ISSN: 0976-3104
SPECIAL ISSUE: (EMERGING TECHNOLOGIES IN NETWORKING AND SECURITY (ETNS)
Suseela and Devi

THE IIOAB JOURNAL

ARTICLE OPEN ACCESS

IMPROVING THE ACCURACY OF IR TECHNIQUES USING TRACEABILITY

MB. Suseela, PP. Devi
Department Of Computer Science and Engineering, Vel Tech Multi tech Dr. RR and Dr.SR Engineering College, INDIA

ABSTRACT

Traceability links between the requirements of a system and its source code are helpful in reducing system comprehension effort. During software maintenance and evolution, requirement traceability links become obsolete because developers do not/cannot devote effort to updating them. Traceability is a sub discipline of requirements management within software development and systems engineering. Consequently, the literature has proposed methods, techniques, and tools to recover these traceability links mechanically or semi automatically. In proposed system the information retrieval technique can automatically recover the traceability links between specified requirements and source code. However, IR techniques lack accuracy. In this paper, we show that mining software repositories and combining mined results with IR techniques can improve the accuracy of IR techniques and we propose a dynamic specification mining technique which used to find the traceability links and also will detect the bugs in the source code.

Published on: 2nd -December-2016

KEY WORDS

Traceability links,Requirement traceability, Information retrieval, dynamic specification mining

***Corresponding author: Email:** suseelambveltech@gmail.com

INTRODUCTION

Traceability is the ability to verify the old process, location, or application of an item by means of documented recorded identification. [1] It is one of important criteria in software engineering quality factors in which it describes and follows the life of a software in term of modelling the relations between software artifacts explicitly. Requirements traceability text file the life of a requirement, tracks every change and links its relationships with other items within a project [2]. Traditionally, a manual traceability matrix connects high-level requirements with design, test plan and test cases. Traceability links have been used to connect requirements gathered resulting from the initial requirements. However, traceability links can be created between any items, be it an artifact, folder or module, in any version [2]. Traceability links can be used for various additional features in a project's development. Trustrace consists of three parts [5]:

1. Histrace stores all the required links between requirements and software repositories it also mines software repositories to create links.
2. Trumo compares the duality of the recovered links provided by the experts and also combines the requirements traceability links from that trumo will ranks/discards.
3. DynWing analyses eavh expert's similarity value for each links and assign values also for each links
We thus show [5] that our trustbased approach indeed improves precision and recall and also that CVS/SVN change logs are useful in the traceability recovery process.

An dynamic specification mining technique is used to recover the requirement traceability and also to detect the bugs in the source code. To control the fault in source code those fault report are been thrown to the developer mail id for future references.

The remaining paper has been followed as. Relatedfocus on the literature review, focus on implementation, focus on the modules, focus on the result and finally focus on conclusion.

THE IIOAB JOURNAL
ISSN: 0976-3104

LITERATURE REVIEW

Trustrace: Trust based traceability

Trustrace uses software repositories e.g. bug tracking system, as experts to trust more or less some base line links recovered by an IR technique.

Definition

In trustrace, we symbolize a traceability links as a triple {source document, target document, and similarity}. In **[Figure- 1]** let R(r1, …rn) be set of requirements or high level document, C(c1….cm) be a set of implementing classes. Let T(t1,..tp) be collections of set where each set Ti is a set of homogeneous piece of information.[4],[5]

Then let us assume that for each Ti €T possible to define function δ_{Ti} consequently, we define R2C as the set of traceability links rebound between R and C and, for each set Ti € T, a set R2CTi generated by each piece of information tk: [4], [5] $R2CT_i(rj, t_k)= \{(r_j, c_{ss},\sigma_{jk})|Cs € \delta Ti(tk) \& tk € Ti\}$

Model

In this trust race use the following equation[4]. Trustrace uses the set of candidate links l_r (r_j, c_s, σ j,s) with j € [1…..N] and s € [1….l] also uses the sets of candidate links l_{rt} =(r_j, t_k, σ' j,k,) j € [1… N] and k € [1,… N_i] generated from set of some other pieces of information T_i and for requirement r_j € R. Indeed, for each set T_i € T, Trustrace builds a trustable links T_{ri} as follows T_{ri} ={(r_j, c_s,σ $_{j,s}$) | ∋ tk € Ti : (r_j, c_s) € α ($R2CT_{i,rj,\ tk}$) & (r_j, c_s) € α (R2C)}

Finally Trustrace merges the trust levels of each T_{ri} into global level of trustworthiness:[5]

Histrace

Histrace produce links between the set of requirements, R and the source code C, using the software repositories Ti. Histrace consider the requirements, textual descriptions, CVS/SVN commit messages, bug reports, and classes as separate documents. It also uses the information from two experts. A bug report can also be produced to improve the accuracy of the IR techniques. Histrace mines the software repositories.

Document Pre processing

Depending on the input information source we perform specific pre-processing steps to remove irrelevant details, split identifiers, and, finally, normalize the text using stop words removal and stemming.

