Architectural Knowledge Management Strategies: Approaches in Research and Industry

Muhammad Ali Babar[1], Remco C. de Boer[2], Torgeir Dingsøyr[3], Rik Farenhorst[2*]
[1]Lero, University of Limerick, Ireland
[2]VU University, Amsterdam, the Netherlands
[3]IDI, Norwegian University of Science and Technology, Norway
[1]Muhammad.AliBabar@ul.ie, [2]{remco,rik}@cs.vu.nl, [3]dingsoyr@idi.ntnu.no

Abstract

The software architecture community has recently gained an increasing interest in managing architectural knowledge. However, up until now there have been no attempts to obtain an overview of the work in the field. In this paper we present a preliminary review on current approaches to architectural knowledge management. To this end, we compare approaches known from literature and encountered in industry with knowledge management theory. We found that in reports from research and practice there appears to be a preference to use the codification strategy. However, our observations of the software architecture industry show that organizations in general tend to use a personalization strategy unintentionally. This paper serves as a call for awareness of this gap between intention and reality, and questions the biased focus on intentional codification alone. We suggest to close this gap through focusing on hybrid approaches.

1. Introduction

Despite making headway in several other knowledge intensive fields, until recently knowledge management has received little attention in the software architecture discipline. In recent years, the software architecture community has begun to recognize that knowledge management is vital for improving an organization's architectural capabilities. There has been an increased demand for suitable methods, techniques, and tools that support organizations in capturing and maintaining the details on which key architecture design decisions are based. Such information represents so-called architectural knowledge, which can be valuable throughout the software development lifecycle [2, 11]. Researchers and practitioners have proposed various approaches to capture and manage architectural knowledge [2, 16, 18]. Many of these approaches have been adapted from knowledge extraction techniques used in artificial intelligence and in social science disciplines. One of the main objectives of these approaches is to help making explicit what is known by architects or implicitly embedded in an architecture. This may include knowledge about the domain analysis, architectural patterns used, design alternatives evaluated, and assumptions underpinning design decisions.

Most of the proposed approaches to manage architectural knowledge can broadly be categorized into codification and personalization [15]. The codification strategy concentrates on identifying, eliciting and storing knowledge in repositories, which makes that knowledge widely available. This strategy promises to support high-quality, reliable, and speedy reuse of knowledge. The downside is that it usually means separating the knowledge from its creators. The personalization strategy emphasizes the interaction among knowledge workers. In this strategy the knowledge is kept with its creator, who is made known as possessor of the required knowledge.

Although management of architectural knowledge is clearly related to management of knowledge in general, up until now no structured efforts have been reported that compare the different approaches with approaches known from knowledge management literature. Moreover, we believe there is a gap between what the research community is focusing on and what the practice is in most organizations. This paper seeks to contribute to the field through providing a preliminary review on the various approaches of architectural knowledge management in research and practice and discusses the pros

*Authors' names appear in alphabetical order; all authors equally contributed to this work.

29th International Conference on Software Engineering Workshops(ICSEW'07)
0-7695-2830-9/07 $20.00 © 2007 IEEE

and cons of each of them. We have not conducted an extensive literature search to identify all contributions in this area, but we do use the authors' knowledge of the field and draw on experience gained in software architecture practice in three different countries.

2. Current Approaches to Architectural Knowledge Management

There is an ongoing trend within the software architecture community to focus on what is called 'architectural knowledge'. This trend is not only visible within the software architecture research community (cf. the SHARK workshop series as well as the WICSA 2007 keynote talk [22]), but also in industry. There seems to be a growing consensus on what architectural knowledge entails. Concepts often considered crucial are architectural design decisions and their rationale.

In the following subsections we first provide an overview of the state of research, followed by a discussion on the state of practice when it comes to managing architectural knowledge. Table 1 provides an overview of all approaches included in this review.

2.1. The Research Community

Over the past few years, the software architecture research community has spent considerable effort to define so-called 'architectural knowledge'. The community's focus on architectural knowledge stems largely from the lack of solid documentation methodologies for the reasoning behind – the rationale of – architectural designs [2]. The consensus seems to be that the key to capturing rationale is an explicit focus on architectural design decisions, summarized for instance in the title of Tyree and Akerman's paper "Architecture Decisions: Demystifying Architecture" [24], as well as in the 'design intent' concept [21]. Work by others (e.g. [2, 16, 18, 25]) further advances this consensus.

