relationships between requirements; those may interfere with each other's achievements. Robinson et al. [13] focused on conflicts between requirements, and recognize the problems with satisfying requirements at requirements definition time. He also gave explanation about different requirements interdependency types. Dahlstedt and Persson [4] in 2003 explained various types of interdependencies through literature and empirical study. He describes nine types of interdependencies exits between requirements. He also group requirement interdependency relationships into two main categories: structural and cost/value. Arda Goknil [5] introduced the way of representation and managing the requirement interdependency through OWL and semantic. He performed inferencing and consistency checking and proposed the mechanism to derived new relationships from existing ones and determined contradiction among requirement. He built a tool called Tool for Requirements Inferencing and Consistency checking (TRIC) that performs automatic inferencing and consistency checking. He also did impact analysis on requirement change with the help First order Logic (FOL). However he worked on only five interdependency relationships (Require, Refines, Partially Refines, Contain and Conflict) and did not focus on visualization and any diagrammatic approach for proper representation. Yuanyuan Zhang [15] worked on five interdependency types (And, Or, Precedence, value related and cost related). He is of the opinion that not much work has been carried out on entire five interdependency types collectively. He has taken Requirement Interdependency problem as a constraint satisfaction problem and has proposed three search based algorithms to tackle requirement interdependencies. He also investigated the impact of requirement interdependency relationship on the automated requirement selection process for release planning.

7 Conclusion and Future Work

Software Requirement Interdependency performs vital role in software development. Software success factor is entirely based on proper understanding of requirement interdependency. Therefore, proper representation of requirement interdependency plays a crucial role in software development. Current research in requirement interdependency mainly pays attention in identifying the interdependency relationship types but the representation and visualization is not being focused upon thoroughly. This research proposed ontology based representation of requirement interdependency relationships and diagrammatic notation called Requirement Interdependency Relationship Diagram (RIRD) for proper visualization of requirement interdependency. The RIRD is used to clarifying the requirement interdependencies aspect of software team, and when the requirement changes the RIRD diagram graphically shows its impact on other requirements. There are still some open issues like verification and validation

of requirement interdependency relationship and ontology based impact analysis of different artifact of software engineering. We are working on developing an ontology based representation of interdependency of different artifacts in Object Oriented Software Engineering.

References

1. Ali, N., Gueneuc, Y., Antoniol, G.: Trustrace: mining software repositories to improve the accuracy of requirement traceability links. IEEE Trans. Softw. Eng. **39**(5), 725–741 (2013)
2. Carlshamre, P., Regnell, B.: Requirements lifecycle management and release planning in market-driven requirements engineering processes. In: Proceedings of the 11th International Workshop on Database and Expert Systems Applications, pp. 961–965. IEEE (2000)
3. Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J.: An industrial survey of requirements interdependencies in software product release planning. In: Proceedings of the Fifth IEEE International Symposium onRequirements Engineering, pp. 84–91. IEEE (2001)
4. Dahlstedt, Å.G., Persson, A.: Requirements interdependencies: state of the art and future challenges. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 95–116. Springer, Heidelberg (2005)
5. Goknil, A., Kurtev, I., van den Berg, K., Veldhuis, J.-W.: Semantics of trace relations in requirements models for consistency checking and inferencing. Softw. Syst. Model. **10**(1), 31–54 (2011)
6. Gotel, O.C., Finkelstein, A.C.: An analysis of the requirements traceability problem. In: Proceedings of the First International Conference on Requirements Engineering, pp. 94–101. IEEE (1994)
7. He, H., Wang, Z., Dong, Q., Zhang, W., Zhu, W.: Ontology-based semantic verification for uml behavioral models. Int. J. Softw. Eng. Knowl. Eng. **23**(02), 117–145 (2013)
8. Hokkanen, M., et al.: Requirements traceability (2001)
9. Karlsson, J., Olsson, S., Ryan, K.: Improved practical support for large-scale requirements prioritising. Requir. Eng. **2**(1), 51–60 (1997)
10. Mc Caffery, F., Casey, V., Sivakumar, M., Coleman, G., Donnelly, P., Burton, J.: Medical device software traceability. In: Cleland-Huang, J., Gotel, O., Zisman, A. (eds.) Software and Systems Traceability, pp. 321–339. Springer, Heidelberg (2012)
11. Pohl, K.: Process-centered Requirements Engineering. Wiley, New York (1996)
12. Ramesh, B., Jarke, M.: Toward reference models for requirements traceability. IEEE Trans. Softw. Eng. **27**(1), 58–93 (2001)
13. Robinson, W.N., Pawlowski, S.D., Volkov, V.: Requirements interaction management. ACM Comput. Surv. (CSUR) **35**(2), 132–190 (2003)
14. Svensson, R.B., Gorschek, T., Regnell, B.: Quality requirements in practice: an interview study in requirements engineering for embedded systems. In: Glinz, M., Heymans, P. (eds.) REFSQ 2009 Amsterdam. LNCS, vol. 5512, pp. 218–232. Springer, Heidelberg (2009)
15. Zhang, Y., Harman, M., Lim, S.L.: Empirical evaluation of search based requirements interaction management. Inf. Softw. Technol. **55**(1), 126–152 (2013)