A Just-In-Time Architectural Knowledge Sharing Portal

Rik Farenhorst, Ronald Izaks, Patricia Lago, Hans van Vliet
Department of Computer Science
VU University Amsterdam, the Netherlands
{rik, crizaks, patricia, hans}@cs.vu.nl

Abstract

In recent years, management of architectural knowledge has become a more prominent theme in software architecture research. Although various specialized tools have been proposed for use in the architecting process, observations show that architects in industry have yet to meet a tool environment that matches their knowledge needs. In order to discover what architectural knowledge needs architects have, we conducted a study in a large organization. In this study we discovered that architects are especially in need for 'Just-In-Time architectural knowledge'. To fulfill this need we designed and implemented an architectural knowledge sharing portal. Our portal's integrated functionality supports architects in their decision-making process, by providing easy access to the right architectural knowledge at any given point in time.

1. Introduction

In recent years, knowledge management has started to play an increasingly prominent role in software architecture. Researchers have proposed various tools that support architects in managing design decisions [7, 16], rationale [2, 10, 20], and other knowledge pertaining to the architecture, all aggregated in the concept of 'architectural knowledge' [9]. However, success stories about the use of these tools in industry are yet to be reported. The main problem seems to be a misalignment between the knowledge managed by these tools and what architects in practice really need for their daily tasks, which could be due to the specialized nature of these tools.

To address this problem, in our research we elicit what architects typically work on and what specific architectural knowledge needs are associated to these activities. To this end, we have conducted action research in the architecture department of a large Dutch software development organization. We assessed the architects' satisfaction with existing tools that support knowledge sharing in this organization, followed by the identification of their requirements for an improved tool environment.

We found that architects are not particularly concerned with specialized architectural knowledge reflected in meta-models, templates or process guidelines. Instead, they seemed primarily interested in support for *'Just-In-Time (JIT) architectural knowledge'*, which we define as access to and delivery of the right architectural knowledge, to the right person, at any given point in time. Such architectural knowledge may include updates on major decisions made or discussions held, but also contact information or expertise of important stakeholders. Since architecting is such a knowledge-intensive decision-making process, JIT architectural knowledge is vitally important for architects to ensure high-quality results.

Based on the requirements for JIT architectural knowledge we designed and implemented a web-based architectural knowledge sharing portal. This portal harbors various types of architectural knowledge, which can be easily retrieved using a number of integrated codification and personalization techniques. Experimentation with our portal indicated that the portal fulfills the architects' needs and that it is a definite improvement over the existing tools in their organization.

The remainder of this paper is organized as follows. In Section 2, related work is discussed. In Section 3, we outline our research design, which essentially is an action research cycle consisting of a diagnostic stage and a therapeutic stage. The results of the diagnostic stage, in which we identify requirements for architectural knowledge sharing tool support, are discussed in Section 4. Results of the therapeutic stage, which focuses on the construction of an architectural knowledge portal, are described in Section 5. In Section 6, we report on the results of experimentation with the portal. We conclude in Section 7 with a discussion about the contribution of our portal and research in general.

2. Related Work

Many practitioners and researchers of the knowledge management community argue that instead of browsing numerous documents and other knowledge sources, ideally people like to compile, capture and receive a smaller and readily digestible volume containing only the really relevant knowledge needed at that moment. The concept of furnishing or making accessible the right knowledge to the right person at any given point in time is known as *"Just-In-Time Knowledge Management"* [8]. The importance of Just-in-Time knowledge is further stressed by Kerschberg and Jeong, who argue that effective decision-making demands that the decision-makers are able to "sift and winnow through the mountains of data to find the right knowledge nuggets at the right time" [17].

Access to and delivery of relevant knowledge at the right time is particularly important for software architects. This need for Just-in-Time knowledge follows from the fact that software architecting inherently is a decision-making process. Over the past few years, this insight has matured in the software architecture domain, triggered by a position paper of Jan Bosch [5]. He argues that we should change the traditional component and connector view on architecture, and start viewing an architecture as the composition of a set of architectural design decisions. Following Bosch' view on software architecture, various researchers have been focusing on knowledge pertaining to the architecture, such as design decisions [15], and their rationale [21]. Establishing ways to manage such 'architectural knowledge' is considered to be one of the key challenges the field of software architecture faces [19], and has resulted in the birth of a workshop series about this topic [3,18].

