Design Assistant for Quality Driven Software Architecture Designs using Architectural Knowledge

Dhaya C, Zayaraz G
1Research Scholar, 2Associate Professor
Department of Computer Science & Engineering
Pondicherry Engineering College, Puducherry
dhaya.c@gmail.com, gzayaraz@pec.edu

Abstract: Software architecture is a standard and essential part of system development. The knowledge produced and consumed during this process needs to be shared and reused among different stakeholders. Therefore, software architecture knowledge (AK) needs to be managed for improving the architectural capabilities. Software architecture design is the first, and most fundamental, stage that addresses the achievement of quality requirements for software-intensive systems. ADUAK (Architectural Development using Architectural Knowledge) model has been developed to support a framework for capturing and using managed architectural knowledge to improve the architecture design process. ADUAK proposes an ontology to conceptualize the issues related to the architectural evaluation process and guides the architect in selecting the most appropriate architecture, where the interactions among quality attributes, tactics should be analyzed for the quality driven architectural design process.

Keywords: Architectural knowledge, Design Decisions, Software Architectures, and Quality attributes, Ontology

1 Introduction

Now-a-days, the software architecture [1] research community puts more importance on 'architectural knowledge'. The so called architectural knowledge concerns the set of design decisions and their rationale [2]. The design decisions are the inherent part of the software architecture [3]. This architectural knowledge plays a major role during architectural development, where architects, developer, and other stakeholders must communicate about the system to be developed. As the size and complexity of systems increases, more people get involved. There often is not a single all-knowing architect. Instead, this role is fulfilled by more than one person. To be able to take well-founded decisions, all stakeholders need to have the relevant AK at the right place, at the right time. AK needs techniques and tool support to effectively manage the knowledge [4]. ADUAK [5] provides a repository-based model support for theoretic support and guidance in the development process. The repository contains information on the ‘best practices’, technology preferences, standards, and expertise
advise to architectural directions that match with the development processes. The repository should be more intelligent, better maintainable and an interactive one.

The focus of the work is the retrieval of architectural knowledge with the creation of ontologies that helps the architect in the architectural design process [6]. To achieve this goal, we look for related knowledge for the set conceivable notional design alternatives, and the factors which lead to the quality of the designs.

The paper is organized as follows. Section 2 describes how the decision making process is applied for different design alternatives with the architectural knowledge. Section 3 describes the formal structure of the ADUAK meta-model and the factors that has maximal articulation in the architectural knowledge. Section 4 describes the development of the meta-model using ontology. Section 5 concludes the paper.

2 Quality driven architecture design process

The main motivation of this framework is to provide a systematic approach of using Architecture Knowledge for improving the organization’s architectural activities and reusing software assets. The ADUAK model entails the use of both implicit and explicit knowledge [7] in the development of architectures. In order to provide an interactive model for selecting architectures using architectural knowledge, ADUAK model is developed. It incorporates the knowledge from previously solved similar examples and the use of expertise advices, experiences in the architecture knowledge repository. The knowledge repository is logically divided into two types of knowledge [8]. Generic architectural knowledge is collected with the general knowledge capturing techniques. Project-specific architectural knowledge gets the artifacts either from the generic knowledge or created during different activities of the software development process. The knowledge concerning the domain analysis, design options evaluated and design decisions [9] made for the similar domains are useful in the design of architectures. The major constraints with architecture design process are (i) Numerous design decisions with multiple design alternatives (ii) Many design alternatives conflict (iv) Multiple stakeholders with competing priorities. It is during architectural design that crucial requirements such as performance, reliability, costs, etc., must be addressed. In other words, software architecture is “shaped” by quality requirements. The task of achieving these quality attributes to meet competing stakeholders’ requirements remains a difficult process. This framework helps the architects in assessing architectural designs based on quality requirements which are completely ‘valuable’.

3 Meta-Model of ADUAK

Based on theoretical base ideas for the quality driven architectural design process we propose the following meta-model of ADUAK. This meta-model for software architectures’ represents the relations between the decisions, stakeholders involved and the quality requirement. The model models the concepts of analysis of architectures based on quality assessments. For the architecting process, the main
concepts are the requirements, stakeholder’s, architectural decisions, and the quality factors[10]. Taking into consideration of the stakeholders’ functional/quality requirements, the ADUAK makes the decisions which are reflected in the selection of the architectures. The dependencies between the decisions [11] and the type of decisions related to software architecture, such as the architectural styles and the patterns [12] are included in the model. Quality factors suitable for selecting the design alternatives [13] are also considered. Determination of the architectural designs based on the given set of quality factors should ideally be the first step in the software development process. It defines the solutions considered and the one selected upon for the given problem definition. Architectural design process is the methodology of analyzing the alternatives, and selecting the alternative that satisfies the stakeholders’ requirement. The chosen alternative becomes the best optimal design.

