POSITION STATEMENT

Extended Requirements Traceability:
A Framework for Changing Requirements

Orlena Gotel & Anthony Finkelstein

Department of Computer Science
City University
Northampton Square
London EC1V 0HB
[olly | acwf]@soi.city.ac.uk

We consider it an undisputed fact that requirements emerge and evolve during the lifetime of most system and software development projects. When they do, a mechanism is needed to assimilate both the change and its repercussions into an existing body of interrelated project material. This has traditionally been made possible, with varying degrees of success, through the ability to carry out *requirements traceability*. Nowadays, the activity of providing for requirements traceability is likely to be achieved using some form of cross referencing or indexing. Dedicated tool support for such techniques provides the added opportunity to exploit the results in interesting ways. However, and despite use of these tools, practitioners still encounter problems with continuous requirements change. We point to some of the more obvious reasons for this situation below:

(1) The inflated claims for requirements traceability that are commonly made by tool vendors are not being realised in practice. This is because there are many difficult issues that need to be considered prior to using such tools. The problem lies firstly in setting up a shared, consistent and coherent requirements traceability scheme for each project. It then lies in the need to obtain total commitment to the scheme from all the stakeholders, coupled with the need for some overall co-ordination. Although advanced technical solutions are undoubtedly needed for those projects with substantial numbers of complex requirements, most of the outstanding problems here are human and organisational in nature.

(2) Requirements traceability has a high start-up cost and needs continued funding throughout a project. Funding is often limited at the onset of a project, restricted to those aspects which are tangible and visible, and subsequently allocated in a phase-by-phase manner. This means that short-cuts are often made with requirements traceability when there are problems with budget or time. In many projects, requirements traceability is not even considered until it is required to start addressing the problems that inevitably arise with a later influx of change requests, by which time it is generally too late. Again, many of the problems here do not necessarily have technical solutions. These problems are likely to remain unless providing for requirements traceability receives higher profile and dedicated project resourcing.

(3) Work on requirements traceability does not tend to be an exemplar of good requirements engineering practice. Techniques and tools are generally developed and put into practical use prior to an understanding of what the type of problems are that the requirements traceability is intended to tackle in any particular project or organisational setting. Indeed, the requirements engineering literature tends to focus on reporting the existence and consequences of requirements traceability problems, then suggesting new and more powerful requirements traceability tools, without seeking to first discover what lies at the heart of these problems.

The first two reasons above suggest the need for changes in organisational and project culture. This we can recommend, and even propose guidelines for, but something we can do little more about in the academic arena. In contrast, our potential to do something actively about the third reason above is much more apparent. An improved understanding of the problems would enable us to examine what really needs to be done to relieve them.

In [1], we reported our findings from an empirical study that investigated the actual problems experienced when practitioners claim to have requirements traceability problems. This led to a working definition of requirements traceability: the ability to describe and follow information about the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases). The nature of the problems we uncovered also led to the identification of two basic types of requirements traceability, namely: (a) *pre-*

requirements traceability, which deals with requirements production and refinement; and (b) *post-requirements traceability*, which deals with requirements deployment and use.

Post-requirements traceability depends on the ability to trace requirements from and back to a requirements specification (or equivalent baseline) through a succession of documents and products in which they are distributed. When changes are made to the baseline, they need to be re-propagated through this chain of distribution. Pre-requirements traceability depends on the ability to trace requirements from and back to their originating sources(s), through the process in which they are integrated into baselined requirements. Any changes in the process need to be re-worked into the baseline. Changes to the baseline itself need to be carried out with reference to this production and on-going refinement process. Together, pre-requirements traceability and post-requirements traceability therefore provide a way to establish and maintain a connection between the information gathered from end-users and customers, the requirements which have been derived from this information by developers, and the subsequent project artifacts in which these requirements have been disseminated and addressed. With such a framework in place, requirements are more likely to emerge and evolve in a disciplined manner.

In the same empirical study we referred to earlier, we found that practitioners predominantly claim to have requirements traceability problems when, being unable to retrieve requirements information they want from a project repository, they have been unable to identify those people in a position to supply it. This was particularly evident with respect to the information produced and exchanged in the requirements production process. We found that such information tends to be discarded in the strive to replace the need for human contact with exhaustive documentation. However, not only is the ability to trace those who have contributed a fundamental working practice, it can provide what is often the only way to explain and assess change. In [2], we outlined an approach to facilitate this practice. The approach is based on modelling the *contribution structure* underlying the requirements engineering process. This describes the overall system of people involved in the process and extends conventional forms of *artifact-based* requirements traceability with *personnel-based* requirements traceability. The approach has been fully documented in [3, 4, 5] and evaluated through case study in [6].

Our current work involves action research with an industrial partner. This aims to investigate how the approach can be taken up in practice, be incorporated in their requirements engineering process, and be supported by general-purpose tools or as part of those schemes supported by dedicated tools. We are also examining how we can extend the approach to encompass the full social picture underlying the process, to account for the participants and non-participants involved in both requirements production and use. Our broader research agenda is investigating other requirements traceability issues. These include: (a) new techniques for trace visualisation and navigation, leading to enhanced forms of impact analysis; (b) the maintenance of requirements traceability across projects and programmes, thus tying requirements into the wider strategic concerns driving a project or suite of projects; (c) enhancing the mechanisms for managing consistency, by mapping the ViewPoints framework [7] on to requirements traceability schemes; and (d) exploiting web technology to provide decentralised architectures for requirements traceability.

References

A full bibliography of related work can be found in the references below:

[1] Gotel, O. C. Z. and Finkelstein, A. C. W. An Analysis of the Requirements Traceability Problem, *Proceedings of the IEEE International Conference on Requirements Engineering*, Colorado Springs, Colorado (April 1994), pp. 94-101.

[2] Gotel, O. C. Z. and Finkelstein, A. C. W. Contribution Structures, *Proceedings of the 2nd IEEE International Symposium on Requirements Engineering*, York (March 1995), pp. 100-107.

[3] Gotel, O. C. Z. *Contribution Structures for Requirements Traceability*, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, University of London (August 1995).

[4] Gotel, O.C.Z. and Finkelstein, A.C.W. *Revisiting Requirements Production*, Software Engineering Journal (to appear July 1996).

[5] Gotel, O.C.Z. and Finkelstein, A.C.W. *Extending Requirements Traceability Through Contribution Structures* (submitted for publication in March 1996).

[6] Gotel, O.C.Z. and Finkelstein, A.C.W. *Extended Requirements Traceability: Lessons from an Industrial Case Study* (in preparation April 1996).

[7] Nuseibeh, B., Kramer, J. and Finkelstein, A. A Framework for Expressing the Relationship Between Multiple Views in Requirements Specification, *IEEE Transactions on Software Engineering*, Volume 20, Number 10 (October 1994), pp. 760-773.