During their decision-making process, architects are in a constant need for access to relevant architectural knowledge in order to make well-founded design decisions. Architects often maintain, implicitly or explicitly, a 'backlog' of smaller needs, issues, problems they need to tackle, and ideas they might want to use in the architecting process [12]. This backlog drives the workflow, helping the architect to determine what to do next. We argue that working on the backlog demands support for JIT architectural knowledge, i.e. access to and delivery of the right architectural knowledge, for the right person, at any given point in time. This way, architects can better discuss open issues, inform other stakeholders, or retrieve specific expertise.

Support for JIT architectural knowledge can be eased by using tools, so that it becomes easier to sift through the vast amounts of architectural knowledge available. Over the past few years, several tools have been proposed to support knowledge sharing in the architecting process, most of which focus specifically on managing architectural design decisions [7, 16] and rationale [2, 10, 20]. All these tools follow a typical codification strategy, which aims to systematically store knowledge in predefined formats so that it can be easily found and reused. However, in order to support access to other kinds of architectural knowledge, such as expertise or experience of colleagues, codification alone does not suffice; architectural knowledge that is hard to articulate is easier shared using a personalization strategy [1]. When using this latter strategy, not the knowledge itself, but information about its source or 'owner' is stored, after which they can use their personal network to share knowledge.

The importance of personal networks in knowledge sharing is also noted by Huysman and Wulf [14]. They conducted extensive studies on practices of knowledge sharing in industry, and they found that when sharing experience, people prefer to look for support from personal networks rather than from electronic networks to gain knowledge about the knowledge. This way, the experience – or other tacit knowledge – does not need to be transformed into explicit knowledge to share it. They argue that knowledge sharing tools should provide an infrastructure for establishing, maintaining or intensifying relationships in communities. Translating this requirement to the architecting process, we argue that JIT architectural knowledge is best supported by tools that not only codify important architectural knowledge, but also help stakeholders to find each other, so that architectural knowledge can be shared using personalization techniques as well.

3. Research Design

The main question we want to answer in our research is what architects' specific architectural knowledge sharing needs are, and how best to fulfill these needs. To this end, we analyzed the architecting process of the central architecture department of NPK, a large Dutch IT organization. This architecture department assists various business lines with developing and maintaining software architectures. Although architects at NPK have access to several tools that support sharing architectural knowledge, they struggle with how best to use them in their daily work.

We have conducted action research at NPK. The essence of this type of research can be described as a two stage process, consisting of a diagnostic stage that involves a collaborative analysis of the current situation, followed by a therapeutic stage that covers collaborative change experiments to improve this sit-

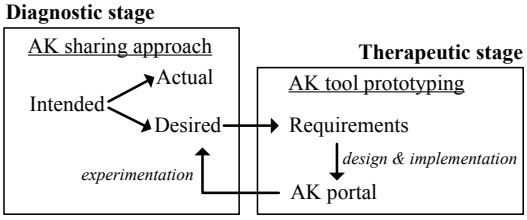

Figure 1. Research Design: Action Research Cycle

uation [4]. The action research cycle we followed is schematically depicted in Figure 1.

The main objective of the diagnostic stage was to analyze how architectural knowledge sharing can best be supported in NPK. For this analysis we used a user-centered design method, which is designed around the assumption that people usually consider it easier to indicate what they dislike, instead of only pointing out positive aspects [13]. Since the architects at NPK already have architectural knowledge sharing tools at their disposal, we were able to use this method to analyze the quality of these tools and identify possible room for improvement. Our analysis consisted of three consecutive steps, which are elaborated in turn below.

1. An analysis of the **intended** approach to architectural knowledge sharing. Over the past few years, four different tools have been introduced at NPK's architecture department to support architectural knowledge sharing. To elicit the original requirements of these tools, we have interviewed four managers from this department who have been responsible for introducing the tools, and who are now responsible for their maintenance. Based on these interviews we could determine how these tools should ideally support architectural knowledge sharing. The results of this first step are elaborated in Section 4.1.

2. An analysis of the **actual** approach to architectural knowledge sharing. The first step helped us to determine how the existing tools should ideally support the architects in sharing architectural knowledge. In this second step, we verified how well the intended support is actually perceived by the architects themselves. The results of this step are elaborated in Section 4.2.

3. An analysis of the **desired** approach to architectural knowledge sharing. In the second step the architects indicated the limitations and issues of the existing architectural knowledge sharing tools. In this third analysis step the same architects were explicitly asked how this situation could best be improved, i.e. how they perceive the ideal tool support for sharing architectural knowledge. This elicitation helped us to identify a set of desired features, based on which we were able to identify a number of requirements that future tools supporting architectural knowledge sharing should meet (see Figure 1). All these requirements are further elaborated in Section 4.3.

