Tracing Back *from* Requirements

Anthony Finkelstein
Imperial College, Department of Computing
180 Queens Gate, London SW7 2BZ.
acwf@doc.ic.ac.uk

The problem of achieving traceability from a design, and its associated documentation, back to a requirements specification has been the subject of a considerable amount of discussion.

A much less frequently discussed and, I shall argue, much more interesting problem, is that of tracing back *from* the requirements specification into the domain which gave rise to those requirements. The tendency, in practice, is to stop conventional traceability with the procurer's signature on the requirements specification. In some senses this avoids the really hard problem. Tracing between successive representations with a defined relation to each other - specification & design, design & implementation, specification & test plan - is much easier than tracing back from the requirements specification to the tangled collection of overlapping and potentially inconsistent perspectives and statements which are the stuff of requirements elicitation. Change, consequent upon system evolution, will commonly require a better understanding of the requirements which can only be achieved by going back to their source.

Despite the amount of discussion of conventional traceability relatively little research devoted to techniques which might support it. An important reason for this is the reasonable argument that problems in traceabilty are an artefact of informal development methods. In a formal development setting an abstract specification is transformed in a correctness preserving fashion into an efficiently executable program. Changes in the specification can be made and the transformations "replayed". There is no need to worry about traceability as a guarantee that the implementation corresponds to the specification because the transformations have been proved secure.

By contrast, the problem of traceability *from* requirements is independent of the development paradigm, formal or otherwise.

In conventional traceability the primary problem that must be overcome is the "distribution" of the specification information. Crudely put, a statement, say a temporal constraint on some behaviour, in the specification tends to get distributed across the program code during the process of implementation. The challenge is to trace back to the originating statement. The reverse is true of requirements specification. In this case a single statement results from a merge or integration of statements from diverse sources. The challenge is to recapture the original distribution.

In order to make tracing back *from* requirements specification easier it is important to try and reflect the "elicitation structure" in the requirements specification itself.

To do this we have developed a range of techniques and associated tools for preserving multiple perspectives or viewpoints during the process of requirements specification and design. In particular we have:

developed a formal model of the process of specification from multiple perspectives based on an account of how "commitments" are established during elicitation (Finkelstein & Fuks 1989);

developed a scheme for organising the process of review and correction of complex specifications (Finkelstein 1991);

developed a framework and environment which supports the use of heterogeneous representation schemes and partial views of complex domains (Finkelstein, Kramer & Goedicke 1991).

Each of these provides explicit support for traceability. Further work in the area of traceability *from* requirements is planned, specifically support for change consequent upon validation.

References

Finkelstein, A. & Fuks H. (1989); Multi-Party Specification; Proc 5th International Workshop on Software Specification & Design, pp 185-195; IEEE CS Press (also as Special Issue of ACM Software Engineering Notes).

Finkelstein, A. (1991); Reviewing and Correcting Specifications; Computers & Writing IV; Kluwer.

Finkelstein A., Kramer J. & Goedicke M. (1990); ViewPoint Oriented Software Development; Proc. of 3rd International Workshop Software Engineering & its Applications; Cigref EC2 V1, pp337-351.