Software Artefact Traceability: the Never-Ending Challenge

Rocco Oliveto*, Giuliano Antoniol*, Andrian Marcus†, Jane Hayes‡

*Department of Mathematics and Informatics, University of Salerno, Fisciano, Salerno, Italy, roliveto@unisa.it
*Département d’Informatique, École Polytechnique de Montréal, Montréal, Québec, Canada, antoniol@ieee.org
†Department of Computer Science, Wayne State University, Detroit, Michigan, USA, amarcus@wayne.edu
‡Department of Computer Science, University of Kentucky, Lexington, Kentucky, USA, hayes@cs.uky.edu

Abstract

Software artefact traceability is widely recognised as an important factor for the effective development and maintenance of a software system. Unfortunately, the lack of automatic or semi-automatic supports makes the task of maintaining links among software artefacts a tedious and time consuming one. For this reason, often traceability information becomes out of date or it is completely absent during software development. In this working session, we discuss problems and challenges related to various aspects of traceability in software systems.

Keywords: software artefact traceability, software maintenance, program comprehension, impact analysis, requirement tracing, information retrieval.

1. Motivations

Software artefact traceability has been defined as “the ability to follow the life of a requirement in a forward and backward direction” [8]. Maintaining traceability across software artefacts can be helpful in a number of tasks. Traceability links between pieces of code and documentation, such as a requirement, a set of design documents, or manual pages, aid both top-down and bottom-up comprehension. In other words, traceability allows one to map high-level documents, and thus abstract concepts, to low-level artefacts. This also improves the software maintainability: once a maintainer has identified the high-level document (e.g., requirement, use case) related to the feature to be changed, traceability helps to locate the pieces of design, code, and whatever needs to be maintained.

Clearly, traceability links efficiently support the impact analysis, as they facilitate the identification of the work products affected by a proposed change. Finally, traceability information can be used to identify reusable software components, to assess the completeness of an implementation with respect to stated requirements, and to devise complete and comprehensive test cases [1].

If on one side the potential benefits of traceability are well known, on the other side it is impractical to maintain traceability links manually during software development and maintenance. This is the reason why developers and maintainers often forgo this activity, do not perform it to an appropriate level of detail, or do not keep traceability information up-to-date. Moreover, until some years ago, there was a lack of dialog between the academic researchers and traceability practitioners that limited researchers’ accessibility to real artefact repositories for testing new techniques, and inhibited feedback loops from industry to researchers. Luckily, such a gap between academia and industry seems mitigated, as confirmed by several industries (see, for instance, NASA and Siemens) that are supporting research on traceability. Clearly, there is still a lot of work to do as the support for traceability management is not enough in real environments, but it seems that the right way has been found.

The aim of this working session is to discuss the main problems related to software artefact traceability and propose possible solutions for such problems. Moreover, the session also aims to identify the key-issues to further improve the dialog between academia and industry concerning the importance of maintaining traceability information during software development and to facilitate technology transfer.

2. Challenges in traceability

During the First Workshop on Grand Challenges in Traceability (GCT’06), members of the traceability community from academia, industry, and the U.S. Government identified several challenges in traceability [3].

By identifying the key issues of each challenge, we propose to articulate the discussion within the working session considering the following topics:

- **Traceability knowledge**: what are the key issues to build a body of knowledge that reflects best practices of traceability experts and practitioners, standard
terminology, and additional information such as case studies on traceability?

- **Training and Certification**: what are the knowledge areas and associated skills for traceability? How to educate managers/developers in the importance and cost-benefits of traceability?

- **Process integration**: traceability informations can be generated and maintained only if they are included into an organizational process; how to stimulate the inclusion of traceability in the development life-cycle? How to build models that define the tracing life-cycle?

- **Link granularity**: what is the correct granularity of links within a project? What are the factors that affect the granularity of links? Is it possible to define guidelines for link granularity?

- **Link evolution**: traceability links evolve with their related artefacts. In particular, during software development might arise no longer valid links or new links as a consequence of some changes made on the artefact repository (insertion of a new artefact, production of a new version of an artefact, etc). Existing change management systems and software configuration management deal with versioning of software artifacts and do not take into account the traceability link evolution problem. Which are the main difficulties to develop change management systems that effectively support the evolution of traceability links across multiple artefact types?

- **Link recovery**: in the last decade several methods [1, 7, 13, 10, 18, 19] and tools [5, 9, 12] have been proposed to support traceability link recovery. Unfortunately, such tools are not able to completely automate the traceability recovery process, as they can miss some correct links or recover incorrect links. Is it possible to define guidelines to develop effective link recovery tools? What about the recall/precision [2] problem? What are the main issues related to link recovery techniques for textual artefacts?

- **Incremental traceability**: recovering traceability links at the end of the development process might be a tedious and error prone task, as the number of links to recover is very high. For this reason, how to stimulate developers to maintain traceability links up-to-date during software development?

- **Scalability**: are the current traceability techniques scalable? Visualization tools are essential to support the comprehension and usage of large amounts of trace information [14]. Are they able to present complex information?

- **Artefact structure**: current traceability methods have been developed to trace well structured data. What are the main issues related to trace large and unstructured documents? And what about multimedia artefacts? An ontological approach [17] can be considered to this aim?

- **Artefact quality**: can traceability information be used to monitor and improve the documentation and source code quality [4, 16] during software development? Textual similarity between related artefact can be considered as quality metric [5, 11]?