The therapeutic stage of our action research cycle, discussed in depth in Section 5, consisted of the design and implementation of an architectural knowledge sharing portal. Input to the design of this portal were the requirements distilled during the last step of the diagnostic stage. To assess the contribution of our portal we let the architects at NPK experiment with it. The key results of this experimentation exercise are elaborated upon in Section 6.

4. Diagnosis: Architectural Knowledge Sharing Approaches in Practice

In this section we elaborate upon the diagnostic stage of our action research cycle. In the following three subsections we respectively discuss the analysis results of the intended, actual and desired approach to architectural knowledge sharing in NPK.

4.1 Intended Approach to Architectural Knowledge Sharing

In this first analysis step we analyzed the four different tools available in NPK to support architectural knowledge sharing. We interviewed the four managers who have been responsible for introducing the tools, in order to retrieve the original requirements of these tools. We classified these requirements as depicted in Figure 2. In the remainder of this section we discuss the main requirements in more detail.

- **Best practice repository.** NPK has developed a knowledge repository that is primarily intended to support the construction of architectural descriptions. This support requires that the architects are offered guidance in their decision-making process. The repository allows storing architectural best practices so that these can be reused in future projects. Example best practices include references to conflicts between technology platforms, reference architectures from customers, or trade-offs between quality criteria. After answering a number of predefined questions, the architect

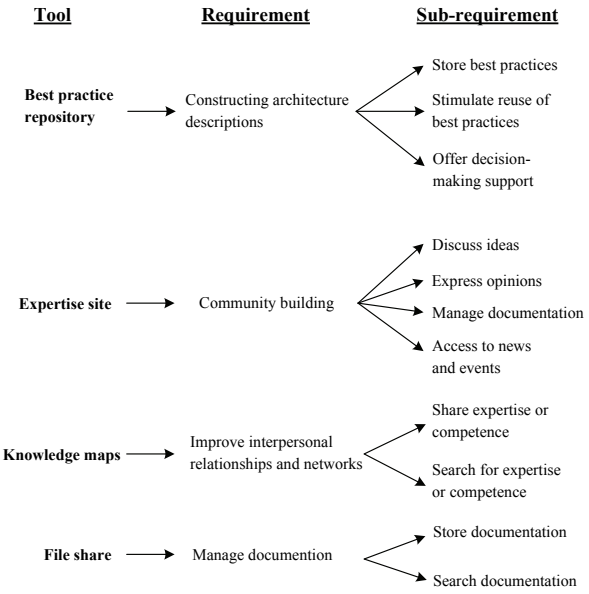

Figure 2. Intended Approach to AK Sharing

is assisted by the repository, which uses its best practice to advise the architect about the architectural solution. During the interview with the manager responsible for the repository, we elicited that reusability of architectural knowledge is envisioned as main strength of the repository. Reusing best practices helps architects to more efficiently arrive at the most suitable architectural solution.

- **Expertise site.** This intranet website uses Microsoft Sharepoint as underlying technology. Its main purpose is to support community building among the architects at NPK. Four sub-requirements were identified during the the interview held with the manager of the Expertise website: the ability to discuss ideas, the possibility to express opinions, a means to manage internal and external documents, and access to news, events, or other external information sources.

- **Knowledge maps system.** NPK has also developed an organization-wide knowledge maps system that aims to connect knowledge and knowledge workers. To meet this requirement, the system offers a place where users publish their expertise with respect to architecture-related topics, by filling in detailed user profiles. Users can use these profiles to search for colleagues with specific expertise or competences.

- **File share.** In addition to the other three – more specialized – architectural knowledge sharing tools, the architecture department of NPK uses a standard file share to manage all documentation. The original requirements of this system mentioned in the interview with its manager are nothing more than storing and searching for documents that contain relevant information for the architects.

4.2 Actual Approach to Architectural Knowledge Sharing

In this analysis step we interviewed eight architects from the total of 15 within the architecture department of NPK. This selected group of interviewees included junior and senior architects with various specialisms. The requirements and sub-requirements identified during the previous step acted as starting point for these interviews. We asked the architects what they liked and – more importantly – what they disliked about the requirements of the four existing tools for architectural knowledge sharing.

The architects were not really satisfied with the best practices repository. Due to a very non-intuitive user interface and low performance, using the tool is a time-consuming task. Moreover, in its current form the tool does not offer much support to decision-making. Although it offers storage for best practices, it does not indicate to the user which best practice is best to follow in a particular situation. Architects therefore see little value in the current implementation of the tool. In addition, the reusability of the repository is low because the content is outdated, and because adding or modifying the best practices is also time-consuming and error-prone, the costs for keeping the content up-to-date outweigh the benefits.

