



(SAKM) Software Architecture Knowledge Management and its recent Practices, Models, Tools and Challenges

Aisha Tahseen, Noor Afshan, M. Tahseen Arshad
Department of Computer Science, Lahore Garrison University, PMAS - UAAR
aishatahseen@lgu.edu.pk noor.afshan@lgu.edu.pk

Abstract:

Management of knowledge for software architecture means to capture convenient experience and then translating it in generalized architectural knowledge. For refining the organizational architectural competences, architectural knowledge management is very much important [5]. Architectural knowledge is valuable in the Software Architecture design process. This knowledge will help the stakeholders for communication in different phases of software development life cycle (SDLC). Properly managing the architectural knowledge is very much important as it is progressively more regarded the same as an organizational positive feature and that is why so many researchers around the world are proposing tools, methods, models and different frameworks for the effective knowledge management [1]. This article contributes in exploring current work in field of software architectural knowledge management (AKM) from 2010 to 2017. This article highlights recent architectural AKM challenges and issues which are still not settled and here we also discuss different AKM tools, practices and models.

Keywords:

Architectural knowledge management AKM tools, models, practices, SOA (Service Oriented Architectures), AK (Architectural Knowledge).

1. INTRODUCTION

Managing Architectural Knowledge of an organization is a huge challenge. AKM is very much important for the betterment of the software architectural means of an institute. AK is iteratively formed, shared, and utilized during the architecture lifecycle by different stakeholders as rapidly as possible [3]. Managing architectural knowledge is necessary as it aids in communication between stakeholders of the same organization and also between the stakeholders of different organizations [1]. AKM is considered as a cornerstone for nonstop integration among the software operational deployment and its development. So there is a strong need to implement AKM using latest methodologies and tools [15]. For managing architectural

knowledge various wikis, approaches, practices, tools, and models have been developed in recent years. Software AKM solutions are not only developed but they are applied successfully [2]. A key building block of architectural knowledge is Architectural Design Decision (ADD) forms, which performs an elementary job in the software architecture process. For the sake of aiding and managing ADDs, different tools are suggested in the literature [6].

For sharing and capturing architectural knowledge, architectural decisions are significantly vital in Information Technology (IT) firms [2]. To aid in AKM, semantic wikis are used in two different contexts which are distributed software development and e-government [9].

Various models of architectural knowledge are utilized for representing domain concepts as well as their relationships. In the distributed environment, these models can also be used for using, sharing and reusing architectural knowledge even in across the organizations [3]. For making user capable of defining ADD models as per their needs, self-preference, we can say that model customization is very much important to achieve perfect fitness of the model by user and the tool provided model [6].

For understanding the recent state of software AKM and further research development, this article has compared 5 AKM tools, 5 models and 4 approaches. The outcome of this comparison of different practices, models and tools would give in sight to the recent focus of AKM support, their plus points and shortcomings. This article would definitely help the future researchers to dig out the research areas and gaps.

Outline of article: Section 1 is about the introduction, Section 2 provides research question, Section 3 gives the in depth literature review of different articles. Section 4 is comparative analysis and discussion of all compared and related papers. Section 5 is conclusion.

1.1. Research Question

What is the recent condition of knowledge management for software architecture particularly in 2010 to 2017?

2. LITERATURE REVIEW

This Literature Review provides details about recent condition of Software AKM. In this section we dig out the strengths, weaknesses of different models, practices and tools. For this purpose we present different comparative matrixes.

2.1. Software Architecture Knowledge Management and AKM Models

For the sake of knowledge management of software architecture here in this section we discuss different models.

2.1.1 Advanced Mapping Quality Prediction Model (AMQPM)

The dominant contribution of this model namely AMQPM reflects that it has provided much more accuracy prediction results as compared to the previous quality prediction models namely RMQPM and SMQPM. This new model AMQPM can help the future practitioners in the way that they can predict the quality for sharing architectural knowledge between the two different models of AK sharing even before actually the efforts has been made to create instances for AK. So we can say that AMQPM contributed a much better solution for predicting quality of architectural knowledge sharing. AMQPM mainly focused on improvement of prediction accuracy of recall rate of architectural knowledge sharing quality and also provided a better balance among prediction effort and accuracy of architectural knowledge sharing quality. Architectural knowledge is shared and maintained using Architectural knowledge Repository [3].

