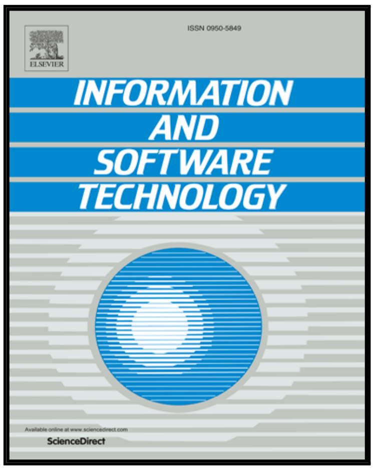

Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review

Rainer Weinreich, Iris Groher

PII: S0950-5849(16)30170-7
DOI: 10.1016/j.infsof.2016.09.007
Reference: INFSOF 5765

To appear in: *Information and Software Technology*

Received date: 11 December 2015
Revised date: 2 September 2016
Accepted date: 22 September 2016

Please cite this article as: Rainer Weinreich, Iris Groher, Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review, *Information and Software Technology* (2016), doi: 10.1016/j.infsof.2016.09.007

Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review

Rainer Weinreich[a,*], Iris Groher[a]

[a] *Johannes Kepler University Linz, Austria*

Abstract

Context: Numerous approaches for Software Architecture Knowledge Management (SAKM) have been developed by the research community over the last decade. Still, these approaches have not yet found widespread use in practice.
Objective: This work identifies existing approaches to SAKM and analyzes them in terms of their support for central architecture knowledge management activities, i.e., capturing, using, maintaining, sharing, and reuse of architectural knowledge, along with presenting the evidence provided for this support.
Method: A systematic literature review has been conducted for identifying and analyzing SAKM approaches, covering work published between January 2004 and August 2015. We identified 56 different approaches to SAKM based on 115 studies. We analyzed each approach in terms of its focus and support for important architecture knowledge management activities and in terms of the provided level of evidence for each supported activity.
Results: Most of the developed approaches focus on using already-captured knowledge. Using is also the best-validated activity. The problem of efficient capturing is still not sufficiently addressed, and only a few approaches specifically address reuse, sharing, and, especially, maintaining.
Conclusions: Without adequate support for other core architecture knowledge management activities besides using, the adoption of SAKM in practice will remain an elusive target. The problem of efficient capturing is still unsolved, as is the problem of maintaining captured knowledge over the long term. We also need more case studies and replication studies providing evidence for the usefulness of developed support for SAKM activities, as well as better reporting on these case studies.

Keywords: software architecture, software architecture knowledge management, software architecture knowledge management activities, software architecture knowledge management approaches, systematic literature review

☆Supplementary material to this review including a description of the identified approaches and quality assessment is available at `https://www.se.jku.at/akm-slr-supplementary-material`

*Corresponding author
Email addresses: `rainer.weinreich@jku.at` (Rainer Weinreich), `iris.groher@jku.at` (Iris Groher)
URL: `www.se.jku.at/weinreich` (Rainer Weinreich), `www.se.jku.at/groher` (Iris Groher)

1. Introduction

Software architecture has been a major research topic over the last two decades. Early definitions of software architecture appeared in the early 1990s (e.g., [1] and [2]). Within the past decade, the view of software architecture has been considerably extended. While early views of software architecture emphasized the structure or structures of a software system and their externally visible properties [3], newer views have started to include the design decisions leading to these solution structures and their rationale [4]. This growing importance of a decision-oriented view of software architecture as opposed to a solution-oriented view also led to Software Architecture Knowledge Management (SAKM) [5] as a still very active sub-discipline of software architecture research.

SAKM deals with identifying and leveraging architectural knowledge within software develpoment (see Section 2). Architectural knowledge, in this case, encompasses not only project-specific knowledge such as as requirements, design decisions, and their rationale but also project-generic knowledge such as experience, expertise, patterns, and architectural tactics. If not explicitly managed, critical knowledge remains tacit knowledge, which erodes over time [5]. Many different approaches have been developed and documented to manage software architecture knowledge (see [6] for some examples). An SAKM approach in this context is any concept, method, tool, and technique (or any combination thereof) that has been specifically developed to support SAKM within software development.

Although SAKM has been a topic of research for more than a decade, formal approaches for SAKM, especially such for capturing decisions and their rationale, have not yet been established in practice. A recent study we performed with 25 software architects from 22 different companies in 10 different countries [7] showed that none of the survey participants use or used a formalized approach to documenting architectural knowledge (AK) even though all participants agreed on the usefulness of capturing AK. Time and cost have been mentioned as the main limiting factors in this case. Another study with open source communities [8] found out that rationale is rarely part of architecture documentation, a potential reason being limited incentives for developers to document AK. A study by Capilla et al. [9] identified lack of motivation and understanding, lack of adequate tools, too much effort in capturing AK, not knowing what to capture, and disruption of the design flow as barriers that hinder widespread adoption of SAKM in practice.

This raises the questions of whether and how existing SAKM approaches have contributed to improvements in this area, along with why these approaches have not yet seen broad application in practice despite more than 10 years of research. To answer these questions, we performed a comprehensive, systematic literature review (SLR) to identify and analyze existing SAKM approaches. We wanted to know how current approaches support SAKM and, in particular, which knowledge-management activities (i.e., capturing, using, maintaining, sharing, and reuse) are well supported and where support is currently lacking.

The contribution of this paper is a comprehensive overview of existing approaches on SAKM developed over the last decade and their focus in terms of the different SAKM activities. Three main aspects distinguish our work from previous, similar work is this area. First, we identify approaches for SAKM based on published studies by aggregating related studies that considered the same approach or variations thereof. Second, for each approach, we determine the exact knowledge-management activity or set of activities supported by that approach, and we also determine the focus of the research performed. Finally, we identify the provided evidence in terms of empirical studies related to a specific activity and not related to an approach in general. This allows us to consider the state of evidence for a particular knowledge-management activity as opposed to the evidence for the approach as a whole.

The main results are a comprehensive picture of the current state of SAKM research in terms of the developed approaches, their main concepts and ideas, the supported knowledge-management activities, and the evidence provided for these activities. Our study shows that core problems SAKM aims to solve, like efficiently capturing rationale and maintaining knowledge over the long term, still remain unsolved. Support and evidence for other activities besides using AK is often weak. Our study also shows that, as researchers in the field, we need to improve our reporting of empirical evidence, the number of empirical studies performed to validate an approach, and the number of studies that replicate earlier results.

The remainder of this paper is structured as follows. In Section 2, we provide an overview of the basic concepts of SAKM and the different SAKM activities. Section 3 presents the main goal and the research questions we answer based on the results of our SLR. In Section 4, we describe the research method we followed during the SLR, and we provide details for each of the steps performed. In Section 5, we present a quality assessment of the studies included in our review and answer the research questions based on the data extracted from the selected studies. We discuss the results and their implications in Section 6. In Section 7, we discuss threats to the validity of this SLR. Section 8 presents related work. The paper is concluded in Section 9.

2. Software Architecture Knowledge Management

The concept of knowledge management was first studied from an organizational perspective [10][11], from which knowledge management refers to identifying and leveraging the collective knowledge in an organization [12].

The two main strategies for knowledge management are codification and personalization [13]. Codification focuses on systematizing and storing information, making this information available for reuse, while personalization focuses on providing information about the sources of knowledge within an enterprise.

In general, knowledge management can largely be regarded as a process involving various activities, including at a minimum creating, storing/retrieving, transferring, and applying knowledge [14]. This process is called the *knowledge management cycle* [15]. Depending on the domain and the overall objective, the knowledge management cycle and the activities related to knowledge management are slightly different, but certain core stages are part of a typical knowledge management cycle, as pointed out by Dalkir [15]: knowledge capture and/or creation, knowledge sharing and dissemination, and knowledge acquisition and application. Knowledge capture/creation concerns the development of new content or the replacement of existing content based on tacit or explicit knowledge. Knowledge sharing and dissemination refers to transferring knowledge. Knowledge acquisition and application refers to using the knowledge in an organization to gain a competitive advantage (see also [14]).

Software Architecture Knowledge Management (SAKM), as a special type of knowledge management [16], is concerned with identifying and leveraging software architectural knowledge within an organization. According to [17] software architectural knowledge can be categorized based on four dimensions: whether it is tacit or explicit [11], and whether it is specific to or independent from a particular application (i.e., generic). The resulting four dimensions are tacit/application-generic (e.g., experience, expertise, skills), tacit/application-specific (e.g., goals, constraints, assumptions), explicit/application-generic (e.g., patterns, reference architectures, ADLs), and explicit/application-specific (e.g., requirements, design decisions, rationale, views). Based on a systematic review on architectural knowledge, Farenhorst and de Boer [17] further identify four different views on architectural knowledge: pattern-centric view, dynamism-centric view, requirements-centric view, and decision-centric view. From these views they identified the decision-centric view as the most dominant view of approaches for SAKM with most of the approaches that have been developed since 2008 being based on this view. Potential reasons are that decisions have gained in importance for defining software architecture since 2004 [4] and that they also seem to be the linking pin between the other views [17].

We define an SAKM approach as any concept, method, tool, or technique (or any combination thereof) that has been specifically developed to support SAKM and SAKM activities within software development. Note that in general, also approaches that have not been developed specifically for SAKM can be used for realizing SAKM activities, but we do not include such approaches in our consideration.

In terms of knowledge-related activities, SAKM approaches follow the core stages of the knowledge management cycle, which includes as its main activities the capture, sharing, and application (use) of software architecture knowledge. To this we add maintenance in order to specifically address the long-term aspect of architectural knowledge and reuse, which highlights the application of captured knowledge in different (project) contexts.

We use the above-described activities as a basis for our analysis of existing SAKM approaches, defining each as follows:

- *Capturing* makes architectural knowledge explicit by formalizing and storing it;
- *Maintenance* refers to the activity of keeping documented knowledge up-to-date over time;
- *Sharing* concerns distributing the documented knowledge among different stakeholders and in different contexts;
- *Application (Using)* refers to using the knowledge in different architecture-related activities, such as analysis and review of architecture; and
- *Reuse* refers to using architecture knowledge from one project in another project.

AK activities supported by recently developed SAKM approaches as discussed by Capilla et al. [9], such as collaborative decision-making and decision assessment, can be classified as sub-activities of the main SAKM activities described above and are thus not treated separately in our study.

3. Goal and research questions

The overall goal of this literature review is to identify existing SAKM approaches and to determine their support for architecture knowledge management activities, along with the evidence provided for this support. Accordingly, we formulate the following three research questions (RQs):

- **RQ1:** Which approaches to SAKM have been developed by the research community within the last decade?
- **RQ2:** Which SAKM activities are supported by these approaches, and what has been the focus of research in terms of SAKM activities?

- **RQ3:** What evidence is provided for the supported SAKM activities (referring to RQ2)?

In **RQ1**, we identify approaches and families of approaches to SAKM by combining related studies obtained from the systematic literature review. The information from **RQ2** helps us to determine which SAKM activities are well supported by current research. In doing so, we can identify gaps in current approaches and needs for future research with respect to support for SAKM activities. With **RQ3**, we want to gain better insight into how well existing approaches have been validated both with respect to academic and industrial studies. The evidence provided for existing approaches is rated with respect to how each supports the SAKM activities. We look at each of the knowledge management activities separately when determining the evidence for their support. Section 4.4.4 explains in detail how we rated the evidence for each approach and activity.

4. Research method and search process

This study was undertaken as a systematic literature review, following the guidelines of Kitchenham [18], in order to identify the current state of research in SAKM by analyzing relevant papers published within the last 12 years. The study was planned in detail and documented in a review protocol in order to reduce researchers bias.[1] This protocol describes the research goals and questions, as well as the search strategy, selection criteria, and the process for data extraction and synthesis. The authors created, reviewed, and revised the protocol. The primary study was conducted from June 2013 to August 2015. The overall design of the performed review is illustrated in Figure 1.

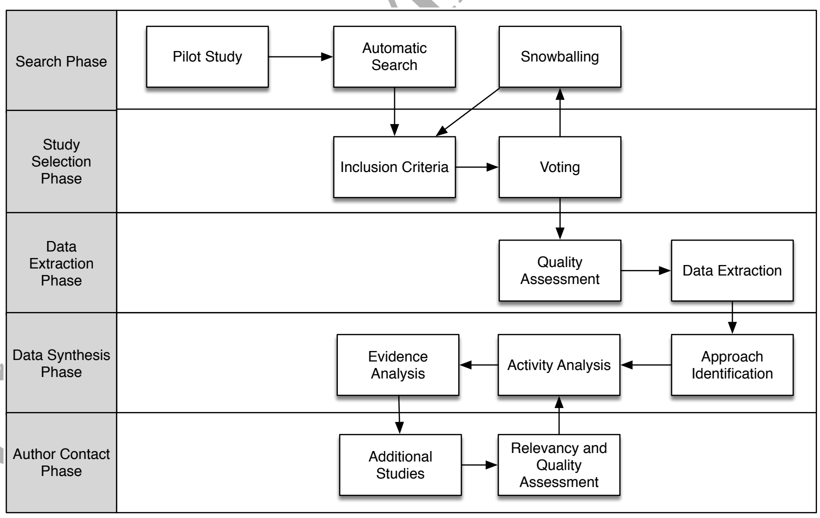

Figure 1: Design of the systematic literature review

The research process is divided into different phases and steps. In the search phase, we automatically searched different scientific databases using a suitable search string in order to retrieve

[1]The review protocol for our study can be found at `https://www.se.jku.at/akm-slr-review-protocol`.

potential papers for review (see Section 4.1). We optimized the search string by performing a pilot study, after which we performed an automatic search based on the refined search terms. We followed a defined study selection process (see Section 4.2). First, we filtered the publications obtained from the automatic search process by excluding papers according to various criteria. Then, four researchers voted based on paper titles and abstracts. Additional, relevant papers were found by applying a snowballing process (see Section 4.3). In the data extraction phase, all remaining papers were read in detail to extract the required information regarding the described approach and the quality of the paper (see Section 4.4). This step led to the further exclusion of some papers. From the final list of primary studies, papers relating to the same approach were grouped (see Section 4.4.2) to perform further analysis of the extracted data and to answer the research questions. We then analyzed the approaches in detail with respect to the supported SAKM activities and the evidence provided for their support of these activities (see Section 4.4.3 and Section 4.4.4). In a final step, we contacted the authors of the papers assigned to the different approaches to discuss the grouping of papers by approach and the assessment of each approach in terms of the supported activities and in terms of the evidence they provided for this support (see Section 4.5). During this last step we obtained additional, potentially relevant papers from the authors, which we reviewed and added to the list of primary studies where appropriate.

