

Establishing a Traceability Links Between The Source Code And Requirement Analysis, A Survey on Traceability

Prashant N. Khetade, Vinod V.Nayyar

Department of Computer Science & Engg.AGPCE , RTM,Nagpur University,Nagpur,India, faculty
Computer Science ,AGPCE,RTM Nagpur University,Nagpur,India,
pkhetade@yahoo.co.in,vinodnayyar12@yahoo.com

ABSTRACT:*In the system and source code development, we develop a source code and documentation is mainly in natural language. The Continuous and frequent development require proper requirements change management Traceability is essential for management of change and analysis of its impact. This research paper presents a technique in the domain of traceability we believe that the application-domain knowledge that programmer's process when writing the code is often captured by the mnemonics for identifiers, the analysis of these mnemonics can help to associate high-level concepts with program concepts and vice-versa. . We propose a method based on information retrieval for traceability links between source code and free text documents. Here we use information retrieval techniques for establishing a links between source code and requirement, documentation and latent semantic indexing, is used to automatically identify traceability links from system code.*

Keywords:*Traceability, Requirement, Requirement Traceability techniques, Traceability Links, Information Retrieval.*

I. INTRODUCTION

Traceability is the most important for software project and when we use this there are much benefits for the development. When we develop a source code for any system then that source code is traceable and identical with the requirement and analysis because we develop a source code as per the requirement. A traceability links is formed between the source code and requirement. For the proper execution of the system. Requirement traceability helps the software engineers to trace a requirement from its emergence to its fulfillment. Traceability would not help us knowing how different components of systems are dependent on each other in the system. We will also fail to find the impact of change to the software and system, a major goal of traceability, in absence of Origin- Requirements and Other Artifacts-Other Artifacts traceability links. Therefore we argue that a definition of traceability must all the aspects of traceability in terms of scope and coverage.

The importance and benefits of recovering and managing traceability links between software artifacts during software evolution has been long establish in the research community and industry. Most such methods consist of several key steps organized in a pipeline architecture, where the output from each step constitutes the input for the next one.

II. TYPES OF THE TRACEABILITY LINKS:

Here in the system execution there are four types of traceability links which are as follows

1. Forward from Requirement:

This is the type of the traceability links where here the components are built from the requirements. In this there is change from high level requirement to the low level requirement.

To trace requirements in this way also allows designers to package requirements into blocks that target a particular module or section of the system that needs to be built, based on that block of requirements. The testing team can also create test cases that directly relate to the requirements, and when this is complete, a person can trace a high-level requirement down to the actual test case, and verify that the test will properly validate the requirement.

2.Backward to Requirement:

In this type of requirement the tracing is done into the backward direction. The design elements are built in backward. Starting from the low level requirement and tracing backwards also assists unit testers with their testing, so they can see what part their module plays in the big picture of the system, based on requirements. While testing, if a defect is found, and the low level requirement is known as the failure point, it can be determined what high-level requirement would then not be satisfied if it were not repaired.

3. Forward to Requirement:

In this type of the requirement we go for the tracing of the documents as per in the higher that is forward to the requirement. This is important regarding the situation where a “what if” analysis is needed, to determine costs and scope of a high-level requirement changing. If one does change, there is enough information to determine the order of magnitude of the overall change needed to the system.

4 .Backward from Requirement:

In this type the traceability is done into the Backward direction that is from the requirement to source code in backward direction. Besides the above types of Requirement there are two types of the requirement traceability these are Pre-RS traceability and Post- RS traceability explain as follows in the diagram.

Traceability Definition

Traceability definitions on the basis of these terms.

A. Scope

We define scope as a boundary for traceability. There are two options for the scope which are widely seen in the survey software , and system . System boundaries incorporate software and software requirements are derived from system requirements

B. Coverage

Coverage is defined as the extent of directional traceability. We have defined four types of coverage in traceability:

This coverage type provides the trace between different forms of requirements that is, requirements specification, design, implementations, test cases, etc . The concept of coverage is more like a dependent concept upon scope. For example, if the scope of traceability is set to Software level, it will automatically restrict the Origin -Requirements traceability to some extent.