2.1.2 A Compliance Metadata Model

For recording and maintaining the business compliance information a Compliance Metadata Model is presented. The concepts presented in this model are purely independent from domain of any application. Authors of this model also suggest a AK corresponding mapping and they also present a tool architecture for the purpose of generating architectural knowledge, recording and documenting this knowledge, storing architectural knowledge in a central architectural knowledge repository and also the referencing system parts which are generated through Model Driven Development (MDD) solutions from AK views [10].

2.1.3 Decision process-oriented knowledge Metamodel

The decision process oriented knowledge Metamodel is produced via DSFR's (domain-specific functional requirements plus quality attributes. This Metamodel is a solution which is workable in the sense that it can be deployed practically in a real setting. Focus of Metamodel is to define a DSL (Domain Specific Language) and decision knowledge processing. The purpose of this DSL lies in structuring and shaping the interfaces plus in the reference

architecture where it discusses about the components and connectors. Here the knowledge is shared and maintained via tool and knowledge base [2].

2.1.4 ArchiMate Model

It is used to represents the chain processes which is used for collecting and recording participant data and this model focuses in storing information to a semantic wiki and so provides powerful search services for querying the wiki pages [1].

2.1.5. Ecosystem Metamodel

The research work [11] is presented in 2017 so this research is the most recent research work which is discussed in this article. Research [11] has presented an ecosystem Metamodel which defines about learning ecosystem via using MDA (Model Driven architecture) as guideline using PIM platform independent model and high level conceptual models. This ecosystem Metamodel is about the MOF (Meta Object Facility). For learning about the ecosystem development, MOF is used to put light on various perspectives. For a PhD program, this Metamodel is used for learning ecosystem. An architectural pattern is used in this Metamodel to describe ecosystems. Static data management, presentation, infrastructure and service are the four layers which are used in this architectural pattern and also introduce the human factor as a key element. For solving various problems this pattern has provided a set of software components as a technology solution. The main objective of this ecosystem Metamodel is to describe a DSL (Domain Specific Language) for developing learning ecosystem. Here for this research University of Salamanca, Spain has played a vital role as this university has provided the institutional repository for the static data management.

2.2 Software Architecture Knowledge Management and AKM Tools

Several tools have been proposed by different researchers in the field of software AKM. All of these tools were presented to cover several problems related to the knowledge management. Here in this section we discuss about some recent tools for the software AKM. We also discuss about the various tool's

contributions plus the shortcomings presented by different research articles.

2.2.1 Tool for safety system architecture design

This tool is Extension of Application Lifecycle Management (ALM) Tool. This tool is typically for safety system architecture design. The main concentration of this tool lies in two things 1) traceability, consistency and decision making 2) developing views for the facilitation of the process of finding the designs of safety associated information. Here in this tool AKM database is used for sharing and maintaining the software architecture knowledge [8].

2.2.2 ADUAK (Architectural Development Using Architectural Knowledge)

ADUAK is a web based tool which is used for the management of software architectural knowledge. For supporting the architectural design to a very large extent this ADUAK tool offers aknowledge repository, templates andalso for capturing, managing and presenting software architectural knowledge ADUAK tool provides different functions. There are logically two portions of the knowledge repository one is application specific knowledge and other is generic knowledge. Architectural repository's access to the developer and users aids in using collected wisdom from various projects and also helps when they devise architectural designs or the decisions for their own projects which are either from similar domain [5]

2.2.3 CADDMS (Customizable Architectural Design Decision Management)

This CADDMS tool determined for proposing a specific solution which aids the users in the way that they can delimit special Architectural Design Decision models as per their own needs or their distinct priorities and their own working situations hence perfect suitability can be attained between the required model by the users and the model which is provided by the tool and here knowledge is preserved with the help of knowledge repository[6].