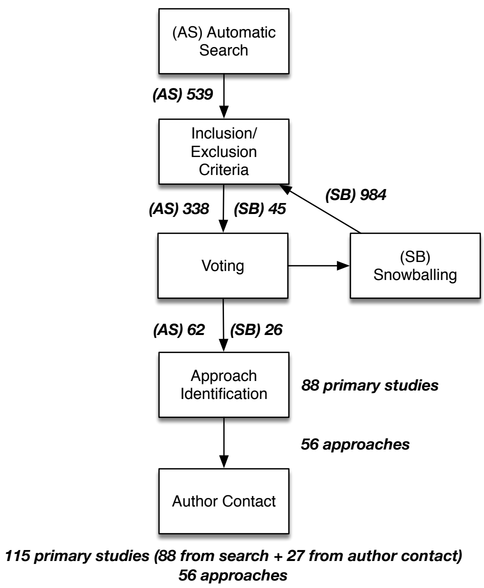

Figure 2: Search and selection process

4.1. Search process

To systematically search for scientific sources, we used an automatic search conducted in four different scientific databases: IEEE, ACM, Springer, and Elsevier. We used population, intervention, comparison, and outcome (PICO) criteria [18] to define the terms for the database search. The population in this SLR is software architecture, because we are only interested in knowledge-management approaches applicable to the area of software architecture. The intervention is architectural knowledge and any synonyms thereof. Comparison is not applicable in this case, as this SLR did not take a comparative approach. The outcome is knowledge representation or management and its synonyms, because in our study we focus on approaches that provide some kind of knowledge representation or support for knowledge management with respect to SAKM activities. These three parts (population, intervention, and outcome) are combined as follows:

”software architecture” AND ”%synonym for architectural knowledge%” AND ”%synonym for knowledge representation or management%”

As suggested by [18] and [19], population, intervention, and outcome are linked with ”AND” and the search terms (alternative spellings and synonyms) within population, intervention, and outcome are linked with ”OR” (see also Table 1 and Appendix A). For the databases that restrict the number of search terms, we broke the search string into substrings (”P AND I AND O” is broken into ”P AND I”, ”P AND O”, and ”O AND I”). [2]

To identify synonyms commonly used in the software architecture community, we manually searched through titles, abstracts, and keywords from different papers on SAKM. We initially identified the following synonyms for architectural knowledge: architectural knowledge; architecture knowledge; architecture decision; architectural decision; architecture decisions; architectural decisions; design issue; design issues; design decision; design decisions; and design rationale. We initially identified the following synonyms for knowledge representation or management: model; models; modeling; modelling; documentation; documenting; decision making; decision-making; decision process; capture; representation; reuse; sharing; recovery; reasoning; evaluation; analysis; and understanding.

To identify a suitable search string, we conducted a pilot study using four conference venues to verify and refine our initial key words, manually examining the four for relevant papers published from 2008 to 2013:

- WICSA – Working IEEE/IFIP Conference on Software Architecture (7th to 10th editions)
- ECSA – European Conference on Software Architecture (2nd to 6th editions)
- MODELS – International Conference on Model Driven Engineering Languages and Systems (2008 2012)
- QoSA – International Conference Series on Quality in Software Architectures (4th to 8th editions)

Through the pilot study, we discovered the following new synonyms for architectural knowledge: decision structure; design reasoning; architecture information; architectural information; knowledge management; and decision management. The following synonyms for representation or management were also identified in the pilot study: metamodeling; metamodelling; metamodel; metamodel; ontology; ontologies; and framework. Table 1 summarizes how the final search string was assembled from the different search terms.

Figure 2 illustrates the search and selection process. As shown in the figure, the automatic search in the four scientific databases mentioned yielded 539 publications. The next step is the selection of publications based on the defined inclusion and exclusion criteria.

[2]The search queries for the different databases can be found in the review protocol for our study at `https://www.se.jku.at/akm-slr-review-protocol`.

Table 1: Search string for automatic search in scientific databases

		OR		OR
software architecture	**AND**	architectural knowledge, architecture knowledge, architecture decision, architectural decision, architecture decisions, architectural decisions, design issue, design issues, design decision, design decisions, design rationale, decision structure, design reasoning, architecture information, architectural information, knowledge management, decision management	**AND**	model, models, modeling, modelling, documentation, documenting, decision making, decision-making, decision process, ontology, ontologies, framework, metamodel, meta-model, metamodeling, metamodelling, capture, representation, reuse, sharing, recovery, reasoning, evaluation, analysis, understanding

4.2. Selection of publications

We defined a number of criteria on the basis of which papers were included in the review or excluded from the review. The defined inclusion (I1-I5) and exclusion (E1-E2) criteria are as follows:

- I1: The study only includes publications available in electronic form.
- I2: The study only includes publications written in English.
- I3: The study only includes publications published since 2004.
- I4: The study only takes into account peer-reviewed publications appearing in books, journals, conferences, or workshops.
- I5: The study can contribute to answering RQ1, RQ2, or RQ3.
- E1: The study excludes introductions to special issues, workshops, tutorials, conferences, and conference tracks as well as editorials.
- E2: The study excludes PowerPoint presentations and short/extended abstract papers (less than five pages).

As shown in Figure 2, 338 publications (about 63% of the initial set of papers obtained in the automatic search) were left after excluding papers on the basis of the criteria defined above (I1-I4, E1-E2). Since this procedure leaves no scope for subjective opinions, a single researcher (PhD student) conducted this part of the selection process. All publications meeting inclusion criteria I1-I4 and not meeting exclusion criteria E1 and E2 proceeded to a voting stage to check I5, which involved four researchers. The four researchers are two senior researchers (the authors of this paper) and two PhD students (mentioned in the acknowledgements at the end of the paper). Based on the title and abstract of each publication, the reviewers rated whether or not the publication could contribute to answering the research questions (I5). To determine the overall inter-rater agreement between the four reviewers, we calculated Fleiss Kappa coefficient [20] based on the researchers initial ratings. The value was 0.78, which, according to Landis and Koch [21], means the reviewers were in substantial agreement. After rating, publications about which the researchers strongly disagreed were discussed by at least two researchers with opposing opinions. The voting stage identified 62 relevant papers (see Figure 2).

4.3. Snowballing

To identify additional relevant publications, we conducted one iteration of backward snowballing [22] in which we went through all references in the 62 selected studies and searched for potentially relevant publications. In total, these referenced 1728 publications. After removing multiple entries, 984 referenced publications remained. From these 984 references, we excluded all publications that did not meet the criteria (I1-I4, E1, and E2) introduced at the beginning of this section. The resulting 45 papers were subject to another voting stage to check for I5, which resulted in 26 newly identified papers. In total, 88 primary studies (62 from the automatic search and 26 from snowballing) were selected in the search and selection process as relevant for answering our research questions (see Figure 2).

4.4. Data collection/extraction process

Each publication from the final set was read in detail to extract information about both the quality of the publication (see Section 4.4.1) and specific approaches or concepts, like the support of models, processes or tools for different SAKM activities and the evidence for each provided in the publication (see Section 4.4.2).

For each paper, the data were extracted by one of the four reviewers, who read the publication in detail. Another reviewer checked the extracted data. The papers were distributed randomly among the four reviewers, who did not extract or check their own publications. To obtain the same level of knowledge with respect to the data to be extracted, we performed a first round of extraction to calibrate. Each reviewer extracted the data from one publication; all four reviewers discussed the data from each of these four publications. The data extraction forms were afterwards adjusted according to the results of this discussion.

4.4.1. Quality assessment

We defined seven criteria (see Table 2) to rate the quality of the publications. These criteria are based on the protocol for a published, systematic literature review [23] and a mapping study [24]. In addition, we added a criterion assessing the discussion of related work (Q4). The quality assessment and the data extraction process were conducted in parallel.

All quality questions were answered using a three-point-scale: ”no,” ”general description or discussion,” and ”explicit description or discussion.” Answers were assigned a numerical value, as follows, with the quality score defined as the sum of the numerical value for each question: ”no” = 0, ”general description or discussion” = 0.5, and ”explicit description or discussion” = 1.

4.4.2. Data extraction and synthesis

To answer our research questions (see Section 3), we read each publication from the final set of papers in detail, extracting specific data about the presented approach or concept. In addition to general information about the selected paper (such as its authors, title, year of publication, publication venue, research context, and relevance of the approach), we extracted data regarding

Table 2: Questions in the quality assessment

#	Question
Q1	Is there a description of the context in which the research was carried out?
Q2	Is there a problem definition for why the study was undertaken?
Q3	Do the authors describe the research design of the study?
Q4	Do the authors compare and evaluate their own results against related work?
Q5	Are the contributions of the study discussed?
Q6	Are insights derived from the study discussed?
Q7	Are limitations of the study discussed explicitly?

the representation of knowledge, the underlying SAKM model, the supported SAKM activities, and the integration of SAKM activities in the decision-making process, as well as validation and tool support. The complete data extraction form including the research questions answered by each category of extracted data is presented in Table 3. We grouped the questions for data extraction in the table into categories (D1-D9) to provide a better overview of the areas that are addressed by the questions.

After extracting the data from all selected primary studies, we grouped the papers by approach. Typically, more than one paper is published for an approach, each having a different goal or focus. In order to draw meaningful conclusions from our study regarding existing SAKM approaches and their support for the different SAKM activities, we aggregated the results from the individual papers by approach. In total, we identified 56 different SAKM approaches.

4.4.3. In-depth activity analysis

After aggregating the papers by approach, we performed an in-depth SAKM activity and evidence analysis for the identified approaches. Two senior researchers analyzed and discussed each approach (and all related papers) in detail. We first analyzed each approach with respect to which SAKM activities it focused on and/or supported, also examining how each activity was supported by the respective approach. The in-depth analysis also included incremental fine-tuning of the SAKM activities.

In order to get a better picture of the level of support provided for particular knowledge management activities, we subdivided support for each of the different activities into general and advanced support. General support, following basic data management, concerns Create/-Read/Update/Delete (CRUD) functionality for architectural knowledge. Advanced support extends basic data management in that it supports a particular knowledge management activity. Examples for advanced support are approaches that explicitly reduce the effort involved in capturing, provide dedicated data models for reuse, or support processes that aim to share architectural knowledge.

In the following, we briefly overview what we considered to be general and advanced support for individual activities. A more detailed definition of what constitutes general and advanced support for the different activities may be found in Appendix D.

Capturing. General support for capturing refers to basic means for capturing architectural knowledge using forms and templates. Advanced capturing refers to support that reduces the effort involved in capturing knowledge through dedicated process support or by using techniques for recovering knowledge from other sources.

Table 3: Data extracted from primary studies

#	RQ	Category	Description/Questions
D1	RQ1	About the publications	Brief description of the publication Research Context (industry, academic, product, industrial research lab, other) What is presented in the paper (approach, concept, idea or other)? Relevance of the approach (research, practice) Which application scenarios are mainly targeted by the approach? Notes about the approach
D2	RQ1	Architecture Knowledge Management Representation	How is architecture knowledge represented (formal, informal, semi-formal)? Is the approach based on a decision model? Which architecture knowledge is captured? Which relations exist within the elements?
D3	RQ2	Capturing Architecture Knowledge	How is knowledge capturing / documenting supported? How is capturing integrated in the decision making process?
D4	RQ2	Maintaining Architecture Knowledge	How is knowledge maintaining supported? Does the approach support the actuality of the documented architecture knowledge (tool or process support)? If yes, how is it supported? Does the approach support evolution of architecture knowledge from a historical perspective? If yes, how is it supported?
D5	RQ2	Sharing Architecture Knowledge	How is knowledge sharing supported? How is knowledge sharing integrated in the decision making process?
D6	RQ2	Using Architecture Knowledge	How is knowledge using supported? How is knowledge using integrated in the decision making process?
D7	RQ2	Reusing Architecture Knowledge	How is knowledge reuse supported? How is knowledge reuse integrated in the decision making process?
D8	RQ3	Validation	How is the approach validated? Is it validated in industry or research? Which SAKM activities have been validated? Size (e.g., number of participants, size of system) Limitations as reported Additional notes
D9	RQ1-2	Tool Support	Is there a tool support for the approach? Which SAKM activities are supported? Additional notes

Using. General support for using refers to basic means for browsing and understanding the stored knowledge by some form of visualization. Advanced support for using refers to dedicated support for specific application scenarios, such as architecture evaluation/review assessment, architecture analysis (e.g., impact analysis), or design reasoning/decision making.

Reuse. General support for reuse refers to basic means to search for previous decisions based on different criteria. Advanced support for reuse refers to dedicated means for separating general from project-specific knowledge in the underlying knowledge representation model, or to specific support in processes or tools for reuse.

Sharing. General sharing is supported if some form of central knowledge repository with remote access for different stakeholders is provided. Advanced sharing support refers to support for specific concepts, processes, or tools for distributing knowledge among stakeholders (e.g., support for notifications or personalization) beyond simply offering centralized access.

Maintaining. General support for maintaining refers to the ability to change the stored knowledge in the sense of create, update, and delete operations. Advanced support for maintaining refers to dedicated support beyond simple data management, such as detecting outdated knowledge, versioning of captured knowledge, or integrating maintenance operations directly into the knowledge management cycle.

4.4.4. Evidence assessment

After in-depth analysis of the supported knowledge management activities, as described in the previous section, we also assessed in detail how each of the 56 approaches was evaluated. Evaluation is explicitly rated with respect to the SAKM activities supported by each approach.

To rate the evidence, we adopted the following evidence hierarchy, as presented in [24]:

1. No evidence (rated 0.0).
2. Evidence obtained from demonstrations or working out toy examples (rated 0.2).
3. Evidence obtained from expert opinions or observations (rated 0.4).
4. Evidence obtained from academic studies, such as controlled lab experiments (rated 0.6).
5. Evidence obtained from industrial studies, such as causal case studies (rated 0.8).
6. Evidence obtained from industrial practice (rated 1.0).

