Architecture Knowledge Management: Challenges, Approaches, and Tools

Muhammad Ali Babar
Lero, University of Limerick, Ireland
Muhammad.Alibabar@ul.ie

Ian Gorton
Pacific Northwest National Laboratory
Ian.gorton@pnl.gov

Abstract

Capturing the technical knowledge, contextual information, and rationale surrounding the design decisions underpinning system architectures can greatly improve the software development process. If not managed, this critical knowledge is implicitly embedded in the architecture, becoming tacit knowledge which erodes as personnel on the project change. Moreover, the unavailability of architecture knowledge precludes organizations from growing their architectural capabilities. In this tutorial, we highlight the benefits and challenges in managing software architecture knowledge. We discuss various approaches to characterize architecture knowledge based on the requirements of a particular domain. We describe various concepts and approaches to manage the architecture knowledge from both management and technical perspectives. We also demonstrate the utility of captured knowledge to support software architecture activities with a case study covering the use of architecture knowledge management techniques and tools in an industrial project.

1. Introduction

The Software Architecture (SA) process consists of several activities (such as design, documentation, and evaluation), which involve complex knowledge intensive tasks. The knowledge that is required to make suitable architectural choices and to rigorously assess those design decisions is broad, complex, and evolving. Such knowledge is often beyond the capabilities of any single architect. The software architecture community has developed several approaches (such as a general model of software architecture design, Architecture Tradeoff Analysis Method (ATAM), and architecture-based development) to support a disciplined architecture process. While these approaches help manage complexity by using systematic approaches, they provide little support to provide or manage the knowledge required or generated during the software architecture process.

Recently, various researchers have proposed different ways to capture the contextual and technical knowledge underpinning architecture design decisions. An essential requirement of all these approaches is to describe software architecture in terms of design decisions and the knowledge surrounding them. However, architecture design decisions and the contextual knowledge are seldom documented in a rigorous manner.

In addition, one of the hurdles to rigorously capturing and managing architecture knowledge is the lack of suitable approaches and supporting tools. Moreover, we have also found that there is little use/reuse of the architectural artifacts (such as scenarios, quality attributes and tactics) that are informally described in architecture patterns documentation. This shortfall is simply because current formats for describing patterns are not suitable for the software architecture process. Also, pattern documentation formats do not explicate the schemas of the relationships among scenarios, quality attributes, and patterns as reusable artifacts of software architecture knowledge.

This tutorial is about approaches, techniques, and tools for Architecture Knowledge Management (AKM) for the software architecture process. We describe the challenges and various approaches to managing architecture knowledge. We also discuss the strengths and weaknesses of different techniques and tools for design, and building repositories of architecture knowledge and share our experiences of using architecture knowledge repository in the industry.

2. AKM problems

One of the key problems in the software architecture process is the lack of access to knowledge underpinning architecture design decisions and process. This type of knowledge involves such diverse items as the impact of certain middleware choices on communication mechanisms, why an API is used instead of a wrapper, and who to contact to discuss the

29th International Conference on Software Engineering (ICSE'07 Companion)
0-7695-2892-9/07 $20.00 © 2007 IEEE

performance issues. Much of this knowledge is episodic and is usually not documented. This absence of a disciplined approach to managing architecture knowledge has many downstream consequences such as:

- The evolution of a system becomes complex and cumbersome, resulting in violations of the fundamental design decisions
- Inability to identify design errors
- Inadequate clarification of arguments and information sharing about the design and process.

All of these factors cause a loss of substantial knowledge generated during the architecture process, thus depriving organizations of a valuable resource. To address this proble, software architecture researchers and practitioners have developed several methods, techniques, and tools to support the software architecture process, though until recently they have not paid significant attention to architecture knowledge management challenges.

Hence, there is dearth of theoretically solid and practically viable methods and tools for effectively capturing and managing software architecture knowledge. Existing approaches to systematic design, documentation and analysis of software architecture do not explicitly state what types of knowledge need to be managed. Also, none of the current approaches provide any conceptual framework to design, develop and maintain a repository of architecture knowledge.

To address these issues, there is a need for a theoretical framework for developing approaches for eliciting, organizing and managing architecture knowledge and providing suitable tool support. Such a framework can exploit the theoretical concepts and practical guidelines from the knowledge management and experience factory paradigms to provide an integrated infrastructure for managing architecture knowledge. Apart from the technical challenges, knowledge management in software architecture process also suffers from non-technical challenges such as lack of motivation, insufficient resources, and lackluster sponsorship by management. Hence, it is equally important to design and deploy effective strategies to deal with these issues.

3. Supporting AKM

The major objective of knowledge management is to improve design processes and practices by utilizing individual and organisational knowledge resources. These include skills, capabilities, experiences, cultural norms, and technologies. The software architecture process generates both explicit and implicit knowledge. These are mutually complementary entities that interact with each other in design activities. Knowledge management support works towards improving the architecture process by systematically acquiring, structuring, storing and maintaining knowledge required to support different activities and tasks.

There are two mail approaches to managing knowledge: 1) codification, aimed at making tacit knowledge explicit and; 2) personalization, intended to support knowledge sharing by describing who knows what. In order to support any of these knowledge management strategies, there are at least three important components: 1) A conceptual data model for characterizing the main architectural constructs and their relationships that form architecture knowledge; 2) Approaches to capturing and organizing knowledge; and 3) A knowledge-repository.

4. Tutorial Outline

This tutorial is divided into four sections. These are briefly described below:

Architecture Knowledge Management Overview: Describes what architecture knowledge management involves, its benefits and associated challenges. Discusses current approaches and tools that are used in practice and research.

Domain modeling: Discusses the various approaches to characterizing architecture knowledge and describes the elements and relationships among them that make architecture knowledge. A schema for architecture knowledge is presented along with a few queries.

Tools and technologies for AKM: Describes various tools and technologies for capturing and managing architecture knowledge and discusses the advantages and disadvantage of different tools. It also shows how to capture and manage architecture knowledge using a tool called BRedB.

Utilizing architecture knowledge: Presents a case study on managing architecture knowledge in the avionics systems domain. It describes the domain and architecture knowledge challenges, and current evaluation methods. It discusses the use of a quality framework managed by BRedB for evaluating the software architecture of an aircraft system.

Acknowledgement: Tutorial material and BRedB came out of the research carried out at the National ICT, Australia by the presenters.