

Traceability Gap Analysis for Assessing the Conformance of Software Traceability to Relevant Guidelines

Patrick Rempel¹, Patrick Mäder¹, Tobias Kuschke¹, and Jane Cleland-Huang²

Technische Universität Ilmenau ¹ Software Systems Group Ilmenau, Germany {patrick.rempel patrick.maeder}@tu-ilmenau.de	DePaul University ² SAREC Chicago, IL, USA jhuang@cs.depaul.edu
--	---

Abstract: Many guidelines for safety-critical industries such as aeronautics, medical devices, and railway communications, specify that traceability must be used to demonstrate that a rigorous process has been followed and to provide evidence that the system is safe for use. In practice, there is a gap between what is prescribed by guidelines and what is implemented in practice, making it difficult for organizations and certifiers to fully evaluate the safety of the software system. We present an approach, which parses a guideline to extract a Traceability Model depicting software artifact types and their prescribed traces. It then analyzes the traceability data within a project to identify areas of traceability failure. Missing traceability paths, redundant and/or inconsistent data, and other problems are highlighted. We used our approach to evaluate the traceability of seven safety-critical software systems and found that none of the evaluated projects contained traceability that fully conformed to its relevant guidelines.

1 Motivation and Traceability Challenges

Developing safety-critical systems is a challenging process. Required features must be delivered in a way that ensures that the system is safe for use. Therefore, stringent guidelines must be met before the systems can be certified for use. Such guidelines typically prescribe activities, deliverable documents, and quality criteria focused around the software development lifecycle. As one important quality criterion, most guidelines mandate traceability. For example, the aviation guideline DO-178B/C mandates traceability from requirements to design, source code and executable object code. In practice, traceability is achieved through *trace links*, defined as “specified associations between a pair of artifacts, one comprising the source artifact and one comprising the target artifact” [CHGHH⁺14]. It is important for demonstrating that a software mitigates all identified safety risks and a rigorous software development process has been followed. However, organizations struggle to establish complete traceability [RMK13]. A prior analysis of submissions to the US Food and Drug Administration (FDA) as part of the medical device approval process, showed a significant *traceability gap* between the traceability expectations as laid out in the FDA’s “Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices”, and the traceability data documented in the submissions [MJZCH13].

2 Traceability Gap Analysis Approach

In this presentation we propose formalisms, metrics, and tool support for evaluating traceability coverage with respect to a project's relevant guidelines in order to identify areas of traceability failure. This work has been originally published at [RMKC14]. Our work is based on a specification of potential traceability problems, such as missing artifacts and missing, redundant, or inconsistent traceability. Our approach can be applied for preparing a system for initial certification or continuously throughout the software development life-cycle and involves three steps. First, those parts of a guideline that are relevant to traceability are translated into formal representation. This is a manual step that produces a formal guideline model that is reusable across different projects. Second, project data such as artifact identifiers, artifact types, and trace links, are automatically parsed and captured in a formal representation. Finally, formally specified rules are used to analyze the captured data for traceability problems within an individual guideline, between guidelines, and between a relevant guideline and project data.

3 Results and Conclusions

To evaluate our approach, we conducted case studies with five safety-critical software guidelines and seven software projects from different domains. Our case studies show that all projects suffer from insufficient traceability. We found missing artifacts in six projects and missing traceability paths as well as missing trace links in all seven projects, implying that none of the projects conform with their relevant guideline(s) and thus cannot be considered ready for certification. Our approach facilitates the identification of such problems for an initial certification and continuously throughout the project's lifecycle.

Acknowledgments We are funded by the German Ministry of Education and Research (BMBF): 16V0116 and 01IS14026A, by the excellence program of the TU Ilmenau, and by the US National Science Foundation Grant CCF-1319680.

References

- [CHGHH⁺14] Jane Cleland-Huang, Orlena Gotel, Jane Huffman Hayes, Patrick Mäder, and Andrea Zisman. Software Traceability: Trends and Future Directions. In *ICSE*, 2014.
- [MJZCH13] Patrick Mäder, Paul L. Jones, Yi Zhang, and Jane Cleland-Huang. Strategic Traceability for Safety-Critical Projects. *IEEE Software*, 30(3):58–66, 2013.
- [RMK13] Patrick Rempel, Patrick Mäder, and Tobias Kuschke. An Empirical Study on Project-Specific Traceability Strategies. In *RE13*, pages 195–204, 2013.
- [RMKC14] Patrick Rempel, Patrick Mäder, Tobias Kuschke, and Jane Cleland-Huang. Mind the gap: assessing the conformance of software traceability to relevant guidelines. In *ICSE*, pages 943–954, 2014.