

# Pragmatic Applications of Architectural Knowledge

A philosophical approach in application of architectural knowledge

**M.Sunil Kumar**

*Associate Professor*

CREngineering College, Tirupati

*Research Scholar*

S V University

Tirupati

sunilmalchi@yahoo.co.in

**Dr. A.Rama Mohan Reddy**

*Member IETE*

*Associate Professor*

S V U College of Engineering

S V University

Tirupati

ramamohansvu@yahoo.com

**A.V.Sriharsha**

*Member ACEEE, IACSIT, IAENG, ISTE*

*Associate Professor*

Sree Vidyanikethan Engineering College

A.Rangampet

avsriharsha@yahoo.com

**Abstract—** A philosophical approach in application of architectural knowledge - As the trends, culture and needs of software are growing, the software at recent, is being developed in distributed setting involving teams and cultures and it has become wide-range, varied and heterogeneous. The role of software architecture in the engineering of software-intensive applications has become more and more important and widespread. Need of architectural knowledge and modeling language has become obligatory, in this direction to support the AKR and anatomy, types and meta-architectural patterns have been discussed in this paper for proposing a new dimension of Architectural Knowledge Modeling Language.

**Keywords—** software architecture, software engineering, architectural knowledge.

## I. INTRODUCTION

“Art does not reproduce the visible; it renders visible.” (Paul Klee et. al.). “The software architecture of a program or computing system is the structure or structures of the system, which comprises of software components, the externally visible properties of those components, and the relationships among them.” – from *Software Architecture in Practice*, Bass, Clements and Kazman. *Architecture*: where non-functional decisions are cast, and functional requirements are partitioned. *Design*: where functional requirements are accomplished. *Heuristic*: it is necessary to go one level deeper to validate choices, so the architect has to do a high-level design to validate the partitioning.

Software architecture has emerged as an important sub-discipline of software engineering encompassing a broad set of

languages, styles, models, tools, and processes. The role of software architecture in the engineering of software-intensive applications has become more and more important and widespread. Indeed, component-based and service-oriented architectures have become key necessities to the design, development, and evolution of most software systems. When an application must evolve to cope with new context and user requirements, integrating new functionalities into its software architecture is necessary. The architect has thus to manually modify the architecture description, which is often tedious and error prone. Nonetheless, the notion of software architecture is one of the key technical advances in the field of software engineering over the last decade. The advantages of using explicit software architecture include early interaction with stakeholders, its basis for a work breakdown structure and the early assessment of quality attributes. Although considerable progress has been made in this area, we still lack techniques for capturing, representing and maintaining knowledge about software architectures. In this context, architectural knowledge is the integrated representation of the software architecture of a component based software-intensive system (or a family of systems), the architectural design decisions that led to it, and the external context/environment in which it functions. While much attention has been given to the identification and documentation of technical aspects of architecture, non-technical knowledge relevant for the software architecture usually remains implicit and is often left to interpersonal, informal communication.

## II. RELATED DISCUSSIONS

In this paper we represent the importance of the following:

Structure of Architectural Knowledge: Vision and Possible Architectural Knowledge Repository, Anatomical Analysis of Architectural Knowledge Repository, Structures to Effective Representation of Architectural Knowledge, Architectural Knowledge types: 1. Actionable Knowledge, 2. Enterprise Knowledge, 3. IT Architecture. Importance of Patterns as Architectural Knowledge, Architectural Knowledge as Meta Data, Knowledge Architecting, Meta-Architectural Knowledge and its representation, Architectural Decisions and Promotions of Developments in Architectures, Comments on Existing Modular Strategies of Architectural Knowledge.

Structure of Architectural Knowledge: Vision and Possible Architectural Knowledge Repository. As the trends, culture and needs of software are growing, the software at recent, is being developed in distributed setting involving teams and cultures and it has become wide-range, varied and heterogeneous [1]. To establish communication among teams and cultures who involve in developing software, for exchange of information and for assessing and searching external components, adequate architectural documentation is a necessity. Incomplete or insufficient representation of architectural knowledge leads to several problems that are generally recognized in any software engineering project.

Design methods stress the importance of documenting architecture design decisions, but experience shows that this documentation often is difficult to interpret and use by individuals not involved in the initial design of the system. Frameworks for describing an architectural maturity of an enterprise have in recent years emerged in literature [2]. These are often biased towards the particular flavor or architectural concept which is the point of interest of the specific author, enterprise, or application, be it a component-based development, platforms, product lines or highly configurable code bases. They can therefore be valid for helping a company reach one particular level of architectural maturity. What they do not provide are guidelines for which such maturity levels are suitable for companies in particular situations. Converging all business domains, applications and feasibilities while using the tool for designing, different resources have to be fulfilled to carry out goals and objectives as a strategy for design.

III. PROPOSED DIMENSIONS

Sharing knowledge is an important issue in the software industry. A lot effort in research and industry has been devoted to find out how knowledge can be coded and stored in artifacts. Architectural Knowledge Repository provides essential supporting component for a software architecture and design. The repositories provide streamlined channels for sharing architectural knowledge. Architectural knowledge is structurally represented as of three types

- 1. Actionable Knowledge
- 2. Enterprise Knowledge
- 3. IT Architecture

Knowledge isn't actionable until organizational entity is in agreement as to what that "Knowledge" really is. The knowledge becomes actionable for the entire system through collaboration with its contents and, conversely, so does theirs by collaborating with system. Many machine learning approaches provide result of machine learning processing as nuggets which are actionable knowledge.

