

Enhancing Requirement Traceability Link Using User's Updating Activity

D.Anandha Kumar, Dr.P.Raviraj

PG Student, Dept of Computer Science and Engineering, Kalaignar Karunanidhi Institute of Technology, India.

Professor and Head , Dept of Computer Science and Engineering, Kalaignar Karunanidhi Institute of Technology, India.

Abstract— Requirement Traceability is one that tracks software project requirement in its source code for maintaining the consistency between them. It is an important process that it could be useful for the construction of software project in time within cost. Therefore it helps to complete the software project in a successful manner. Such an important process is needed to be enhanced for providing better accuracy of trust to requirement traceability link. In this paper, the user's updating activity like bug tracking system, forum, cvs/svn repositories are used to improve the trust of requirement traceability link so as to complete the software project in more successful manner.

Keywords— Requirement traceability link, Experts, Bug tracking system, Forum, Repositories.

I. INTRODUCTION

Requirements traceability could be used to describe and track the life of a requirement in software development process. That is in all phase of software development the requirements are tracked to find whether the source code is consistent to its requirement which is provided by the client in association with the requirement specification engineer. This process if done manually will lead to a costly task in software engineering. There comes the need for automatically tracing requirement in the source code. In traceability, it is important to consider of what and how to access and present the software artifacts. This lead us to know who want that software artifact, when or why that is wanted, also the software project characteristics. In requirement traceability, there arises a need to know the working practice. That is to know that whether there were sufficient resources, time support, ongoing cooperation and coordination or not. Requirement traceability also depends on the awareness of information required to trace, the ability to obtain and

document required information, and the ability to organize and then maintain required information for flexible traceability of requirements [1].

A. Enhancement By Information Retrieval Techniques

Information Retrieval (IR) approaches [2] are studied to be useful in recovering traceability links between free-text documentation and source code. This IR-based

traceability recovery approaches could provide ranked lists of traceability links. Yet, the precision and recall value did not meet the saturation point. In reference [3], they have used the web trust models [4], [5], [6], [7] to improve precision and recall of traceability links. In this

web trust model the cvs/svn repository are used as a main data to improve the value of precision and recall value. In this requirement traceability link's improvement process the cvs/svn change log and the bug tracking system could be taken as expert. Based on the opinion of the expercould be used to remove and or adjust the rankings of the IR-based approaches produce traceability link. They [3] used Vector Space Model (VSM) based information retrieval approaches to produce the baseline requirement traceability link. Then the baseline link is combined with the trust model to give more surety to the baseline link. Thus it is studied that the combined use of IR-based approaches and trust model will result in the improvement of precision and recall value of requirement traceability link.

B. Uses Of Traceability Link

The traceability links provide hints on where to look for requirement's implementation in code. The main role of the traceability links is to assist programmers in the assignment of a concept to the code and in the aggregation to abstract concepts. Fig. 1 shows that the requirement is tracked in the source code for its consistency. Here the requirement is given by the customer and the source code are stored in cvs/ svn repository. The CVS/SVN repository contains the same project with different version. These versions are created by the developers, whenever they update and compile the project. The source code with particular version which

satisfies the requirement is used to make a link with the appropriate requirement. This link is called requirement traceability link (RTL) which is a set composed of three elements, they are requirement, source code and similarity value.

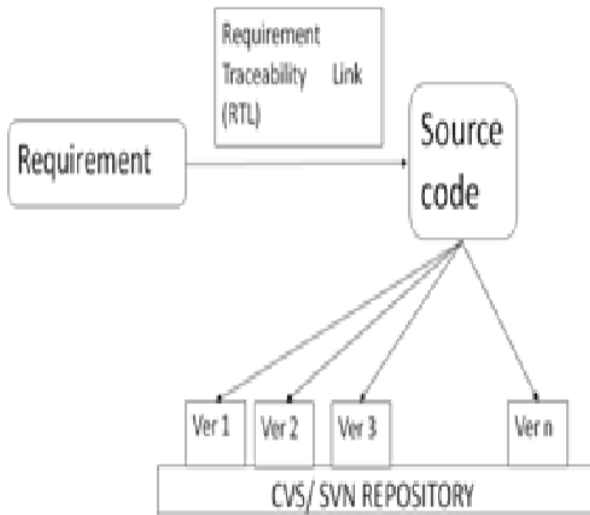


Fig. 1. Requirement Traceability Link (RTL)

Traceability links between code and other sources of information are a reasonable aid to perform the combined study of heterogeneous information and, finally, to associate domain concepts with code and vice-versa.

