University of Groningen

Master Thesis

Architectural Knowledge Sharing: a Systematic Mapping Study

Author:
Alexandra Cătălina **Mătreață**

Supervisors:
Paris **Avgeriou** and
Christian **Manteuffel**
Nicolae **Goga**

Declaration of Authorship

I, Alexandra Cătălina Mătreață, declare that this thesis titled, "Architectural Knowledge Sharing: a Systematic Mapping Study" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Contents

List of Figures

List of Tables

Abbreviations

AK	**A**rchitectural **K**nowledge
AKM	**A**rchitectural **K**nowledge **M**anagement
SA	**S**oftware **A**rchitecture
SLR	**S**ystematical **L**iterature **R**eview
SMS	**S**ystematic **M**apping **S**tudy
QGS	**Q**uasi-**G**old **S**tandard

1 Introduction

Architecture represents the backbone of any system and what may appear as a small aspect in the architectural design process may decide the success or failure of a project(software related or otherwise), proving to be of significant value at a later moment.

As opposed to other phases of a project's or systems' life cycle, architecting is significantly dependant on the collaboration and communication between the parties involved. This communication represents the transfer and sharing of knowledge between architects, stakeholders and developers and is essential for a thorough understanding of the system to be built. In software architecture, this knowledge is also referred to as architectural knowledge.

According to [56], software architecture entails that a large amount of knowledge is being continuously produced and consumed. The work of [49] helps introduce a distinction between two types of knowledge: implicit and explicit. Implicit (or tacit) knowledge is the knowledge residing in people's heads, whereas explicit knowledge is the knowledge which has been codified in certain form, such as documents or models([56]). Best practices in software architecture often rely more on explicit knowledge rather than implicit knowledge and over the last decade there has been an increasing trend in trying to recover and/or capture implicit knowledge into explicit formats[11].

[39] define architectural knowledge as "the integrated representation of the software architecture of a software-intensive system (or a family of systems), the architectural design decisions, and the external context/environment". The CORE model suggests that architectural knowledge is a set of relationships between decisions, people, architectural design, and processes([19]). As this model tries to explain, architectural knowledge and the architecting process are a set of intricate and complex relationships between different entities. In order for this process to succeed and for the knowledge to get across from one entity to another, a good communication tactic is needed. A collection of such tactics used for sharing, capturing and understanding architectural knowledge is referred to as architectural knowledge management.

It has been estimated that communication with stakeholders can amount up to 50% of an architect's time effort ([36]). A thorough management of architecting knowledge can significantly decrease the effort needed for evaluating and maintaining a system by introducing traceability between architectural aspects and the actual implementation of the system, preventing knowledge vaporization and architectural drift, discovering new tactics and patterns for

architecture and most importantly, help communication between interested parties responsible for different domains(e.g. stakeholders, architects, developers). Additionally, capturing design aspects and architectural knowledge can help architects reuse successful experiences further on in other projects or even for solving of similar problems in the same project.

However, in practical scenarios, "architects are not likely to document their decisions and rationale, despite the well-established benefits of doing so"([11]). The most important aspect contributing to this lack of thorough and complete architectural documents is related to the costly nature of the process. Since little or no functionality is attributed to the documentation besides its value in architectural knowledge(AK) sharing and architectural issues in general, the value of AK management tends to be overlooked and the time attributed to the project tends to be used for more practical and concrete processes([11]).

As stated in [11] and [58], many approaches and tools for AK sharing and management have been proposed over the last decade, aiming to decrease the amount of time needed for the capturing of AK aspects and to facilitate the work of software architects. This new attention towards this research topic is linked to a change in the way of perceiving software architects and their work. As described by [11], "at the core of Architecture Knowledge Management, lies the principle of considering the architect as a decision maker instead of someone 'drawing boxes and lines'". When taking this into consideration, the high dimensionality and complexity of the architecting process become more clear, which in turn explains the need for more detailed research in the field.

