

Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent

| | | | |
|--|---|--|--|
| Paris Avgeriou University of Groningen, The Netherlands paris@cs.rug.nl | Philippe Kruchten University of British Columbia, Canada pbk@ece.ubc.ca | Patricia Lago Vrije Universiteit Amsterdam, The Netherlands patricia@cs.vu.nl | Paul Grisham and Dewayne Perry The University of Texas at Austin, USA {grisham, perry}@ece.utexas.edu |
|--|---|--|--|

Abstract

The shift of the software architecture community towards architectural knowledge has brought along some promising research directions. In this workshop we discuss the issues that lead to the application of architectural knowledge in research and industrial practice as well as presenting ongoing research and new ideas to advance the field. We expect to examine the state of the art and practice and gauge future challenges and trends. This year's workshop has a strong emphasis on documenting, sharing, and reusing architectural rationale and design intent.

1. Theme and goals

Software architecture plays an important role in managing the complex interactions and dependencies between stakeholders and providing a central artifact that can be used for reference. Current research trends in software architecture focus on the treatment of architectural decisions as first-class entities and their explicit representation in architectural documentation. From this point of view, a software system's architecture is no longer perceived as interacting components and connectors only, but also as a set of architectural decisions that convey the architectural principles underlying a particular design [1] [3]. This new paradigm is a response to current design practices that fail to document design decisions and their underlying rationale. The consequences of the lack of architectural decisions are well known in the software industry: expensive system evolution; poor stakeholder communication; limited reusability of architectural assets; and poor traceability between requirements, architecture and implementation.

The focus of this workshop is on current software engineering approaches that attempt to tackle this

problem by extracting, representing, sharing, using and re-using architectural knowledge. Architectural Knowledge (AK) is defined as the integrated representation of the software architecture of a software-intensive system (or a family of systems), the architectural design decisions, and the external context/environment.

This workshop aims to bring together researchers and practitioners (especially architects) of the software architecture community that are interested in sharing and reusing architectural knowledge. It will foster a presentation of the latest approaches in the field, both from industry and academia, as well as a creative discussion between the participants in specific themes.

2. Topics

The subject of architectural knowledge is complex and covers many issues, both general and domain-specific. It is truly a multi-disciplinary domain within software engineering and knowledge engineering. Overall, it involves at least the following topics:

- Notations, languages and tools to model, visualize, extract and share architectural knowledge
- Ontologies, domain models and meta-models for architectural knowledge
- Technical, social, and management factors in communicating architectural knowledge
- Architectural knowledge as decision support for both new and evolving designs
- Traceability between requirements, architectural design decisions and architectural solutions
- Evolution of, and reconstruction from legacy systems of, architecture knowledge and rationale
- Empirical studies of the use and reuse of, as well as the impact on design and maintenance by using, architectural knowledge and design rationale

- Impact of intent and rationale for design evaluation, change analysis, component reuse, and project communication
- Using intent and rationale to manage evolution
- Documenting dependencies between design decisions and projecting impact of change
- Design and intent recovery and reverse engineering
- Decision support and capture tools

3. Selected Papers for Presentation

The papers went through a thorough review process, where each paper was evaluated by three members of the programme committee. We selected 12 papers to be presented in the workshop [4] that we believe will help stimulate discussion and further research. Authors of accepted papers will present short position statements and raise the topics that will be discussed during the workshop. To give a taste of the current state of the art and practice in this domain, we present here short summaries of the accepted papers.

The paper by Carmen Zannier, *et al.*, presents the human and social aspects of the architectural decision making process. They support the thesis – often met in the different software engineering disciplines – that any technical solution proposed needs to take into account the social context of development organizations.

Muhammad Ali Babar, *et al.*, conducted a survey of current approaches for architectural knowledge management both in the research and the industrial context. They compare these approaches with the knowledge management theory and reflect upon the resulting gap.

Charles Chen, *et al.*, discuss capturing the rationale behind design decisions in a systematic way. They explore and compare two research architecting methods that support rationale documentation. On the same topic, Paul Grisham, *et al.*, discuss the application of design methods and notations with respect to capturing design rationale. They present the results of a classroom experiment, demonstrating the usefulness as well as the challenges of rationale modeling.

Ibrahim Habli and Tim Kelly tackle the topic of recovering architectural knowledge in an existing system architecture. They propose to use derivational analogy to reconstruct the decision-making process and document architectural drivers, decisions, and subsequent analysis.

Lenin Babu Thummalapalli, *et al.*, focus on application-generic architectural knowledge [2] that can be reused across systems or application domains.

They present their early efforts to derive an ontology that captures such knowledge as an invaluable reusable asset for architecture knowledge management. Sebastian Herold, *et al.*, also tackle the topic of application-domain knowledge and specifically generic architectural drivers. They propose to reuse this knowledge and thus couple requirements engineering and architecture design during the early design decisions.

Anders Mattsson, *et al.*, present an industrial perspective of attempting to capture architectural knowledge formally through Model-Driven Development techniques. They highlight the shortcomings of such techniques in industrial practice and discuss the formalization of design rules.

Liming Zhu and Ian Gorton propose modeling both architectural design decisions and non-functional requirements as first-class entities by means of UML profiles. They integrate the two associated models in the software architecture document and offer traceability with other architectural artifacts.

Rafael Capilla, *et al.*, have worked on the documentation of design decisions and the subsequent maintenance operations during system evolution. They propose achieving that with the use of a specific description format for decisions as well as a conceptual model and associated tool support. In a similar context, Salvador Trujillo, *et al.*, suggest extending an existing architecture by customizing the design decisions in a fashion similar to variability management in software product lines. They reuse existing architectural knowledge and further promote the traceability between architectural decisions and solutions. Muhammad Ali Babar presents another example of tool support for capturing, presenting, using and reusing architectural knowledge and demonstrates how this tool can be used with existing architecture design and evaluation methods.

10. References

- [1] Kruchten, P., Lago, P., and van Vliet, H. Building up and Reasoning about Architectural Knowledge, 2nd International Conference on Quality of Software Architectures (QoSA), pp. 43-58, 2006.
- [2] Lago, P. & Aygeriou, P. First workshop on sharing and reusing architectural knowledge SIGSOFT Softw. Eng. Notes, ACM Press, pp. 32-36, 2006, 31.
- [3] Dewayne E. Perry, and Paul Grisham. Architecture and Design Intent in Component and COTS Based Systems, International Conference on COTS Based Software Systems, February 2006, Orlando FL.
- [4] <http://www.cs.rug.nl/~paris/SHARK-ADI2007>