C. Analysis

The analysis has defined traceability as “the ability to relate requirements specifications with other artifacts created in the development life-cycle of a software system”.

We, in this section, critically evaluate the existing traceability techniques on the basis of usability of techniques criteria. The parameters defined, including scope and coverage, consider the usability of existing techniques. The following are the evaluation parameters and their details. The link representation technique is of two types. The links that represent item must be trace.

III. INFORMATION RETRIEVAL BASED TRACEABILITY:

In the Information Based-based traceability recovery approaches produce ranked lists of traceability links between pieces of documentation and source code. This is purely based on the information Retrieval technique In this technique we proposed a model named Trustrace. Trustrace uses any traceability recovery approach to obtain a set of traceability links, which rankings are then re-evaluated using a set of other traceability recovery approaches. We propose a novel traceability recovery approach, Histrace, to identify traceability links between requirements and source code through CVS/SVN change logs using a Vector Space Model (VSM).

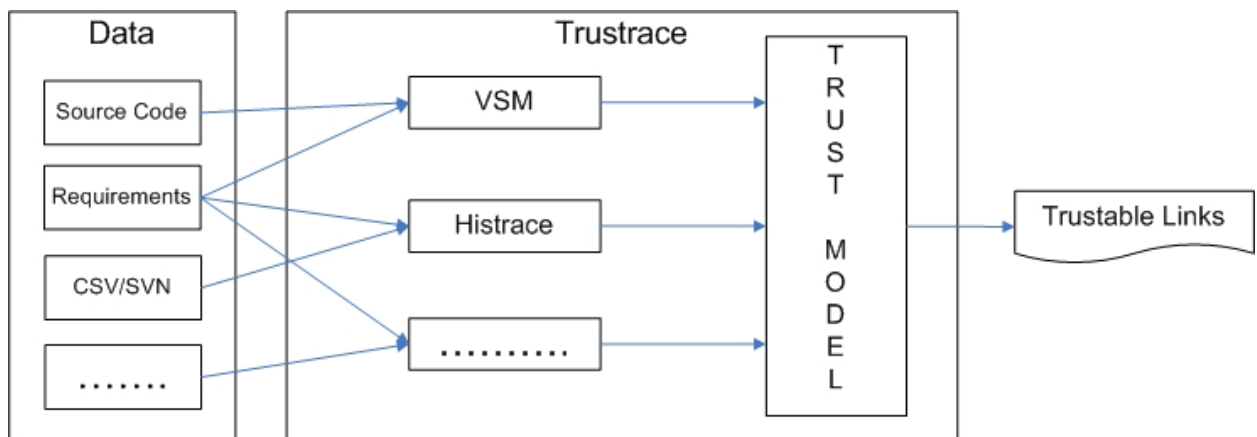
Traceability recovery approaches assume the availability of high-level free text documentation and infer links between pieces of this documentation and source code entities using an indexing process producing a ranked list of traceability links In the In Information Retrieval method we use Vector Method and the Shenon Method for IR.

We propose two contributions to improve the precision and recall of traceability links. First, we propose a novel approach, Trustrace, inspired by Web trust models [1] to improve precision and recall of traceability links: Trustrace uses any traceability recovery approach as the basis on which it applies various experts' opinions to remove and/or adjust the rankings of the traceability links. The experts can be human experts or other traceability recovery approaches.

IV. TRUSTTRACE BASED MODEL:

We describe a trustrace model as follows:This trustrace based model consist of the following parameters

As Source Code,Requirement,CSV/SVN messages,Trust Model.Trustable links.



1.Source Code:

That is source code of the system. It contain all the data and attributes about the source code.

2.Requirement:

It is the requirement of system define by the client which contain a requirement analysis of whole system.

3.CVS?SVN:

These are the commit messages and histrace bugs uses bug report to built traceability links between requirement and source code.

4.Histrace:

Histrace is a technique to create experts supporting the identification of traceability links between requirements and source code.