4.5. Author contact

As a final step, we contacted the authors of the papers to discuss with each author whether we correctly mapped his or her papers to an approach. We also asked the authors to confirm our assessment of supported activities and provided evidence for their approach (or approaches in case of multiple approaches per author). We attached documents describing how we rated the evidence and how we defined the different SAKM activities and their sub-activities.[3]

We received answers from 30 authors, corresponding to 54% of all 56 approaches. All authors agreed with our assignment of papers to approaches and largely agreed with our assessment of the approaches' support for SAKM activities and evidence for that support. More than half

[3]The documents we sent to the authors can be found at `https://www.se.jku.at/akm-slr-documents-for-authors`.

of the authors that responded (18 authors) suggested additional papers for review, which typically included more case studies or support for additional SAKM activities. 27 new papers were identified during the author contact phase. Two senior researchers (i.e., the authors of this paper) checked the relevancy of each newly considered paper and performed quality assessment and data extraction as described in Sections 4.4.1 and 4.4.2. Two newly added papers did not meet the formal criteria, because they were shorter than 5 pages. We nevertheless added these papers, as the contacted authors explicitly classified them as relevant.

In total, we included 115 publications in our study, which we aggregated to a final set of 56 different SAKM approaches. The paper selection process and the resulting number of papers from each stage is summarized in Figure 2.

5. Study results

This section presents the results of search and selection. First, the studies are classified by publication year and venue. Second, the results of the assessment of the quality of the studies are presented. Finally, the three research questions (see Section 3) are answered based on these results.

5.1. Demographic data

In total, our SLR included 115 studies. Of these studies, 29 were published in journals, 60 were published in conference proceedings, three studies were book chapters, and 23 studies were published in workshop proceedings. Figure 3 shows the distribution of studies included in our SLR.

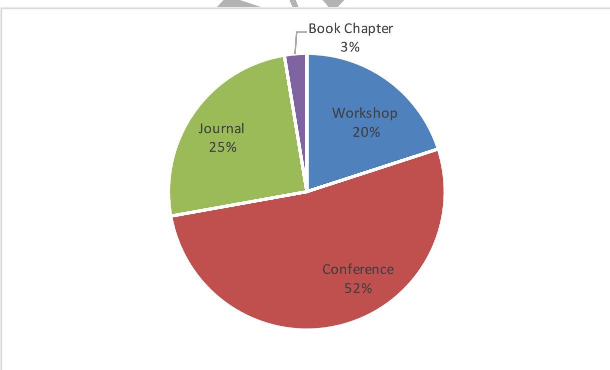

Figure 3: Type of publication

Table 4 shows the number of papers included in the SLR by publication venue. In all, the identified studies were published in 55 different venues. The Journal of Systems and Software (JSS) was the most prominent venue to publish the SAKM-related research reviewed here, followed by a workshop focusing on the topic of SAKM (SHARK/ADI). Conferences on software architecture (WICSA, WICSA/ECSA, ECSA) and software quality (QoSA, QSIC) were also attractive venues for SAKM research.

Table 4: Distribution of primary studies among publication venues

Venue	# of Studies
Journal of Systems and Software (JSS)	14
Workshop on Sharing and Reusing Architectural Knowledge: Architecture, Rationale, and Design Intent (SHARK/ADI)	11
Working Conference on Software Architecture (WICSA)	8
Joint Working Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA)	7
International Conference on Quality of Software Architectures (QoSA)	6
European Conference on Software Architecture (ECSA)	6
IEEE Software	4
International Conference on Quality Software (QSIC)	3
Symposium on Applied Computing (SAC)	3
Australian Software Engineering Conference (ASWEC)	2
International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS)	2
International Conference on Software Engineering (ICSE)	2
International Multitopic Conference (INMIC)	2
Science of Computer Programming	2
Workshop on Managing Requirements Knowledge (MARK)	2
International Conference on Software Engineering and Knowledge Engineering (SEKE)	2
Automated Software Engineering (ASE)	1
Canadian Conference on Electrical and Computer Engineering (CCECE)	1
Collaborative Software Engineering	1
Conference on Hypertext and Hypermedia (HT)	1
European Conference on Pattern Languages of Programming (EuroPLoP)	1
European Conference on Software Maintenance and Reengineering (CSMR)	1
Future Generation Computer Systems (FGCS)	1
Groningen Workshop on Software Variability Management	1
International Conference on Communications and Signal Processing (ICCSP)	1
International Conference on Computer Software and Applications (COMPSAC)	1
International Conference on Engineering of Complex Computer Systems (ECBS)	1
International Conference on Evaluation and Assessment in Software Engineering (EASE)	1
International Conference on Information Society (i-Society)	1
International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)	1
International Conference on Program Comprehension (ICPC)	1
International Enterprise Distributed Object Computing Conference (EDOC)	1
International Symposium on Empirical Software Engineering (ISESE)	1

Journal of Software	1
Software Architecture Knowledge Management	1
The Knowledge Engineering Review	1
Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE)	1
European Workshop on Software Architecture (EWSA)	1
Workshop on Software Evolvability	1
Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks)	1
Workshop on Traceability, Dependencies and Software Architecture (TDSA)	1
Workshop on Wikis for Software Engineering (WIKIS4SE)	1
Journal on SoftwarePractice and Experience	1
Journal of Software Maintenance	1
Computers in Industry	1
International Journal of Information Systems Modeling and Design	1
Relating System Quality and Software Architecture	1
Workshop on Software Engineering for Systems-of-Systems (SESoS)	1
Conference on Emerging Technologies & Factory Automation (EFTA)	1
Working Conference on the Practice of Enterprise Modeling (PoEM)	1
International Journal of Electronic Commerce	1
Workshop on Future of Software Architecture Design Assistants (FoSADA)	1
International Conference on Collaboration Technologies and Systems (CTS)	1
Latin American Computing Conference (CLEI)	1
	115

Figure 4 on the left shows the distribution of studies over the years from 2004 to 2014. We did not include the year 2015 in the figure, as only papers published until August were considered in our review. This figure shows a trend of increasing publications on SAKM approaches until 2008, after which publications on SAKM approaches are published on a nearly constant level with slight variations between the years. Other studies on similar topics (discussed in related work) also show that SAKM and related topics like design decisions and documentation are still an active research discipline in general (see Figure 4 on the right). All of them show a similar increase until 2008 with slight increases and decreases in the following years. The studies by Ding et al. [25] and Li et al. [24] identified a similar number of published papers per year as we did in our study. The mapping study by Tofan et al. [26] detected even a more significant upward trend and more papers per year, which we attribute to the broader scope of their study, which looks at studies on design decisions in general and not only in the context of approaches for software architecture knowledge management.

5.2. Quality assessment

Figure 5 shows the total quality scores for the studies and the distribution of total quality scores (see Section 4.4.1 and Section 2 in the supplementary material to this review). The mean score of all reviewed studies is 3.7 (with a maximum of 7.0). The right side of Figure 5 shows the frequency of the total quality scores. Most studies received a score between 2.0 and 5.5, with a peak at 3.5. Given that score, we conclude that the overall set of studies are of medium quality. The left side of Figure 5 shows the total quality scores of the primary studies. This figure again

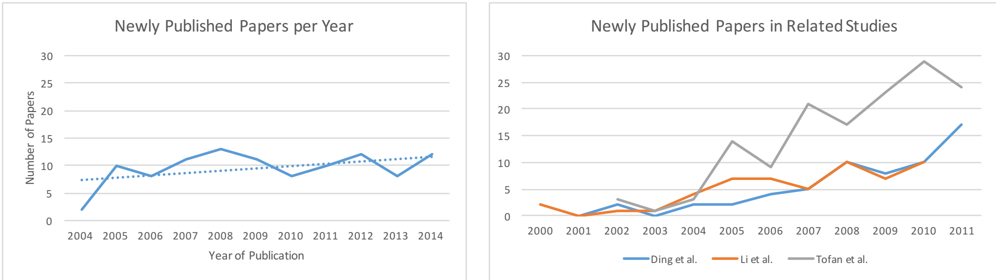

Figure 4: Papers published per year in our study and related studies

shows a high frequency of scores between 2.0 and 5.5. Four studies received the maximum score of 7.0 and one study received a score of 0.0. The papers that received a score below 2.0 were in most of the cases part of an approach that included additional papers that received a higher score.

Figure 6 shows a frequency analysis of scores on each quality question. Most studies describe the context in which the research was carried out (Q1). 86% of studies at least partially describe the research context. The mean score for Q1 was 0.7. The problem of why the study was undertaken (Q2) is sometimes not explicitly described (16% of the studies do not provide a description). The mean score for Q2 was 0.6. Only a few studies describe the research design (Q3); 68% of all studies do not provide a description of the research design. The mean score for Q3 was 0.3. An evaluation against related work (Q4) and an explicit listing of the study's contributions (Q5) are also quite often missing. 22% of the studies do not discuss related work, and 16% of the studies do not list contributions. The mean scores for both Q4 and Q5 were 0.6. The derived insights are often not discussed (Q6); 42% of all studies do not discuss insights derived. The mean score of Q6 was 0.5. Only a few authors describe the limitations of their studies; 68% of studies do not describe limitations. This was also confirmed by the mapping study performed by Li et al. [24]. The mean score of Q7 was 0.3.

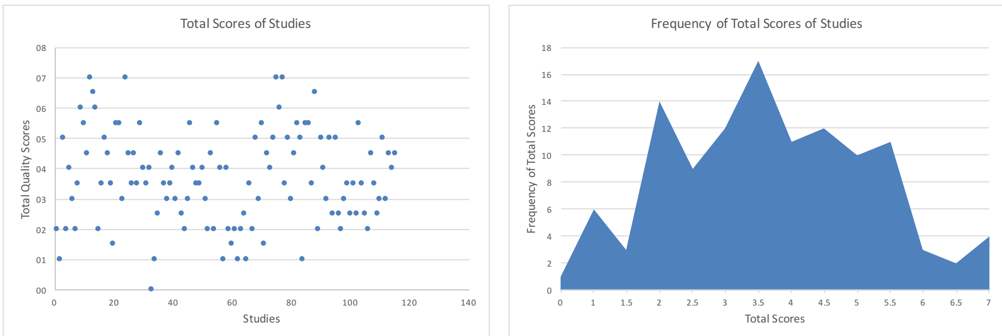

Figure 5: Quality assessment of studies

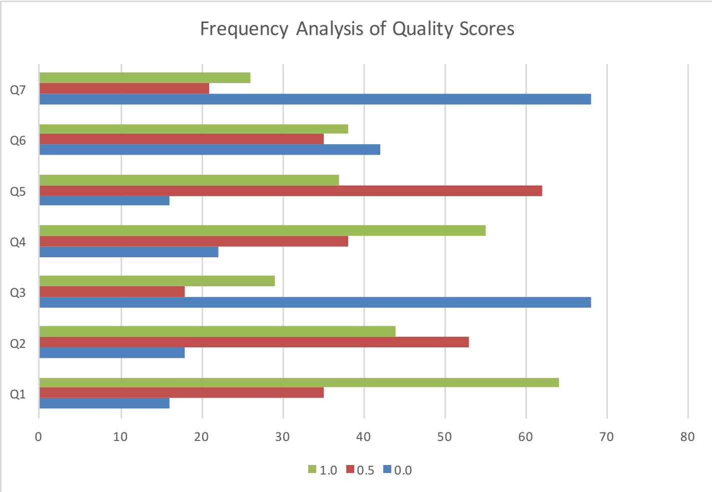

Figure 6: Quality scores per question

5.3. *RQ1: Which approaches to SAKM have been developed by the research community within the last decade?*

We identified 56 different SAKM approaches.[4] The 56 approaches we identified are very diverse with respect to their characteristics and main features, covering a large spectrum of software architecture knowledge management. In the following, we provide some general information about important concepts supported by a significant number of the identified approaches. We separate the description into two parts: one relating to the supported architecture knowledge management activities, and the other referring to important technological foundations on which the approaches are based. Table 5 provides an overview of important concepts and related SAKM approaches.

Table 5: Features and characteristics of approaches

Knowledge capture	**template-based**	PAKME, ADDSS, Decision Capture Tool, LISA, ADManager, A28, A44, A45
	schema-based	CORE, Tyree Template, Kruchten Ontology
	view-based	ADF, Decision Capture Tool, AR-diagram, TVM, A34
	annotations	Knowledge Architect, Decision Capture Tool, ArchiMind
	value-based	DDRD/DGA, CADDMS
	reuse-based	PAKME, ADDSS, RADM, ADMD3, ADUAK, SEURAT_Architecture, A26, ADvISE, A33, A34, A40, ArchPad, ADManager, ADMentor

[4]Information on how the primary studies have been aggregated to approaches can be found in Appendix B. Additionally, Section 1 in the supplementary material contains a brief description of each approach.

	automation / generation	ABC/DD, STREAM-ADD, TopDocs, A36, Latent Semantic Analysis, A44, LISA
	recovery	NDR, TopDocs, A25, A26, A28, ADDRA, Latent Sematic Analysis, A45, A41, DVIA
Knowledge application / presentation	**visualization**	Knowledge Architect, LISA, Decision Capture Tool, NDR, Compendium, QuOnt, EA Anamnesis, ShyWiki
	analysis	AREL, LISA, ArchiMind, Decision Capture Tool, NDR, Archium, AR-diagram, TopDocs, A46, AQUA, ADManager
	evaluation	TopDocs, QuOnt, PAKME, AREL, ADF, Knowledge Architect, Latent Semantic Analysis, AQUA, LISA, NDR, AR-diagram, A41
	decision-making	AREL, ADF, ABC/DD, DDRD/DGA, RADM, ShyWiki, ADMD3, SEURAT_Architecture, A27, ADvISE, A33, ArchPad, EA Anamnesis, Software Architecture Warehouse, A52, TDD/Decision Buddy, ISARCS
Knowledge maintenance	**history tracking**	PAKME, AREL, ADF, ADDSS, Decision Capture Tool, RADM, ShyWiki
	process	TVM, Decision Documentation Model
	transformation	ADvISE, AQUA
Knowledge sharing	**central access**	PAKME, AK Sharing Portal, ADDSS, ADUAK, Software Architecture Warehouse, TDD/Decision Buddy, ISARCS, Knowledge Architect, ShyWiki, ArchiMind, RADM
	knowledge base/repository	TopDocs, A45, A52, NDR, A28, A25, Decision Capture Tool
	model-focused	*MQPM, CORE, CADDMS
Knowledge reuse	**generic/project-specific**	RADM, ADMentor, TDD/Decision Buddy
	pattern-based	ArchPad, A40, A33, ADvISE, ADUAK
	partial solutions	A36, A44, Archium
Technology	**web-based**	AK Sharing Portal, PAKME, ADDSS, ADUAK, Software Architecture Warehouse, TDD/Decision Buddy, ISARCS, Knowledge Architect
	wiki-based	ShyWiki, ArchiMind, RADM
	Eclipse-based	LISA, SEURAT_Architecture, ADManager, ABC/DD, ADvISE, Decision Capture Tool
	UML-based	AREL, UML profile, EA Anamnesis, ADMentor
	DSL-based	Archium

Knowledge capture. Many of the analyzed approaches either focus on or support capturing, the most basic form of which is the manual entry of architectural knowledge, such as patterns and design decisions, using templates and forms. This form of capture is supported by web-based approaches, dedicated SAKM toolsets, and UML-based solutions. Entry is either guided by a schema or meta-model describing the required information or by a view-framework describing the information required for different views of the captured knowledge. Examples for approaches providing general schemas for architectural knowledge are the Tyree template, the Kruchten ontology, and the CORE framework. Some examples of approaches providing capturing support through manual entry according to a schema or template are PAKME, EA Anamnesis, AREL, AD-Manager, and LISA. Examples of view-based approaches are ADF and TVM. Capturing knowledge in this way is a rather tedious process, especially if the knowledge is captured as completely as possible. If capturing knowledge requires a lot of effort, architectural knowledge is either not captured at all or only when dedicated processes are in place. Fully capturing project-specific knowledge, such as design decisions, is also a major disturbing factor, since design is an inherently creative process that is easily interrupted by a formalized approach to documentation.