Traceability links between the requirement and the code are a solution to locate the areas of code that contribute to implement specific user's functionality [8], [9], [10]. This is helpful to appraise the comprehensiveness of an implementation with respect to stated requirements, to devise complete and comprehensive test cases, and to infer requirement exposure from structure exposure during test time. Traceability links between requirements and code can also help to identify the code areas directly affected by a repairs request as confirmed by an end user. Finally, they are useful during code check up, on condition that inspectors with clues about the goals of the code at and in quality assessment, for a case in point, singling out loosely-coupled areas of source code that implement diverse requirements [15].

III. ENHANCEMENT BY PREPROCESSING REQUIREMENT AND SOURCE CODE

The requirement specification and the source code are the two documents which are considered as the higher level and lower level document. Software artifact's documentations are mostly informally expressed. Such documents possibly related to requirement, bug logs, source code and related maintenance reports. Information retrieval could be used to recover traceability links between source code and free text requirement specification documents. For that it is needed to preprocess meaningful name like class name. Class will mostly implement most of the requirement in the source code. Therefore it is important to know the application-domain knowledge which would help to improve the

semantic of the class identifiers. These identifiers can help to associate between high-level document and low-level document.

II. COMPARING THE INFORMATION RETRIEVAL MODELS

A. Information Retrieval (IR)

The software engineer needs to trace the requirement in the source code which was developed. This is done to ensure that the developer is only doing what the client required. Here the information needed to retrieve is that the link between the client's requirement and the source code developed. This link is called requirement traceability link. In information retrieval the major models to retrieve information are the Boolean model, the statistical model which also includes the vector space and probabilistic retrieved model [11]. While retrieving the requirement traceability link there may be two problems [12]. First, they retrieve irrelevant links between the requirement and source code. Second they do not retrieve all the relevant links.

B. Basic Measures For Retrieval Methods

The two measures mostly used to evaluate the effectiveness of a retrieval method are precision and recall. Let the set of links relevant to the requirement traceability be denoted as {relevant links} and the set of links retrieved be denoted as {retrieved links}. The set of links that are both relevant and retrieved is represented as {relevant links} ∩ {retrieved links}. Precision is identified as the percentage of retrieved links that are in fact relevant to the traceable requirement. This indicates the correct link retrieved response. It is formally depicted [3] as

$$precision = \frac{| \{relevant links\} \cap \{retrieved links\} |}{| \{retrieved links\} |}$$

Recall is the percentage of links that are relevant to the traceable requirement and were in fact retrieved. It is formally depicted [3] as

$$recall = \frac{| \{relevant links\} \cap \{retrieved links\} |}{| \{relevant links\} |}$$

An information retrieval system often requires to trade off recall for precision or precision for recall. Usually f-score is used as trade off which is defined as the harmonic mean of recall and precision measure. It is formally depicted [3] as

$$fscore = \frac{\{recall measure * precision measure\}}{\{\{recall measure + precision measure\}/2\}}$$

Precision, recall and fscore are only the measure for a retrieved set of links. These three measures are not mainly useful for comparing ranked links for various versions of cvs/svn repository because they are not more detective to the ranking of the links in a retrieved set. If the qualities of a ranked list of links are need to measure then we need to compute average of precisions at all the ranks where a new relevant links is returned. One can also plot a graph of precision at many different levels of recall; higher curve depicts a better quality information retrieval system. Precision and recall are the very useful in measuring the efficiency of information retrieval methods such that it is possible to predict how much

enhancement is provided for retrieving requirement traceability link.

IV. REQUIREMENT TRACEABILITY RETRIEVAL PROBLEM

A. Problem in Requirement Traceability

Let us consider the problem with some assumption. For example, in a banking project in version 1 the developer mainly concentrated on “balance enquiry module” and let it uncompleted. .