The Expertise site did also not particularly please the architects. This site, which is built as an intranet website, is not well accessible and its performance on NPK's network is low, too. As a result, the Expertise site is not often visited by the architects. Consequently, new discussion topics are seldom started, because architects doubt whether anybody will read them anyway. Another main problem of this tool is its non-intuitive user interface, which makes publishing knowledge on the site especially cumbersome. Architects therefore often resort to traditional communication means, such as email or phone, to communicate their ideas and experience.

The architects were particularly harsh on the knowledge maps system. In their opinion the main problem with this tool is that it lacks efficient search mechanisms. Consequently, the architects consider it difficult to quickly find the right knowledge workers within

the organization. Likewise, they doubt whether their knowledge profile would be read often by colleagues. Due to the perceived low return on investment, architects often skip filling in such a profile, which was deemed a very time-consuming process, too.

The file share is used as the primary way of document management in NPK. Nevertheless, the architects are not really positive about its implementation. The architects' main problem with this tool is not in storing the documents (this is done using the standard Windows Explorer in Windows), but in retrieving them. Except for a standard folder structure there is no way to add meta-data. Moreover, the standard search functionality in Windows is not very flexible, which makes retrieving the right document a painful task.

In addition to the issues specific to the four existing tools, the architects reported one major problem of the current situation: the abundance of different information sources. As a result, architects have difficulty to easily retrieve specific architectural knowledge needed at a particular point in time, because they do not know where to start looking. It is not clear which source to trust more. A lack of trust and overview also results in a lack of motivation of contributing architectural knowledge to these knowledge sharing tools. After all, where can you best publish your knowledge?

From the above analysis we conclude that there is quite a mismatch between the intended and actual use of the four tools. All four tools have specific flaws that hinder widespread success and the lack of integration between the tools confuses architects which tools to use in which situation.

4.3 Desired Approach to Architectural Knowledge Sharing

During the interview round with the eight architects we also elicited their desired way of sharing architectural knowledge. We followed a similar approach as while identifying the intended approach to architectural knowledge sharing (see Section 4.1), only this time we focused on what the architects consider important requirements for any (future) architectural knowledge sharing tool. These requirements are further decomposed into sub-requirements whenever possible, after which we ranked them in order of importance based on how often they were mentioned by the interviewees. The resulting ranked classification of requirements is depicted in Figure 3.

Figure 3. Desired Approach to AK Sharing

1. **Integration.** The requirement considered most important by the architects is that a tool environment should offer a central point of access to the various types of functionality available. This central point of access should be both attractive and intuitive. Attractiveness is key in the sense that it increases the chance for the portal's widespread adoption. Intuitiveness decreases the learning curve and makes using the tool fun. In addition, the architects noted that customization is important, because not all users have the same knowledge needs. Depending on current experience or interests, you might want to adapt the content shown or the user interface itself to your liking. Another requirement mentioned is that if new architectural knowledge emerges, a notification should be sent. This improves the overview users have on newly published architectural knowledge, which keeps them up-to-date. Finally, to further add to the integration strength, the architects also mentioned the need for links to external information sources, such as white papers, seminars and trainings, or other corporate communication.

2. **Project view.** The architects indicated that one major improvement for the current situation would be the support for a project view that enables management of project-specific architectural knowledge. The main advantage of such a project view is that it offers a central point of access to easily search all architectural knowledge related

to a particular project. For stakeholders that join a project at a later point in time, such a central point of access is helpful to quickly become acquainted with the ins and outs of the project. A sub-requirement that follows from this search requirement, however, is that the maintainability of documents is high. In addition, architects required that the project view should contain information about the project stakeholders. This information may include standard personal contact information, but also more architectural knowledge related content such as expertise areas of people. Finally, the architects indicated a need for discussion board functionality to be used by project stakeholders, so that issues, design decisions or conflicts can be quickly communicated.

3. **Manage documentation.** Related to the previous category is support for managing documentation. The difference with the project view is that the scope may be (much) broader, including all sorts of company documents. As with the project view requirement, searching documents was considered of prime importance by the architects, since this is one of the things that is currently implemented poorly. Consequently, sufficient metadata has to be added to the documents in order to support intelligent search queries.

4. **Community building.** In contrast with the need for document management is the architects' wish to support building a community within their department. Although the architects acknowledge the power and importance of traditional conversations and meetings – both formal and informal – with respect to tool support they reckon it would be very helpful if there were facilities in place that help people to connect with each other. Consequently, requirements in this category include support for discussions and sharing expertise, but also overviews of 'who knows what' and 'who is doing what' in the organization. Finally, the ability to share news and events with colleagues would further add to the community feeling.

