Towards Automated Tool Support for Extracting Information from Knowledge Repository

C. Dhaya and G. Zayaraz

1Research Scholar, 2Associate Professor
Pondicherry Engineering College, E-mails: dhaya.c@gmail.com; gzayaraz@pec.edu

Abstract: Architectural knowledge Management is vital in improving an organization's architectural capabilities. Different approaches have been developed to support such architectural knowledge management activities. Some approaches maintain a repository based tool support for managing architectural knowledge. The organized architectural knowledge in the architectural knowledge repository is a reusable artifact. Simply having a repository of architectural knowledge is insufficient in delivering the knowledge to the people. In this paper, a framework is proposed for extracting relevant knowledge from repositories using our architectural knowledge management tool - ADUAK (Architectural development using architectural knowledge). This extracted knowledge, implicitly or explicitly offer solutions for the specific problem. The retrieval framework receives a user’s request and extracts the relevant knowledge from the entire software documents. Searching various documents in the knowledge repository is now made easier.

Keywords: Architectural Knowledge (AK), Architecture Knowledge Management (AKM), Design decisions, Functional Requirements (FR), Non-Functional Requirements (NFR)

1. INTRODUCTION

The software architecture of a program or computing system is the structure or structures of the system, consisting of interrelated components and connectors [1]. They fail to document design decisions and their rationale. This may lead to (i) costly support for system development (ii) poor communication among stakeholders and (iii) reduced reusability of software [2, 3]. The architecting process can be considered as a decision making process through which the appropriate decisions must be made at the right time [4]. Recent research in software architecture treats architectural decisions as first class entities. Hence software system’s architecture is no longer observed as interacting components and connectors, but rather as a set of architectural design decisions [5]. The design decisions, rationale, alternatives, and architectural design collectively make a concept called architectural knowledge (AK) [6]. These decision-based approaches of developing software architecture manage architectural knowledge, emphasizing the explicit organization and documentation of design decisions and the rationale.

A comparison on nine different decision based approaches [7] has been made. (i) Archium (ii) Akerman and Tyree’s Ontology (ATO) (iii) AREL (iv) ArchDesigner (v) Bayesian Belief Network based Alternatives Selection Method (BBNb ASM); (vi) AQUA (vii) Automated Solution Synthesis Method (ASSM) (viii) PAKME (ix) ADDSS. Each decision-centric approaches record different architectural knowledge and employ different reasoning techniques. However the different notions of and emphasis is reflected in the design and implementation of existing AK tools [8].

ADUAK (Architectural Development using Architectural Knowledge) tool captures AK as reusable objects. The main objective is to provide a good support and theoretical guidance to design and implement repository-based tool support for managing architectural knowledge. The architectural design developed with the architecture knowledge viz., architecture styles and patterns [9-12], design patterns [11],
architecture and design tactics [12] improves the architecture-based software development process for their software applications.

But the repository technology is challenging for two reasons: (i) people are not able to identify their problems correctly because they are not familiar with the terms used in the repository [13,14]; (ii) people do not get the assistance from repositories [15,16]. There are no helpful suggestions for searching, a person has to either browse the entire document set or have a strong sense of what they are looking.

In this paper, a model is proposed for retrieving relevant information from knowledge repositories to develop an appropriate design for the given problem specification.

The rest of this paper is organized as follows: Section 2 specifies the features of ADUAK. Section 3 explains about the semi-automatic tool model for the extraction of relevant AK from repositories. Section 4 ends up with the conclusion.

2. MANAGING ARCHITECTURAL KNOWLEDGE

ADUAK, an architecture knowledge management tool aims in providing knowledge management support for the software architecting process. Similar to PAKME [17], ADUAK provides knowledge repository, templates and various functions to capture, manage, and present architectural knowledge. As represented in Figure 1, ADUAK provides a generic solution to address the issues during the software architecting process with its repository.

ADUAK’s knowledge repository is logically divided into two types of knowledge: Generic and Project specific. Generic architectural knowledge is collected with the general knowledge capturing techniques [18] Project-specific architectural knowledge gets the artifacts either from the generic knowledge or created during different activities of the software development process. Project-specific knowledge helps designers, developers and maintainers to better understand the architectural decisions, their constraints and reasoning behind it. The main objective of this approach is to help making explicit what is known by architects or implicitly embedded in architecture. This may include knowledge about the domain analysis, architectural patterns used, design alternatives evaluated, quality factors considered and assumptions underpinning design decisions.

