

University of Groningen

Wishes and Boundaries for a Software Architecture Knowledge Community

Lago, Patricia; Avgeriou, Paris; Capilla, Rafael; Kruchten, Philippe

Published in:
EPRINTS-BOOK-TITLE

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2008

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Lago, P., Avgeriou, P., Capilla, R., & Kruchten, P. (2008). Wishes and Boundaries for a Software Architecture Knowledge Community. In *EPRINTS-BOOK-TITLE* University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Wishes and Boundaries for a Software Architecture Knowledge Community

Patricia Lago
VU Univ.
Amsterdam,
The Netherlands
patricia@cs.vu.nl

Paris Avgeriou
Univ. Groningen,
The Netherlands
paris@cs.rug.nl

Rafael Capilla
University Rey
Juan Carlos,
Spain
rafael.capilla@urjc.es

Philippe Kruchten
Univ. British Columbia,
Canada
pbk@ece.ubc.ca

Abstract

Software architecting is a highly knowledge-intensive process demanding and producing a large and rich amount of information. To remain competitive, companies and organizations working in the IT sector must be able to manage this knowledge portfolio and effectively exploit and reuse it. In the era of Web 2.0, knowledge grids, social networking, global development and semantic web, this working session addresses the problem of building a knowledge community in the field of software architecture. To this end, we aim at exploring the wishes of academics and industrial organizations, on the one hand, and their boundaries on the other. Our goal is to compare and contrast the inputs from academia and industry, and gain a shared understanding about what can be done now, and in the near future.

1. Introduction

Software architecting is a highly knowledge-intensive process [13] demanding and producing a large and rich amount of information. To remain competitive, companies and organizations working in the IT sector must be able to manage this knowledge portfolio [4] and effectively exploit and reuse it. If not, expensive resources and significant investments must be allocated to e.g. recollect forgotten information, and go over previously decided (or discarded) architecture design options for which the rationale has been lost in the gears of memory, time, or which left the company [12].

The knowledge about software architecture is rich and multi-faceted, being domain- and organisation-specific, crossing cultural and geographic boundaries, and reflecting the creativity, initiative, experience and

investments of its creators.

In the last decade, the software architecture field underwent a number of evolution steps. Firstly, software architecture was considered as a blueprint to be used for further development [14], modeling one software system in terms of computational components and the interactions among them. Secondly, the notion of software architecture changed to guide the construction of multiple systems [2] and to address the concerns of multiple stakeholders via a multi-view approach [9]. More recently, the need to manage multiple configurations of the same type of systems at the same time, resulted in software product lines, or families of systems, for which each individual product has its own product architecture, and this must comply with the more generic (and shared) properties defined in a common architecture [11].

The next evolution step represents a fundamental change in the notion of software architecture. By recognizing that architects are mainly decision makers, software architecture can be seen as a set of design decisions [3]. This definition covers a broader perspective of the profession, trying to solve problems like governance of know-how within the company. Recent work addresses an even broader definition, which focuses on the complete knowledge produced around the software architecting process, and its integration [1, 6, 7, 10, 13].

The software engineering community, both in industry and academia, is therefore gradually acknowledging capturing and codifying architectural knowledge as an especially critical task. Furthermore, the codified architecture know-how is seen as a valuable company asset, and therefore needs to be transferred in knowledge bases. This approach would allow for efficient sharing and reusing of architecture knowledge. However, this codification strategy does not work in practice. The people involved in the architecting process (who

own the knowledge) often do not document it [8]: they lack the motivation to document and maintain architecture knowledge as the benefits do not seem substantial enough to justify the effort; they have a short-term interest in the project and are thus uninterested in the long-term architectural knowledge reuse; they sometimes make decisions without realizing it or without reflecting explicitly upon them, as they are absorbed in the creative flow of design; they are not used to, or trained in documenting their decisions. To make matters worse, even when the architecture knowledge *is* documented, it is often not sufficiently shared within the organization: the knowledge is not disseminated to the appropriate stakeholders; the recipients of knowledge do not use it in their own tasks either intentionally or because there is no provision in the processes; it is cumbersome to search and locate the appropriate knowledge and adapt it in one's needs etc.

Sharing knowledge is not easy to achieve, in particular in distributed and global projects. Some companies that participate in virtual communities like inner- or open-source communities are starting to realize the challenges of sharing the architectural knowledge between the communities. In specific, the vision of a knowledge grid for making architectural knowledge available within the distributed teams turns out to be a rather cumbersome goal.

These issues can only be resolved by architectural knowledge management strategies that support the knowledge producers in efficiently documenting the knowledge and the consumers in using it. These strategies are required to convince the management that they will receive a worthwhile return of investments, as well as the employees that they can better perform their daily work.