Requirements and source code

Histrace first processes source code files to extract all the identifiers and comments in each class [4] [5].
Histrace then performs the Following steps to normalize source code documents and requirements: (i) compute all upper case letters into lower case and remove punctuation: (ii) remove all stop words.

CVS/SVN commits

To build To, histrace extracts CVS/SVN commits and discards those that (i) are tagged as "delete" (ii) does not concern source code (iii) have messages of length shorter or equal to two words [4] [5].

Bug reports

To create histrace bug, histrace extracts all the bug reports from a bug tracking system. Usually, bug reports do not contain declared information about the source code files that developers updated to fix a bug [5]. Histrace use regular expression, i.e, a simple text machine approach but with reasonable results to link CVS/SVN commit messages to bug reports.

Trumo

Trumo assumes that different experts know useful information to discard or rerank the traceability links between two documents, e.g., requirements and source code classes. By the definitions, section 2.1.1 rj is a requirement with rj € R, and class cs € $\delta_{Ti}(t_k)$ [5], [6]

Dynwing

Dynamic weighting technique automatically decides the weights. Choosing a right weight per link is a problem that we articulate as a more problem. Essentially we have different experts, i.e, CVS/SVN commits, bug reports, and others, to trust a link. By maximizing the similarity value $\psi r_j,c_s(Tr)$. Dynwing automatically identifies the experts that are most trust worthy and those that are less trustworthy. [5]

$\lambda_1(r_j,c_s)..max..\lambda_{p+1}(r_j,c_s)\{\psi r_j,c_s(\ Tr)\}$
With the following constraints: [5]
$0 \leq \lambda_i(r_j,c_s) \leq 1,i=1,\ldots,P+1\ \lambda_1(r_j,c_s) + \lambda_2(r_j,c_s)+\ldots+ \lambda_{p+1}(r_j,c_s) = \lambda_{k1}(r_j,c_s) \geq \lambda_{k2}(r_j,c_s) \geq\ldots\geq \lambda_{kp+1}(r_j,c_s).$

Therefore, developer may further constraint [5] by imposing
λ commits (rj,cs)≥ λ bugs (rj,cs) > 0
To mechanically decide the weights $\lambda_i(r_j,c_s)$ for each expert.

Topic modeling

Topic modeling is a widely-used machine learning technique for automatically [3] inferring semantic topics from a text corpus.

Latent dirichlet allocation

In LSI (latent semantic indexing), each document in the corpus is represented as a word count vector of length W, where W is the number of words in the principal vocabulary. When the vectors of all D documents are placed side by side, one obtain a W x D matrix of counts.

It is possible to learn an LDA model in near real time on a moderately-sized set of documents. The algorithm consists of iteratively performing variational updates in a systematic scan over tokens [3],
q(Znd=t)αNwd¬nd+β÷Nt¬nd+¬nd) [3]
Wβ(Ntd
Nwd¬nd and Ntd¬nd are expected counts derived from q(z).

Current support for requiremnets traceability

It has been noted that most tools do not cover RT.[6] They differ mainly in cosmetics, and in time, effort, and manual intervention they require to achieve RT.

Basic techniques

Various techniques used for providing RT, are as follows: cross referencing schemes, key phrase dependencies, templates, RT matrices, matrix sequences, hypertext, consolidation documents, assumption-based truth maintenance networks and constraint networks.
Moreover, some form of RT can result from using certain languages, models, and methods for development.

Automated support

More commercial tools and research products support RT, we high spot some representative examples: they are as follows [6]

General- Purpose tools: These incorporate; hypertext editors, word processors, spread sheets, database systems and even more. They can be hand-configured to allow previously manual and paper-based RT tasks to be carried out on-line.

Special-Purpose Tools: foundation dedicated activities related to RE and some achieve restricted RT.
Workbenches: It limits a collection of the above to support coherent sets of activities. Less restricted RT can be achieved.

Environment: RT can be acclaimed by the basis of integration; a common language, a common structure, a common method or a specialized RT tool or repository structure.

IMPLEMENTATION

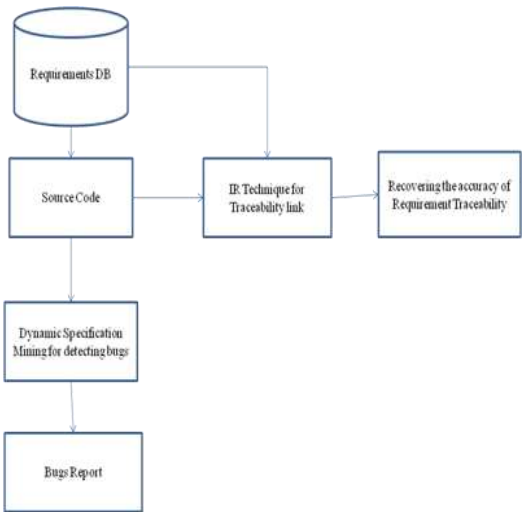

Fig: 1. Work flow diagram

Requirements are stored in a database and then a link is created between the source code and the specified requirements. Then a bugs can be detected from the source code and that bugs are been thrown to the developer mail id.