A number of research initiatives take an architectural knowledge perspective to devise tools and methods to manage architectural knowledge. These initiatives are further elaborated below.

2.1.1 DGA DDR

Falessi et al. have proposed a framework that focuses on the reasons why design decisions have been taken [13]. The framework contains a specific design decision rationale documentation technique called DGA DDR, which is driven by the decision goals and design alternatives available.

The framework aims not only to document decisions previously taken, but also to support decision makers in taking these decisions. The framework consists of two main activities. In the first activity the project objectives and constraints are defined and it is investigated which decision relationships are appropriate for the project. In the second activity the knowledge is further refined and described in tables.

2.1.2 PAKME

Researchers at National ICT Australia (NICTA) have proposed an architectural knowledge management framework, which incorporates concepts from knowledge management, experience factory, and pattern-mining [1]. This framework consists of various approaches to capture design decisions and contextual information, an approach to distill and document architecturally significant information from patterns, and a data model to characterize architectural constructs, their attributes and relationships.

The main objective of the framework is to provide a theoretical underpinning and conceptual guidance to design and implement a repository-based tool support for managing architectural knowledge. A web-based knowledge management tool, called Process-based Architecture Knowledge Management Environment (PAKME), has been developed to support the proposed framework.

2.1.3 GRIFFIN

For the last two years, researchers of the Griffin consortium have been working on methods, tools, and techniques to manage architectural knowledge. One of the results of this project is a structure for software architecture project memories [5]. A software architecture project memory stores architectural knowledge, such as the design decisions embodied in the architecture as well as the rationale underlying the design decisions. This allows management of know-why and know-how of software architectures, in addition to know-what already targeted by most existing notational and documentation approaches in software architecture, which typically focus on components and connectors.

2.1.4 ADDSS

Two collaborating universities in Spain have recently proposed to extend traditional architectural view methods, such as the 4+1 view method, with a decision view that allows capturing design decisions in the architecture process [10]. Capilla et al. [3] have proposed a web-based tool called ADDSS for recording

Approach	Description
ADDSS	A web-based tool for recording architectural design decisions.
PAKME	A process based knowledge management environment for generic and project-specific knowledge.
DGA DDR	A design decision rationale documentation technique for decision goals and design alternatives.
GRIFFIN	A software architecture project memory to manage know-why and know-how.
RFP	A knowledge repository for reusing best practices with a questionnaire as a front-end.
VCC	Architectural rules disseminated by means of small text-based documents.
RBS	A knowledge base harboring reusable quality criteria.
DSTO	An architectural knowledge management tool to improve architectural evaluation practices.

Table 1. Current Approaches to Architectural Knowledge Management

architectural design decisions. In this work a meta-model and a web-based tool are proposed to record, maintain and manage the architectural decisions taken. The tool allows for modeling traces between decisions and supports traceability between design decisions and artifacts such as architecture diagrams.

2.2. The industry

In parallel to the the research community, industry has on itself tried to employ – sometimes unconsciously – knowledge management practices to the software architecture process. Below we elaborate on a number of knowledge management initiatives we have encountered in industry that are related to software architecture and architectural knowledge.

2.2.1 RBS

RBS is a medium-sized consultancy firm that performs independent software product quality audits for third parties. One of the key architectural knowledge elements that plays a role in such an audit is the set of applicable quality criteria. These criteria are a special kind of architectural design decisions that correspond to the desired architecture of the software product [6].

Many quality criteria are applicable to multiple projects. Some projects share certain desired quality characteristics for which more or less standard architectural approaches exist. Although over the years some sort of best practices regarding the documentation of quality criteria have grown within RBS, there is no mandatory format or template that is used to express quality criteria. Each project can adopt its own style to document the quality criteria that are used in the audit. Part of these quality criteria is harvested from documentation belonging to previous audit projects.

In the past, RBS has tried to employ a knowledge management approach to better support reuse of quality criteria. Quality criteria were to be made explicit and captured in a knowledge base that could be queried to find applicable quality criteria at the start of a new audit project. Unfortunately, construction of this knowledge base eventually was discontinued because it was hard to capture all relations that exist between different quality criteria.

2.2.2 DSTO

Defence Science and Technology Organisation (DSTO) is a research and development organization, which provides scientific and technical advice on the acquisition of material to the Australian Defence Organisation. The Airborne Mission Systems (AMS) division of DSTO is responsible for evaluating software architectures for aircraft acquisition projects. AMS is required to understand and organize large amounts of architectural knowledge for a mission system's architecture to support the evaluation process. Currently, the architectural evaluation process mainly relies on the domain knowledge of local experts.

