

Rebuilding and Maintaining Architecture Traces

General Information

Supervisors: Prof. Dr. Holger Giese (Holger.Giese@hpi.de) and Dr. Dominique Blouin (Dominique.Blouin@hpi.de).

Research Context: International collaboration with Dr. Dalila Tamzalit (University of Nantes, Nantes, France) and Dr. Joost Noppen (University of East Anglia, Norwich, UK).

Location: Mainly at the HPI but there may be an option to visit the University of Nantes or / and the University of East Anglia for collaboration.

Keywords: Traceability, Traceability Management, Software Architectures, Software Evolution, Requirements Engineering, Model-Based Engineering.

Background and Motivation

Traceability within software development is the ability to establish and maintain the relationships between artefacts of various development phases, such as determining for a component of a software architecture from which requirement(s) it originates. While this information is essential for determining the impact of changing requirements on the actual system, it is also a vital part of the certification processes that safety-critical systems must undergo to ensure that no malfunction will lead to catastrophic consequences or endanger human lives.

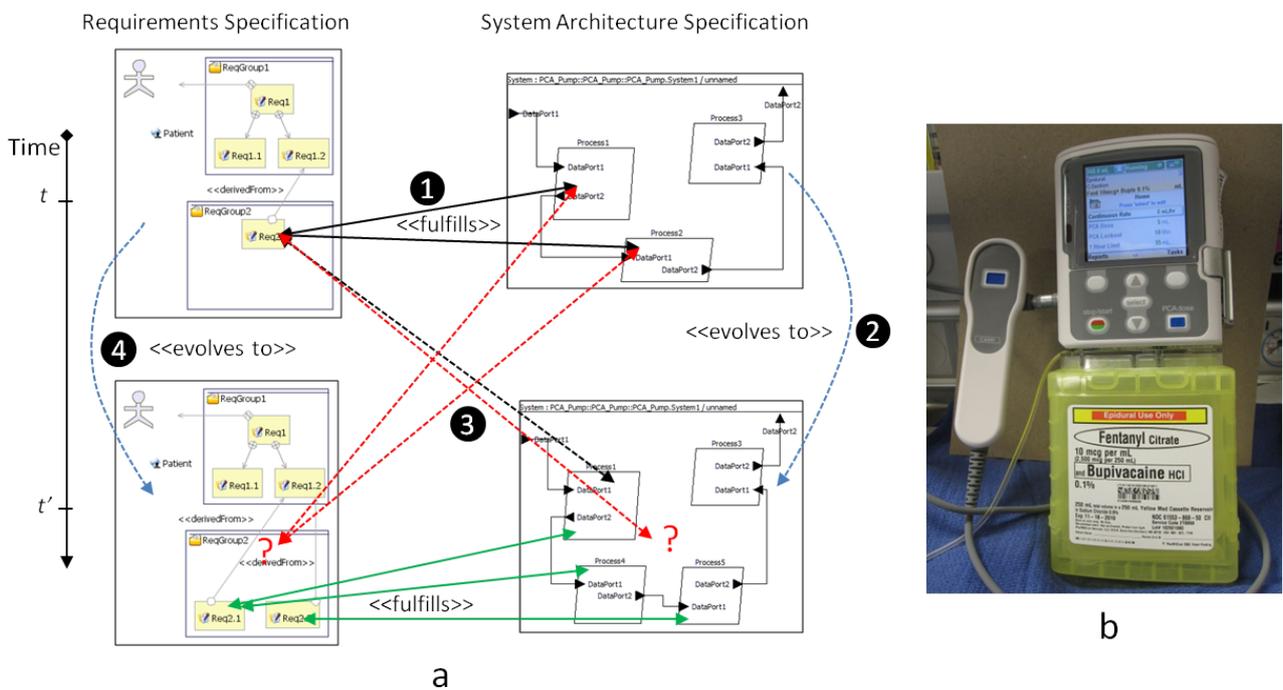


Figure 1: a) Erosion of the traceability links between a software architecture and its requirements specification, and b) example of a patient-Controlled Analgesia infusion pump for the case study

Maintaining accurate tracing information between complex systems and their requirements is currently a major problem. Traceability needs to be updated for every change that occurs in either the requirement

or the architecture of the system or both. Under ideal circumstances, a requirement specification, which consists of a set of individual requirements can be completely traced forward and backwards to elements of a software architecture that fulfills them (figure 1 a ①). Each requirement is linked to a set of associated architectural elements involved in implementing the requirement, and conversely, each architectural element is linked to a set of associated requirements from which it originated. However, over time often architectures evolve ② to meet new requirements or to improve their quality attributes such as maintainability, performance, resilience, etc. For example, component interfaces may be relocated to ease access, or components may be split or restructured to improve adaptability and evolvability. While these changes do not (or at least should not) affect the functional behaviour, the trace information between requirements and architecture elements can be damaged ③ (red dashed arrows) and it may no longer be possible to determine for each architecture element to which requirement(s) it belongs, and for requirements by which architecture elements they are realised.

Description

The goal of this project is to develop a method to rebuild and maintain the traces between requirements specifications and system architectures when at least one of them evolves. In a first step, the scenario where traces have been eroded following the refactoring of an architecture will be considered. Assuming that the original requirements and architecture specifications and their trace relations are still available, specific insights can be used to rebuild the traces:

- The amount of tracing information that was lost as a result of refactoring the architecture
- The difference between the architectures as obtained from the trace relations between the elements of the original and the refactored architecture

In a second step, the scenario where the requirements specification is modified after the refactoring of the software architecture ④ will be considered, and the developed approach will be integrated into the reasoning framework developed for the first step.

To validate the developed method, a case study from the medical domain will be used, which puts into practice traceability management between requirements modeled with the RDAL language and system architectures modeled with the AADL language. The system consists of a Patient-Controlled Analgesia (PCA) infusion pump (figure 1 b). PCA pumps are used to provide safe and accurate drug delivery to patients. Such devices have been associated with frequent persistent safety problems that lead to severe injuries and deaths. To improve their safety, the US Food and Drug Administration (FDA) has issued special guidance for their development including model-based engineering.

As a starting point, traceability forensics in legacy software systems developed at the University of East Anglia on the top of the DUEA-ST platform and traceability management approaches developed at the System Analysis and Modeling group of the Hasso-Plattner Institute as well as software architecture restructuring approaches from the University of Nantes will be studied.

The developed approach will be implemented with the Eclipse Modeling Framework and released under the Eclipse platform integrating the Open Source AADL Tool Environment for editing AADL and RDAL models.

Technical Skills

Knowledge of Java is highly recommended. Model-Driven Engineering, Eclipse Modeling Framework (EMF) and Eclipse Plugin Development would be a plus but otherwise training will be provided. There may be an opportunity to continue to work for the group as a student volunteer after the project.

Contact

For any question about this project, please contact Dr. Dominique Blouin.