2.2.4 SDA (Solution Decision Advisor)

The main focus of SDA tool is focusing

on providing easing in decision making and due to its knowledge engineer can easily model comprehensive design variants and see the relationship among them. Due to this tool a knowledge engineer can also discuss about the user criticism and implementation of the concepts for decision modeling. Decision Point knowledge base is used for sharing and managing the knowledge [2].

2.2.5 A new tool as an add-in

This new tool [12] serves as an add-in which is specially developed for Sparx System's Enterprise Architect (EA). This add-in is view point based implementation of a decision documentation framework. This add-in is useful as it addresses many of the stakeholders' concerns during all the architecting stages. This tool is specially developed for providing ease in the documentation of architectural decisions in a very efficient and user friendly manner. The tool is beneficiary in the sense that it can provide the benefit of even creating diagrams, and managing the projects. For knowledge management a central project repository is used. This tool also supports some high-level use-cases for example tracing of the decisions to further modeling elements in the EA (such as components, classes, and requirements), Exporting of decision views in, power point, Word and Excel, and also documenting the architectural decisions as per five viewpoints (Decision Relationship Viewpoint, Chronology Viewpoint, Stakeholder Involvement Viewpoint, Forces Viewpoint and Detail Viewpoint) from the conceptual framework. This tool has another advantage that it gives a very user-friendly method for creating traces among the decisions and some of the other elements of modeling. Moreover this tool provides bidirectional tracing hence the user can traverse from decisions towards design elements and also vice versa via context menu provided for the decision element [12].

2.3 Software Architecture Knowledge Management approaches and software AKM

For the proper management of Software AKM throughout the world many research studies have been conducted and we also discuss here some of them.

3.3.1 Extension of AKM database

This approach [8] aims to assist the architectural design of the safety systems. This approach has extended an AKM database for the development of safety system. This approach helps different certification authorities, architects and developers for presenting and finding safety associated architectural information. Architectural knowledge is maintained and shared through AKM database. This approach also provides traceability of in-between artifacts. This approach has another dominant feature that it can notify the developers for missing implementations for the requirements and also about the incompatibilities among components, patterns and decisions.

3.3.2 Peered Sites

This approach peered sites [9] is a new and very useful practice which covers various activities that provides a balance for decision making power even across sites. This approach also delivers a balance in cross site architecture for repetition of information exchange. This approach is also helpful in encouraging and setting the goal which is common for the project activities.

3.3.3 Approach LISA

This approach [13] LISA (Language for Integrated Software Architecture) targets at broad support in software architectural activities in software development life cycle. LISA model plus the LISA toolkit are two important fundamentals of this LISA approach. The LISA model is a meta-model which describes the software architecture and also architectural information. LISA model is specially designed for the automatic processing and for analysis. This model is useful for providing base for architectural documentation, AKM and then connecting the system implementation with the architecture. The LISA toolkit is used for providing various views for visualization and manipulation of many aspects of a particular architecture. In the LISA approach to assist evolution and reprocessing architectural documentation and architectural information, a knowledge repository is used. Extension of LISA approach [7] focuses on enhancement, discovering, capturing and

maintaining an architectural knowledge which is based on contextual information while knowledge is shared or maintained using context provider

3.3.4 An approach for building exploratory search systems

This research study [14] has presented an approach for software architecture knowledge management. On Enterprise Knowledge Graphs (EKGs) this approach helps in the exploratory search. This research study finds that EKG's ontology agrees in making organization's earlier implicit knowledge to explicit. Hence this knowledge notifies the design of appropriate affiliated metrics to aid exploration.

4. COMPARATIVE ANALYSIS AND DISCUSSION

This section is Comprehensive and Comparative analysis is covering several things presented by different authors. Our article has also summarized different contributions from different researchers for the sake of knowledge management. Here we describe various perspectives of different practices, models and tools of software architecture knowledge management.