Recently, AMS's technical leadership has become increasingly interested in building its capabilities in systematically evaluating system and software architectures and managing architectural knowledge for aircraft mission systems. Hence, AMS has decided to improve its architectural evaluation practices by codifying and reusing an architecture evaluation process, architectural knowledge, and contextual knowledge.

2.2.3 RFP

RFP is a large Dutch organization that develops and maintains software systems. These systems are typically critical for the public, large in size and complexity, and long lasting. In a recent study [14] we investigated and assessed the organization's current mechanisms for sharing architectural knowledge.

RFP has acknowledged the need to support sharing and reuse of architectural knowledge. To this end, the organization has deployed a knowledge repository in which commonly used architectural knowledge is stored

and made reusable in the form of questions and answers. At the start of a project, architects use the tool to create a very first version of the architecture description. The tool will provide the architects with the stored questions. The architects' answers – the initial architectural design decisions for the project – are then structured and reflected in a text that provides the basis for the remainder of the architecting process.

2.2.4 VCC

VCC is a multi-national software development organization. Development teams within this organization are located at multiple sites spread throughout the globe. Architectural knowledge is shared in the form of 'architectural rules'; architectural decisions that need to be complied with throughout the organization. A study within this organization focused on dissemination of and compliance with architectural rules in this multi-site environment [4]. Dissemination of architectural rules takes place by means of small, text-based documents; so-called architectural notes, or *archnotes*.

3. A Comparison with Knowledge Management Theory

Knowledge management is a large interdisciplinary field, and has as such fostered a number of different approaches. Earl [12] proposes different 'schools' of knowledge management, broadly characterized as the *technocratic*, the *economic*, and the *behavioral* school. Hansen et al. [15] divide between two strategies: codification and personalization. Codification refers to organizations aiming their strategy on codifying knowledge and making it easily available for anyone through so-called knowledge repositories [19]. Knowledge from individuals or groups can be codified (or acquired) through a number of means, such as interviews, questionnaires and architecture reviews. Personalization, on the other hand, focuses on helping people communicate knowledge, instead of storing it. This can be facilitated through 'yellow-page' indexes of experts or skills management systems, or a focus on company processes to share knowledge such as postmortem reviews. Alternative approaches are to focus on collaborative work such as pair programming, to design informal meeting-spaces or to use open-plan offices.

In software engineering, most research and most industry practice has been associated with codification [8]; personalization has been given less attention in knowledge management initiatives. This observation seems to hold true for software architecture research and architectural knowledge management as well.

The research projects listed in Section 2.1 all exhibit to a large extent a technocratic focus on codification of architectural knowledge, largely neglecting economic and behavioral aspects. Both ADDSS and DGA DDR aim to record and subsequently maintain architectural design decisions. Researchers from NICTA mention both codification and personalization as ways to manage knowledge; nevertheless, the focus of their framework is on mining and capturing (i.e. codifying) implicit knowledge in the form of architectural patterns. The Griffin project reports on their efforts to construct software architecture project memories that contain codified architectural knowledge in the form of architectural design decisions and related entities.

Industry also seems to favor an explicit choice for codification of architectural knowledge, although in practice some personalization aspects are also present. However, none of the organizations we observed seems to have intentionally made a choice for the latter.

The strategy chosen for quality criteria reuse in RBS is a classic example of a codification strategy. DSTO's goal to organize architectural knowledge denotes a focus on codification as well, and RFP has clearly chosen a (centralized) codification approach for architectural knowledge management. Personalization does not play a significant role in the chosen architectural knowledge management strategy, although interviews with employees of RFP brought to light that much architectural knowledge sharing takes place through formal and informal meetings. Important (implicit) knowledge within this organization therefore includes who knows what and which architectural knowledge can be found where.

In VCC, the use of archnotes denotes an explicit choice of the organization for a codification strategy for architectural knowledge distribution. This is probably not a surprising choice, since the geographic dispersion of the development teams to a large extent inhibits colloquial knowledge sharing. Nevertheless, to our surprise one of the most important 'archnotes' (in terms of usage frequency as well as utility in daily practice) turned out to be a 'yellow-pages'-like document that contains names, locations, and phone numbers of key personnel involved in the project. The users of the archnotes system seem to have exploited the lack of mandatory structure of the archnotes to introduce elements of a personalization strategy in the system. Although the organization explicitly chose to follow a codification strategy, there is apparently an implicit undertone that personalization is very important as well. However, VCC has never explicitly chosen to follow (or reject) a personalization strategy.