For this reason, some of the analyzed approaches focus on light-weight and efficient (e.g., incremental) capturing or on recovering architectural knowledge from existing documentation and artifacts. Approaches to efficient capturing include annotating existing knowledge in documents and Wikis (Knowledge Architect, Decision Capture Tool), annotating code with decision documentation (Decision Capture Tool), semi-automatic capturing of decisions and decision traces from design and development activities (LISA), enriching a generated architecture model with decisions (STREAM-ADD), and following a value-based approach by customizing the application context and the data required to be captured (ADDSS, CADDM).

Another technique for addressing the capturing problem is knowledge recovery. Approaches supporting knowledge recovery include recovering design decisions from architectural deltas (ADDRA), from various information sources (A41), from transcribed meeting logs (DVIA), and from text documents (NDR).

Finally, generic or reusable knowledge can be used to speed up the capturing process. Examples of such approaches are ADMentor, RADM, ADvISE, and A40, which use patterns as reusable decisions that can be customized for a specific context.

Knowledge application/presentation. Capturing architectural knowledge only makes sense if one can benefit from the captured knowledge by applying it to a specific context or use case. Thus, a large number of the analyzed approaches provide support for applying or using the captured knowledge. Different kinds of visualization, such as decision-graphs (Knowledge Architect, LISA), color-based coding (ShyWiki), and issue maps (Compendium) are provided in addition to simple lists of patterns, decisions, and solution structures. Impact analysis based on captured traces is supported by a number of approaches (e.g., AREL, Compendium, Knowledge Architect, LISA, ArchiMind). Some approaches provide support for specific use cases, like architecture evaluation. For example, TopDocs supports tailoring the captured knowledge for a particular stakeholder, QuOnt provides ontology-driven visualization for software audits, and PAKME supports evaluations through quality attributes and ATAM utility trees.

An important use case is the application of captured architectural knowledge to decision-making itself. Examples are approaches for the automatic synthesis of candidate architectural solutions into solutions to specific issues (ABC/DD), the guidance of pattern-selection based on questions and/or quality attributes (ADMD3, ADvISE, ArchPad), proposing patterns and styles based on quality-attribute requirements (SEURAT_Architecture, ADUAK, A52), constraint-based decision-making based on a solution repository (TDD/Decision Buddy), and support for collaborative decision-making and evaluation of alternatives (Software Architecture Warehouse, ISARCS).

Knowledge maintenance. Most approaches that provide support for maintaining the captured knowledge track the knowledge history. AREL, for example, provides an extension called eAREL that explicitly keeps track of the versions of rationale and architectural elements. ADDSS captures decisions made during different iterations of a project. Some approaches provide dedicated process support for keeping the captured knowledge up to date. TVM, for example, defines a scenario-based documentation and evolution method that explicitly considers decision evolution. Maintaining mappings between decisions and designs by transforming reusable knowledge to design views is supported by ADvISE. AQUA transforms the architecture as a result of changing design decisions.

Knowledge sharing. Most approaches provide central access to captured knowledge through a web interface or a wiki (e.g., PAKME, AK Sharing Portal, and ADDSS). Others like TopDocs and the Decision Capture Tool provide a repository or knowledge base to store the captured knowledge for later use. In addition to knowledge storing and access, there are approaches that focus on different aspects of sharing AK models. *MQPM, for example, calculates the semantic distance among different architectural knowledge models to predict in advance the quality of sharing of architectural knowledge stored in different models between sites. CORE defines a minimal architectural knowledge model to support sharing knowledge. CADDMS supports tailoring models to adapt the model to a specific context or domain. Other approaches (e.g. A36, A44, Archium) store partial solutions that are combined to form a final architecture.

Knowledge reuse. Reuse of captured knowledge is mainly supported through separating between generic and project-specific knowledge. The generic knowledge can be instantiated for a concrete project. Examples for approaches supporting this are RADM and ADMentor. Other approaches explicitly capture patterns as generic knowledge. The patterns can be applied as solutions for concrete problems. ArchPad, for example, is a design method that supports the identification of decisions in requirements models and provides guidance for the selection of domain-specific patterns. ADvISE supports the architectural decision-making process through a set of questions with potential options for recurring design issues. Different criteria are evaluated for each option, which leads to a suggested pattern-based solution.

Technology. In terms of the technological foundations, the analyzed approaches leverage Web technologies, Wikis, the Eclipse platform, UML, and DSLs. Examples of approaches providing a Web UI and a central repository for storing architectural knowledge are ISARCS, the Software Architecture Warehouse, PAKME, and the Knowledge Architect. Approaches leveraging a Wiki or a semantic Wiki infrastructure are ArchiMind, RADM, and ShyWiki. Approaches providing SAKM extensions to UML tools are EA Anamnesis, AREL, and ADMentor. Archium is an example of a DSL-based approach. Finally, ADManager, the Decision Capture Tool, ABC/DD, ADvISE, and LISA are examples of Eclipse/EMF-based approaches.

5.4. RQ2: Which SAKM activities are supported by these approaches and what has been the focus of research in terms of SAKM activities?

With this research question, we wanted to know what has been the focus of research in terms of SAKM activities (capturing, using, maintaining, sharing, and reuse). In this regard, we determined the research aims (focus) of the approaches described above and also their support for particular SAKM activities. Usually, an approach's focus is reflected in which SAKM activities it supports, but support for additional activities is often needed to pursue the primary research goal (e.g., capturing is a prerequisite to support using) or to provide a comprehensive approach for the whole knowledge management cycle. This means, in addition to the SAKM activities supported because they are the focus of research, some activities might be supported as a side-activity.

As described in Section 4.4.3 we distinguish between general and advanced support for SAKM activities. Appendix C provides a more detailed description of what we considered to be general and advanced support of SAKM activities. Appendix C provides an overview of all SAKM activities and of the approaches supporting a particular activity (either as a focus or as a side activity).

Figure 7: Focus and support for SAKM activities

Figure 7 shows the distribution of focus and support for the SAKM activities of the various approaches (cf. Section 3 in the supplementary material). The top-left chart shows that half of the approaches (28 out of 56) focus on using, while about 30% (16 out of 56) of the approaches support using as a side activity. About one-third of the approaches (20 out of 56) focus on capturing, and nearly half of the approaches (26 out of 56) support capturing as a side activity, likely because capturing knowledge is a prerequisite for all other activities. Only a very few approaches (4 out of 56) focus on maintaining and sharing, though more support them as side activities (19 support maintaining and 15 support sharing). About 20% of the approaches focus on reuse (11 out of 56) or support it as a side activity (12 out of 56).

The bottom-left chart shows that only a few approaches focus on general SAKM activities. A small spike indicates a focus on general using for eight approaches that typically provide some dedicated form of visualization. However, general SAKM activities are mainly supported as side activities.

Advanced activities are typically the focus of research (see the bottom-right chart) and are rarely supported as side activities. Almost half of the approaches (23 out of 56) focus on advanced using activities. One-third of the approaches (18 out of 56) focus on advanced capturing, and 20% (11 out of 56) focus on advanced reuse. Only a few approaches (7%, or 4 out of 56) focus on advanced maintaining or sharing.

The top-right chart details those approaches supporting the use of architectural knowledge. Half of these focus on decision-making (14 out of 28), 25% focus on evaluation/review/assessment (7), and only four approaches focus on some kind of analysis. General using, in the form of browsing and understanding knowledge, is mainly supported as a side activity. Only about 25% of the approaches that focus on using focus on general using (8 out of 28).

5.5. RQ3: What evidence is provided for the supported SAKM activities (referring to RQ2)?

With this research question, we want to gain insight into the evidence provided for the SAKM activities. We analyzed the studies related to a particular approach with regards to the evidence they provide for each specific SAKM activity.

The results of this analysis are presented in Figure 8, which shows the percentages and the absolute numbers of approaches with evidence for a specific SAKM activity (cf. Section 4 in the supplementary material). Three pairs of charts are provided: one focusing on general SAKM activities, one focusing on advanced SAKM activities, and one for sub-activities of using. Here, we only consider validation studies with an evidence level greater than 0.2; in other words, demonstrations and toy examples are not included in the charts.

The charts at the top show that only a few validation studies have been performed that evaluate the support for general activities. Only about 28% of the approaches supporting general capturing have been validated (17% with an academic study, only 11% with an industrial case study). About 29% of the approaches supporting general using provide an evaluation (13% academic, 16% industrial). Only one approach supporting general reuse has been evaluated. No evaluation has been performed for approaches supporting general sharing and maintaining.

The charts in the middle show that advanced activities are generally better validated than are general activities. The highest number of validation studies (20) is provided for approaches supporting advanced using activities (see charts at the bottom for a breakdown of this category). Fifty percent of all approaches supporting advanced using have been evaluated in academic or industrial studies, the majority of which are industrial studies (35% industrial, 15% academic).

Looking at the bottom charts reveals that, within advanced using, support for decision-making is by far the best-validated activity. About 70% of the approaches supporting decision-making provide an evaluation (40% industrial, 30% academic). Forty-two percent of the approaches supporting evaluation/review/assessment provide an evaluation (only industrial, no academic). Only 27% of the approaches supporting analysis provide an evaluation (18% industrial, 9% academic).

Approaches supporting advanced capturing are also better validated than approaches supporting general capturing. About 50% of the approaches with support for advanced capturing have been evaluated, but in this case the majority of these studies (about 70%) are academic studies. The overall number of approaches supporting advanced capturing (21) is only two-thirds of the number of approaches supporting advanced using (32).

The number of approaches with evidence supporting advanced reuse (5), sharing (2), and maintaining (2) is even less than the number of approaches supporting advanced capturing. Of

Figure 8: Approaches with evidence related to each SAKM activity

the overall number of approaches supporting the respective activity, 39% (reuse), 33% (sharing), and 18% (maintaining) provide supporting evidence. Only industrial studies are provided to evaluate support for advanced sharing, reuse, and maintaining.

We should also note that, in most cases, at most one validation study has been performed per approach and supported activity. There are only three exceptions: one approach for general capturing (ADvISE), one approach for evaluation/review/assessment (Knowledge Architect) and another approach for decision-making (ADF) have each been evaluated in two different case studies. We only found one replication study [27], which focused on the importance of the elements of design rationale documentation and not on any particular SAKM activity. In general, the lack of replication studies may be due to a lack of reporting on the analyzed studies, as discussed in Section 5.2. This means that little empirical evidence is available for most of the approaches.

6. Discussion of results

Active area of research. The demographic data presented in Section 5.1 reveal that approaches on SAKM are still very actively investigated. Since 2004, research has been published in many different venues, especially in the Journal of Systems and Software (JSS), software architecture conferences (ECSA, WICSA, QoSA), and the workshop on sharing and reusing architectural knowledge (SHARK).

Broad spectrum of approaches. In considering the approaches to SAKM that have been developed by the research community over the last decade (RQ1), we identified 56 different approaches to SAKM, which we qualitatively analyzed in terms of concepts, support for architectural knowledge management activities, and technological foundations. The identified approaches are very diverse with respect to their characteristics and main features (see Section 5.3). For example, capturing knowledge is supported through manual entry of data using templates and forms, through specific processes and tools, and through recovery of information from various sources. There is still no general agreement for addressing the problem of capturing this knowledge. The analysis also reveals a broad spectrum of approaches supporting the use of architectural knowledge for basic understanding, evaluation, and review; for change-impact analysis; and especially for decision-making. In terms of technologies, the analyzed approaches leverage Web technologies, Wikis, the Eclipse platform, UML, and DSLs.

Capturing and using are in focus. Since we wanted to identify the focus of research and the current support for central SAKM activities (RQ2), we analyzed each SAKM approach in terms of the activities on which each focuses and additionally the activities each supports. We found that most approaches focus on or support two main activities: capturing and using architectural knowledge. Half of the approaches focus on using, and 30% support some kind of using as a side activity. Conversely, 36% of the approaches focus on capturing, and 46% support capturing as a side activity. The high percentage of approaches supporting capturing as a side activity is likely due to the fact that support for capturing knowledge is a prerequisite for the other activities. Already at this level of analysis, it is apparent that only a very few approaches focus on the remaining knowledge management activities: maintaining, sharing, and reuse. The number of approaches supporting maintaining is particularly low, an indication that this particular topic has thus far been largely neglected by the research community.

The high level of support for capturing (82%) and using (80%) in existing approaches to SAKM might suggest that both activities are well-researched, perhaps even solved in current research. But even this is misleading; just providing tools for capturing knowledge through templates or forms does not solve the problem of capturing. When knowledge is not captured, it is often for reasons others than missing technical support, such as the effort involved, the lack of immediate benefits, or the lack of required skills [9]. Similar, just providing technical means to access the stored knowledge for any kind of use may be insufficient. Instead, dedicated support for specific forms of using architectural knowledge, like decision-making, architectural review, and impact analysis are often needed. Therefore, we further distinguished between general and advanced support for SAKM activities. The aim is to separate general bookkeeping functionality (e.g., storing, retrieving, and browsing) from distinctive support for a specific SAKM activity. This includes additional means for addressing efficiency and usability through specific kinds of processes and tools for a particular activity.

Advanced activities are typically the focus of research, whereas general activities are typically supported as side activities. Of 20 approaches focusing on capturing, for example, 18 focus on advanced capturing, and of 28 approaches focusing on using, 23 focus on advanced using. This is actually good, since it suggests that SAKM research typically focuses on the real problem of improving support for a specific SAKM activity.