We mention here two promising industrial examples of such strategies. The first comes from a large international organization offering business consulting, systems integration and IT&business process outsourcing solutions worldwide. At the Dutch conference on software architecture 2007, they presented an architecture utilizing peer-to-peer, web service technologies & standards, and community-building principles to support autonomous communities in industrial contexts. The idea is to re-think on-line business as open communities. The architecture can offer a variety of business services and business, integration and communication patterns. The business actors can freely join a community, regulated by the patterns they select (e.g. business pattern offer-request, integration pattern scatter-gather) and by the services they want to use (from e-shopping to consulting or back-office management). This initiative introduces innovation in combining the

openness of virtual communities, the flexibility of peer-to-peer, and the necessary regulation via well-known and trusted patterns.

A second example addresses knowledge as a valuable, industrial business asset. A large international company with activities in the healthcare, lifestyle and technology domains, recently re-focused the way they develop software according to its observed *commodification*: for most products, only a small part (5 to 10 percent) of the software is differentiating. The remainder is more or less a commodity. Effective and efficient software development only focuses on producing the differentiating parts. This motivates the industry in turning part of their software into OSS and in actively participating in open communities to influence the development of the commodity software to be integrated with their products. Here architectural knowledge is used to reason about the components to turn into commodity, and to re-engineer the whole product and enterprise architectures.

This working session aims at discussing the issues around the effective and successful management of software architectural knowledge. The discussion will provide a first step towards understanding the problems and potential solutions of a *knowledge community in the field of software architecture*. In particular, assuming that we are willing to create such a global knowledge community, we want to address two main questions:

- What is the relevant architectural knowledge we want (and *can*) share?
- How can we share architectural knowledge?

The working session will attempt to answer these questions in three steps:

- We will first outline and select the topics for discussion by prioritizing on the industrial needs and wishes, as we aim at focusing on the problem of architectural knowledge (AK) management in the real world.
- We will then split into two working groups, industry and academia, to discuss the topics. In this way, the participants from the industry can express the challenges and promises they see in such a knowledge community, the industrial obstacles, and the existing solutions and current practice. In a similar way, the participants from academia can provide their perspective, together with the technological and research promises, results and challenges.
- At the end, a plenary session will put the pieces together, so that the two perspectives can be compared and contrasted.

By comparing the two visions on a knowledge community in the field of software architecture, we can hopefully gain a realistic understanding about how to proceed in the dream of building such a community, and a research agenda for future collaboration.

2. Discussion points

As pointed out in Section 1, the first main question is, “*What is the relevant architectural knowledge we want (and can) share?*”. This can be further detailed as:

- Is it possible to define a shared body of knowledge about software architecture?
- Is it possible to standardize the meta-models for architecture knowledge through a generic core meta-model?
- How to bridge the gaps between the different architecture knowledge meta-models of various organizations?
- What are the different categories of architectural knowledge? Is generalized, domain-, organization-, or project-specific architectural knowledge a good categorization?
- What is the boundary of architecture knowledge in the problem space and in the solution space?

The second question is, “*How can we share architectural knowledge?*”. This can be further refined into:

- How can we deliver or make accessible the *right* knowledge to the *right* person at any given point in time? [5]. And what is the right knowledge?
- How can we realize the necessary knowledge management strategies?
- Can we build a common knowledge base for a web community?
- What can the community contribute by populating the knowledge base?
- Do managers and architects realize the significance of storing such architectural knowledge for their architectural projects?
- How can we motivate the knowledge producers to put the effort in documenting their knowledge?
- How open can the knowledge base be? Public access or restricted to communities?
- What is the aim, goals and intended audience of such a knowledge base?
- Can we initiate a wiki-based encyclopedia for architectural knowledge?
- What are the relevant issues in populating such architectural knowledge encyclopedia?

3. Possible contributions

We foresee the following possible inputs from the industrial and academic participants.