In summary, current (reported) architectural knowl-

edge management approaches have a tendency to heavily rely on codification. This is true both in research and in industry. However, industry practice clearly shows that personalization is important as well. Different organizations seem to apply personalization aspects to management of architectural knowledge. Nevertheless, whenever personalization is encountered, this is more the result of an implicit choice rather than an explicit preference.

4. Reflection on Current Approaches

The main argument for choosing a codification strategy has been that you can invest once in knowledge assets, and reuse them many times [15]. A common critique of the codification strategy is that it may create 'information junkyards'. McDermott [20] claims that "if people working in a group don't already share knowledge, don't already have plenty of contact, don't already understand what insights and information will be useful to each other, information technology is not likely to create it". In addition, Swan et al. [23] criticize the knowledge management field for being too occupied with tools and techniques. They claim that researchers tend to overstate the codifiability of knowledge and to overemphasize the utility of IT to give organizational performance improvement. They also warn that "codification of tacit knowledge into formal systems may generate its own pathology: the informal and locally situated practices that allow the firm to cope with uncertainty, may become rigidified by the system".

One can also question the strong focus on codification when it comes to managing knowledge related to software architectures. Frequent technological changes make up a force which makes knowledge reuse difficult in the software engineering field. For knowledge that is not to be reused many times, personalization is inexpensive compared to codification.

Then again, the high prevalence of codification in architectural knowledge management might be the result of a number of distinctive characteristics of the software architecture field. First, software architects are used to codify knowledge, through work with modeling techniques and through identifying architectural patterns. Second, software architects are very mature users of information technology, and should be able to use technical tools more efficiently than employees in other domains. Third, architectural knowledge is the earliest design knowledge, which can be expected to be used and revisited often throughout the whole software development life cycle. With a high level of knowledge reuse, the cost of codification may be outweighed by its benefits. These characteristics of the software architecture field seem to favor a codification strategy for managing architectural knowledge.

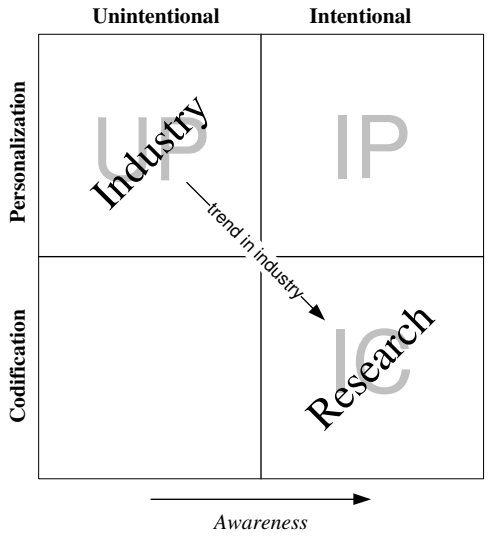

Figure 1. Architectural Knowledge Management Strategies in Research and Industry

But in spite of a favor for using a codification strategy in software architecture, in the software engineering field there are very few reports of organizations that have opted for a codification strategy where the developed knowledge repositories have been used to a large extent over time [9]. The apparent issues with and discontinuation of various codification efforts in industry reported in Section 2.2 further illustrate the difficulty of sustained exploitation of knowledge repositories. So the question is: is codification really *the* answer?

Kankanhalli et al. [17] have developed a model to explain the use of knowledge repositories, a central element in the codification strategy. Their study indicates that social incentives are important in order to increase knowledge repository usage. This is a topic which is commonly ignored in the software engineering and software architecture literature, which tends to focus on technical issues related to knowledge codification. Moreover, some knowledge is difficult or impossible to codify, and there can be problems interpreting knowledge if sufficient context information is not codified as well. A personalization strategy can stimulate discussion within a company and may be more appropriate in certain situations.

Our investigation of current approaches in architectural knowledge management suggests that there is an ongoing trend in industry of increasing awareness of the need for a strategic choice for architectural knowledge management and, simultaneously, increasing efforts to

express such knowledge. This results in a movement from unintentional personalization (UP) - used by most organizations - to intentional codification (IC). This situation is graphically depicted in Fig. 1.

However, the reported IC initiatives from industry all seem to encounter their own problems. Current research tries to alleviate such problems by inventing structured approaches to architectural knowledge management, using an IC strategy. In the meantime, both industry and research seem to ignore (or forget about) the existence of intentional personalization (IP).