The model AMQPM has contributed to predict the quality of knowledge being shared before actually sharing between two different models that share the architectural knowledge but this research study is case dependent as the weight of each architectural knowledge concept could be different for every case. With respect to the compliance Metadata model, it maintained a domain specific knowledge repository and also presented a tool for maintaining and recording the business compliance information. This compliance Metadata model is also not validated on a larger scale. Decision process-oriented knowledge Metamodel that is decision process oriented it has presented a Reference Architecture for discussing user feedback, implemented the decision modeling concept and also presented a tool but the short coming of this research study is that it has not provided any interface for project management or risk management. This research study also not delivered some contrast judgment of fresh and earlier design hence risk mitigation actions can be recognized and semi-automatically be reported and also described application of this

approach external of IT domain. ArchiMate Model focused on Collection, organization, searching and queering software AKM through semantic wikis. ArchiMate targeted various domains and aspects of software AKM while using semantic wikis. Solutions for alignment of knowledge models and knowledge versioning are missing in this ArchiMate model's research work. Ecosystem Metamodel research work typically concentrated on learning ecosystem through MDA (Model Driven architecture) but this model is not evaluated on any real context. However great efforts are been done recently via AKM models but most of them are not validated on broader context so here is the gap that could be picked up by future practitioners. While talking about the Extension of AKM database, it concentrated on software architectures for safety systems by means of extension of a broad purpose software AKM database with merging views and functions supportive development of safety systems. But in extension of AKM database there are some shortcomings although numerous abstract levels and phases of design that are stable collectively and held with modeling tools but here is the need that modeling should be utilized properly. Approach peered sites emphasized on maintaining a balance for controlling of frequency of information exchange. But approach peered sites has not delivered some empirical study on use of software AKM practices for GSD for improvement of quality for software architectures which are developed. An approach for building exploratory search systems has contributed for exploratory search on EKG. This research study also went outside recent faceted search tools of software architecture knowledge management. But this approach for building exploratory search has the short coming that this approach needs to be tested in more than one company's setting therefore only one case study is not reliable. So as per the approaches discussed here they need to be tested in real setting on a large scale so the approaches could be evaluated properly. ALM tool for software architecture concentrated on security systems. Fundamental focus includes two stages i.e. traceability, consistency & decision-making and developing views for facilitation of the process of finding the design of safety associated information. Its shortcoming summarizes that a combination of modeling support for architectural and detailed design is required in model-based development of future safety

related control systems. ADUAK, a web based tool for managing software architectural knowledge has offered some knowledge repositories and templates along with different functioning. ADUAK tool has focused over chain processes. ADUAK tool is implemented in a real scenario through case study. However, this work was unable to provide any solution for choosing best design comparing with other architectural designs. SDA has provided easiness in decision-making. This research presented a decision oriented meta-model, reference architecture that helps in discussing feedback and implementing decision-modeling concepts. However, this research does not provided any interface for project/risk management. It was also unable to provide the comparison among fresh and earlier design that leads to identification of risk mitigation actions. A browser add-in developed for Sparx System's Enterprise Architect (EA), a viewpoint based implementation of a decision documentation framework provided ease in architectural decisions documentation. However, the limitation of the study involves validation of the add-in empirically in some broader industrial setting. LISA approach that targets broad support in software architectural activities during SDLC. It focuses on enhancement, discovering, capturing and maintain architectural knowledge. LISA approach involves LISA model for automatic process analysis and LISA toolkit that provide various views for visualization and manipulation of a particular architecture. Furthermore, LISA approach focuses on enhancement. Limitation of LISA involves testing required in more than one company's setting.

Our research will contribute in the sense that the future researchers can take quick picture with the help of following comparative analysis if they adopt any these practices, tools or models in their own projects for the sake of knowledge management. Anybody can take help of these comparisons to judge the strengths and weaknesses of different recent tools, models and practices of software AKM.

To take deep vision of fresh research work related to AKM approaches, models and tools, here we present the Analysis from current work done in different research studies. Our research analysis will aid a researcher to discover out on what areas of software AKM research will be advantageous in future.

5. CONCLUSION

Software AK is very much important and positive characteristic of the organization. Software architectural knowledge is the building block for accomplishing architectural activities. Software AKM serves for processing and sharing information of an organization. Our research article discusses various software AKM tools, models and approaches. These software AKM tools, models and approaches are recent and used for aiding in management of software architectural knowledge. Here in our research we compare these tools, models and approaches. Our research article would provide a comparison on these recent tools, models and approaches. Our Comparison would help the future practitioners in the sense that if someone wants to adopt any of these tools, model or approach in his own project, then he/she would get an initial picture that what is main contribution, focus, shortcoming of the tools, approaches, or models presented by different authors. Our article has not only presented different tools, approaches and models but also pointed out some recent areas in the form of short comings we presented from different tools, models and approaches. These shortcomings could be used to fulfill the gap of research.

