ABSTRACT

Enterprise Architecture (EA) modeling languages can express the business-to-IT-stack for an organization, showing how changes in the IT landscape impact business aspects and vice versa. Yet EA languages provide only the final architectural design, not the rationale behind this design. In earlier work, the authors presented the EA Anamnesis approach for EA rationalization. The authors discussed how EA Anamnesis forms a complement to current EA modeling languages, showing for example design alternatives, EA artifact selection criteria and the decision making strategy that was used. In this paper, the authors extend EA Anamnesis with a capability for organizational learning. In particular, the authors present an integration of two viewpoints presented in earlier work: (1) an ex-ante decision making viewpoint for rationalizing EA during decision making, which for example captures a decision and its anticipated consequences, and (2) an ex-post decision making viewpoint, which for example captures the unanticipated decision consequences, and possible adjustments in criteria. The authors use a fictitious, yet realistic, case study to illustrate their approach.

Keywords: Decision Design Graph, Design Rationale, Enterprise Architecture, Organizational Learning

1. INTRODUCTION

Enterprise Architecture (EA) languages are considered as an instrument for expressing an enterprise holistically (Lankhorst, 2005), linking perspectives on an organization that are usually considered in isolation. In doing so, one can consider the organization-wide impact of a change (Lankhorst, 2005; Op’t Land, Proper, Waage, Cloo and Steghuis, 2008), expressing its complete business-to-IT-stack (Aier and Winter, 2009). For example, for a newly introduced IT application, EA modeling languages can be used to consider implications on business processes, human resources, organizational goals, and more.

While EA modeling languages can be used to express the holistic design of an organization,
they do not express the design decisions behind the resulting models. Although we should be careful with the analogy, experience from the field of software architecture shows that leaving design rationales implicit leads to “architectural knowledge vaporization” (cf. Jansen and Bosch, 2005). This means that, without design rationale, one leaves implicit design criteria, reasons for a design, and design alternatives.

As a result of lacking rationalization architects are unable to justify their designs (Tang, Jin and Han, 2005). Furthermore new designs are constructed in an ad hoc manner without taking into consideration constraints implied by past design decisions (Tang et al., 2007). Here, constraints refer to boundaries implied by the design. These boundaries can be of a business or of a technical nature, such as the choice for a programming language implied by choosing a particular application environment.

Moreover, a survey amongst enterprise architecture practitioners (Plataniotis, 2013c) provides indications of the usefulness of design rationalization for motivating design decisions, and for architectural maintenance. At the same time, however, the survey shows that practitioners often forego the use of a structured template/approach when rationalizing an architecture, relying instead on ad hoc information capturing in tools such as Microsoft Office.

In earlier work (Plataniotis, de Kinderen and Proper, 2012; Plataniotis, de Kinderen and Proper, 2013a; Plataniotis, de Kinderen and Proper, 2013b), we introduce the EA Anamnesis approach for architectural rationalization. EA Anamnesis derives from the ancient greek word ανάµνησις (/ænæm“niːsɪs/), which denotes memory and repair of forgetfulness. Our earlier work addresses two separate viewpoints for architectural rationalization: ex-ante and ex-post. On the one hand, (Plataniotis et al., 2013a) addresses ex-ante decision rationalization. In so doing, it focuses on concepts relevant during or just after a decision making process, such as decision criteria and the anticipated consequences of a decision. On the other hand, (Plataniotis et al., 2013b) focuses on a reflection on past decisions. This ex-post viewpoint focuses on concepts relevant some time after the decision making process, such as unanticipated decision consequences and a possible subsequent change in the importance score of criteria.

In this paper, we focus on relating the ex-ante and ex-post viewpoints of EA Anamnesis. We do so because these viewpoints naturally complement each other. For one, by a comparison of ex-post unanticipated consequences to ex-ante decision criteria and anticipated consequences, we can inform future decision making. In so doing, we essentially provide a good first step in using EA Anamnesis for “organizational learning” (cf. Conklin, 1996) in the area of decision making.

As such, the main contribution of this paper is twofold: (1) to integrate two viewpoints that we considered previously in isolation, (2) to show how this integration can be useful for learning from past decision making. For now, we focus our approach on a single decision maker: the enterprise architect.

At this point, we would like to note that existing rationalization approaches in software architecture, such as (IEEE, 2005; Kruchten, 2004; Tang et al., 2007), do not consider business issues, such as decisions related to business processes. Furthermore, while the EA language ArchiMate 2.0 (The Open Group, 2012) has a motivational layer, it lacks concepts important for rationalization such as considered alternatives, decision criteria, etc. As such, ArchiMate 2.0 is not a suitable language for architectural rationalization. We discuss this in further detail in Section 2.

This article is structured as follows. Section 2 presents the related work in design rationale, decision strategies and challenges, Section 3 introduces our central artifact, a metamodel for capturing decision making. Section 4 illustrates the usefulness of EA Anamnesis with an insurance industry example. Finally Section 5 concludes.
Evaluation of Kernel Based Atanassov's Intuitionistic Fuzzy Clustering for Network Forensics and Intrusion Detection
www.igi-global.com/article/evaluation-of-kernel-based-atanassovs-intuitionistic-fuzzy-clustering-for-network-forensics-and-intrusion-detection/144138?camid=4v1a

Quality-Driven Software Development for Maintenance
www.igi-global.com/chapter/quality-driven-software-development-maintenance/77773?camid=4v1a