LUND UNIVERSITY

In vivo evaluation of large-scale IR-based traceability recovery

Borg, Markus

Link to publication

Total number of authors:
1

In Vivo Evaluation of Large-scale IR-based Traceability Recovery

Markus Borg

Department of Computer Science
Lund University
Lund, Sweden
markus.borg@cs.lth.se

Abstract—**Modern large-scale software development is a complex undertaking and coordinating various processes is crucial to achieve efficiency. The alignment between requirements and test activities is one very important aspect. Production and maintenance of software result in an ever-increasing amount of information. To be able to work efficiently under such circumstances, navigation in all available data needs support. Maintaining traceability links between software artifacts is one approach to structure the information space and support this challenge. Many researchers have proposed traceability recovery by applying information retrieval (IR) methods, utilizing the fact that artifacts often have textual content in natural language. Case studies have showed promising results, but no large-scale in vivo evaluations have been made. Currently, there is a trend among our industrial partners to move to a specific new software engineering tool. Their aim is to collect different pieces of information in one system. Our ambition is to develop an IR-based traceability recovery plug-in to this tool. From this position, right in the middle of a real industrial setting, many interesting observations could be made. This would allow a unique evaluation of the usefulness of the IR-based approach.**

Keywords-**requirements-test alignment; traceability; information retrieval; empirical software engineering.**

I. INTRODUCTION

In large-scale software development, coordination between different organizational units is a key success factor to develop high-quality products on time and within budget. Especially the alignment between the requirements and verification processes is a critical aspect [23]. Software development results in a myriad of textual information; requirements and design specifications at various abstraction levels and test descriptions, test results and defect reports to mention some. Developing techniques to navigate all this growing information is crucial.

One approach to structure the information space is to maintain traceability links. This is widely recognized as an important factor for efficient development, as it supports tasks, such as verification, change impact analysis, program comprehension, and software reuse [3]. Lack of traceability has been identified as one of the top factors causing delays in software engineering projects [6].

Textual content in natural language (NL) is the common form of information representation among software artifacts and also source code has NL content in identifiers and comments [20]. This enables information retrieval (IR) approaches to be applied. IR-based traceability recovery has been the focus of many researchers, enabling semi-automated tracing between artifacts based on textual similarity.

A trend in this research has been to benchmark methods using the established metrics recall and precision. The race for good values in traceability recovery has resulted in studies using different IR-methods such as the *vector space model* (VSM), *probabilistic models* and *latent semantic indexing* (LSI). To support comparisons, there has been work done on enabling benchmarking of tools. The *Center of Excellence for Traceability*[1] has made five datasets containing artifacts and correct links publicly available, containing between 68 and 455 artifacts and between 41 and 1005 correct links. It is however common in case studies to base evaluations on artifacts from the university environment. Recently, case studies have been conducted using proprietary data from the industry [1], [16], but they are still in minority.

The primary goal of our research is not to study how IR methods can be improved and configured to perform better in an industrial setting, but rather to evaluate the IR-based approach in general and study how developers can benefit from the additional traceability information in an alignment context. De Lucia et al. conducted a controlled experiment where the usefulness of IR-based traceability methods was investigated with student subjects [18]. By conducting similar evaluations in a real company conducting large-scale software development, many unanswered questions could be studied with a real-world validity.

This leads us to the following research questions:

RQ1 What is state-of-the-art in IR-based traceability recovery?

RQ2 How much can additional traceability information support alignment between requirements and test activities and how do developers work with and benefit from access to additional traceability information?

RQ3 How do existing IR-approaches to traceability recovery perform in a large-scale industrial setting?

The first research question will be studied by a systematic

---

[1]COEST, http://www.traceabilitycenter.org/

mapping study, described in section III-B. In our region, we know that a number of companies are in a transition to use Quality Center (QC) from Hewlett Packard[2] to collect requirements, test case descriptions and issue tracking in one system. This tool can be extended by an IR-based traceability recovery plug-in. Its possible benefits, and the second research question, will be explored by case studies presented in section III-A and IV-B. Finally, the last research question will be addressed by a master thesis project described in section III-C and the in vivo case study in section IV-B.