References

- [1] DE Boer, R.C.; van Vliet, H., "Experiences with Semantic Wikis for Architectural Knowledge Management," Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on , vol., no., pp.32,41, 20-24 June 2011.
- [2] Miksovic, C.; Zimmermann, O., "Architecturally Significant Requirements, Reference Architecture, and Metamodel for Knowledge Management in Information Technology Services," Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on , vol., no., pp.270,279, 20-24 June 2011
- [3] Peng Liang, Anton Jansen, Paris Avgeriou, Antony Tang and Lai Xu "Advanced quality prediction model for software architectural knowledge sharing" The Journal of Systems and Software, ACM, 2011.
- [4] Olaf Zimmermann, Christoph Miksovic,

Jochen M. Küster, Reference architecture, metamodel, and modeling principles for architectural knowledge management in information technology services, *Journal of Systems and Software*, Volume 85, Issue 9, September 2012, Pages 2014-2033.

[5] Dhaya, C.; Zayaraz, G., "Development of multiple architectural designs using ADUAK," *Communications and Signal Processing (ICCSP)*, 2012 International Conference on , vol., no., pp.93,97, 4-5 April 2012

[6] Lianping Chen; Babar, M.A.; Haiqi Liang, "Model-Centered Customizable Architectural Design Decisions Management," *Software Engineering Conference (ASWEC)*, 2010 21st Australian , vol., no., pp.23,32, 6-9 April 2010.

[7] Miesbauer, C.; Weinreich, R., "Capturing and Maintaining Architectural Knowledge Using Context Information," *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA)*, 2012 Joint Working IEEE/IFIP Conference on , vol., no., pp.206,210, 20-24 Aug. 2012.

[8] TimoVepsäläinen, SeppoKuikka, Veli-PekkaEloranta, "Software Architecture Knowledge Management for Safety Systems", *IEEE*, 2012.

[9] Clerc, V.; Lago, P.; van Vliet, H., "Architectural Knowledge Management Practices in Agile Global Software Development," *Global Software Engineering Workshop (ICGSEW)*, 2011 Sixth IEEE International Conference on , vol., no., pp.1,8, 15-18 Aug. 2011.

[10] Huy Tran, IoannaLytra and UweZdun, "Derivation of Domain-specific Architectural Knowledge Views from Governance and Security ComplianceMetadata", *ACM*, 2013.

[11] García-Holgado A, García-Peñalvo FJ. A metamodel proposal for developing learning ecosystems. In *International Conference on Learning and Collaboration Technologies 2017* Jul 9 (pp. 100-109). Springer, Cham.

[12] Manteuffel C, Tofan D, Koziolk H, Goldschmidt T, Avgeriou P. Industrial implementation of a documentation framework

for architectural decisions. In *Software Architecture (WICSA)*, 2014 IEEE/IFIP Conference on 2014 Apr 7 (pp. 225-234). IEEE.

[13] Manteuffel C, Tofan D, Koziolk H, Goldschmidt T, Avgeriou P. Industrial implementation of a documentation framework for architectural decisions. In *Software Architecture (WICSA)*, 2014 IEEE/IFIP Conference on 2014 Apr 7 (pp. 225-234). IEEE.

[14] Sabou M, Ekaputra FJ, Ionescu T, Musil J, Schall D, Haller K, Friedl A, Biffl S. Exploring Enterprise Knowledge Graphs: A Use Case in Software Engineering. In *European Semantic Web Conference 2018* Jun 3 (pp. 560-575). Springer, Cham.

[15] Colomo-Palacios R, Fernandes E, Soto-Acosta P, Larrucea X. A case analysis of enabling continuous software deployment through knowledge management. *International Journal of Information Management*. 2018 Jun 1;40:186-9.