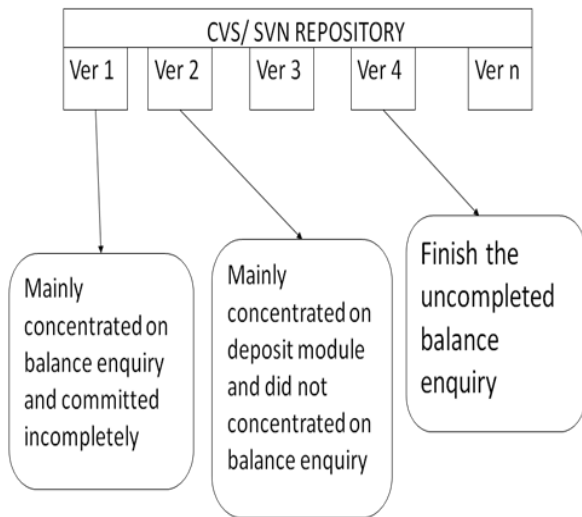


Fig. 2. Problem in Tracing Requirements

Let us consider the problem with some assumption. For example, in a banking project in version 1 the developer mainly concentrated on “balance enquiry module” and let it uncompleted. In version 2 , the developer mainly concentrated on “ deposit module” and did not concentrated on balance enquiry. In version 4, they finished the “balance enquiry module” which was incomplete in version 1. Likewise, in some version of cvs/svn repository developer took some requirement and kept which is in completed but in future the developer complete those requirement left incomplete in future version. Here the solution needed is to be trace the particular cvs/svn version that contains the completed requirement’s source code.

B. Ranking Process of RTL

The ranking process is required to give suggestion to the developer for to find automatically which version contains the source code of fulfilled requirements. Since the accuracy of textual similarity is less, ranking is used for second, third options to satisfy the develop need of requirement traceability. The ranking process is performed by maintaining the requirement by maintaining the requirement traceability link (RTL).

Requirement traceability link is a set contains the requirement, source code and its similarity value as elements. Here for illustration, version1 having requirement traceability link as { R1, S1,SV=6% } and

version 2 {R1, S1, SV=9%} and version 3 {R1, S1, SV=5%}. While comparing the three versions version 1 is ranked in second position, version 2 is in first position and version 3 ranked as third position.

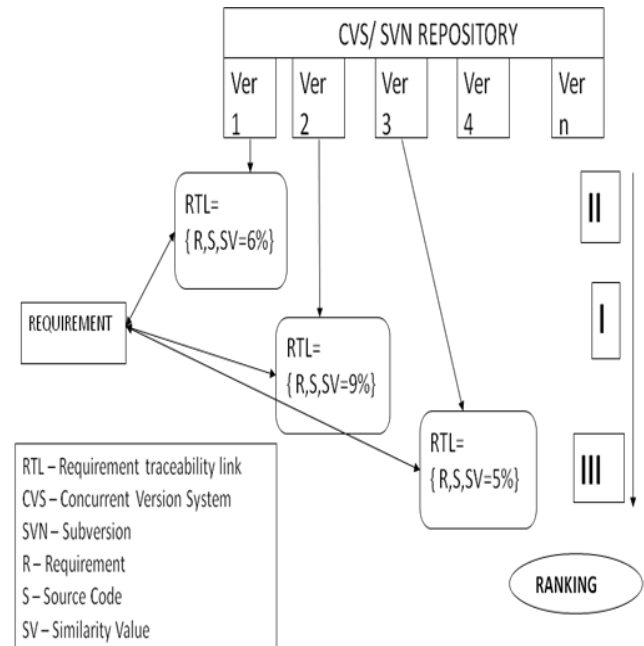


Fig. 3. Ranking Process of RTL

This ranking gives the suggestion of particular requirement traced to find its full-fledged or completed source code version to frame the final output software and also in the development phase of software engineering.

C. Using Bug Tracking System

Since the textual similarity is not more accurately produce the best matched requirement traceability link, one have to move to some other means. So, the Bug tracking system is taken to improve the already made requirement traceability link with textual similarity measure.

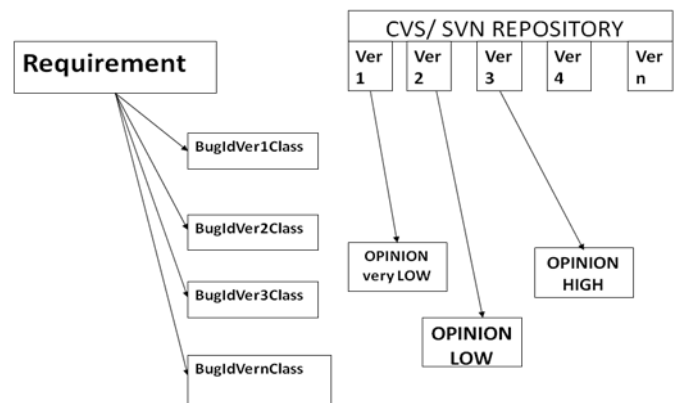


Fig. 4. Using Bug Tracking System

The developer’s updating activity may be tracking the different modules’ bug. This bug is identified by bug tracking IDs. For illustration for a particular requirement’s coding in different version, we may count

in which version; the bug post is found more. We may ReLink the textual similarity based ranking to this bug count based ranking which may future improve the accuracy.