- **Evaluation and Technology transfer**: a research tool has more chances to be transferred to practitioners if the usefulness of such a tool is investigated through empirical user studies [15]. Are we ready to bring traceability studies to industry? Preliminary results seem promising [6] but more empirical studies and industrial case studies are needed to demonstrate the effectiveness of traceability methods. How to stimulate industry to experiment with the research methods/tools? What are the key issues to consider to organise the technology transfer process?

It is worth noting that during the working session the discussion of other challenges related to traceability will be welcome.

### 3. Working session organisation

The working session will consist of three parts. It will start with a limited set of very short presentations (5/10 minutes) given by some of the participants, which will be solicited in advance and selected by the organisers. These presentations will present provocative statements or ideas related to any of the topics reported in Section 2. The aim is not only to present results, but also stimulate the discussion and favour networking within participants.

Following these presentations, all the participants will participate in an open brainstorming session, which will focus on identifying possible solutions, new challenges, etc. Questions will be asked and answers provided by the participants. One of the organisers will act as a moderator while the other will collect a list of solutions and issues, taking into account presentations, questions and the discussion held.

In the final part, the slides with the collected issues will be displayed to recapitulate and reiterate the unanswered items from the previous two parts and to build a roadmap for future events, research, and collaborations among the participants.
4. Intended participants

Different kind of participants will be foreseen:

- **Speakers**, aiming to raise issues with short presentations spread across the different dimensions to be discussed. Also, speakers putting on the industry perspective and needs are highly desired as well.

- **Attendees**, i.e. people coming from both academia and industry interested to debate on the topic, to collect useful insights and to create networks with other participants.

To invite some speakers and participants, we plan to issue a call for participation. In addition, we plan to invite some invited speakers.

5. Expected outcome of the session

A web-site (http://www.sesa.dmi.unisa.it/wst07) for the working session will be developed and maintained by the organisers. The discussions and presentations from the session will be summarised and publicised on the web-site and other appropriate venues.

We hope that this working session encourages research collaborations with the aim to develop new technology to meet tracing needs. Moreover, we expect that this session will be the first in a succession of future events that will focus on this research area and will also include related fields.

6. Short biography of the organisers

**Rocco Oliveto** received (cum laude) the Laurea degree in computer science from the University of Salerno (Italy) in July 2004. Since 2005 is a PhD student at the University of Salerno under the supervision of Prof. Andrea De Lucia. From October 2006 to February 2007 he has been a visiting researcher at the Department of Computer Science of the University College London (UK) under the supervisor of prof. Anthony Finkelstein. His research interests include traceability management, information retrieval, cooperative supports for software engineering, and empirical software engineering. He is a student member of the IEEE and the ACM.

**Giuliano Antoniol** research on traceability is partially supported by the Natural Sciences and Engineering Research Council of Canada Research Chair in Software Change and Evolution. He is a member of the IEEE.

**Jane Hayes** holds a M.S. in Computer Science from the University of Southern Mississippi and a PhD in Information Technology from George Mason University (GMU). She is currently an Associate Professor in Computer Science at the University of Kentucky. She is assisting the university in establishing a software engineering. Previously, she was a Corporate Vice President and the Manager of the Integrated System Technologies Operation of Science Applications International Corporation (SAIC). She has over 16 years of experience in the field of verification and validation, testing, software development, and process improvement. She is the current Director of the Center of Excellence for Traceability. Her research interests include requirements engineering, requirements traceability, maintainability, and software verification and validation. She is on the editorial board of the STVR Journal, she serves on many Program Committees, and reviews for many journals, including TSE.

7. Acknowledgments

Giuliano Antoniol research on traceability is partially supported by the Natural Sciences and Engineering Research Council of Canada Research Chair in Software Change and Evolution.

Rocco Oliveto is supported by the project METAMORPHOS (MEthods and Tools for migrAting software sys-
teMs towards web and service Oriented aRchitectures: ex-
Perimental evaluation, usability, and techHnOlogy tranSfer),
 funded by MiUR (Ministero dell’Universit e della Ricerca)
under grant PRIN2006-2006098097.

References

E. Merlo. Recovering traceability links between code and
documentation. IEEE Transactions on Software Engineering,
B. Berenbach, A. Eyged, S. Ferguson, J. Maletic, and A. Zis-
COET-GCT-06-01-0.9, Center of Excellence for Traceabil-
ity, September 2006.
proving comprehensibility of source code via traceability in-
formation: a controlled experiment. In Proceedings of the
14th IEEE International Conference on Program Compre-
hension, pages 317–326, Athens, Greece, 2006. IEEE CS
Press.
Recovering traceability links in software artefact manage-
ment systems using information retrieval methods. ACM
Transactions on Software Engineering and Methodology, 16(4),
to appear (available for download at
http://www.sesa.dmi.unisa.it/TOSEM_06.pdf).
ability links using information retrieval tools: a controlled
experiment. In Proceedings of ACM International Sym-
posium on Grand Challenges in Traceability, pages 46–55,
coupling based on product release history. In Proceedings of
the 14th IEEE International Conference on Software Main-
tenance, pages 190–198, Bethesda, Maryland, USA, 1998.
IEEE CS Press.
[8] O. Gotel and A. Finkelstein. An analysis of the require-
ments traceability problem. In Proceedings of the 1st In-
ternational Conference on Requirements Engineering, pages
Press.
ing candidate link generation for requirements tracing: The
study of methods. IEEE Transactions on Software Engineer-
for traceability links. In Proceedings of ACM International
Symposium on Grand Challenges in Traceability, pages 32–
ity assessment using information retrieval techniques. In
Proceedings of the 14th IEEE International Conference on
Program Comprehension, pages 149–158, Athens, Greece,