3. EXTRACTING KNOWLEDGE FROM REPOSITORIES

Access to a repository of AK facilitates the designers to use accumulated “wisdom” from different projects and users when devising or evaluating architecture decisions or designs for projects in the same or similar domains [19].

But there is no single source that contains or provides all relevant architectural knowledge. Instead, architectural knowledge is reflected in various artifacts such as patterns, source code, and documentation. It’s a complicated process to distill relevant AK from the repository. In general, the designers have three major questions regarding knowledge repository and the architectural knowledge contained in it. The three questions are:

(1) From where I should start reading?
(2) Which documents I should consult for more information for the given problem specification?
(3) What is the useful ‘route’ through the documentation for a sufficient level of Architectural Knowledge?

Figure 1: Architectural Design Using ADUAK
Towards Automated Tool Support for Extracting Information from Knowledge Repository

There is a need for framework to extract relevant Architectural Knowledge from the knowledge repository, which helps the novice to provide an architectural solution for their specific requirements. The semiautomated framework consists of the following components.

3.1. Keyword Pre-processor and Document Filter

The designer prompts the terms for searching, based on that the relevant knowledge is extracted. This keyword is used to generate a set of semantic list of keys by using the DISCO (extracting DIStributionally related words using Co-occurrences) [20]. The documents having the semantic list of keys are used to filter particular documents from the block of documents.

3.2. Content word Extractor

It uses the set of key matched documents to separate the content words from it. The content words represent the document contents, which are represented in the documents. Stop words are common words that carry less important meaning than keywords. Removing stop words return the most relevant result. This content word extraction process removes the stop words with the stop word recognizer.

3.3. LSA

It uses the list of content word as keys to generate count vector matrix based on the appearance of the content word in the title of the document. It is the method that can be used to capture the meaning of a collection of documents by the construction of a vector-space model. Vector-space models are based on the assumption that the meaning of a document can be derived from the terms that are used in that document. Two problems that arose using the vector space model:

- **Synonymy** - Many ways to refer to the same object, e.g. car and automobile which leads to poor recall.

- **Polysemy** - Most words have more than one distinct meaning, e.g. model, python, chip which leads to poor precision.

To overcome these problems Latent Semantic Analysis (LSA) method can be used. Latent Semantic Analysis [21], uses a mathematical technique called Singular Value Decomposition

---

**Figure 2:** Extracting Related Documents from Knowledge Repository
(SVD) to discover the semantic structure underlying a set of documents. LSA infers the meaning of words and the contexts in which these words appear by statistical analysis. Latent semantic structure is used to guide the designers through the documentation to the architectural knowledge needed.

3.4. Rank Analyzer

It uses the cluster based method to do the ranking process on the list of related documents given by latent semantic analysis module. This ranking aids the designer to read the documents in an efficient way.

3.5. Related Paragraph Extractor

Then the knowledge is extracted from the ranked based on the paragraph extractor. The paragraph extraction tool such as Apache POI can be used.

The knowledge obtained from the documents can give rise to new terms to explore, and the discovered semantic structure can be used to determine the order in which the terms - and corresponding documents – should be explored.

With the framework explained, extracting knowledge from repositories intend to reduce the time, resources and skill level required to effectively and efficiently capture AK. Our aim is to provide a tool like ADUAK to help the organizations and the architects to reduce the time and effort required for capturing relevant AK from repositories.

4. CONCLUSION

Software architecting is essentially a collaborative process in which reusing architectural knowledge is a vital and essential part. The central objective of this framework is to extract relevant knowledge from repository-based architecture knowledge management tool. Document inspection is a method to distill architectural knowledge from the knowledge repositories. But, document inspection is very much difficult to perform. The designers are in need of a process to extract relevant knowledge in an easier manner which helps them in giving an architectural solution.

This paper aims to provide collaborative architecting process by integrating knowledge extraction model that correspond to the various activities of the process. There is no use in having tools working separately. A holistic tool is necessary that aims in reducing the searching time, and skill level to effectively and efficiently manage AK is the need of the hour.

References