D. Using Forum System

In forum, individual messages are organized by topic and discussion title. A discussion (also known as a "thread") is a series of messages along a specific topic of conversation. Each discussion begins with a comment or question that elicits feedback from others. By reading each post—that is, each individual message—in a discussion, one can see how it has evolved.

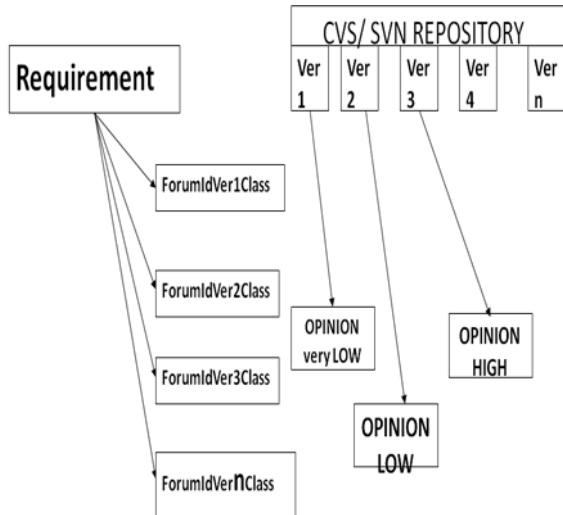


Fig. 5. Using Forum System

By using forum as an expert system one can enhance the trust of the baseline requirement traceability links.

E. Necessity of ReLink

The relink or reranking is required to improve the requirement traceability link produced with textual similarity between the source code and requirement.

TABLE1. RELINKUBG PROCESS

Rank By Textual Similarity	CVS / SVN	Relink
II	SOURCE CODE VERSION 1	III
I	SOURCE CODE VERSION 2	II
III	SOURCE CODE VERSION 3	I

This relink is performed by reranking the already ranked links. Based on the bug count the importance of particular version's rate is increased and it is made possible to relink the requirement traceability link.

V.CONCLUSION

In finding or tracing the link between requirement and the source code automatically is the aim. But the accuracy differs from different methods. The efficiency of

requirement traceability link is measured by the well known measure like precision, recall and fscore; through these measures the efficiency of the information retrieval techniques for the requirement traceability can be found. This technique of relink is used by the authors in the reference paper given below and used the same here for this technique that serves different need of requirement traceability.

REFERENCES

- [1] N. Ali, Y.-G. Gue'he'neuc, and G. Antoniol, "Trustrace: Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links" IEEE Trans. Software Eng., vol. 39, no. 5, pp. 725-741, May 2013.
- [2] O. C. Z. Gotel and C. W. Finkelstein, "An analysis of the requirements traceability problem," Requirements Engineering., Proceedings of the First International Conference on, pp. 94-101, April 1994.
- [3] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison-Wesley, 1999.
- [4] N. Ali, Y.-G. Gu'eh'eneuc, and G. Antoniol, "Trust-based requirements traceability," in Proceedings of the 19th International Conference on Program Comprehension, S. E. Sim and F. Ricca, Eds. IEEE Computer Society Press, 10 pages, June 2011.
- [5] R. Berg and J. M. L. Van, "Finding symbolons for cyberspace: addressing the issues of trust in electronic commerce," Production Planning and Control, vol. 12, pp. 514-524(11), 2001.
- [6] D. H. McKnight, V. C., and C. K., "The impact of initial consumer trust on intentions to transact with a web site: a trust building model," The Journal of Strategic Information Systems, vol. 11, no. 3-4, pp. 297 - 323, 2002.
- [7] J. W. Palmer, J. P. Bailey, and S. Faraj, "The role of intermediaries in the development of trust on the www: The use and prominence of trusted third parties and privacy statements," Journal of Computer-Mediated Communication, vol. 5, no. 3, 2000.
- [8] K. M. and H. William, "The development of initial trust in an online company by new customers," Information & Management, vol. 41, no. 3, pp. 377 - 397, 2004.
- [9] B. Ramesh and V. Dhar, "Supporting Systems Development Using Knowledge Captured During Requirements Engineering," IEEE Trans. Software Eng., vol. 9, no. 2, pp. 498-510, June 1992.
- [10] F.A.C. Pinhero and J.A. Goguen, "An Object-Oriented Tool for Tracing Requirements," IEEE Software, vol. 13, no. 2, pp. 52-64, Mar. 1996.
- [11] J. Konclin and M. Bergen, "Gibis: A Hypertext Tool for Exploratory Policy Discussion," ACM Trans. Office Information Systems, vol. 6, no. 4, pp. 303-331, Oct. 1988.
- [12] Belkin, N. & Croft, B. (1992) "Information Filtering and Information Retrieval: Two Sides of the Same Coin," Communication of the ACM, Dec., 1992.
- [13] http://comminfo.rutgers.edu/~aspoerri/InfoCrystal/Ch_2.html
- [14] Salton, G. & McGill, M. (1983) Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
- [15] G. Antoniol, G. Canfora, G. Casazza, A.D. Lucia, and E. Merlo, "Recovering Traceability Links between Code and Documentation," IEEE Trans. Software Eng., vol. 28, no. 10, pp. 970-983, Oct. 2002.
- [16] D. Artza and Y. Gil, "A Survey of Trust in Computer Science and the Semantic Web," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, no. 2, pp. 58-71, 2007.
- [17] H. Kagdi, S. Yusuf, and J.I. Maletic, "Mining Sequences of Changed-Files from Version Histories," Proc. Int'l Workshop Mining Software Repositories, pp. 47-53, 2006.
- [18] H. Kagdi and J. Maletic, "Software Repositories: A Source for Traceability Links," Proc. Int'l Workshop Traceability in Emerging Forms of Software Eng., pp. 32 -39, Mar. 2007.
- [19] P. Mader, O. Gotel, and I. Philippow, "Enabling Automated Traceability Maintenance by Recognizing Development Activities Applied to Models," Proc. 23rd IEEE/ACM Int'l Conf. Automated Software Eng., pp. 49-58, 2008.