Codification has the potential to contribute to better management of architectural knowledge, given that the research field takes into consideration how important social aspects are to get such systems into use. There are good reasons for many organizations to opt for a personalization approach in order to facilitate innovative solutions with minimal bureaucracy. However, choosing between codification and personalization need not be a black or white choice. A combination of the two strategies, what Desouza et al. [7] call a hybrid approach, probably suits typical architecting activities best.

Although it is beyond the scope of this paper to discuss how a hybrid approach can best be utilized in software architecture practice, we do want to share some of our ideas on this topic. Notwithstanding many other factors – such as organizational size or geographic separation – that also influence the best choice of strategy, we believe two phases of architecting should be distinguished. In the first phase, the decision making process is a rather unstructured process in which the architectural solution space is explored and ideas are coined. While this phase could already very much benefit from codified knowledge (such as architectural styles, patterns, and tactics), this phase seems to be particularly suited for a personalization strategy. Although it is important that architectural knowledge such as expertise, options under consideration, and the like can be located when needed, this need not necessarily be through structured knowledge repositories. In this phase, meetings between architects (and other stakeholders) and ad-hoc communication by means of for instance email discussions are little intrusive and pose no limitations on the options that can be considered. Architects might feel such limitations when forced to codify each and every option they consider.

In the second phase, the design space is outlined by approved architectural decisions, and a stable architectural design emerges. This phase lends itself for a rationalization of the earlier 'unstructured' decision making process. Traces from the first phase, present in for instance emails, discussion fora, and meeting minutes, can be used to 'reconstruct' the rationale for the current architectural design. In this phase, the rationalized decision making process can be codified by making explicit for instance the architectural decisions taken and other options considered (and rejected). This codified knowledge can then easily be consulted throughout the remainder of the development project – and possibly even be reused in similar future projects.

We urge researchers to be clearer on addressing in which contexts they think their work will be appropriate. Over time, this enables our community to identify in which circumstances the various architectural knowledge management approaches can best be applied.

5. Conclusion and Future Work

In this preliminary review of various approaches to managing architectural knowledge, we have discovered that both research and industry have an intentional focus on the codification strategy. However, seeing that most intentional codification efforts in industry suffer from various shortcomings, most organizations seem to rely on unintentional personalization as the primary architectural knowledge management strategy. This suggests that there is a gap between what researchers are working on and the practice in the industry. We think both personalization and codification serve purposes for managing architectural knowledge, and both strategies as well as hybrid combinations should be investigated in more detail in the future. Furthermore, the industry should be aware of the implicit choices they make in architectural knowledge management, and strive for an explicit, intentional, choice of strategy. As a final remark, although we have investigated 'technocratic' (i.e. codification) and 'behavioral' (i.e. personalization) aspects of architectural knowledge management, 'economic' aspects have not been touched upon at all in this paper. This warrants future research in the direction of commercialization of architectural knowledge, i.e. the protection and exploitation of architectural knowledge to produce revenue streams.

Our ongoing work in this research domain focuses on two main aspects. First, we plan to conduct a more extensive review on architectural knowledge management in both research and industry, with which we aim to provide a more complete picture of the current state of the art and state of the practice in this area. Second, we see the need for guidelines that support organizations in increasing their awareness of knowledge management strategies. We intend to devise such guidelines that should assist in a potential transition from the current unintentional personalization strategy and intentional codification strategy to a more hybrid

approach that includes an intentional focus on both personalization and codification.

Acknowledgment

This research has been partially sponsored by the Dutch Joint Academic and Commercial Quality Research & Development (Jacquard) program on Software Engineering Research via contract 638.001.406 GRIFFIN: a GRId For inFormatIoN about architectural knowledge. Muhammad Ali Babar was working with National ICT Australia when he contributed to this manuscript based on his involved in the development and trial of PAKME by DSTO.

References

[1] M. Ali Babar, I. Gorton, and R. Jeffery. Toward a Framework for Capturing and Using Architecture Design Knowledge. Technical report unsw-cse-tr-0513, The University of New South Wales, June 2005.

[2] J. Bosch. Software Architecture: The Next Step. In F. Oquendo, B. Warboys, and R. Morrison, editors, *1st European Workshop on Software Architectures (EWSA)*, volume 3074, LNCS, pages 194–199, St. Andrews, UK, 2004. Springer-Verlag.