A great majority of the approaches proposed so far present the disadvantage of being conceived or tested for a specific context or environment. This introduces the need for classifying and synthesizing these approaches in a format which facilitates their reuse and application in similar contexts to the ones they were designed for.

This thesis aims to analyse a wide range of approaches used in architectural knowledge sharing and summarize existing research in this field by performing a Systematic Mapping Study(SMS).

This type of study may help researchers in the field understand the state of research, obtain an overview of past research and to plan future research on areas which are not yet covered or which may need more detailed attention. Its main goals are to provide a detailed review on the topics covered by the primary studies analysed and also a qualitative analysis of each primary

study.

[51] describe SMSs as providing a structure of the type of research reports and results that have been published by categorizing them and often giving a visual summary, the map, of its results. According to [33], any secondary literature study needs to be conducted according to a well-defined methodology such that it is auditable and repeatable. [51] also states that while systematic reviews are frequent in medical research, they were quite neglected in the context of software engineering and up to the year 2008 when they published their paper, only one clear example of such a study was evident in research([4]). This points out the need for new research of this type in the domain of software engineering and more specifically, software architecture.

The SMS described in this thesis represents a preliminary research for a Secondary Literature Review(SLR) conducted in the context of a project aiming to synthesize existing approaches for AKS into a pattern language.

Patterns are considered to be a valid format for capturing and sharing knowledge, easy to use and understand([9]). As stated in multiple papers relevant for the field of software architecture([30], [26], [14], [11]), software architecture can be neatly described by common architectural patterns or styles, and the application of those patterns constitutes some of the most important architectural decisions([30]). By using patterns for synthesizing a particular type of problem-solution context, architects are able to reuse knowledge that proved to be of use in a previous similar situation.

While the initial plan was to directly conduct an SLR for the purpose of selecting and extracting information related to AKS from primary studies and then synthesizing this information into a pattern language, the topic proved to be broader than expected. As suggested by [32], considering these observations, a systematic mapping study proved to be a more appropriate exercise than a systematic review.

2 State of the art

[44] introduces knowledge as "an essential property for companies in contemporary economies" and stresses its importance especially for the case of software-development companies due to the knowledge-intensive nature of the products they sell. In order to exploit and develop this asset, they argue that it is important for companies to both acquire new knowledge and maintain, share and reuse the one already obtained. This need for knowledge management and more specifically knowledge sharing within a company has led to the birth of new technologies such as 'social software'. "Social software is a term for software systems that support human communication, collaboration and interaction in large communities"[34]. Social software helps workers deal with information overload, integration of new knowledge and reuse of existing practices by simplifying the organization of diverse content sources[44].

Researchers have split this knowledge into two main types: tacit and explicit[16]. Tacit knowledge, as opposed to formal or explicit knowledge, refers to a category of knowledge that is difficult to transfer to another person by means of writing it down or verbalising it([55]). The concept of tacit knowledge was introduced by Polanyi ([53]) who described it as knowledge that cannot be articulated, or the fact that "we can know more than we can tell". Over the last years, with the growth of software engineering and more specifically software architecture practices, it has become more evident that dealing with tacit knowledge for large projects as opposed to explicit knowledge brings many problems related to knowledge vaporisation and technical debt[11].

Architectural knowledge refers to knowledge related to software architecting practices, rationale regarding design decisions and design alternatives([39]). As stressed upon by many researchers in the field([11], [12], [25]), architectural knowledge needs to be managed in a well defined manner in order to facilitate reuse, project and system maintenance and evolution processes, etc.

According to researchers([3], [2], [29]), one can define two main practices for knowledge management, namely codification and personalization. While the codification process involves mostly storing knowledge in an explicit format such as a type of information repository, personalization focusses more on knowledge sharing and acts more similar to a set of connections between people and the knowledge they have("a list of who knows what", [3]). Thus, while both personalization and codification are mechanisms which enable knowledge sharing, the first one relies more on experts which retain the knowledge 'in their heads' and are able to transmit the knowledge through

verbal communication and the second focuses more on capturing the knowledge in specific formats and making it available to all members of the team. Each strategy brings its own benefits, codification helping with knowledge management on a long term basis and capturing knowledge in an easy understandable and easy to use format, while personalization enables a fast transfer of information[29]. Given their different areas of action, experts suggest that a combination of both strategies will bring a better results for organizations([29]).