Fig 1: Metamodel of ADUAK

4 Ontology for Software Architecture Design process

The main reasons for developing an ontology are to share a common understanding of the structure of the information among people, to enable the reuse of domain knowledge [14]. Ontologies express the semantics and vocabularies of the given problem in an explicit, expressive, and well-defined manner [15]. They fill the semantic gaps between the display of the information and the concepts they deliver, provides features for the reusability of the knowledge by sharing among the systems,
thereby increasing the use of architectural knowledge at different phases of the architectural design. The architectural design is implemented by several architectural decisions which are concerned by different quality attributes [1]. Ontology offers a common terminology which enables the users for making effective architecture decisions thereby assisting them in the architectural design process [16]. Choosing a software architecture design still remains a complex task. The non-functional requirements should be addressed early so as to select an optimal design. Different architectural designs to achieve the quality attributes can be identified with a capability of the architectural knowledge. Since different stakeholders have different quality requirements, the extent to which the stakeholders’ quality requirement was satisfied is a tedious process. Identifying different users and classifying them is done using segmentation [17]. Segmentation refers to grouping of people based on similar characteristics. Participating stakeholders are given the choice for selecting their relevant quality attributes. Existing requirement gathering techniques viz., WinWin requirement gathering method [18] and Goal oriented requirement gathering method [19] can be employed to identify the stakeholders’ quality requirements. For the given concern, there exist multiple architectural designs.

The software architecture team can simply query ADUAK for the architectural design that has good effects on the different quality attributes mentioned by different stakeholders. ADUAK proves the need to rely on architectural knowledge for the decision-making process within ontology design. Several architectural designs are selected for the given concern with the architectural knowledge. Each tactics can have good or bad effects on one or more quality attributes. The architect traces to determine the architectural design that has good effect on the combination of quality attributes defined. The impact of the quality attribute on the specific architectural design appears as the result of the query. An evaluation method will be much more effective if the architectural knowledge of the evaluations is reused by the software architecture team during the architecture development process with low costs [20].

The ontology shown in the Fig 1 drive the architecture evaluation process. As the evaluation team gains more experiences, the knowledge in the ontology can be evolved and refined. By this manner, the reusability of AK is achieved with the ontology. The ontology is implemented through the open source tool called protégé [21]. The rules to be considered for the architecture evaluation process are as follows:

Rule 1: The instance of architectural decision making must be taken from the Architectural_Knowledge class.
Rule 2: The architectural design must be selected from the available Architectural_Style class.
Rule 3: The Quality attribute is selected from the Quality_Characteristics class.
Rule 5: The different Quality Characteristics have the impact on the Architectural designs selected through the Architectural_decision_making process.

\[
\text{ADUAK : Architectural_Knowledge (?)AK) ^ }
\text{ADUAK : Architectural_Styles (?)AS) ^ }
\text{ADUAK : implements(?)AS, ?AK) ^ }
\text{ADUAK : Quality_Characteristics(?)QA) ^ }
\text{Xsd:string(?getValueOf)}
\]
[Fair, Good, Excellent] (?QA)
ADUAK :getValueOf (?AK, ?QA) ∧
Quality_Characteristics(?QA) ∧ Architectural_Styles (?AS) ∧
Architectural_Decision_Making (?AD) ∧ hasCriteria(?AS, ?AK) →
getValueOf(?QA)

The use of ADUAK speeded up the modeling process, as decisions were more easily reached. It is worth to know that ontology tool like protégé, executing simple queries and extracting the utility tree during the analysis is very simple. ADUAK, an ontological based meta-model contains the relationship among the architectural decisions, quality attributes and the effect of them on the architectural designs. It effectively helps the architects in the selection of architectures, thereby saving time and energy.

5 Conclusion

Software Architecture is a knowledge intensive process. Architectural Knowledge is defined as the collection of architectural design decisions and architectural forces that influence these decisions. The main purpose of the ontology based ADUAK model is to reuse the architectural knowledge and to share that knowledge. The importance of managing architectural design decisions and its rationale has been gained significant recognition. This paper aims at improving the efficiency of the architectural design process through architectural knowledge management (AKM) support. A meta-model called ADUAK is developed to support the AKM. The success of any system depends on the attention paid to the design process. ADUAK has the potential to help the architects in improving the software architecting process by providing competing design alternatives to design software architecture. Helping the architects in the architectural design process in a more expressive and effective manner if it utilizes the best practices and past architectural knowledge in a simple well-formed and expressive way.

References