5. **Constructing architecture descriptions.** The last main category relates to one of the primary deliverables of the architects in NPK: architecture descriptions. These documents usually contain a variety of architectural knowledge, and usually take multiple days or weeks to construct. All sort of automated support during the process of making well founded decisions, followed by reflecting these decisions in the architecture description is highly appreciated.

If we compare the requirements classifications of the intended and desired approach to architectural knowledge sharing (see Figure 2 and 3), we can make a few interesting observations. First of all, the requirements related to *'constructing architecture descriptions'* are mentioned both in the intended approach as in the desired approach. Obviously, the architects still like the underlying concepts, but are unhappy with the way the current tools implement these concepts. Secondly, although *'manage documentation'* was already an original requirement, architects at NPK take a consumer perspective and desire more focus on access to stored documents, instead of just storing them. For *'community building'* we observe the opposite trend. Here, the architects put more emphasis on publishing architectural knowledge, such as ideas, news, and other information; something which was poorly implemented in the current Expertise site. In addition, the sub-requirement related to finding colleagues based on expertise or competence suggests that architects not only rely on codification mechanisms, but also desire personalization strategies to share architectural knowledge. This wish for 'hybrid' architectural knowledge sharing is further stressed by the *'project view'* requirements, that indicate a need for both codification (e.g. document management) and personalization (discussion boards) techniques. Finally, the desire for *'integration'* is something that was obviously overlooked when designing the four existing tools.

5. Therapy: An Architectural Knowledge Sharing Portal

The requirements identified in the previous section provide us with a good overview of the architectural knowledge sharing needs of the architects in NPK. The most important requirement is that of an integrated environment to share architectural knowledge. In addition to this need for integration, we conclude that architects are in need for what we defined earlier as Just-In-Time architectural knowledge. Requirement categories 2 till 5 of Figure 3 demand various mechanisms to get easy access to available architectural knowledge. As discussed in the previous section, a hybrid strategy is needed to support both codification and personalization of architectural knowledge. The project view and community building requirements further show the need architects have for a tool environment that supports them in using their personal networks. Meeting these requirements demands specific personalization techniques.

Based on the identified requirements we have designed and implemented a web-based architectural