Requirement traceability

The requirements and bug report from the user is given a link and that link is given to the source code if any requirement is missing by doing the traceability method the missed requirements will be specified.

Bug detection

The fault that are been present in the code will be detected and the misspell synax word can be found in this step.

Recovery process

In this process the words that are misspelled will be recovered automatically by clicking the recover process.

Accuracy of traceability links

The accuracy of the traceability links between requirements and source code recovered by an IR technique are improved by
1. Mining software repositories
2. Trust model
3. Weighting the experts' opinions

SMTP mail server

THE IIOAB JOURNAL
ISSN: 0976-3104

Simple mail transfer protocol is used to send the error report to the developer mail id.

Expected Outcome

Fig: 2. Code Correction Report

The source code fault will be detected in this process and that report will be send to the developer mail id.

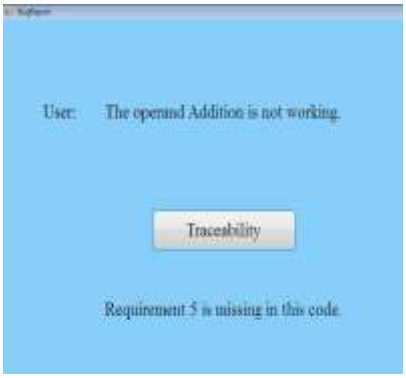

Fig: 3. Requirement Traceability

The requirement missing will be specified in this screen shot.

CONCULSION

Requirement tracing is inevitable. The concepts of requirements tracing is quite simple: to follow relationships or links. Requirement traceability concern to the ability to describe and follow the life of requirements in both forwards and backwards direction. In this paper, detailed explanation given about various information retrieval techniques. First we proposed Trust race, inspired by web trust models to improve precisions and recall of traceability links: trust race use any traceability recovery approach as a basis on which it applies various expert's opinion. Second we proposed Histrace an expert supporting the identification of traceability links between requirements and source code. Third we proposed Trumo, inspired by web trust models improves the precision and recall values of some baseline traceability links. Fourth we proposed Dynwing combines and assigns weights to Histrace experts' opinion using a dynamic weighting technique. Fifth we proposed Topic modelling is used for automatic machine learning technique.

FUTURE ENCHANCEMENT

In future work, we plan more experiments with other combinations of IR-based approaches to future improve the precisions and recall values. We will also perform more experiments on heterogeneous software arti facts to measure the usefulness of these other arti facts for a traceability recovery process. In particular, we are currently

building new traceability approaches using BCR techniques (binary class relationships) this will improve the precision and recall values of the traceability links.

CONFLICT OF INTEREST

The authors declare no conflict of interests.

ACKNOWLEDGEMENT

None

FINANCIAL DISCLOSURE

None

REFERENCES

[1] Nuv Adila A, Rodziah A. [2012] Traceability Method for Software Engineering Documentation IJSC international journal of computer science issues. 9(2):2.

[2] Ale Ksandra K, Kalman G, Andy S, Ralf S. [2011] Traceability Link Evolution with Version Control in: IEvolution are software- und system entwickleny- method und erfahurgen workshop I Am Rahmen der Konferenz SE2011.

[3] Asuncion H, Asuncion A, Taylor R. [2010] software Traceability with Topic modeling proc.32ndACM/IEEE int'l conf. Software eng. 1:95-104.

[4] Ali N, Gueheneuc YG, Antoniol G. [2011] Trust-Based Requirements traceability, Proc. 19th IEEE Int'l Conf. Program Comrehension, S.E. Sim and F.Ricca, eds. 111-120.

[5] Nasir A. [2013] Student Member, IEE, Yann-Gael Gueheneuc, Senior Member, IEEE, and Giuliano Antoniol, Member, IEEE, Tustrace: Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links, of Software Engineering. 39(5).

[6] Gotel OCZ, Finkelstein CW. [1994] An Analysis of the Requirements traceability Problem, Proc. First Int'l Conf. Requirements Eng. A. Marcus and J.I. Maletic. 94-101.

[7] [2003] Recovering Documentation-to source-Code Traceability Links Using Latent Semantic Indexing, proc. 25th Int'l Conf. software Eng. 125-135.

[8] McKnight DH, Choudhury V, Kacmar CK. [2002] The Impact of Initial Consumer Trust on Intentions to Transact with a Web Site: A Trust Building Model, The J. Strategic Information Systems. 11(¾):297-323.

[9] Palmer JW, Bailey JP, Faraj S. [2000] The Role of Intermediaries in the Development of Trust on the WWW: The Use and Prominence of Trusted Third Parties and Privacy Statements, J. Computer-Mediated Comm. 5(3).