Architectural knowledge sharing, a smaller part of knowledge management, is relevant in the sense of supporting architects in communicating with different stakeholders with different backgrounds and helping them to understand the requirements of a system in the same manner. "AKS makes architectural knowledge available to others"([39]). [46] distinguish between two types of knowledge sharing:

- Passive sharing: the retrieval and reviewing of codified knowledge (e.g.: in the form of books, web pages, documentation)
- Active sharing: meetings between stakeholders, publisher-subscriber strategies(e.g., RSS, contents for distributed teams) under a collaborative environment, direct transfer of information through verbal communication or practical examples in the form of prototypes.

The next section presents the state of the art regarding current approaches in architectural knowledge sharing.

2.1 Existing approaches for architectural knowledge sharing

According to [13], the various techniques proposed for aiding the software architect in managing architectural knowledge may be categorized as:

- theoretical: centred towards proving a specific theory, argumentation regarding the usage of a particular technology, providing rationale behind certain architectural decisions
- experimental: focusing on practical demonstrations, experiments which aim to measure or prove that the response of the system in a specific context related to particular quality attributes is as expected or desired

[11] give a short summary and comparison of several theoretical approaches proposed in the last decades. The common elements of architectural knowl-

edge captured with the help of these models were design issues, design alternatives, and a justification for the decisions taken. They argue that these models, including gIBIS ([18]), DRL ([40]), QOC ([43]) were "not widely accepted by practitioners". They later strenghten this viewpoint by introducing an explanation given by Conklin and Burgess-Yakemovic which describe architecture models as "unwieldy of loosely organized textual information that is difficult to use" ([17]).

The limitations of textual documentation have been thoroughly investigated over the last years. [69] describe textual documentation as "error prone thus often inconsistent and ambiguous, difficult to analyse and verify, inefficient in presenting complex concepts". They introduce a scheme for the categorization of documents according to the level of formality used:

- textual (informal)
- semi-formal (diagrammatical)
- formal (mathematical)

According to [69], an increase in the level of formality of a textual document helps resolve some of the drawbacks, however, they also stress on the fact that "increasing the level of formality of the models of architectural decisions seems to be an important research challenge".

Various scientific studies have tried to present alternative approaches which address the drawbacks of textual documentation. [20] address the limitations of "file-based documents such as text files, diagrams, source code, and meeting notes" and introduce an ontology-based documentation which enables AK to be described "unambiguously and comprehensively for all of its different uses" while also making it "explicit and unambiguous by applying a semantic structure".

[61] suggest that most companies prefer a different, less formal approach for architectural knowledge sharing, namely the concept of 'walking architecture'. This relies on one person or a group of persons which are responsible for maintaining and updating the architecture of a product, are always a part of discussions regarding that product and help introduce new employees to the existing architecture at a specific point in time. Although the usage of this concept results in "representations of the architecture thus are temporary and partial, e.g., sketches on whiteboard and scrap paper used in a specific situation", this practice is extensively used and the authors suggest that future research should "focus not only on documentation and tools when improving architectural practices, but also on the development of social

protocols around such methods and tools".

Several surveys and qualitative studies are also present in the literature which aim to compare and analyse approaches for different aspects of AKM, including AKS.

