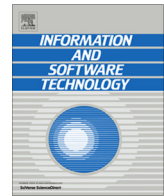




Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Two controlled experiments on model-based architectural decision making



Ioanna Lytra*, Patrick Gaubatz, Uwe Zdun

Software Architecture Research Group, University of Vienna, Austria

ARTICLE INFO

Article history:

Received 4 September 2014
 Received in revised form 16 March 2015
 Accepted 16 March 2015
 Available online 21 March 2015

Keywords:

Architectural design decision
 Architectural decision model
 Architectural knowledge
 Controlled experiment

ABSTRACT

Context: In recent years, architectural design decisions are becoming more and more common for documenting software architectures. Rather than describing the structure of software systems, architectural decisions capture the design rationale and – often reusable – architectural knowledge. Many approaches and tools have been proposed in the literature to support architectural decision making and documentation (for instance, based on models, ontologies, or templates). In this context, the capturing, organization, and effective reuse of architectural knowledge has gained a lot of attention.

Objective: However, there is little empirical evidence about the supportive effect of reusable architectural knowledge on the effectiveness and efficiency of architectural decision making.

Method: To investigate these aspects, we conducted two separate controlled experiments with software architecture students in which we tested the supportive effect of reusable decision models in decision making and documentation.

Results: Our results show that the use of reusable decision models can significantly increase both the efficiency and the effectiveness of novice architects.

Conclusion: We can report, that our findings are in line with similar studies and support the claims regarding reusable architectural design decisions in principle.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, architectural design decisions (ADDs) have been promoted to first class citizens in software architecture documentations [1]. Rather than documenting the structure of software systems (e.g., components and connectors), ADDs contribute to the capturing of design rationale. There are numerous attempts on documentation and leveraging of design rationales with focus on the reduction of architectural knowledge (AK) vaporization [2], reusability of ADDs [3], and AK sharing [4]. Apart from that, the documentation of ADDs for providing architectural guidance in software projects has gained much attention in industrial practice [5,6], lately. In this context, capturing the design solutions and their rationale is important not only for the experienced software architects but also for novice software designers who need to be educated on the existing AK and the systematic reasoning on ADDs to avoid both reinventing the wheel and making ADDs of bad quality.

Reusing ADDs can contribute to simplifying architecting [7]. Thus, addressing systematic documentation of ADDs and providing guidance during decision making for recurring design issues, the use of reusable ADD models has been proposed in the literature [3]. Similar to leveraging patterns for architectural decision making [2], reusable ADD models provide proven solutions – both application generic and technology specific – to various design issues along with their forces and consequences. Examples of reusable ADD models that have been documented cover solutions for designing service-oriented architectures (SOA) [8] and service-based platform integration solutions [9].

A few reusable ADD models and related tools that support their management (such as [10]) have been evaluated in real-life contexts. For instance, Zimmermann et al.'s ADD model consisting of 300 ADDs from the SOA domain covering various aspects such as Web service integration and transaction management has been used by practitioner communities and in industrial projects [8]. However, no feedback or empirical evidence has been gathered on whether and to which extent reusable ADD models are beneficial (i.e., they support effectiveness and efficiency of architects) in the architectural decision making process. While a few studies have investigated how reusable AK management and sharing is practiced in industrial contexts [5,11] and have validated the

* Corresponding author at: Faculty of Computer Science, University of Vienna, Währingerstraße 29, 1090 Vienna, Austria. Tel.: +43 1 4277 78523.

E-mail addresses: ioanna.lytra@univie.ac.at (I. Lytra), patrick.gaubatz@univie.ac.at (P. Gaubatz), uwe.zdun@univie.ac.at (U. Zdun).

supportive effect of pattern-based reusable AK in the decision making process [12], the software architecture community lacks empirical evidence on the positive impact of reusable AK on ADD making and documentation. Such empirically-grounded findings are important not only for validating the benefits of reusable AK in practice, but also for understanding, improving, and supporting the management and leveraging of reusable ADDs.

Therefore, we conducted two controlled experiments with students to test whether the use of reusable ADD models increases the efficiency and effectiveness of architects in the decision making and documentation process. We explicitly considered software architecture students in our evaluation, as reusable ADD models are supposed to be used as guidance models by trainers for systematically teaching patterns and technology best practices to new or inexperienced members in a software development team [13]. In the two controlled experiments, 49 and 122 students, respectively, with background in software architecture and design patterns, were asked to make and document ADDs while designing the architecture of two different software systems. For this, a Web-based tool support, called CoCoADvISE,¹ was provided to the experiment and control groups. Both the experiment and control group received material with related patterns and technology documentations and could use the tool to make and document decisions. Contrary to the control group, the tool provided additional semi-automated decision making guidance based on reusable ADD models for the participants of the experiment group. We found that participants who were supported by our semi-automated decision making guidance approach ...