II. SOLUTION IDEA AND RELATED WORK

The last decade, many researchers have applied information retrieval to the task of traceability recovery. There is also work on comparing how methods perform when data scales up.

A. IR-based traceability recovery

Fiutem and Antoniol did early work on recovering traceability links between design and source code in 1998. They used basic string comparisons and edit distances to find links between design and code, both expressed in Abstract Object Language (AOL) [8]. In the following years, Antoniol et al. continued by using VSM and probabilistic models to recover traceability links between source code and textual documentation in natural language [2].

Marcus and Maletic introduced another IR-based technique to recover traceability in 2003. They used another vector space approach, Latent Semantic Indexing (LSI). They also showed that LSI can achieve good results without the need for stemming, which is fundamental in VSM and the probabilistic model [20].

Starting in 2003, Hayes et al. have done much work on requirements tracing using IR. They have applied both VSM [11] and LSI [13] for traceability recovery. Apart from the more technical aspects, Hayes et al. have done research on the overall quality of the tracing process [12] and how human analysts participate in the tracing loop [10].

From 2005 and onwards, De Lucia et al. have also published much on IR-based traceability recovery. For example, they have done research on how to configure and apply the IR methods to traceability management [5]. They have also done empirical studies on the usefulness of supported traceability recovery [18].

B. Growing data

The risk of spending too much effort on improving techniques for document retrieval without considering the actual needs of the users has been known for decades [15]. In the field of natural language processing, research has shown that when corpora grow bigger, the results of various techniques tend to converge. In 2001, Banko and Brill

compared different learning methods on a natural language disambiguation task and found that the performance of their studied approaches all benefited significantly from much larger training sets [4]. They argued that the research community should direct efforts on increasing the sizes of datasets, and spend less time on optimizing algorithms on small corpora.

Recently authors have seen similar results in IR-based traceability recovery. Oliveto et al. presented a traceability case study where VSM, LSI and JS were compared and the results were almost equivalent [21] and Falessi et al. have showed that simple techniques outperform more advanced techniques [7].

III. EARLY RESULTS AND ONGOING WORK

Our research so far has resulted in findings from an empirical exploratory interview study, a systematic mapping surveying previous case studies on IR-based traceability recovery and a master thesis project evaluating existing tools on real industrial documentation.

A. Interview study on alignment

The first part of this project is based on qualitative research. A large in-depth exploratory interview study was initiated in 2009 to investigate practitioners' views on requirements and verification and validation alignment (REVAL). We have conducted 30 interviews in 6 different companies with interviewees with different roles. They shared their views on the importance of REVAL, possible benefits, current REVAL practices and challenges they experience. The interview format was semi-open and the interviews lasted about one hour each.

The overall goal of the study is to identify concrete and general challenges to help us focus our future research. Intermediate results from this study, where the challenges of REVAL in one of the companies were studied, has been published [23], while results from the other companies are pending. Conclusions of the study included:

- Challenges come from many areas
- Poor tools and interoperability are often mentioned as causing obsolete information
- Traceability is mentioned as an important problem
- Communication, cooperation and other people issues are intrinsic in many challenges

B. Systematic mapping on IR-based traceability recovery

In parallel, we have studied previous work on IR-based traceability recovery. This has been done as a systematic mapping study. This approach to software engineering research has been described by Kitchenham [14], where they also compare it to systematic reviews and discuss the research contribution. In our study, the search for publications started with an exploratory phase. We used snowball sampling, exploratory searching and scanning of interesting fora

to understand the terminology of the field. Sixty research articles were the result of this step. These articles were considered our *gold standard* and enabled us to iteratively develop a valid search string. The search string was then used in the databases *Inspec & Compendex*, *ACM Digital Library*, *IEEE eXplore* and *Web of Science*.