Second, we show how Trustrace improves traceability recovery accuracy using Histrace, an expert supporting the identification of traceability links between requirements and source code through CVS/SVN change logs and a Vector Space Model (VSM). In any organization developing software, there are many heterogeneous sources of information available, including CVS/SVN repositories, bug-tracking systems, mailing lists, forums, and blogs. Histrace uses CVS/SVN change logs to build Traceability links between high-level documentation and source code entities, observing that log messages are tied to changed entities and, thus, can be used to infer traceability links.[2].

V. CONCLUSIONS

According to the literature Traceability is important. The Information Retrieval (IR) approach is most useful for the establishing the traceability links and also for recovering the traceability links. We found that there are many barriers to a company implementing and using traceability links these are to be overcome by using the Information Retrieval approach which uses a Trustrace based model up to some extent. It can be further improved to some extent with solutions such as: value based requirements tracing; industry/organisational policies; customised tools; using different approaches and techniques including a semi-automated approach, VSM approach in further implementation and in improving the accuracy. Hence from the above description of the IR approach and the Trustrace Based Model. It is conclude that the Information retrieval is the better technique in terms of the establishing the traceable links between the Source code and requirement analysis.

VI. ACKNOWLEDGEMENTS

Much of this work was carried out by the principle author Prashant Khetade, PG student of computer Science & Engineering, Abha college of Engineering, RTM Nagpur University, with my co-author Vinod V. Nayyar, faculty of computer Science & Engineering, Abha college of Engineering, RTM Nagpur University, Nagpur, India. This work is done by referring the various papers and through study.

REFERENCES

- [1] Glean S.Stout "Requirement Traceability" Research paper on Traceability publish in spring cluster 2001.
- [2] N. Ali, Y.-G. Gue'he'neuc, and G. Antoniol, "Trust-Based Requirements Traceability," Proc. 19th IEEE Int'l Conf. Program Comprehension, S.E. Sim and F. Ricca, eds., pp. 111-120, June 2011.
- [3] 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.
- [4] Marcus and J.I. Maletic, "Recovering Documentation-to- Source-Code Traceability Links Using Latent Semantic Indexing," Proc. 25th Int'l Conf. Software Eng., pp. 125-135, 2003
- [5] J.H. Hayes, A. Dekhtyar, S.K. Sundaram, and S. Howard, "Helping Analysts Trace Requirements: An Objective Look," O.C.Z. Gotel and C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem," Proc. First Int'l Conf. Proc. 12th IEEE Int'l Requirements Eng. Conf., pp. 249-259, 2004.
- [6] J.I. Maletic and M.L. Collard, "TQL: A Query Language to Support Traceability," Proc. ICSE Workshop Traceability in Emerging Forms of Software Eng., pp. 16-20, 2009
- [7] R. Wu, H. Zhang, S. Kim, and S. Cheung, "Relink: Recovering Links between Bugs and Changes," Proc. 19th ACM SIGSOFT Symp. and 13th European Conf. Foundations of Software Eng., pp. 15-25, 2011
- [8] 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.
- [9] M. Eaddy, T. Zimmermann, K.D. Sherwood, V. Garg, G.C. Murphy, N. Nagappan, and A.V. Aho, "Do Crosscutting Concerns Cause Defects?" IEEE Trans. Software Eng., vol. 34, no. 4, pp. 497-515, July/Aug. 2008.
- [10] De Lucia, M. Di Penta, R. Oliveto, A. Panichella, and S.Panichella, "Improving IR-Based Traceability Recovery Using Smoothing Filters," Proc. 19th IEEE Int'l Conf. Program Comprehension, pp. 21-30, June 2011.

- [11] Jaoul paul, Maria Eocob “Requirement Traceability in the Model Driven Approach Applying Model and Transformation Confermence. July 2008
- [12] Richard Taokar, Kashif Kamran “Requirement Traceability the State Of Art Systematic Review” in aug 2009
- [13] Nasir Ali “Analysis Source Code Structure And Mining Software Repositories to Create Requirement Traceability Links” Montreal in December 2012.
- [14] Huzefa Kadgi, Maletic, Bonita Sarif “Mining Software Repositories for Traceable Links” in aug 2009.
- [15] Rocco Olivato, Andrea Dee Lucia “On the Equivalence of Information Retrieval Method for Automated Traceability Link Recovery” in 2010