- [20] A.D. Lucia, M.D. Penta, and R. Oliveto, "Improving Source Code Lexicon via Traceability and Information Retrieval," *IEEE Trans. Software Eng.*, vol. 37, no. 2, pp. 205-227, Mar.2011.
- [21] X. Zou, R. Settini, and J. Cleland-Huang, "Improving Automated Requirements Trace Retrieval: A Study of Term-Based Enhancement Methods," *Empirical Software Eng.*, vol. 15, no. 2, pp. 119-146, 2010.
- [22] A.D. Lucia, F. Fasano, R. Oliveto, and G. Tortora, "Recovering Traceability Links in Software Artifact Management Systems Using Information Retrieval Methods," *ACM Trans. Software Eng. Methodology*, vol. 16, no. 4, article 13, 2007.
- [23] W.B. Frakes and R. Baeza-Yates, *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, 1992.
- [24] N. Ali, Y.-G. Gue'he'neuc, and G. Antoniol, "Requirements Traceability for Object Oriented Systems by Partitioning Source Code," *Proc. 18th Working Conf. Reverse Eng.*, pp. 45-54, Oct. 2011.
- [25] A. Marcus and J.I. Maletic, "Recovering Documentation-to-Source-Code Traceability Links Using Latent Semantic Indexing," *Proc. Int'l Conf. Software Eng.*, pp. 125-135, May 2003.
- [26] J.H. Hayes, G. Antoniol, and Y.-G. Gue'he'neuc, "PREREQIR: Recovering Pre-Requirements via Cluster Analysis," *Proc. 15th Working Conf. Reverse Eng.*, pp. 165-174, Oct. 2008.
- [27] D. Poshyvanyk, Y.-G. Gue'he'neuc, A. Marcus, G. Antoniol, and V. Rajlich, "Feature Location Using Probabilistic Ranking of Methods Based on Execution Scenarios and Information Retrieval," *IEEE Trans. Software Eng.*, vol. 33, no. 6, pp. 420-432, June 2007.
- [28] M. Gethers, R. Oliveto, D. Poshyvanyk, and A.D. Lucia, "On Integrating Orthogonal Information Retrieval Methods to Improve Traceability Recovery," *Proc. 27th IEEE Int'l Conf. Software Maintenance*, pp. 133-142, Sept. 2011.
- [29] N. Ali, Y.-G. Gue'he'neuc, and G. Antoniol, *Factors Impacting the Inputs of Traceability Recovery Approaches*, A. Zisman, J. Cleland-Huang, and O. Gotel, eds. Springer-Verlag, 2011.
- [30] J.H. Hayes, A. Dekhtyar, S.K. Sundaram, and S. Howard, "Helping Analysts Trace Requirements: An Objective Look," *Proc. 12th IEEE Int'l Requirements Eng. Conf.*, pp. 249-259, 2004.