3.1. Input from the Industry

- industrial experience, successes and failures (e.g. in distributed architecting or global development)
- economics of architectural knowledge documentation and sharing
- industrial experimentation (including case studies and pilot projects)
- existing meta-models of architectural knowledge used formally or informally
- use cases currently applied in practice
- empirical evaluation and measurement of architectural knowledge in organizations
- domain-specific architectural knowledge (including views/viewpoints, aspects, etc.)
- domain-specific architectures

3.2. Input from Academia

- meta-models and ontologies for architectural knowledge modeling
- discovery tools and techniques
- elicitation tools and techniques
- sharing tools and techniques
- integration of tools and techniques with current software engineering processes
- conceptual issues in sharing and reusing architectural knowledge
- standardization of architectural knowledge concepts through international standards (e.g. IEEE 1471)

4. Expected Results

We expect several results from the working session:

1. The initial establishment of an architecture knowledge community that is comprised of stakeholders from both industry and academia. The working session will form and maintain the links between the community members through the WICSA Wiki initially and later on through its own web portal.
2. A report detailing the current state of the art with the relevant problems, the main challenges companies need to overcome, and the solutions able to meet the current needs of modern software architects. This report may be used in at least the following two ways:

- (a) As a research agenda that is comprised of: an overview of the state of practice; a list of challenges and needs of industry partners; future research directions to meet those challenges in order to improve current architecting practices within the *software architecture knowledge community*.
- (b) As a “political” argument: demonstrating the economics of AK management in an organization and making architectural knowledge available to the community can serve to convince managers about the expected benefits of introducing and using such architectural knowledge.

5. Related efforts

This working session is one among a number of initiatives towards the more efficient documentation and sharing of architectural knowledge in the software architecture field. Other related initiatives are:

- The IFIP WG 2.10 [www.softwarearchitecture-portal.org], aims to further the practice of software architecture by integrating software architecture research and practice. The aspects of software architecture within the working group’s scope are: identifying common problems encountered by practitioners, investigating notations, languages, techniques, tools, and methodologies for improving the practice of software architecture, and training, education, and certification of software architects.
- The IASA repository [www.iasahome.org] provides contents and resources created by architects for architects. IASA tries to create a global architecture community of practicing architects and allow that body to drive standards, according to their needs and issues. The benefits to identify and unify common perspectives in the industry instead of standalone practices of IT architects is highly recommended to face similar problems by sharing architectural knowledge. Hence, the IASA provides an appropriate forum to meet the challenges of the profession.
- The Architecture Handbook by Grady Booch [www.booch.com/architecture] includes a collection of architecture documentation of many software systems, forming a rich library of diverse systems, representing many different domains.
- Previous SHARK’06 and SHARK/ADI’07 workshops [www.cs.rug.nl/~paris/SHARK-ADI2007] have brought a number of open issues and solutions for supporting architectural knowledge and

design rationale in software architecture. Various papers published in these two workshops summarize a significant part of the current state of the art and efforts made in the field. Several authors have contributed proactively for the success of the past SHARK workshops.

References

- [1] P. Avgeriou, P. Kruchten, P. Lago, P. Grisham, and D. Perry. Architectural knowledge and rationale: issues, trends, challenges. *SIGSOFT Softw. Eng. Notes*, 32(4):41–46, July 2007.
- [2] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. SEI Series in Software Engineering. Addison-Wesley, second edition, 2003.
- [3] J. Bosch. Software architecture: The next step. In *European Workshop on Software Architecture*, pages 194–199, 2004.
- [4] P. Clements, R. Kazman, M. Klein, D. Devesh, S. Reddy, and P. Verma. The duties, skills, and knowledge of software architects. *Proceedings of the Working IEEE/IFIP Conference on Software Architecture*, 0:20, 2007.
- [5] N. Conteh, G. Forgionne, W. Schulte, and K. O’Sullivan. The Merits of a Just-In-Time Knowledge Management (JITKM) Approach to Decision-Making Support. *Journal of Information & Knowledge Management*, 5(4):269–277, 2006.
- [6] R. C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A. Jansen. Architectural knowledge: Getting to the core. In *Third International Conference on the Quality of Software Architectures (QoSA)*, pages 188–206, 2007.
- [7] M. T. Hansen, N. Nohria, and T. Tierney. What’s your strategy for managing knowledge? *Harvard Business Review*, 77(2), 1999.
- [8] N. B. Harrison, P. Avgeriou, and U. Zdun. Using patterns to capture architectural decisions. *IEEE Software*, 24(4):38–45, 2007.
- [9] IEEE Computer Society. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, 2002. IEEE Std-1471-2002.
- [10] P. Lago and P. Avgeriou. First workshop on sharing and reusing architectural knowledge. *SIGSOFT Softw. Eng. Notes*, 31(5):32–36, 2006.
- [11] K. Pohl, G. Böckle, and F. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, Sept. 2005.
- [12] I. Rus and M. Lindvall. Guest editors’ introduction: Knowledge management in software engineering. *IEEE Software*, 19(3):26–38, 2002.
- [13] M. Shaw and P. Clements. The golden age of software architecture. *IEEE Software*, 23(2):31–39, 2006.
- [14] M. Shaw and D. Garlan. *Software architecture: perspectives on an emerging discipline*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.