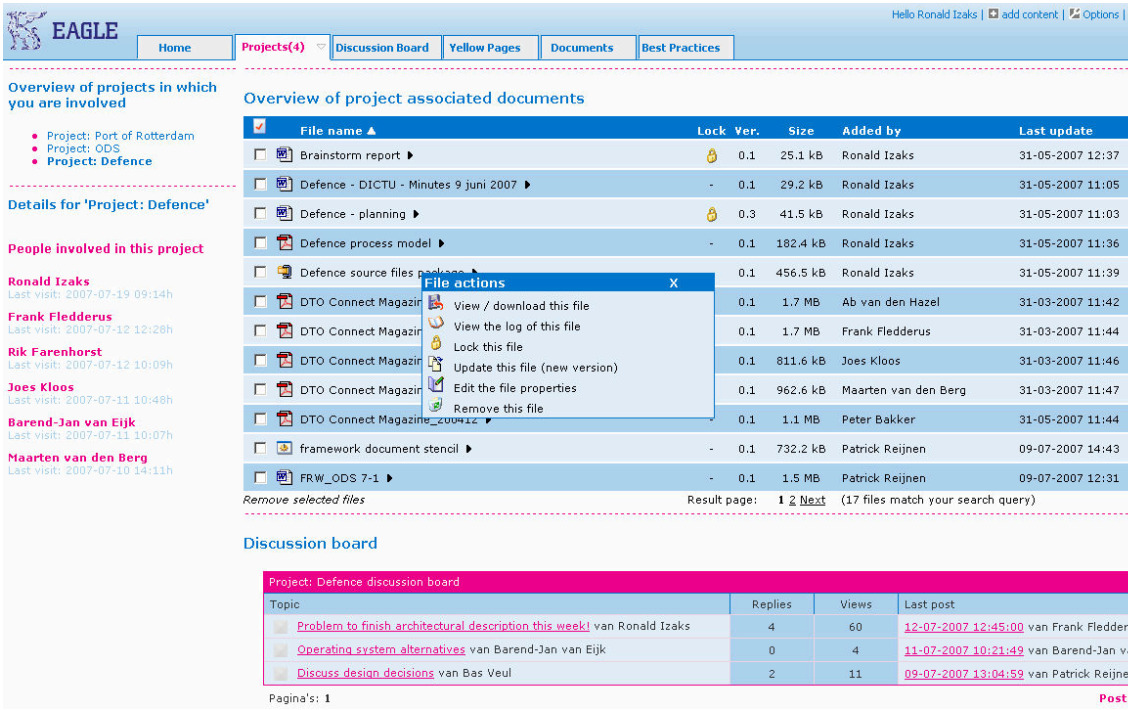

Figure 4. AK Portal: Project Environment

knowledge portal, of which a screenshot is shown in Figure 4. The portal is in essence a client server system; a web browser communicates with an Apache web server. Asynchronous communication between the client and server is applied whenever possible, to foster the speed and usability of the application. All architectural knowledge is stored in a relational database. Additional meta-data is stored with this architectural knowledge to make retrieval easier. For the client side of our portal, we selected a suitable open source framework: Portaneo [1], a Rich Internet Application. Portaneo is highly modifiable, has a flexible plugin system – making the portal highly extensible – and, above all, is free. These characteristics make it a better choice compared to existing commercial software such as Microsoft Sharepoint, because with Portaneo we are able to experiment more easily with the portal, while using little resources. For a more detailed discussion about the portal's architecture we refer to [11].

An important plugin of our portal is the project environment, which was one of the key requirements identified in Section 4.3. In a project environment, of which an example is shown in Figure 4, architectural knowledge is available in various forms, such as a list of the major project deliverables (center), a list of involved stakeholders (left), and an integrated discussion board where project stakeholders can discuss open issues (bottom). If necessary, access control measures can be used to ensure that only specific architects have access to the architectural knowledge stored.

Document management is supported in the portal by the document repository plugin. Instead of merely storing the documents, additional meta-data can be added to the underlying data model and documents can be classified using a tailored architectural knowledge category model that we designed together with the architects of NPK. Consequently, advanced search functionality is offered, such as searching for "all documents about Project X that have the status Final" or "all documents related to security written by John Doe". Using this search functionality, architects can quickly retrieve the documents that match their need.

Architectural best practices are stored in a repository that is added as a plugin to the portal. In this repository, architectural knowledge is codified in predefined formats, and could be retrieved for various purposes, such as reusing past design decisions, or to find out what best practices exist on a certain topic. In order to overcome the issues with the repository that were mentioned in Section 4.2, we are currently making the repository more intelligent, better maintainable, and better-looking. A first version of the improved repository is currently being trialled within NPK.

[1] http://www.portaneo.com/solutions/en/

Whereas the document repository and best practices repository are good examples of plugins that follow the architectural knowledge codification strategy, our portal also supports architectural knowledge personalization to fully comply to the community building requirements. To this end, the portal contains a 'yellow pages' plugin. On the yellow pages architects can get an overview of all other architects. By selecting the name, a more detailed information page is shown with personal information and contact information of that person. We are currently extending this information with more detailed information, such as the expertise areas of the architect, and which projects and activities he is assigned to. This allows retrieving knowledge about 'who is doing what', and 'who is knowing what' in the organization. Although at first sight this information is not directly pertaining to the architecture being designed, it can still be valuable information for architects, because it might tell them who to contact if they require help with specific architectural topics.

All the plugins mentioned above are accessible from the portal's start page. This start page acts as central point of access, and offers an intuitive user interface to ensure easy navigation. In addition, the portal also incorporates functionality to add personalized links to various information sources. Various RSS feeds can be loaded in the portal, allowing architects to access all sorts of non-architectural information via the portal as well, such as the daily news headlines, the weather report, etc. This coherence between all knowledge – architectural or not – is in line with the integration requirements identified in Section 4.3.

In addition to the main portal plugins described above, the portal has three main features, which will be elaborated upon below in turn:

1. **Integrated functionality.** Our portal offers a central access point to various types of functionality by means of a start page. From this start page all important functionality can be accessed by the architects by one mouse click, after which they can quickly retrieve the architectural knowledge they need, using codification techniques, personalization techniques, or a combination.

2. **Stakeholder-specific content.** The portal offers an intuitive and attractive user interface. Since architects are already familiar with web pages, navigating the portal is easy. Both the user interface and the content can be customized by architects. Different architects can thus focus on different types of architectural knowledge. A lead architect supervising a project for example would be mainly interested in what all architects are currently working on, and what their specific expertise areas are. A security architect on the other hand wants to be kept posted on specific developments in his domain, so he would be interested in documentation, discussions or news feeds related to this topic.