Capturing problem is still unsolved. But separating general from advanced support also paints a completely different picture of how mature support for specific SAKM activities really is. While 82% of the approaches support capturing, most of those provide capturing support through bookkeeping functionality, which does not address the capturing problem. Only 38% of all approaches provide some support for advanced capturing. In then considering the number of validation studies for these approaches (RQ3), while about 50% of the approaches supporting advanced capturing have been validated, the majority of these validation studies (70%) are academic. This means that only 14% of approaches with advanced capturing support have been validated through industrial case studies, and none of the analyzed approaches derives from industrial practice. The overview of the approaches in Section 5.3 (RQ1) additionally reveals a great diversity in terms of the concepts and technologies used for advanced capturing. All of this leads to the conclusion that while quite a significant number of SAKM approaches provide some support for advanced capturing, the capturing problem remains unsolved. There is still a high diversity of approaches, indicating active, ongoing research, with only weak evidence that any of the developed approaches work in industrial practice. Solving the capturing problem through approaches with minimal capturing overhead and a solid cost/benefit ratio is a prerequisite for successful industrial adoption.

Using is well supported. The picture is a little bit different for using as an SAKM activity. Here, 57% of all approaches go beyond basic functionality for this activity. Also, stronger evidence is provided for the approaches supporting using activities. The predominantly supported using scenario is decision-making (with 50% of the using approaches focusing on decision-making), followed by support for evaluation/review/assessment (25%) and some kind of analysis (15%). The provided evidence is also best for decision-making, followed by evaluation/review/assessment and analysis. In addition, the majority of the provided evidence for using activities was obtained through industrial case studies.

Few approaches exist for reuse, maintaining, and sharing. Finally, taking a closer look at the remaining knowledge management activities – reuse, maintaining, and sharing – the number of approaches with evidence provided for advanced activities is even lower than for capturing. Generally, this is due to the fact that fewer approaches focus on these activities. Industrial case studies for validation are found in 39%, 18%, and 33% for approaches supporting reuse, maintaining, and sharing activities, respectively. Although there are few approaches supporting these activities, then, there is at least some validation, because the approaches typically focus on each respective activity. Here, the least supported, least validated activity again is maintaining. This is also a problem that needs to be addressed by the research community, since managing architectural knowledge is typically a long-term activity. Proper support for maintaining already-captured knowledge over the long term is thus a prerequisite for industrial adoption of SAKM approaches.

Few case studies performed per approach. With respect to the evidence provided for the different SAKM activities, our study also reveals that generally few validation studies have been performed per approach. Except for three cases, at most one study (academic or industrial) has been performed for any approach and evaluated SAKM activity. Multiple studies evaluating one particular aspect are hardly ever performed but are essential to obtain more evidence for the SAKM activities supported by the approaches.

Case study reporting is weak. When assessing the evidence for a particular activity supported by the approaches, we also found that the reporting of performed case studies is often very weak. Our quality assessment (see Section 5.2) shows that 68% of the reviewed studies do not properly describe their research designs. In assessing the evidence provided by the approaches, it was often very hard to judge what was actually evaluated in the study and how. It was, for example, often unclear who performed the actions during the evaluation (e.g., for an industrial case, the researcher who developed the approach or someone from the company?). Also, the papers often did not provide the number of participants in the case study nor the numbers with respect to the size of the system under investigation (e.g., number of decisions). This means that our assessment of the level of the provided evidence (see Section 4.4.4) is quite generous, as is the resulting analysis in terms of the evidence available for a particular activity. The actual situation might well be worse, which just emphasizes the need for additional research and evidence for the identified areas.

7. Study validity and assessment of review

In the following section, we first discuss threats to validity and then evaluate our review against a set of four quality questions proposed by Kitchenham et al. [28] for evaluating systematic literature reviews.

7.1. Threats to validity

Several factors may have influenced the results of this study. These factors may have influenced the search, the study selection, and the extraction of the data from the selected studies. The four threats to validity that apply in our case are discussed below.

Reliability refers to the question of whether the study is reproducible by other researchers [29]. To ensure reliability, we presented the search terms, the sources for our automatic search, and the defined inclusion and exclusion criteria. The general search string for the automatic search is presented in Appendix A and the exact search queries for the different databases are provided in the review protocol for our study. The voting process of course leaves room for variation, as different researchers are likely to have different opinions about whether or not a publication can contribute to answering the research questions. To reduce this source of bias, four researchers voted in parallel and discussed any publications about which they strongly disagreed. The data extraction and especially the quality assessment of the studies strongly affect the classification of approaches in this review. To reduce personal bias in study assessment, at least two reviewers checked the extracted data. Also, researchers did not extract or check their own publications, and we performed a pilot extraction study. Furthermore, we looked at related studies (such as [24]), analyzing how papers that were contained in both our study and the related review study were rated. We found strong agreement between the results of quality assessment in most cases. For each approach we list which of the related papers have been identified through the automatic search or the author contact (see Appendix B). The author contact phase did not result in new approaches, all 56 approaches have been identified through the search process. The papers obtained from the authors provided additional case studies or described the respective approach in more detail.

Construct validity refers to the question of whether the constructs are measured and interpreted correctly [29]. To ensure a common understanding of the relevant concepts and terms, we checked the relevant literature and analyzed the definitions therein. To ensure that our search terms were accurate, we conducted a pilot study to verify and refine the initial key words. To ensure a common understanding of the data to be extracted from the studies, we performed a pilot extraction. Each reviewer extracted the data from one publication, and four reviewers discussed the data resulting from these four publications. The data extraction forms were adjusted according to the results of these discussions. To ensure that the assessment of the support for SAKM activities and the provided evidence from each study was correct, we contacted the authors of the analyzed approaches.

Internal validity is the question of whether the study results really follow from the data [29]. Since we only use descriptive statistics in our data analysis, the threats to internal validity are minimal.

External validity refers to the question of whether claims for the generality of the results are justified [29]. The goal of this study was to analyze SAKM approaches in terms of their support for knowledge management activities and the evidence provided for this support. The approaches included in this study were identified in a systematic literature review, following the guidelines of Kitchenham [18]. In the next section, we evaluate the systematic literature review we performed against the quality questions for systematic literature reviews proposed by Kitchenham et al. [28].

7.2. Assessment of review

Kitchenham et al. [28] propose four quality questions for evaluating systematic literature reviews. These are presented below, along with an evaluation of our study against these questions:

QA1: Are the reviews inclusion and exclusion criteria described and appropriate?
We explicitly defined and discussed the inclusion and exclusion criteria used in Section 4.2, so this quality criterion is met.

QA2: Is the literature search likely to have covered all relevant studies?
According to Kitchenham et al., this criterion is met if four or more digital libraries have been searched and additional search strategies have been included. This quality criterion is met, as we performed an automatic search in four scientific databases (IEEE, ACM, Springer, and Elsevier) and additionally conducted a snowball sampling process. In a final step, we also contacted the authors of the papers and asked for other relevant papers.

QA3: Did the reviewers assess the quality/validity of the included studies?
We explicitly assessed the quality of each primary study according to defined quality criteria. In total, we defined seven criteria (see Section 4.4.1) to rate their quality. As part of the data synthesis, we explicitly rated the level of evidence presented for each approach with respect to the SAKM activities each supported.

QA4: Were the basic data/studies adequately described?
We consider this quality criterion as met as we used a detailed data collection form for each study. The data collection form was piloted. We also explicitly list the results of the merging of papers to approaches (see Appendix B), their focus and support in terms of the different SAKM activities (see Section 3 in the supplementary material), as well as their individual evidence levels for the supported SAKM activities (see Section 4 in the supplementary material).

8. Related work

While working on the SLR presented here, we had already published preliminary results related to SAKM models and variability management. In [30], we compared SAKM approaches with respect to their aims and underlying models for SAKM representation. In [31], we specifically analyzed how existing SAKM approaches support variability management. Below, we comment on work by other authors assessing the state of the art in SAKM and related areas.

Li et al. [24] conducted a systematic mapping study to assess the state of the art in how knowledge-based approaches are applied in software architecture. The authors assessed the use of knowledge management activities (which they call approaches) in architecting. In contrast to our study, Li et al. did not analyze knowledge management activities in terms of knowledge management approaches covering the entire knowledge management cycle. Their mapping study overviews architectural activities, where knowledge-based activities currently play a role, but it does not provide any conclusions on the state of SAKM research. For example, they analyzed how often a particular knowledge management activity has been used, but they do not provide information about any evidence for the application of SAKM activities. The authors did assess the evidence provided by a study as part of their general assessment of study quality, but they did not analyze the aim of each study's evaluation, which may be related to the architecting activity or some other aspect of the study and not to a knowledge management activity. Our study explicitly assesses the evidence for a particular SAKM activity that is supported by a particular approach.

Tofan et al. [26] present a systematic mapping study of the state of research on architectural decisions. They looked at various aspects of architectural decisions, such as support for functional requirements, quality attributes, and group decision-making. In contrast to our study, their mapping study did not investigate how existing SAKM approaches support SAKM activities, nor did it include any evidence provided for this support. As part of their study, Tofan et al. identified papers regarding the documentation of architectural decisions. Eight-three percent of all papers proposed some form of documentary approach for architectural decisions, and of those papers 63% proposed a process for documenting decisions. In our study, only 38% of the identified SAKM approaches provide some support for advanced capturing (which includes approaches providing process support for capturing architectural decisions). A possible reason for this discrepancy is that Tofan et al. included any process supporting any decision-related activity (examples given are processes for decision-making and impact analysis) and not, as we did here, including only processes for decision capturing. Our study deals with managing decisions as part of a knowledge management cycle, while the mapping study by Tofan et al. is more about the role of decisions in architecting. Tofan et al. also stated that there is a need to consolidate research regarding documenting decisions with respect to the provided evidence, and they recommended more clarity with respect to reporting case study settings. This agrees with our findings.

Ding et al. [25] performed a systematic literature review to assess the state of research on knowledge-based approaches to software documentation. In particular, the authors analyzed how the application of knowledge-based approaches influences documentation quality attributes, along with describing the costs and benefits of knowledge-based approaches. The review performed by Ding et al. is not specific to software architecture but rather covers software documentation in general, thus having a broader scope than does our study. The authors did not look at existing knowledge management approaches, as we did here, but at how knowledge-based approaches (activities, in our study) are applied to software documentation. They distinguish between general and specific knowledge-based approaches, where general approaches correspond to the main activities in our study. A specific knowledge-based approach in their study refers to a technique for supporting a general knowledge management activity (i.e., a general knowledge-based approach, to take the terminology of Ding et al.). There is also an important semantic difference regarding the mentioned knowledge-based approaches (activities, in our study). The authors determine the application of a particular knowledge-based approach (capture, sharing, reuse, etc.) in their selected studies, but they did not analyze the degree of support provided by a particular approach (general/advanced support, in our study), which may lead to completely different analytical results. For example, the authors combine knowledge capture and representation, concluding that 90% of the studies in their review employed capturing. They argue that this is the case because knowledge capture is the basis for the other knowledge-based approaches (activities). Considering capturing from the perspective of the activity rather than the concept (outcome), as we do here, enables a more targeted view of how well a particular approach actually supports the activity related to the concept of capturing. This also leads to a completely different conclusion. While Ding et al.'s results suggest that capture is a well-covered research topic, we conclude that the problem of capture is yet unsolved, which may well be one of the reasons why SAKM approaches have not yet found wide use in practice.

Capilla et al. [9] also presented a review of research on software architecture knowledge management published within the last 10 years. Their motivation is similar to ours: despite increasing SAKM research, SAKM approaches have not yet found broad acceptance in practice. The aim of their study was thus to identify successes and shortcomings of current SAKM approaches, and also to describe what industry needs from architectural knowledge. Their research design is different, as they provide an informal, retrospective analysis from the point of view of researchers who were involved in the development of a number of SAKM research tools and not an SLR. Their analysis is mainly based on selected SAKM approaches (some of them developed by the authors). Upon comparing the selected approaches, the authors conclude that all approaches support capturing and that many support sharing and evolution (maintaining). This might indicate that these activities are supported well enough by SAKM approaches, which contrasts with our findings. One reason for this may be that the authors did not distinguish between general and advanced activity support, as we did here. The authors also did not look at empirical evidence provided in support of a particular activity, instead relating their decisions mainly to their own experiences. On the other hand, the authors discussed some potential barriers to the adoption of SAKM in industry and provided some suggestions for addressing these barriers, which relate to what we classified as advanced activity support.

Finally, some studies have compared existing knowledge management approaches based on different criteria. For example, Tang et al. [32] presented a comparative study of five architecture knowledge management tools based on a defined comparison framework. They compared the tools for architectural knowledge representation and relationships, along with support for analysis, synthesis, and evaluation of architectural knowledge. Shahin et al. [33] analyzed nine architectural decision models and tools supporting these models with respect to their major elements and characteristics. An investigation of the support and evidence provided for a particular architectural knowledge management activity, as provided here, is not part of the work by Tang et al. [32] or Shahin et al. [33].

9. Conclusion

We performed a systematic literature review to obtain a detailed picture of the state of research on approaches for SAKM. Important aspects distinguishing our study are the identification of approaches based on related primary studies, the separation of focus and support, the separation of general and advanced activities, and the assessment of evidence related to specific SAKM activities.

We identified 115 studies, which we aggregated to 56 SAKM approaches. We then analyzed these approaches in detail with regard to the main SAKM activities they support and the evidence provided for this support. The highest possible evidence rating was industrial practice (1.0), followed by industrial case studies (0.8) and academic studies (0.6).

None of the analyzed approaches is industrial practice, aligning with a previous study we performed with industry [7], which showed that, as of publication, none of the study participants had ever used an SAKM approach to document their decisions. This also agrees with experiences reported by other authors (e.g., [9]).

Most approaches developed for SAKM focus on using knowledge by taking captured knowledge for granted. Using existing architectural knowledge is also the best-validated knowledge activity and the activity with the highest number of industrial studies.

Capturing still seems to be the major hindrance for industrial adoption. Approaches to capturing are diverse, with no universally agreed upon method. Capturing is often only supported through bookkeeping functionally. Approaches focusing specifically on enhancing capturing beyond data entry have mainly been validated in an academic context.

Only a very few approaches specifically address maintaining, reuse, or sharing of architectural knowledge. In terms of maintaining and sharing, only two approaches have provided some evidence, and the total number of approaches addressing these activities specifically is very low.

Validation is generally weak in terms of the number of validation studies performed for individual approaches. Most approaches with evidence only provide one academic or industrial study with respect to a specific activity.