During the entire process the following inclusion and exclusion criteria were applied:

- Inclusion: Peer-reviewed with a clear focus on recovering traceability links using an IR approach
- Exclusion: Publications about concept location, code clone detection or code clustering. Non-peer reviewed publications.

The final step, the synthesis of data, will map case studies according to IR technique, validity of data set and type of links established.

C. Tool evaluation on industrial dataset

The main objective of this master thesis project is to study traceability as a REVAL supporting concept. Many IR-based tools have been developed to support traceability recovery, but few studies have used proprietary industrial data for evaluations. There are also few reported studies that have a focus on links between requirements and test cases, Lormans and van Deursen in 2006 being an exception [17].

We have collected requirements and test documentation from a company involved in embedded software development. The company has safety requirements on their development process, and a well-defined requirement-test traceability is one aspect. Our plan is to use this documentation as input to a number of already existing IR-based tools to compare our results with previous findings. Since the company provided us with a key with the correct links, we will be able to evaluate the output from the tools. Hayes et. al have proposed a framework for comparing tools intended for traceability recovery [9], which will be followed.

We will use the search function in *Google Desktop* as a benchmark when comparing the results from different tools. Since the usefulness of the Google search engine is well known, it will help us understand the results better and go beyond graphs showing recall and precision.

IV. PLANNED FUTURE STEPS

Apart from finishing the remaining work in section III, our main steps are presented below.

A. Develop plugin in state-of-the-practice software engineering tool

Some of our closest industrial partners are working on introducing *HP Quality Center* as a new CASE-tool. A direct outcome of this transition will be that requirements, test cases and defect reports will be accessible in the same tool. This means the issue of poor tool interoperability stressed by practitioners in our early results [23] will no longer be a major obstacle. Another major advantage of this tool change in industry is that QC has good support for plug-in development, thus it can be used as a test bed. This would enable us to implement an IR-based traceability tool within the system, right in the center of the information hub.

B. Evaluate approach in an in vivo industrial environment

The aim of this study will be to evaluate how well the IR-based approach to traceability recovery works in a real industrial setting. With the plug-in described in section IV-A in place, we will be able to study the performance of IR-based approaches for traceability recovery with an industrial validity. It will also enable us to study developers and their information without introducing any additional external tools. The focus will be less on recall and precision, instead aspects such as how much you benefit from additional traceability information, how it affects the way developers work, how much time can be saved etc. and our research questions should be addressed.

A suitable method for the empirical evaluation is a case study [22]. In vivo studies are hard to conduct as experiments, since the level of control usually is too low. Collected data will include tool usage statistics complemented by answers from a questionnaire distributed among involved practitioners. The plug-in solution would also simplify expanding to multiple companies to strengthen the external validity.

C. Secondary planned steps

Our research is focusing on empirical studies of benefits of IR-based traceability, but a by-product will be more general results on the applications of IR in software engineering. Our ideas include measures, techniques and benchmarking related to IR-based traceability.

As also identified by other researchers in the field the widely used measures recall and precision are not enough to compare the results from tracing experiments [12]. Other qualities must be considered as well, but no other measures have been well-established. We are currently working on developing new alternatives.

When new traceability tools are developed, it is important to modularize them in a good way. This supports easy replacement of underlying IR technique. In small scale we have tried some other approaches with inspiration from information theory and modern search engines and our work will enable us to try this out on real industrial data.

Hopefully our work will also contribute to the benchmarking issues in the community, identified as a required future research direction by Marcus and Antoniol [19].

REFERENCES

[1] A. Abadi, M. Nisenson, and Y. Simionovici. A traceability technique for specifications. pages 103 – 112, Amsterdam, Netherlands, 2008.

[2] G. Antoniol, G. Canfora, A. De Lucia, and G. Casazza. Information retrieval models for recovering traceability links between code and documentation. *Software Maintenance, IEEE International Conference on*, 0:40, 2000.