3. **Notifications and subscriptions.** The portal has a built-in subscription and notification system. Architects can subscribe to specific architectural knowledge topics (e.g. a topic of a discussion forum) or artifacts (e.g. a document). As soon as relevant architectural knowledge is published (e.g. another architect posts a message on the forum) or changed (e.g. a document expires or is replaced by a newer version) a notification is sent to all subscribed architects.

We argue that the above three features together ensure that our portal offers support for what we defined as JIT architectural knowledge. The integrated functionality provides access to *'the right architectural knowledge'*. The support for stakeholder-specific content ensures that *'the right person'* finds what he wants. Finally, the subscription and notification mechanisms allow architects to stay up-to-date by delivering the relevant architectural knowledge to them when needed.

6. Experimentation

In order to assess the value of our architectural knowledge portal, we let 11 architects of NPK experiment with it. Among these 11 architects were the eight we had interviewed in our diagnostic stage, plus three additional ones. These latter three architects were included because we deemed them as more objective, so that the assessment results are even more representative for the whole population.

The experimentation consisted of executing predefined scenarios that mapped on the requirements identified in Section 4.3. The architects had to execute each scenario using the portal (e.g. the scenario *"retrieve the newest version of the technical design of Project X, using the document repository plugin."*), after which they had to give scores for the implementation using a 5-points Likert scale. When comparing the scores from the three new architects with those of the eight others, we did not see any significant differences. The main results of the experiment are discussed below.

In its current form the portal is already an improvement over the existing tools that were in place in NPK. Most architects (82%) indicated that the document management properties of the portal are an improvement over the existing fileshare. Because of the cate-

gorization model and metadata that can be added to documents, retrieving documents is much easier. However, architects mentioned that a change in mindset is required before everyone is used to the new way of uploading and tagging documents.

Although the majority of architects (91%) was particularly fond of the integration aspects of the portal, in which document management, project environments, discussion boards and personal contact information is integrated, they wanted the portal to integrate even more with existing tools of the department, such as email clients (send emails to colleague, send invitations for meetings, attach documents to emails, store documents from emails in the repository), calendars (todo lists in the portal), or project tools (assign people to tasks or activities using the portal).

All architects liked the way the combination of the notifications and subscriptions of the portal work. They deemed it considerably useful to stay up-to-date on architectural knowledge available that might be of interest. The fact that the portal has different types of notifications (e.g. 'document expired', 'new forum post') is highly appreciated, and the fact that architects are free to subscribe to architectural knowledge reflected in various ways (e.g. news, discussion boards, documents) is liked as well. Some architects mentioned that the notification and subscription system might also add to the attractiveness of the portal, in the sense that architects are motivated to visit it on a regular basis (to see if new relevant architectural knowledge is present). It therefore appears that our portal is to a certain extent 'sticky' to its users, which is considered an important prerequisite for successful adoption of knowledge management tools in practice [6]

The portal's emphasis on providing access to the organization's vast amount of architectural knowledge is appreciated by all the architects. This portal in its current form supports access to and delivery of the "*right architectural knowledge on the right time*", and leaves sufficient freedom to the architects on how to visualize this knowledge. Apart from this support for JIT architectural knowledge, the portal emphasizes the social capital, i.e. supporting sharing in a community as opposed to individually consuming knowledge. As a result, by improving collaboration between architects of NPK, the portal is a good first step to create a real 'community of architects'.

7. Conclusions

In this paper we have investigated what are the typical architectural knowledge needs of architects at a large software development organization, and how these needs can best be fulfilled. By following an action research cycle we have identified five main requirements for an architectural knowledge sharing environment: 1) integration, 2) project views, 3) manage documentation, 4) community building, and 5) constructing architecture descriptions. Based on these requirements we have concluded that architects are best supported by an integrated tool environment that supports Just-In-Time architectural knowledge.

To meet the above requirements we have designed and implemented an architectural knowledge sharing portal. Main features of our portal include integrated functionality to retrieve architectural knowledge, support for stakeholder-specific content, and a notification and subscription system. Architects can use the portal to connect to colleagues or other involved stakeholders by retrieving 'who is doing what' and 'who knows what'. In addition, codified architectural knowledge in a document repository or best practice repository can easily be accessed using advanced search mechanisms. Finally, collaboration is explicitly supported by the portal's discussion board and project environment.

By offering an integrated environment for architects that incorporates various functionality to easily get access to available architectural knowledge, we argue that our portal is able to deliver the continuous flow of relevant information that architects need when working on their backlog [12]. Moreover, our portal's functionality to share architectural knowledge follows a hybrid architectural knowledge sharing strategy, combining both codification and personalization techniques [1]. Because explicit attention to personalization is incorporated in the portal, our portal supports architectural knowledge sharing by focusing on social capital [14].