We conclude that despite more than a decade of SAKM research, there are still many critical open issues that need to be addressed to enable widespread adoption of SAKM in industry. In the following we summarize some important directions for further research based on our findings:

- *Cost-effcient capturing*. While general capturing is addressed by most approaches considered here, we are still far from a working solution that fosters industrial adoption. We especially need to investigate ideas that go beyond simple data entry and focus on efficient capturing with a good cost/benefit ratio. Some of the identified approaches already point to promising research directions such as process models for capturing, capturing alongside other activities like evaluation and decision-making, and recovering knowledge from other documentation (e.g., meeting minutes, commit logs).

- *Long-term perspective*. There exist few approaches focusing on maintaining, i.e., on the long-term perspective of architectural knowledge. Since architectural knowledge is inherently long-term, we need additional work focusing on this aspect together with long-term studies showing the usefulness of such approaches.

- *Holistic approaches*. Most of the identified approaches focus on one or two particular knowledge management activities (mainly capturing and using). As we have shown, there is also a high diversity in how a particular activity is supported by the identified approaches. In order to be applicable in practice, all activities of the knowledge management cycle need to form a well-integrated solution that is validated as a whole. Thus, we need more research that focuses on a holistic view on SAKM.

- *Industrial validation*. Some of the analyzed approaches present industrial evidence, but we also showed that even these approaches are validated by mostly one study per activity. To improve industrial validation, we need to perform multiple industrial studies, ideally in different industrial contexts, per approach.

Recent advances in SAKM research (see, e.g., [9]) show trends that align with some of the research directions listed above. Collaborative features of groupware as well as open source software infrastructure (e.g., repositories) may be used for improving knowledge sharing. A focus on sustainability of architectural knowledge may strengthen the long-term perspective and improve maintainability. We think that integrating knowledge capturing as part of other activities like group decision-making and architecture evaluation are promising and should be investigated more. Another promising way for dealing with the capturing problem is synthesizing new information based on incomplete information (see [9]). Mining repositories for recovering knowledge is particularly well supported through software infrastructure like distributed repositories, but still requires research on the kind of decisions and the rationale that can be extracted from such repositories. These are only some examples, but they show new ways of dealing with the identified challenges. For making SAKM successful in practice, the whole knowledge management cycle needs to be addressed. This does not necessarily mean that we need one approach addressing all SAKM activities; it could also be a set of approaches that are integrated within a particular context or domain.

Finally, we also found that there is a need to improve reporting of empirical studies that have been performed to validate the developed approaches. Reporting differed considerably in quality, making it hard to judge the level of validation based on the analyzed papers alone. We thus requested feedback and comments from authors regarding supported activities and level of evidence provided. Still, the evidence ratings presented here are rather generous, so the strength of the available evidence might actually be weaker than described in this paper.

Acknowledgements

The authors would like to thank Cornelia Neumüller, Georg Buchgeher, and Jürgen Krammer for their help during the search process.

Appendix A. Search string

(software architecture) AND (architectural knowledge OR architecture knowledge OR architecture decision OR architectural decision OR architecture decisions OR architectural decisions OR design issue OR design issues OR design decision OR design decisions OR design rationale OR decision structure OR design reasoning OR architecture information OR architectural information OR knowledge management OR decision management) AND (model OR models OR modeling OR modelling OR documentation OR documenting OR decision making OR decision-making OR decision process OR ontology OR ontologies OR framework OR metamodel OR meta-model, metamodeling OR metamodelling OR capture OR representation OR reuse OR sharing OR recovery OR reasoning OR evaluation OR analysis OR understanding)

Appendix B. Approaches and aggregated primary studies

ID	Approach	Primary Studies Identified through Search	Studies Obtained from Author Contact	Total Number of Aggregated Studies
A1	PAKME	[S1-5]		5
A2	AREL	[S6-8]	[S9]	4
A3	AK Sharing Portal	[S10-11]		2
A4	ADF	[S12-14]	[S15-16]	5
A5	ABC/DD	[S17-18]		2
A6	Kruchten Ontology	[S19-20]		2
A7	*MQPM	[S21-22]	[S23]	3
A8	Knowledge Architect	[S24-25]	[S26]	3
A9	DDRD/DGA	[S27-30]		4
A10	ADDSS	[S31-35]	[S36]	6
A11	Decision Capture Tool	[S37-38]		2
A12	LISA	[S39-40]		2
A13	RADM	[S41-43]	[S44-45]	5
A14	NDR	[S46-48]		3
A15	Archium	[S49-50]		2
A16	ShyWiki	[S51-52]		2
A17	ADMD3	[S53]		1
A18	AR-diagram	[S54]		1
A19	TVM	[S55]		1
A20	Tyree Template	[S56]		1
A21	ADUAK	[S57]		1
A22	STREAM-ADD	[S58]	[S59]	2
A23	SEURAT Architecture	[S60]		1
A24	TopDocs	[S61]	[S62-63]	3
A25	Figueiredo et al.	[S64]		1

A26	Haqqie et al.	[S65]		1
A27	Al-Naeem et al.	[S66]	[S67-68]	3
A28	Tran et al. – Domain-specific AK	[S69]		1
A29	ADDRA	[S70]		1
A30	UML profile	[S71]		1
A31	CADDMS	[S72]		1
A32	ADvISE	[S73]	[S74-76]	4
A33	Lytra et al. – Platform Integration	[S77]		1
A34	Zdun et al. – Patterns and Primitives	[S78]		1
A35	CORE	[S79]		1
A36	Trujillo et al. – Synthesis	[S80]		1
A37	Compendium	[S81]	[S82]	2
A38	ArchiMind	[S83]	[S84-86]	4
A39	Latent Semantic Analysis	[S87]		1
A40	van Heesch et al. – Pattern-based AKM	[S88-89]		2
A41	Lago et al. – Assumptions	[S90-91]		2
A42	QuOnt	[S92]		1
A43	ArchPad	[S93]		1
A44	Garcia et al.	[S94]	[S95]	2
A45	Sundaravadivelu et al.	[S96]		1
A46	Luo et al.	[S97]		1
A47	EA Anamnesis	[S98]	[S99-100]	3
A48	Decision Documentation Model	[S101]	[S102-103]	3
A49	Software Architecture Warehouse	[S104]		1
A50	AQUA	[S105]		1
A51	ADManager	[S106]	[S107]	2
A52	Silva et al.	[S108]	[S109]	2
A53	TDD/Decision Buddy	[S110-112]		3
A54	ADMentor	[S113]		1
A55	ISARCS	[S114]		1
A56	DVIA	[S115]		1

Appendix C. Support for general and advanced SAKM activities

Capturing	**general**	PAKME, AREL, AK Sharing Portal, ADF, ABC/DD, ADDSS, Decision Capture Tool, LISA, RADM, Archium, ShyWiki, AR-diagram, Tyree Template, ADUAK, STREAM-ADD, SEURAT_Architecture, TopDocs, A27, A28, UML profile, CADDMS, ADvISE, CORE, A36, ArchiMind, A44, A45, EA Anamnesis, Decision Documentation Model, Software Architecture Warehouse, ADManager, A52, TDD/Decision Buddy, ADMentor, ISARCS

	advanced	Knowledge Architect, DDRD/DGA, Decision Capture Tool, LISA, NDR, Archium, TVM, STREAM-ADD, SEURAT_Architecture, TopDocs, A25, A26, ADDRA, A34, A40, A41, Decision Documentation Model, Software Architecture Warehouse, ADManager, ISARCS, DVIA
Using	**general**	PAKME, AREL, AK Sharing Portal, ADF, Knowledge Architect, ADDSS, Decision Capture Tool, LISA, RADM, NDR, Archium, ShyWiki, AR-diagram, ADUAK, SEURAT_Architecture, TopDocs, A25, A28, UML profile, CADDMS, CORE, Compendium, ArchiMind, QuOnt, A45, EA Anamnesis, Decision Documentation Model, Software Architecture Warehouse, ADManager, ADMentor, ISARCS
	evaluation review assessment	PAKME, AREL, ADF, Knowledge Architect, LISA, NDR, AR-diagram, TopDocs, Latent Semantic Analysis, A41, QuOnt, AQUA
	analysis	AREL, Decision Capture Tool, LISA, NDR, Archium, AR-diagram, TopDocs, ArchiMind, A46, AQUA, ADManager
	decision-making	AREL, ADF, ABC/DD, DDRD/DGA, RADM, ShyWiki, ADMD3, SEURAT_Architecture, A27, ADvISE, A33, ArchPad, EA Anamnesis, Software Architecture Warehouse, A52, TDD/Decision Buddy, ISARCS
Maintaining	**general**	PAKME, AREL, AK Sharing Portal, Knowledge Architect, ADDSS, Decision Capture Tool, LISA, RADM, Archium, ShyWiki, TopDocs, A28, UML profile, A36, ArchiMind, Decision Documentation Model, Software Architecture Warehouse, A52, TDD/Decision Buddy, ISARCS
	advanced	PAKME, AREL, ADF, ADDSS, Decision Capture Tool, RADM, ShyWiki, TVM, ADvISE, Decision Documentation Model, AQUA
Sharing	**general**	PAKME, AK Sharing Portal, Knowledge Architect, ADDSS, Decision Capture Tool, RADM, NDR, ShyWiki, ADUAK, TopDocs, A25, A28, CADDMS, ArchiMind, A45, Software Architecture Warehouse, A52
	advanced	AK Sharing Portal, *MQPM, NDR, ShyWiki, CORE, ArchiMind
Reuse	**general**	PAKME, AREL, AK Sharing Portal, ADF, ABC/DD, ADDSS, LISA, RADM, Archium, ShyWiki, ArchiMind, A45, ADMentor
	advanced	RADM, NDR, Archium, ADUAK, TopDocs, ADvISE, A33, A36, A40, ArchPad, A44, TDD/Decision Buddy, ADMentor

Appendix D. Definition of SAKM activities

Capturing

- *General support:* Approach provides some basic means for capturing architectural knowledge, such as forms and templates.
- *Advanced support:* Approach supports the process of capturing architecture knowledge (during development and design), in a way that goes beyond entering information in structured forms. Examples are explicit process models for capturing, retrieving knowledge from various sources (after the fact), or providing some other way of (semi-) automatically supporting capturing.

Using

- *General support:* Approach supports visualizing and browsing the captured knowledge. The knowledge is visualized to support the manual inspection of information (e.g., through diagrams or views).
- *Evaluation/review/assessment:* Approach supports the evaluation of architecture, its review, or its assessment by explicitly providing process or tool support for these activities.
- *Analysis:* Approach supports different forms of architectural analysis (e.g., change impact analysis) by explicitly providing process or tool support for these activities.
- *Design reasoning (decision-making):* Approach supports making new decisions by using the captured knowledge from previous decisions, explicitly providing process or tool support for this activity.

Reuse

- (Cross-project reuse of architectural information)
- *General support:* Approach supports reuse of decisions by providing a means to search for previous decisions based on different criteria.
- *Advanced support:* Approach explicitly addresses reuse by separating general and project-specific knowledge, and through specific process or tool support.

Maintaining

- *General support:* Approach supports changing the captured knowledge, supporting basic operations for deleting, creating, or editing knowledge.
- *Advanced support:* Approach supports the process of knowledge maintenance by, for example, detecting outdated knowledge, versioning captured knowledge, or explicitly integrating maintenance operations into the process.

Sharing

- (making knowledge easily accessible to involved stakeholders)
- *General support:* Approach provides some form of central knowledge repository, with remote access for different stakeholders.
- *Advanced support:* Approach provides specific concepts, processes, or tools for distributing knowledge among stakeholders (e.g., personalization), which goes beyond centralized access.

Appendix E. Identified Primary Studies

[S1] M.A. Babar, I. Gorton, and R. Jeffery, Capturing and Using Software Architecture Knowledge for Architecture-Based Software Development, Fifth International Conference on Quality Software (QSIC 2005) , IEEE, 2005, p. 169-176.

[S2] M.A. Babar, and I. Gorton, A Tool for Managing Software Architecture Knowledge, Second Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent (SHARK/ADI'07) , IEEE, 2007, p. 11.

[S3] M.A. Babar, A. Northway, I. Gorton, P. Heuer, and T. Nguyen, Introducing Tool Support for Managing Architectural Knowledge: An Experience Report, 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2008) , IEEE, 2008, p. 105-113.

[S4] M.A. Babar, X. Wang, and I. Gorton, PAKME: A Tool for Capturing and Using Architecture Design Knowledge, 9th International Multitopic Conference (INMIC 2005), IEEE, 2005, p. 1-6.

[S5] M.A. Babar, and R. Capilla, Capturing and Using Quality Attributes Knowledge in Software Architecture Evaluation Process, First International Workshop on Managing Requirements Knowledge (MARK'08) , IEEE, 2008, p. 53-62.

[S6] A. Tang, A. Nicholson, Y. Jin, and J. Han, Using Bayesian belief networks for change impact analysis in Architecture Design, Journal of Systems and Software. 80 (2007) 127-148.

[S7] A. Tang, J. Han, and R. Vasa, Software Architecture Design Reasoning: A Case for Improved Methodology Support, IEEE Software. 26 (2009) 43-49.

[S8] A. Tang, Y. Jin, and J. Han, A Rationale-Based Architecture Model for Design Traceability and Reasoning, Journal of Systems and Software. 80 (2007) 918-934.

[S9] A. Tang, and M.F. Lau, Software Architecture Review by Association, Journal of Systems and Software. 88 (2014) 87-101.

[S10] R. Farenhorst, R. Izaks, P. Lago, and H. van Vliet, A Just-In-Time Architectural Knowledge Sharing Portal, Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008) , IEEE, 2008, p. 125-134.

[S11] R. Farenhorst, P. Lago, and H. Van Vliet, Effective Tool Support for Architectural Knowledge Sharing, First European Conference on Software Architecture (ECSA 2007), Springer Berlin Heidelberg, 2007, p. 123-138.

[S12] U. van Heesch, P. Avgeriou, and R. Hilliard, A Documentation Framework for Architecture Decisions, Journal of Systems and Software. 85 (2012) 795-820.

[S13] U. van Heesch, P. Avgeriou, and A. Tang, Does Decision Documentation Help Junior Designers Rationalize Their Decisions? A Comparative Multiple-Case Study, Journal of Systems and Software. 86 (2013) p. 1545-1565.

[S14] U. van Heesch, P. Avgeriou, and R. Hilliard, Forces on Architecture Decisions - A Viewpoint, 2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and 6th European Conference on Software Architecture (ECSA), IEEE, 2012, p. 101-110.