[3] G. Antoniol, G. Canfora, A. De Lucia, and E. Merlo. Recovering code to documentation links in OO systems. In *Sixth Working Conference on Reverse Engineering (Cat. No.PR00303)*, page 136 44, Los Alamitos, CA, USA, 1999.

[4] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 26–33, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[5] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora. Recovering traceability links in software artifact management systems using information retrieval methods. *ACM Trans. Softw. Eng. Methodol.*, 16(4):13, 2007.

[6] R. Dömges and K. Pohl. Adapting traceability environments to project-specific needs. *Commun. ACM*, 41(12):54–62, 1998.

[7] D. Falessi, G. Cantone, and G. Canfora. A comprehensive characterization of NLP techniques for identifying equivalent requirements. In *ESEM '10: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–10, New York, NY, USA, 2010. ACM.

[8] R. Fiutem and G. Antoniol. Identifying design-code inconsistencies in object-oriented software: a case study. In *Conference on Software Maintenance*, pages 94 – 102, Bethesda, MD, USA, 1998.

[9] J. Huffman Hayes and A. Dekhtyar. A framework for comparing requirements tracing experiments. *International Journal of Software Engineering and Knowledge Engineering*, 15(5):751 – 781, 2005.

[10] J. Huffman Hayes and A. Dekhtyar. Humans in the traceability loop: can't live with 'em, can't live without 'em. In *TEFSE '05: Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering*, pages 20–23, New York, NY, USA, 2005. ACM.

[11] J. Huffman Hayes, A. Dekhtyar, and J. Osborne. Improving requirements tracing via information retrieval. In *in Proceedings of the International Conference on Requirements Engineering*, pages 151–161, 2003.

[12] J. Huffman Hayes, A. Dekhtyar, and S. Sundaram. Advancing candidate link generation for requirements tracing: The study of methods. *IEEE Transactions on Software Engineering*, 32:4–19, 2006.

[13] J. Huffman Hayes, A. Dekhtyar, S. Sundaram, E. Holbrook, S. Vadlamudi, and A. April. REquirements TRacing on target (RETRO): improving software maintenance through traceability recovery. *Innovations in Systems and Software Engineering*, 3:193–202, 2007. 10.1007/s11334-007-0024-1.

[14] B. Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.

[15] F. W. Lancaster and W. D. Climenson. Evaluating the economic efficiency of a document retrieval system. *Journal of Documentation*, 24(1):16–40, 1968.

[16] J. Leuser. Challenges for semi-automatic trace recovery in the automotive domain. In *Proceedings of the 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering, TEFSE 2009*, pages 31 – 35, Vancouver, BC, Canada, 2009.

[17] M. Lormans and A. Van Deursen. Can LSI help reconstructing requirements traceability in design and test? In *CSMR '06: Proceedings of the Conference on Software Maintenance and Reengineering*, pages 47–56, Washington, DC, USA, 2006. IEEE Computer Society.

[18] A. De Lucia, R. Oliveto, and G. Tortora. Assessing IR based traceability recovery tools through controlled experiments. *Empirical Software Engineering*, 14(1):57 – 92, 2009.

[19] A. Marcus and G. Antoniol. The use of text retrieval techniques in software engineering. Tutorial, International Conference on Automated Software Engineering, Antwerp, 2010.

[20] A. Marcus and J.I. Maletic. Recovering documentation-to-source-code traceability links using latent semantic indexing. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, pages 125–135, Washington, DC, USA, 2003. IEEE Computer Society.

[21] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia. On the equivalence of information retrieval methods for automated traceability link recovery. In *2010 IEEE 18th International Conference on Program Comprehension*, pages 68–71, 2010.

[22] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Eng.*, 14(2):131–164, 2009.

[23] G. Sabaliauskaite, A. Loconsole, E. Engström, M. Unterkalmsteiner, B. Regnell, P. Runeson, T. Gorschek, and R. Feldt. Challenges in aligning requirements engineering and verification in a large-scale industrial context. In *The 16th International Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ 2010)*, 2010.