Reference: **SAR-TS**

Topic: **Software Architecture Reconstruction – Tool Survey**

Lecture-Type: Practical course, seminar, Bachelor thesis

Start: 02.04.2017

End: To be defined

Contact: Felix Rinker (felix.rinker@tuwien.ac.at)
Stefan Biffl (stefan.biffl@tuwien.ac.at)

Background

Software architecture of complex systems tends to erode over time unless effort is spent to analyze the actual software architecture and reduce non-essential complexity. If the actual systems architecture is not clear, software architecture reconstruction becomes necessary.

Software architecture acts as a shared mental model of a system expressed at a high level of abstraction. By leaving details aside, this model plays a key role as a bridge between requirements and implementation. According to Garlan [1], software architecture plays an important role in at least six aspects of software development: *understanding*, *reuse*, *construction*, *evolution*, *analysis*, and *management*. To understand, document, and maintain large applications, it is important to know their architecture. However, reasoning about a system's intended architecture must be recognized as distinct from reasoning about its realized architecture. This is particularly true in cases where the intended architecture is not completely specified, documented, or disseminated to all of the project members[1].

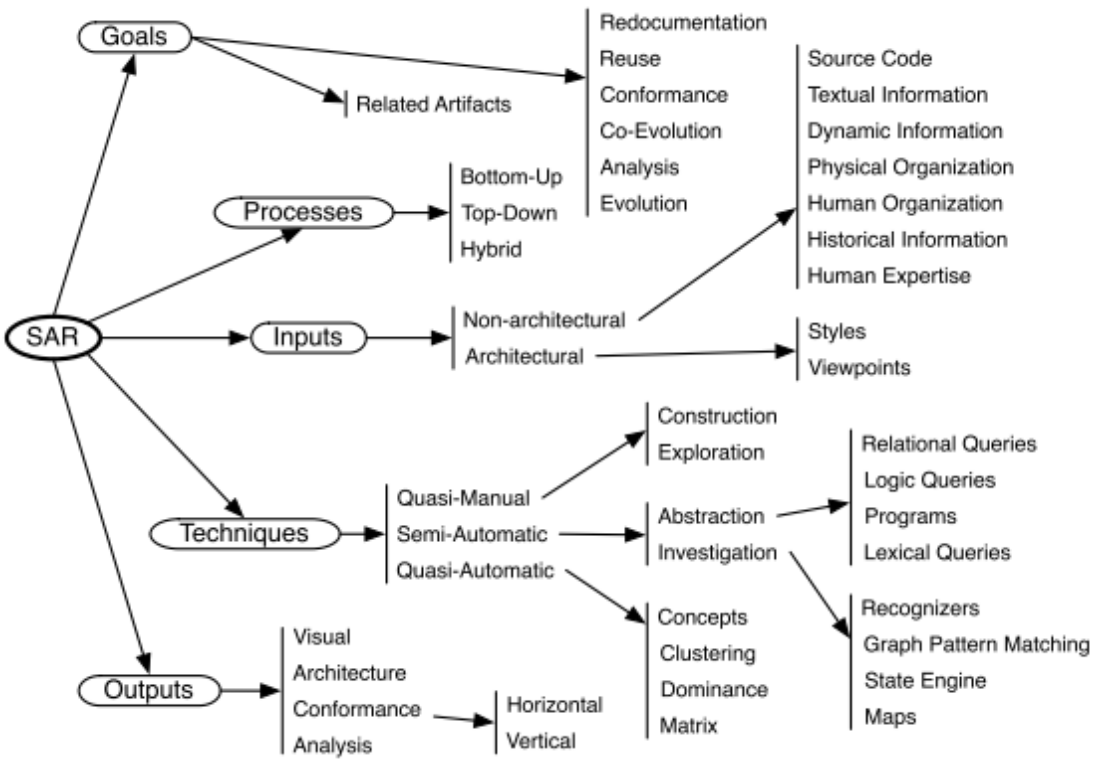

Fig. 1: Categorization of Software Architecture Reconstruction approaches

[1] SEI: http://www.sei.cmu.edu/architecture/research/previousresearch/reconstruction.cfm

Ducasse and Pollet [2] provide a process-oriented taxonomy on *Software Architecture Reconstruction* that evaluates available approaches and tools, and categorized these according to their *goals*, *processes*, *inputs*, *techniques* and *outputs* (see Figure 1).

Main goal of this topic is to identify and evaluate tools in selected categories with focus on their feasibility to reconstruct a software architecture from source code and documentation, as well as their usefulness in today's technology stacks.

Tasks

Based on the goal specific tasks include:

- Basic literature search and review based on [2] and [4].
- Evaluation of approaches and tools in an existing Java project.
- Documentation of findings and challenges.

Expertise and Skills Needed

Based on the selected tasks, required expertise and skills may vary. For this project, the following skill set is recommended:

- Empirical evaluation of tools
- Software Engineering Skills
- Java and the standard technology stack (e.g., Build Tools, Issue tracker, SCM)
- Interest in software architecture
- Willingness to explore tools with different levels of granularity of documentation

You will learn

- Basics on systematic literature reviews.
- Tool evaluation approaches for software architecture investigations

References

[1] Garlan D.: "Software architecture: a roadmap". In Proceedings of ICSE, pages 91–101, 2000

[2] Ducasse S. and Pollet D.: "Software Architecture Reconstruction: A Process-Oriented Taxonomy", In: IEEE Transactions Software Engineering, pp.573-591, 2009

[3] van Deursen A., Hofmeister C., Koschke R., Moonen L. and Riva C.: "Symphony: view-driven software architecture reconstruction", In: Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), pp. 122-132, 2004,