[S15] V.-P. Eloranta, U. van Heesch, P. Avgeriou, N. Harrison, and K. Koskimies, Lightweight Evaluation of Software Architecture Decisions, Relating System Quality and Software Architecture, Relating System Quality and Software Architecture, Elsevier Science, 2014, p. 157-179.

[S16] U. van Heesch, V.-P. Eloranta, P. Avgeriou, K. Koskimies, and N. Harrison, Decision-Centric Architecture Reviews, IEEE Software. 31 (2014) 69-76.

[S17] X. Cui, Y. Sun, S. Xiao, and H. Mei, Architecture Design for the Large-Scale Software-Intensive Systems: A Decision-Oriented Approach and the Experience, 14th IEEE International Conference on Engineering of Complex Computer Systems , IEEE, 2009, p. 30-39.

[S18] X. Cui, Y. Sun, and H. Mei, Towards Automated Solution Synthesis and Rationale Capture in Decision-Centric Architecture Design, Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008) , IEEE, 2008, p. 221-230.

[S19] P. Kruchten, An Ontology of Architectural Design Decisions in Software-Intensive Systems, 2nd Groningen Workshop on Software Variability, 2004, p. 54-61.

[S20] P. Kruchten, P. Lago, and H. Van Vliet, Building Up and Reasoning About Architectural Knowledge, Second International Conference on Quality of Software Architectures (QoSA 2006), Springer Berlin Heidelberg, 2006, p. 43-58.

[S21] P. Liang, A. Jansen, P. Avgeriou, A. Tang, and L. Xu, Advanced Quality Prediction Model for Software Architectural Knowledge Sharing, Journal of Systems and Software. 84 (2011) 786-802.

[S22] P. Liang, A. Jansen, and P. Avgeriou, Selecting a High-Quality Central Model for Sharing Architectural Knowledge, The Eighth International Conference on Quality Software (QSIC'08), IEEE, 2008, p. 357-365.

[S23] P. Liang, A. Jansen, and P. Avgeriou, Sharing Architecture Knowledge Through Models: Quality and Cost, The Knowledge Engineering Review. 24 (2009) 225-244.

[S24] A. Jansen, P. Avgeriou, and J.S. van der Ven, Enriching Software Architecture Documentation, Journal of Systems and Software. 82 (2009) 1232-1248.

[S25] A. Jansen, T. de Vries, P. Avgeriou, and M. van Veelen, Sharing the Architectural Knowledge of Quantitative Analysis, 4th International Conference on the Quality of Software Architectures (QoSA 2008), Springer Berlin Heidelberg, 2008, p. 220-234.

[S26] P. Liang, A. Jansen, and P. Avgeriou, Collaborative Software Architecting Through Knowledge Sharing, Collaborative Software Engineering, Springer Berlin Heidelberg, 2010, p. 343-367.

[S27] D. Falessi, R. Capilla, and G. Cantone, A Value-Based Approach for Documenting Design Decisions Rationale: A Replicated Eexperiment, 3rd International Workshop on Sharing and Reusing Architectural Knowledge (SHARK'08), ACM, 2008, p. 63-70.

[S28] D. Falessi, M. Becker, and G. Cantone, Design Decision Rationale: Experiences and Steps Ahead Towards Systematic Use, ACM SIGSOFT Software Engineering Notes. 31 (2006).

[S29] D. Falessi, G. Cantone, and M. Becker, Documenting Design Decision Rationale to Improve Individual and Team Design Decision Making: An Experimental Evaluation, ACM/IEEE International Symposium on Empirical Software Engineering, 2006, p. 134-143.

[S30] D. Falessi, G. Cantone, and P. Kruchten, Value-Based Design Decision Rationale Documentation: Principles and Empirical Feasibility Study, Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008) , IEEE, 2008, p. 189-198.

[S31] R. Capilla, F. Nava, and J.C. Dueñas, Modeling and Documenting the Evolution of Architectural Design Decisions, Second Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent (SHARK/ADI'07) , IEEE, 2007, p. 9.

[S32] P. Kruchten, R. Capilla, and J.C. Dueñas, The Decision View's Role in Software Architecture Practice, IEEE Software. 26 (2009) 36-42.

[S33] J.C. Dueñas, and R. Capilla, The Decision View of Software Architecture, 2nd European Workshop on Software Architecture (EWSA 2005), Springer Berlin Heidelberg, 2005, p. 222-230.

[S34] R. Capilla, F. Nava, and A. Tang, Attributes for Characterizing the Evolution of Architectural Design Decisions, Third International IEEE Workshop on Software Evolvability, IEEE, 2007, p. 15-22.

[S35] R. Capilla, F. Nava, S. Prez, and J.C. Dueñas, A Web-Based Tool for Managing Architectural Design Decisions, ACM SIGSOFT Software Engineering Notes. 31 (2006).

[S36] R. Capilla, J.C. Dueñas, and F. Nava, Viability for Codifying and Documenting Architectural Design Decisions with Tool Support, Journal of Software Maintenance and Evolution: Research and Practice. 22 (2010) 81-119.

[S37] L. Lee, and P. Kruchten, A Tool to Visualize Architectural Design Decisions, 4th International Conference on the Quality of Software-Architectures (QoSA 2008), Springer Berlin Heidelberg, 2008, p. 43-54.

[S38] L. Lee, and P. Kruchten, Customizing the Capture of Software Architectural Design Decisions, Canadian Conference on Electrical and Computer Engineering (CCECE 2008), IEEE, 2008, p. 693-698.

[S39] R. Weinreich, and G. Buchgeher, Towards Supporting the Software Architecture Life Cycle, Journal of Systems and Software. 85 (2012) 546-561.

[S40] R. Weinreich, and G. Buchgeher, Integrating requirements and design decisions in architecture representation, Software Architecture. (2010) 86-101.

[S41] O. Zimmermann, J. Koehler, F. Leymann, R. Polley, and N. Schuster, Managing Architectural Decision Models with Dependency Relations, Integrity Constraints, and Production Rules, Journal of Systems and Software. 82 (2009) 1249-1267.

[S42] O. Zimmermann, T. Gschwind, J. Küster, F. Leymann, and N. Schuster, Reusable Architectural Decision Models for Enterprise Application Development, Third International Conference on Quality of Software Architectures (QoSA 2007), Springer Berlin Heidelberg, 2007, p. 15-32.

[S43] R. Capilla, O. Zimmermann, U. Zdun, P. Avgeriou, and J.M. Kster, An Enhanced Architectural Knowledge Metamodel Linking Architectural Design Decisions to other Artifacts in the Software Engineering Lifecycle, 5th European Conference on Software Architecture (ECSA), Springer Berlin Heidelberg, 2011, p. 303-318.

[S44] O. Zimmermann, P. Kopp, and S. Pappe, Architectural Knowledge in an SOA Infrastructure Reference Architecture, Software Architecture Knowledge Management: Theory and Practice, Springer Berlin Heidelberg, 2009, p. 217-241.

[S45] P. Koenemann, and O. Zimmermann, Linking Design Decisions to Design Models in Model-Based Software Development, 4th European Conference on Software Architecture (ECSA 2010), Springer Berlin Heidelberg, 2010, p. 246-262.

[S46] C. Lopez, V. Codocedo, H. Astudillo, and L.M. Cysneiros, Bridging the Gap Between Software Architecture Rationale Formalisms and Actual Architecture Documents: An Ontology-Driven Approach, Science of Computer Programming. 77 (2012) 66-80.

[S47] C. Lopez, P. Inostroza, L.M. Cysneiros, and H. Astudillo, Visualization and Comparison of Architecture Rationale with Semantic Web Technologies, Journal of Systems and Software. 82 (2009) 1198-1210.

[S48] C. Lopez, L.M. Cysneiros, and H. Astudillo, NDR Ontology: Sharing and Reusing NFR and Design Rationale Knowledge, First International Workshop on Managing Requirements Knowledge (MARK'08) , IEEE, 2008, p. 1-10.

[S49] A. Jansen, J. Van Der Ven, P. Avgeriou, and D.K. Hammer, Tool Support for Architectural Decisions, The Working IEEE/IFIP Conference on Software Architecture (WICSA'07) , IEEE, 2007, p. 4.

[S50] A. Jansen, and J. Bosch, Software Architecture as a Set of Architectural Design Decisions, 5th Working IEEE/IFIP Conference on Software Architecture (WICSA 2005) , IEEE, 2005, p. 109-120.

[S51] C. Solis, and N. Ali, An Experience Using a Spatial Hypertext Wiki, 22nd ACM Conference on Hypertext and Hypermedia (HT'11), ACM, 2011, p. 133-142.

[S52] C. Solis, N. Ali, and M.A. Babar, A Spatial Hypertext Wiki for Architectural Knowledge Management, ICSE Workshop on Wikis for Software Engineering (WIKIS4SE'09) , IEEE, 2009, p. 36-46.

[S53] Z. Durdik, and R. Reussner, Position Paper: Approach for Architectural Design and Modelling with Documented Design Decisions (ADMD3), 8th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA'12), ACM, 2012, p. 49-54.

[S54] B. Orlic, R. Mak, I. David, and J. Lukkien, Concepts and Diagram Elements for Architectural Knowledge Management, 5th European Conference on Software Architecture (ECSA 2011): Companion Volume, ACM, 2011, p. 3.

[S55] M. Che, and D.E. Perry, Scenario-Based Architectural Design Decisions Documentation and Evolution, 18th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS 2011), IEEE, 2011, p. 216-225.

[S56] J. Tyree, and A. Akerman, Architecture Decisions: Demystifying Architecture, IEEE Software. 22 (2005) 19-27.

[S57] C. Dhaya, and G. Zayaraz, Development of Multiple Architectural Designs using ADUAK, International Conference on Communications and Signal Processing (ICCSP 2012), IEEE, 2012, p. 93-97.

[S58] D. Dermeval, J. Pimentel, C. Silva, J. Castro, E. Santos, G. Guedes, and A. Finkelstein, STREAM-ADD-Supporting the Documentation of Architectural Design Decisions in an Architecture Derivation Process, 2012 IEEE 36th International Conference on Computer Software and Applications (COMPSAC), , IEEE, 2012, p. 602-611.

[S59] D. Dermeval, J. Castro, C. Silva, J. Pimentel, I.I. Bittencourt, P. Brito, E. Elias, T. Tenrio, and A. Pedro, On the Use of Metamodeling for Relating Requirements and Architectural Design Decisions, 28th Annual ACM Symposium on Applied Computing (SAC'13), ACM, 2013, p. 1278-1283.

[S60] W. Wang, and J.E. Burge, Using Rationale to Support Pattern-Based Architectural Design, 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge (SHARK), ACM, 2010, p. 1-8.

[S61] V.-P. Eloranta, O. Hylli, T. Vepsalainen, and K. Koskimies, TopDocs: Using Software Architecture Knowledge Base for Generating Topical Documents, 2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and 6th European Conference on Software Architecture (ECSA), IEEE, 2012, p. 191-195.

[S62] T. Vepsalainen, S. Kuikka, and V. Eloranta, Software Architecture Knowledge Management for Safety Systems, 2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA) , IEEE , 2012, p. 1-8.

[S63] V.-P. Eloranta, and K. Koskimies, Aligning Architecture Knowledge Management with Scrum, WICSA/ECSA 2012 Companion Volume, ACM, 2012, p. 112-115.

[S64] A.M.C. Figueiredo, J.C. dos Reis, and M.A. Rodrigues, Semantic Search for Software Architecture Knowledge: A Proposal for Virtual Communities Environment, 2011 International Conference on Information Society (i-Society) , IEEE, 2011, p. 270-274.

[S65] S. Haqqie, and A.A. Shahid, Mining Design Patterns for Architecture Reconstruction using an Expert System, 9th International Multitopic Conference (IEEE INMIC 2005), IEEE, 2005, p. 1-6.

[S66] T. Al-Naeem, F.T. Dabous, F.A. Rabhi, and B. Benatallah, Formulating the Architectural Design of Enterprise Applications as a Search Problem, 2005 Australian Software Engineering Conference (ASWEC'05), IEEE, 2005, p. 282-291.

[S67] T. Al-Naeem, I. Gorton, M.A. Babar, F. Rabhi, and B. Benatallah, A Quality-Driven Systematic Approach for Architecting Distributed Software Applications, 27th International Conference on Software Engineering (ICSE 2005), IEEE, 2005, p. 244-253.

[S68] T. Al-Naeem, F.A. Rabhi, B. Benatallah, and P.K. Ray, Systematic approaches for designing B2B applications, International Journal of Electronic Commerce. 9 (2005) 41-70.

[S69] H. Tran, I. Lytra, and U. Zdun, Derivation of Domain-specific Architectural Knowledge Views from Governance and Security Compliance Metadata, 28th Annual ACM Symposium on Applied Computing (SAC'13), ACM, 2013, p. 1728-1733.

[S70] A. Jansen, J. Bosch, and P. Avgeriou, Documenting After the Fact: Recovering Architectural Design Decisions, Journal of Systems and Software. 81 (2008) 536-557.

[S71] L. Zhu, and I. Gorton, UML Profiles for Design Decisions and Non-Functional Requirements, Second Workshop on SHAring and Reusing Architectural Knowledge Architecture, Rationale, and Design Intent (SHARK-ADI'07), IEEE, 2007, p. 8.

[S72] L. Chen, M.A. Babar, and H. Liang, Model-Centered Customizable Architectural Design Decisions Management, 21st Australian Software Engineering Conference (ASWEC 2010), IEEE, 2010, p. 23-32.

[S73] I. Lytra, H. Tran, and U. Zdun, Constraint-Based Consistency Checking between Design Decisions and Component Models for Supporting Software Architecture Evolution, 16th European Conference on Software Maintenance and Reengineering (CSMR 2012) , IEEE, 2012, p. 287-296.

[S74] I. Lytra, H. Tran, and U. Zdun, Harmonizing Architectural Decisions with Component View Models using Reusable Architectural Knowledge Transformations and Constraints, Future Generation Computer Systems. 47 (2015) 80-96.

[S75] P. Gaubatz, I. Lytra, and U. Zdun, Automatic Enforcement of Constraints in Real-Time Collaborative Architectural Decision Making, Journal of Systems and Software. 103 (2015) 128-149.

[S76] I. Lytra, G. Engelbrecht, D. Schall, and U. Zdun, Reusable Architectural Decision Models for Quality-driven Decision Support: A Case Study from a Smart Cities Software Ecosystem, Third International Workshop on Software Engineering for Systems-of-Systems (SESoS 2015), IEEE, 2015, p. 37-43.