Enterprise Knowledge Base development is based on the redefined concept of the knowledge-based enterprise and follows the Knowledge based Information System engineering paradigm [3]. Enterprise Knowledge Space is formed for the analysis of the semantics and granularity of enterprise knowledge items. The IT Architecture is an organized set of consensus decisions on policies & principles, services & common solutions, standards & guidelines as well as specific vendor products used by IT providers both inside and outside the Information Technology Branch (ITB). One of the major activities associated with producing IT Architecture will be the process of achieving such consensus decisions. It is understood that reaching consensus may constrain purchase and design options, hopefully in the interest of enhancing interoperability. It is given that the greater the consensus achieved, the greater the organizational benefits attained.

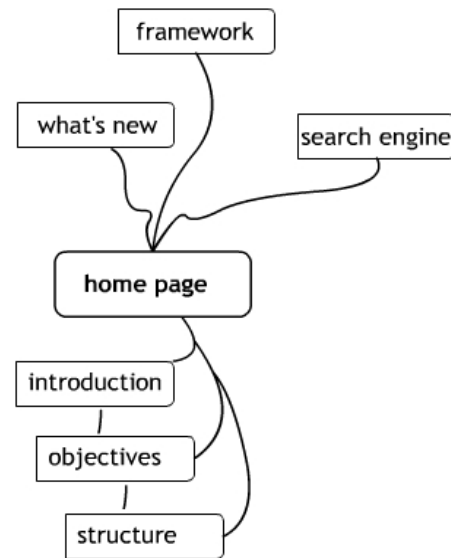


Figure 1. An Illustration of Architectural Knowledge

The above illustration projects the genericities of various types of knowledge. (Actionable Knowledge, Enterprise Knowledge, IT Architecture)

In other way patterns are the key elements of representing the architectural knowledge. Importance of Patterns as Architectural Knowledge has its own strategy of projecting the wide gamut of the system architecture. In other words, patterns are one of the many ways to put on paper knowledge that otherwise would be, perhaps only, in the expert's mind. A

pattern is “a rule showing the relation among a given context, a problem and a solution”. Patterns are proven solutions to repeated problems in determined contexts and thus patterns can help also to improve expertise. To be reusable and to help the architect to find the right solution, patterns must be understandable. Therefore, patterns can include objective, context, factors influencing the design, analysis of advantages and pitfalls, solution, implementation factors and other issues that the composer considers philosophically necessary to facilitate the use and application of the pattern. *Causal Patterns* are composite in their constitution that consists of problem and solution definitions.

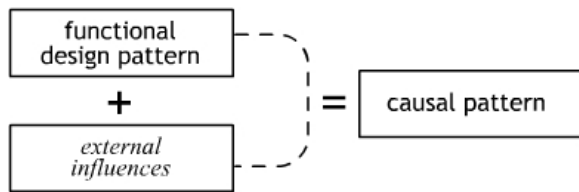


Figure 2. Causal Pattern as Architectural Knowledge

IV. REFERENCES

[1] Patricia Lago, Hans van Vliet , Towards an Ontology Web Service providing Architectural Knowledge, Vrije Universiteit, Amsterdam, The Netherlands, Feb 2005.

[2] Josef Nedstam, Even-André Karlsson, Experiences from Architectural Evolution, STRAW'05, Portland, 2005

[3] Saulius Gudas, Rasa Brundzaite, Approach to Enterprise Knowledge Base Development : Advances in Information Systems Development, Springer US, © 2007 ISBN 978-0-387-70760-0 (Print) 978-0-387-70761-7 (Online) pp 61-72, SpringerLink

V. ABOUT AUTHORS



Mr. M Sunil Kumar has completed B.Tech in Computer Science & Information Technology from JNT University and M.Tech in Computer Science from JNT University. Presently he is pursuing Ph.D in Computer Science and Engineering, S.V.University, TIRUPATI. He is currently working as Assistant Professor in the Department of CSE, Sree Vidyanikethan Engineering College, A. Rangampet, Tirupati, A.P. His main research interest includes Software Engineering, Software Architecture, Information Retrieval and Database Management Systems.



A.V. Sriharsha is B.Tech in Computer Science & Engineering from Andhra University and M.Tech in Information Technology from Sathyabhama University, Chennai. He is currently working as Assistant Professor in the Department of CSE, Sree Vidyanikethan Engineering College, A.

Rangampet, Tirupati, A.P. His main research interest includes Data Mining, Information Retrieval and Database Management Systems, Knowledge Based Software Architecture.

Dr. A. Rama Mohan Reddy is currently working as Professor & Head, Department of CSE, Sree Vidyanikethan Engineering College, A. Rangampet, Tirupati, A.P. He has completed his B.Tech from JNT University, Anantapur and M.Tech from NIT, Warangal. He has received his Ph.D. from SV University in Software Architecture. His other areas of interests include Data Mining and Object Oriented Analysis & Design.

