

# Establishing Traceability Links among Software Artefacts

Diunuge B. Wijesinghe, Karthigesu Kamalabalan, Thanuja Uruththirakodeeswaran, Gitanjali Thiyagalingam, Indika Perera, Dulani Meedeniya

*Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka*

diunuge.10@cse.mrt.ac.lk  
kamalan.10@cse.mrt.ac.lk  
thanuja.10@cse.mrt.ac.lk  
anjali.10@cse.mrt.ac.lk  
indika@cse.mrt.ac.lk  
dulanim@cse.mrt.ac.lk

**Abstract**— Artefact management in a software development process is a difficult problem in software engineering. Usually there is a wide variety of artefacts, which are maintained separately within a software development process such as requirement specifications, architectural concerns, design specifications, source codes and test cases to name a few. Artefact inconsistency is a major problem since these artefacts evolve at different rates. Maintaining traceability links among these artefacts and updating those artefacts accordingly can be a solution to address artefact inconsistency. There is a need for establishing these artefact traceability links in semi-automatic way. Proper management and visualization tool is required for effective software artefact management in an incremental software development. We provide a prototype tool to establish artefact traceability links and visualization. This paper describes the research methodology and relevant research carried out for semi-automatic traceability link establishment and visualization of software artefacts.

**Keywords**— Software Artefacts, Traceability Links, Visualisation, Traceability Management

## I. INTRODUCTION

Artefact consistency management is a complex challenge in software engineering. Different process activities in a software lifecycle that result in artefacts such as requirements specification, software architecture, design specification, source code, test cases for verification and validation of the system etc. are usually maintained in isolation and evolve at different rates. To elaborate further, although, all of these artefacts are aimed at facilitating software product developed with the expected quality parameters and fulfilling functional and domain requirements, once produced they often are subjected to different priority levels for their maintenance; some artefacts are hardly maintained and updated as the project progresses whereas certain high priority artefacts, such as the source code, are regularly updated and maintained. This can lead to artefacts rapidly becoming inconsistent with one another and losing

their value for development and documentation purposes [1].

Research on artefact consistency management is an important area of study in software engineering; it is still getting popular attracting many researchers who are interested in different aspects of the artefact consistency, however [2]. Some research studies are currently being carried out to tackle software artefact inconsistency problem. A main area of study to attack this problem is the establishment of traceability links among software artefacts. For example, we may wish to indicate that a particular requirement is the reason for the existence of certain design elements in the design specification and such design element can result in a particular type of code snippet thereby creating a faint link between the three artefacts (Fig. 1 shows an overview of having interconnected artefacts of a software process). This may be more straightforward in cases where a generative approach such as Model Driven Development is used to produce one artefact from another, since a mapping is likely to be created and can be used to establish and maintain links in between. However this single process of generating one artefact from another, which allows clear and easily manageable traceability links, is not always feasible and links may have to be established between existing artefacts and between existing and new artefacts. The default way of doing this is to define the inter-artefact links manually, which is a labour-intensive and potentially error-prone process. Therefore, we establish our main objective of this research as to explore the semi-automatic identification and specification of traceability links among the various software artefacts.

Effective artefact relationship management is a key point for the success of the applied process.