[S77] I. Lytra, S. Sobernig, and U. Zdun, Architectural Decision Making for Service-Based Platform Integration: A Qualitative Multi-Method Study, 2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), IEEE, 2012, p. 111-120.

[S78] U. Zdun, P. Avgeriou, C. Hentrich, and S. Dustdar, Architecting as Decision Making with Patterns and Primitives, 3rd International Workshop on Sharing and Reusing Architectural Knowledge (SHARK'08), ACM, 2008, p. 11-18.

[S79] R.C. De Boer, R. Farenhorst, P. Lago, H. Van Vliet, V. Clerc, and A. Jansen, Architectural Knowledge: Getting to the Core, Third International Conference on Quality of Software Architectures (QoSA 2007), Springer Berlin Heidelberg, 2007, p. 197-214.

[S80] S. Trujillo, M. Azanza, O. Diaz, and R. Capilla, Exploring Extensibility of Architectural Design Decisions, Second Workshop on Sharing and Reusing Architectural Knowledge-Architecture, Rationale, and Design Intent (SHARK/ADI'07) , IEEE, 2007, p. 10.

[S81] M. Shahin, P. Liang, and M.R. Khayyambashi, Improving Understandability of Architecture Design through Visualization of Architectural Design Decision, 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge (SHARK'10), ACM, 2010, p. 88-95.

[S82] M. Shahin, P. Liang, and Z. Li, Do Architectural Design Decisions Improve the Understanding of Software Architecture? Two Controlled Experiments, 22nd International Conference on Program Comprehension (ICPC 2014), ACM, 2014, p. 3-13.

[S83] K.A. de Graaf, A. Tang, P. Liang, and H. van Vliet, Ontology-based Software Architecture Documentation, Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on, 2012, p. 121-130.

[S84] K.A. De Graaf, P. Liang, A. Tang, W.R. Van Hage, and H. Van Vliet, An Exploratory Study on Ontology Engineering for Software Architecture Documentation, Computers in Industry. 65 (2014) 1053-1064.

[S85] K.A. de Graaf, Annotating Software Documentation in Semantic Wikis, Fourth Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR'11), ACM, 2011, p. 5-6.

[S86] K.A. de Graaf, P. Liang, A. Tang, and H. van Vliet, Supporting Architecture Documentation: A Comparison of Two Ontologies for Knowledge Retrieval, 19th International Conference on Evaluation and Assessment in Software Engineering (EASE'15), ACM, 2015, p. 3.

[S87] R.C. De Boer, and H. van Vliet, Architectural Knowledge Discovery with Latent Semantic Analysis: Constructing a Reading Guide for Software Product Audits, Journal of Systems and Software. 81 (2008) 1456-1469.

[S88] U. van Heesch, and P. Avgeriou, A Pattern-based Approach Against Architectural Knowledge Vaporization, 14th Annual European Conference on Pattern Languages of Programs (EuroPLoP 2009), 2009.

[S89] U. van Heesch, P. Avgeriou, U. Zdun, and N. Harrison, The Supportive Effect of Patterns in Architecture Decision Recovery - A Controlled Experiment, Science of Computer Programming. 77 (2012) 551-576.

[S90] P. Lago, and H. van Vliet, Explicit Assumptions Enrich Architectural Models, 27th International Conference on Software Engineering (ICSE 2005), ACM, 2005, p. 206-214.

[S91] R. Roeller, P. Lago, and H. van Vliet, Recovering Architectural Assumptions, Journal of Systems and Software. 79 (2006) 552-573.

[S92] R.C. De Boer, P. Lago, A. Telea, and H. Van Vliet, Ontology-Driven Visualization of Architectural Design Decisions, Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA 2009) , IEEE, 2009, p. 51-60.

[S93] O. Zimmermann, U. Zdun, T. Gschwind, and F. Leymann, Combining Pattern Languages and Reusable Architectural Decision Models into a Comprehensive and Comprehensible Design Method, Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), IEEE, 2008, p. 157-166.

[S94] A. Garcia, T. Batista, A. Rashid, and C. Sant'Anna, Driving and Managing Architectural Decisions with Aspects, ACM SIGSOFT Software Engineering Notes. 31 (2006) 6.

[S95] C. Sant'Anna, A. Garcia, T. Batista, and A. Rashid, Mastering Crosscutting Architectural Decisions with Aspects, Software: Practice and Experience. 43 (2013) 305-332.

[S96] S. Sundaravadivelu, A. Vaidyanathan, and S. Ramaswamy, Knowledge Reuse of Software Architecture Design Decisions and Rationale within the Enterprise, 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), IEEE, 2014, p. 253-261.

[S97] J. Luo, R. Huang, and X. Chen, Automatic Validation of Design Decision Based on Multiple Analysis Methods Integration, 13th International Conference on Quality Software (QSIC 2013) , IEEE, 2013, p. 333-340.

[S98] G. Plataniotis, S. de Kinderen, and H.A. Proper, Relating Decisions in Enterprise Architecture using Decision Design Graphs, 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2013) , IEEE, 2013, p. 139-146.

[S99] G. Plataniotis, S. De Kinderen, and H.A. Proper, EA Anamnesis: An Approach for Decision Making Analysis in Enterprise Architecture, International Journal of Information System Modeling and Design (IJISMD). 5 (2014) 75-95.

[S100] G. Plataniotis, S. De Kinderen, and H.A. Proper, Capturing Design Rationales in Enterprise Architecture: A Case Study, 7th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2014), Springer Berlin Heidelberg, 2014, p. 133-147.

[S101] T.-M. Hesse, and B. Paech, Supporting the Collaborative Development of Requirements and Architecture Documentation, 3rd International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks 2013) , IEEE, 2013, p. 22-26.

[S102] T.-M. Hesse, S. Gartner, T. Roehm, B. Paech, K. Schneider, and B. Bruegge, Semiautomatic Security Requirements Engineering and Evolution using Decision Documentation, Heuristics, and User Monitoring, IEEE 1st Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE 2014) , IEEE, 2014, p. 1-6.

[S103] T.-M. Hesse, A. Kuehlwein, B. Paech, T. Roehm, and B. Bruegge, Documenting Implementation Decisions with Code Annotations, 27th International Conference on Software Engineering and Knowledge Engineering (SEKE 2015), KSI Research Inc. and Knowledge Systems Institute Graduate School, n.d., p. 152-157.

[S104] M. Nowak, and C. Pautasso, Team Situational Awareness and Architectural Decision Making with the Software Architecture Warehouse, 7th European Conference on Software Architecture (ECSA 2013), Springer Berlin Heidelberg, 2013, p. 146-161.

[S105] H. Choi, Y. Choi, and K. Yeom, An Integrated Approach to Quality Achievement with Architectural Design Decisions, Journal of Software. 1 (2006) 40-49.

[S106] M.T.T. That, S. Sadou, and F. Oquendo, Using Architectural Patterns to Define Architectural Decisions, 2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA) , IEEE, 2012, p. 196-200.

[S107] M.T.T. That, S. Sadou, F. Oquendo, and R. Fleurquin, Preserving Architectural Decisions Through Architectural Patterns, Automated Software Engineering. (2014) 1-41.

[S108] I.C.L. Silva, P.H.S. Brito, B.F. dos S Neto, E. Costa, and A.A. Silva, A Decision-Making Tool to Support Architectural Designs Based on Quality Attributes, 30th Annual ACM Symposium on Applied Computing (SAC'15), ACM, 2015, p. 1457-1463.

[S109] I.C.L. Silva, P.H.S. Brito, E. Costa, and H. Rocha, A Tool for Trade-off Resolution on Architecture-Centered Software Development, The 26th International Conference on Software Engineering and Knowledge Engineering, Hyatt Regency, Vancouver, BC, Canada, July 1-3, 2013, 2014, p. 35-38.

[S110] S. Gerdes, M. Soliman, and M. Riebisch, Decision Buddy: Tool Support for Constraint-Based Design Decisions during System Evolution, 1st International Workshop on Future of Software Architecture Design Assistants (FoSADA'15), ACM, 2015, p. 13-18.

[S111] M. Soliman, M. Riebisch, and U. Zdun, Enriching Architecture Knowledge with Technology Design Decisions, Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on, 2015, p. 135-144.

[S112] M. Soliman, and M. Riebisch, Modeling the Interactions between Decisions within Software Architecture Knowledge, 8th European Conference on Software Architecture (ECSA 2014), Springer Berlin Heidelberg, 2014, p. 33-40.

[S113] O. Zimmermann, L. Wegmann, H. Koziolek, and T. Goldschmidt, Architectural Decision Guidance Across Projects - Problem Space Modeling, Decision Backlog Management and Cloud Computing Knowledge, 12th Working IEEE/IFIP Conference on Software Architecture (WICSA 2015), IEEE, 2015, p. 85-94.

[S114] N. Chanda, and X.F. Liu, Intelligent Analysis of Software Architecture Rationale for Collaborative Software Design, International Conference on Collaboration Technologies and Systems (CTS 2015) , IEEE, 2015, p. 287-294.

[S115] G. Pedraza-Garcia, H. Astudillo, and D. Correal, Analysis of Design Meetings for Understanding Software Architecture Decisions, 2014 XL Latin American Computing Conference (CLEI), IEEE, 2014, p. 1-10.

References

[1] D. E. Perry, A. L. Wolf, Foundations for the study of software architecture, SIGSOFT Softw. Eng. Notes 17 (4) (1992) 40–52.

[2] M. Shaw, D. Garlan, Software Architecture: Perspectives on an Emerging Discipline, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[3] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, 3rd Edition, SEI Series in Software Engineering, Addison-Wesley Professional, 2003.

[4] J. Bosch, Software architecture: The next step, in: First European Workshop on Software Architecture (EWSA 2004), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 194–199.

[5] M. Ali Babar, T. Dingsyr, P. Lago, H. Van Vliet (Eds.), Software Architecture Knowledge Management : Theory and Practice, Springer Berlin Heidelberg, 2009.

[6] P. Liang, P. Avgeriou, Tools and Technologies for Architecture KnowledgeManagement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, Ch. 6, pp. 91–112.

[7] R. Weinreich, I. Groher, C. Miesbauer, An expert survey on kinds, influence factors and documentation of design decisions in practice, Future Generation Computer Systems 47 (2015) 145–160.

[8] W. Ding, P. Liang, A. Tang, H. v. Vliet, M. Shahin, How do open source communities document software architecture: An exploratory survey, in: Engineering of Complex Computer Systems (ICECCS), 2014 19th International Conference on, 2014, pp. 136–145.

[9] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, M. A. Babar, 10 years of software architecture knowledge management: Practice and future, Journal of Systems and Software 116 (2016) 191–205.

[10] T. H. Davenport, Saving it's soul: Human-centered information management, Harvard Business Review 72 (2) (1994) 119–131.

[11] I. Nonaka, H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, 1995.

[12] G. von Krogh, Care in knowledge creation, California Management Review 40 (3) (1998) 133–153.

[13] M. T. Hansen, N. Nohria, T. Tierney, What's your strategy for managing knowledge?, Harvard Business Review 77 (2) (1999) 106–116.

[14] M. Alavi, D. E. Leidner, Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues, MIS Quarterly 25 (1) (2001) 107–136.

[15] K. Dalkir, Knowledge Management in Theory and Practice, 2nd Edition, MIT Press, 2011.

[16] T. Dingsøyr, H. van Vliet, Introduction to Software Architecture and Knowledge Management, Springer Berlin Heidelberg, 2009, Ch. 1, pp. 1–17.

[17] R. Farenhorst, R. C. Boer, Knowledge Management in Software Architecture: State of the Art, Springer Berlin Heidelberg, 2009, Ch. 2, pp. 21–38.

[18] B. A. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. rep., EBSE Technical Report, EBSE-2007-01, Software Engineering Group, School of Computer Science and Mathematics, Keele University, UK, and Department of Computer Science, University of Durham, Durham, UK (7 2007).

[19] B. A. Kitchenham, E. Mendes, G. H. Travassos, Cross versus within-company cost estimation studies: A systematic review, IEEE Transactions on Software Engineering 33 (5) (2007) 316–329.

[20] J. L. Fleiss, Measuring nominal scale agreement among many raters., Psychological Bulletin 76 (5) (1971) 378–382.

[21] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1) (1977) 159–174.

[22] J. Webster, R. T. Watson, Analyzing the past to prepare for the future: Writing a literature review, MIS Q. 26 (2) (2002) xiii–xxiii.

[23] D. Weyns, T. Ahmad, Claims and evidence for architecture-based self-adaptation: A systematic literature review, in: 7th European Conference on Software Architecture (ECSA 2013), Lecture Notes in Computer Science, 2013, pp. 249–265.

[24] Z. Li, P. Liang, P. Avgeriou, Application of knowledge-based approaches in software architecture: A systematic mapping study, Information and Software Technology 55 (5) (2013) 777 – 794.

[25] W. Ding, P. Liang, A. Tang, H. v. Vliet, Knowledge-based approaches in software documentation: A systematic literature review, Information and Software Technology 56 (6) (2014) 545–567.

[26] D. Tofan, M. Galster, P. Avgeriou, W. Schuitema, Past and future of software architectural decisions - a systematic mapping study, Information and Software Technology 56 (8) (2014) 850–872.

[27] D. Falessi, R. Capilla, G. Cantone, A value-based approach for documenting design decisions rationale: A replicated eexperiment, in: 3rd International Workshop on Sharing and Reusing Architectural Knowledge (SHARK'08), ACM, 2008, pp. 63–70.

[28] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - a systematic literature review, Information and Software Technology 51 (1) (2009) 7–15.
[29] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting Empirical Methods for Software Engineering Research, Springer London, 2008, pp. 285–311.
[30] R. Weinreich, I. G. Groher, A fresh look at codification approaches for sakm: A systematic literature review, in: 8th European Conference on Software Architecture (ECSA 2014), Lecture Notes in Computer Science, 2014, pp. 1–16.
[31] I. Groher, R. Weinreich, Variability support in architecture knowledge management approaches: A systematic literature review, in: 48th Hawaii International Conference on System Sciences (HICSS 2015), IEEE, pp. 5393–5402.
[32] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, M. A. Babar, A comparative study of architecture knowledge management tools, Journal of Systems and Software 83 (3) (2010) 352 – 370.
[33] M. Shahin, P. Liang, M.-R. . R. Khayyambashi, Architectural design decision: Existing models and tools, in: Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA 2009), IEEE, 2009, pp. 293–296.