- delivered more documented ADDs.
- delivered ADDs of better quality.
- invested less time for documenting ADDs.

The remainder of the paper is structured as follows. We give an overview of the approaches related to architectural decision making and documentation and compare existing architectural decision management tools with CoCoADvISE in Section 2. We present our Web-based tool CoCoADvISE for decision making and documentation and discuss its main concepts in Section 3. In Sections 4 and 5 we describe our experimental settings, as well as the analysis of the results for the two controlled experiments we conducted. Our findings, implications, and validity threats are discussed in Section 6, and finally, Section 7 concludes with a summary of our contributions and discusses future work.

2. Related work

In this section, we discuss the concept of ADDs, present existing tools and methods for decision making and documentation, and summarize the few empirical studies related to ADDs that exist in the literature.

2.1. Architectural design decisions

ADD documentations contain not only the resulting designs but also the justification, the strengths and weaknesses, as well as alternatives for the selected design solutions. Thus, software architects capture ADDs for analyzing and understanding, as well as sharing and communicating the rationale and implications of these decisions. Apart from that, the documentation of ADDs prevents

the potential loss of AK, a phenomenon which is known as *architectural knowledge vaporization* [1,2]. There are numerous attempts on supporting ADDs and capturing their design rationales. Clements et al. suggest a general outline for documenting architectures and guidelines for justifying design decisions [15] while Tyree and Akerman present a rich template for capturing and documenting several aspects of ADDs [16]. A different technique proposed by Lee and Kruchten aims at establishing formalized ontological descriptions of architectural decisions and their relationships [17]. Zimmermann et al. use decision meta-models to capture reusable AK [3] to be reused among different projects of the same domain. In addition, patterns are regarded proven knowledge for capturing ADDs and their rationale [2] and are considered often in the aforementioned approaches.

Numerous tools for the management of ADDs have been proposed in the literature [18–20]. In addition, a substantial amount of work has been done in the direction of documenting the AK using architectural decision modeling (refer to [18] for a comparison of existing architectural decision models and tools). For instance, Jansen and Bosch propose a meta-model for capturing decisions that consist of problems, solutions and attributes of the AK [1]. Zimmermann et al.'s meta-model for capturing ADDs [6] consists of three core domain entities: *Architectural Decision (AD)* related to one or more *ADTopics* organized in *ADLevels*, entailing *ADAlternatives*, the selection of which leads to an *ADOutcome*. The advantage of such ADD models is that they are reusable and can be used as guidance for architectural decision making activities, whenever recurring design issues emerge. Reusable ADD models share common concepts with patterns (see [2]), that is, they both provide proven solutions for specific design issues along with their motivation and rationale. The main difference is that reusable ADD models provide the means for defining formally more complex relationships for ADDs (e.g., the selection of a design option may exclude a design solution). Furthermore, they allow us to capture except for generic knowledge – usually addressed by patterns – also domain and technology specific AK. Yet, the relationship between architectural patterns and reusable ADD models can be eventually synergetic [3], for instance, reusable decision models can be integrated with patterns and guide the selection of patterns. Various reusable ADD models have been documented in the literature, covering SOA-related solutions [8], service-based platform integration [9], the design of domain specific languages [21], and model and metadata repositories [22].

In our empirical study, we focus on the evaluation of reusable AK in the form of reusable ADD models. For this, we provide reusable ADD models for the participants of the experiment groups of the two controlled experiments similar to the aforementioned reusable ADD models.

2.2. Tools for architectural decision making and documentation

Several tools have been developed to ease capturing, managing and sharing of architectural decisions. In most of the cases, the focus is set on the manipulation of architectural decision artifacts and their relationships, and the capturing and reuse of AK, as well as collaboration aspects. In our work, we do not intend to develop “yet another tool” for ADD management but rather to implement existing concepts in architectural decision support such as reusable architectural decision models [3] and the Questions–Options–Criteria (QOC) approach [23] and provide semi-automated tool support integrating these concepts. Our main goal is to gather empirical evidence on the supportive effect of reusable ADDs in architectural decision making. In this section, we discuss existing tools for architectural decision making and documentation and compare these to CoCoADvISE, the Web-based tool we have evaluated in our empirical study.

¹ CoCoADvISE is the Web-based version of our previous tool ADvISE for decision making and documentation [14] and shares common concepts with other reusable ADD approaches that have been documented in the literature (such as [3]). CoCoADvISE was developed for the needs of the controlled experiments and in order to provide additionally collaboration support.

Download English Version:

<https://daneshyari.com/en/article/549810>

Download Persian Version:

<https://daneshyari.com/article/549810>

[Daneshyari.com](https://daneshyari.com)