[3] R. Capilla, F. Nava, S. Pérez, and J. C. Dueñas. A Web-based Tool for Managing Architectural Design Decisions. In *1st ACM Workshop on SHaring ARchitectural Knowledge (SHARK)*, Torino, Italy, 2006.

[4] V. Clerc, P. Lago, and H. van Vliet. Assessing a Multi-Site Development Organization for Architectural Compliance. In *6th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, Mumbai, India, 2007.

[5] R. C. de Boer, R. Farenhorst, J. S. van der Ven, V. Clerc, R. Deckers, P. Lago, and H. van Vliet. Structuring Software Architecture Project Memories. In *8th International Workshop on Learning Software Organizations (LSO)*, Rio de Janeiro, Brazil, 2006.

[6] R. C. de Boer and H. van Vliet. Constructing a Reading Guide for Software Product Audits. In *6th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, Mumbai, India, 2007.

[7] K. C. Desouza, Y. Awazu, and P. Baloh. Managing Knowledge in Global Software Development Efforts: Issues and Practices. *IEEE Software*, 23(5):30–37, 2006.

[8] T. Dingsøyr and R. Conradi. A Survey of Case Studies of the Use of Knowledge Management in Software Engineering. *International Journal of Software Engineering and Knowledge Engineering*, 12(4):391–414, 2002.

[9] T. Dingsøyr and E. Røyrvik. An Empirical Study of an Informal Knowledge Repository in a Medium-Sized Software Consulting Company. In *25th International Conference on Software Engineering (ICSE)*, pages 84–92, Portland, USA, 2003.

[10] J. C. Dueñas and R. Capilla. The Decision View of Software Architecture. In *2nd European Workshop on Software Architecture (EWSA)*, Pisa, Italy, 2005.

[11] A. H. Dutoit and B. Paech. Rationale Management in Software Engineering. In S. K. Chang, editor, *Handbook of Software Engineering and Knowledge Engineering*, volume 1. 2001.

[12] M. Earl. Knowledge Management Strategies: Toward a Taxonomy. *Journal of Management Information Systems*, 18(1):215–233, 2001.

[13] D. Falessi, M. Becker, and G. Cantone. Design Decision Rationale: Experiences and Steps Ahead Towards Systematic Use. In *1st ACM Workshop on SHaring ARchitectural Knowledge (SHARK)*, Torino, Italy, 2006.

[14] R. Farenhorst, P. Lago, and H. van Vliet. Prerequisites for Successful Architectural Knowledge Sharing. In *18th Australian Software Engineering Conference*, Melbourne, Australia, 2007.

[15] M. T. Hansen, N. Nohria, and T. Tierney. What's Your Strategy for Managing Knowledge? *Harvard Business Review*, 77(2):106–116, 1999.

[16] A. Jansen and J. Bosch. Software Architecture as a Set of Architectural Design Decisions. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 109–120, Pittsburgh, USA, 2005.

[17] A. Kankanhalli, B. C. Y. Tan, and K.-K. Wei. Contributing Knowledge to Electronic Knowledge Repositories: An Empirical Investigation. *MIS Quarterly*, 29(1):113–143, 2005.

[18] P. Kruchten, P. Lago, and H. van Vliet. Building up and Reasoning about Architectural Knowledge. In *2nd International Conference on the Quality of Software Architectures (QoSA)*, 2006.

[19] J. Liebowitz and T. J. Beckman. *Knowledge Organizations: What Every Manager Should Know.* CRC, 1998.

[20] R. McDermott. Why Information Technology Inspired But Cannot Deliver Knowledge Management. *California Management Review*, 41(4):103–117, 1999.

[21] D. E. Perry and P. S. Grisham. Architecture and Design Intent in Component & COTS Based Systems. In *International Conference on COTS Based Software Systems*, Orlando FL., 2006.

[22] T. Prabhakar. A Knowledge Management Perspective of Software Architecture. WICSA Keynote, 2007.

[23] J. Swan, H. Scarbrough, and J. Preston. Knowledge Management - The Next Fad to Forget People? In *7th European Conference on Information Systems*, Copenhagen, Denmark, 1999.

[24] J. Tyree and A. Akerman. Architecture Decisions: Demystifying Architecture. *IEEE Software*, 22(2):19–27, 2005.

[25] J. S. van der Ven, A. Jansen, J. Nijhuis, and J. Bosch. Design decisions: The Bridge between Rationale and Architecture. In A. Dutoit, editor, *Rationale Management in Software Engineering*, pages 329–346. Springer-Verlag, 2006.