A last important characteristic of our portal is the balanced focus on architectural knowledge consumption (i.e. retrieving architectural knowledge), and production (i.e. publishing architectural knowledge). This characteristic distinguishes our portal from existing architectural knowledge sharing tools, which often focus solely on the producing side [3].

Experimentation with our portal further indicated that it is already a definite improvement over the existing tools within NPK. Nevertheless, as future work we plan to extend our portal with additional plugins that further ease sharing of architectural knowledge. One of these plugins features the use of Wikis and Blogs allowing architects to easily produce architectural knowledge. On the technology side we will investigate how well existing commercial tools, such as Microsoft Sharepoint 2007^{TM}, support our architectural knowledge sharing requirements, and how such tools could be employed to further improve our portal.

A final insight, of which the software architecture research community should take notice, is the fact that architects – at least the ones at NPK – apparently are not particularly interested in very 'specialized' architectural knowledge support, such as detailed metamodels, templates or process guidelines. They seem to already have their techniques and processes in place to design and maintain software architectures. What *is* important, however, is the facilitating support during their everyday decision-making process by means of continuous access to and delivery of relevant architectural knowledge.

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.

References

[1] M. Ali Babar, R. C. de Boer, T. Dingsøyr, and R. Farenhorst. Architectural Knowledge Management Strategies: Approaches in Research and Industry. In *Second Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI)*, Minneapolis, USA, 2007.

[2] M. Ali Babar and I. Gorton. A Tool for Managing Software Architecture Knowledge. In *2nd Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI)*, Minneapolis, USA, 2007.

[3] P. Avgeriou, P. Kruchten, P. Lago, P. Grisham, and D. Perry. Architectural Knowledge and Rationale - Issues, Trends, Challenges. *ACM SIGSOFT Software Engineering Notes*, 32(4):41–46, 2007.

[4] R. L. Baskerville. Investigating Information Systems with Action Research. *Communications of the AIS*, 2(3es), 1999.

[5] J. Bosch. Software Architecture: The Next Step. In *1st European Workshop on Software Architectures (EWSA)*, pages 194–199, St. Andrews, UK, 2004.

[6] A. A. Bush and A. Tiwana. Designing Sticky Knowledge Networks. *Communications of the ACM*, 48(5):66–71, 2005.

[7] R. Capilla, F. Nava1, and J. C. Dueñas. Modeling and Documenting the Evolution of Architectural Design Decisions. In *2nd Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI)*, Minneapolis, USA, 2007.

[8] N. Y. Conteh, G. Forgionne, W. D. Schulte, and K. J. 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.

[9] R. C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A. Jansen. Architectural Knowledge: Getting to the Core. In *3rd International Conference on the Quality of Software-Architectures (QoSA)*, Boston, USA, 2007.

[10] 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.

[11] R. Farenhorst, P. Lago, and H. van Vliet. EAGLE: Effective Tool Support for Sharing Architectural Knowledge. under submission.

[12] C. Hofmeister, P. Kruchten, R. L. Nord, H. Obbink, A. Ran, and P. America. Generalizing a Model of Software Architecture Design from Five Industrial Approaches. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 77–86, Pittsburgh, USA, 2005.

[13] J. F. Hoorn. *Software Requirements: Update, Upgrade, Redesign. Towards a Theory of Requirements Change.* PhD thesis, VU University Amsterdam, the Netherlands, 2006.

[14] M. Huysman and V. Wulf. IT to Support Knowledge Sharing in Communities, Towards a Social Capital Analysis. *Journal of Information Technology*, 21:40–51, 2006.

[15] 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.

[16] A. Jansen, J. S. van der Ven, P. Avgeriou, and D. K. Hammer. Tool Support for Architectural Decisions. In *6th Working IEEE/IFIP Conference on Software Architecture*, Mumbai, India, 2007.

[17] L. Kerschberg and H. Jeong. Just-in-Time Knowledge Management. In *Professional Knowledge Management*, volume 3782, pages 1–18. Springer Berlin / Heidelberg, 2005.

[18] P. Lago and P. Avgeriou. 1st Workshop on SHaring and Reusing ARchitectural Knowledge, Final Workshop Report. *ACM SIGSOFT Software Engineering Notes*, 31(5):32–36, 2006.

[19] M. Shaw and P. Clements. The Golden Age of Software Architecture. *IEEE Software*, 23(2):31–39, 2006.

[20] A. Tang, Y. Jin, and J. Han. A Rationale-Based Architecture Model for Design Traceability and Reasoning. *The Journal of Systems and Software*, 80(6):2007, 2007.

[21] 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.