[64] present an expert survey aiming to gain insights into the different kinds, influence factors, and sources for design decisions while also analysing the way in which ADDs are captured in practice. Some of the results presented include the fact that architects seem to be heavily structure- and technology-minded when thinking about and documenting architecture, while "nonexistence" decisions(bans) are rarely documented. These results also apply to the case of AKS in the sense that the knowledge transfers related to design decisions revolve mostly on decisions which were taken and less on the decisions that were discarded. An important observation made in the study which confirms previous findings is that "documenting decisions makes sense even in smaller teams because knowledge vaporization is also present at the level of individuals". While mainly revolving on documentation and the capturing aspects of AKM with regards to design decisions and their rationale, the study also offers some information about the AKS related functionalities of the analysed approaches. However, the focus on AKS remains of little standing next to AK capturing.

[54] present a survey that analyses the suitability of existing SA decision-making methods to support group decision making. In this context, the authors also analyse the support of the selected approaches with regard to AKS and present several methods for group decision-making which also present functionalities for AKS. The study focusses on ADD related tools since the process of taking decisions within the scope of SA is highly collaborative and involves different stakeholders. While the study presents interesting results with regards to the ability of the evaluated methods to cope with different GDM problems, it also outlines liabilities and alternatives for each approach. The main difference between this study and the one presented in this thesis revolves around the differences between GDM and AKS. GDM is concentrated on the decision making process and might involve knowledge sharing activities, it's main goal is to produce a set of ADDs which best represent the needs of all stakeholders involved, while AKS focuses on knowledge sharing as a main purpose and has a value on a longer term.

2.2 Challenges and motivation

Many problems have been attributed to a lack of explicitly documented knowledge regarding the software architecture of different projects[57]. Researchers agree that a thorough management of architectural knowledge greatly decreases the amount of effort needed in understanding the architecture of a system when the experts or the creators of the architecture are no longer available([11], [10], [19]). [31] draws the attention on two major problems in missing or incomplete AK sharing: "awareness is often missing of which AK is relevant to share and the multi-disciplinary context, creating an obstacle to sharing this knowledge".

Numerous works provide solutions for capturing and representing knowledge in order to help facilitate the process of AKM: [35] [60] use list of attributes for characterizing important AK elements like design decisions, [12] lean towards more flexible models defined as lists of mandatory and optional attributes easy to personalize according to the users' needs, etc.

However, not many of these papers focus on bringing a concrete solution for sharing knowledge in particular and the ones that do are usually limited at addressing a project-wide context and do not cover a general view on the domain. Also, while the work presented in recent years presents numerous approaches for AKS, there is no evaluation of the quality of these works and no research which summarizes aspects related to validity. Given the amount of studies published in recent years which seem to follow this new interest for AKM and AKS practices([11]), a need for collecting, analysing and summarizing this research is evident.

Towards this purpose, the aim of the SMS presented in this thesis is to gather and select a list of primary studies specifically related to AKS practices with the help of several inclusion and exclusion criteria, to extract quality information from these studies and to obtain an overview map which is able to answer specific points of interest (presented in the research questions).

2.3 Related studies

The state of the art in the domain of AKM and more specifically AKS presents several secondary literature studies focussed mainly on AKM aspects in general or on different tools and approaches for supporting AKM processes.

[44] present a survey conducted in a group of Brazilian software development companies in order to assess the usage of new tools and technologies and in what way these tools and technologies help knowledge-sharing. This survey is backed up by a previously conducted systematic literature study([45]). The results of the study were validated against data from the literature and three knowledge-sharing and learning theories: the double-loop learning theory proposed by Argyris and Schön([27]), Wenger's communities of practice theory([65]) and Nonaka and Takeuchi's theory of knowledge creation([48]). The findings presented by [44] suggest that most tools used by the companies analysed do not follow the concepts described by these theories although the article, with evidence from [45] considers them to be the most used in software engineering related studies. While this study presents important findings with regard to knowledge sharing in practical scenarios, its focus is on knowledge sharing in general and the tools presented have little functionality related to AKS in particular.

[22] presents a systematic literature review exploring different knowledge-based approaches for software documentation, their impact on the quality of such documents and the benefits and cost of using them. After analysing 60 finally selected studies with regards to twelve quality attributes of software documents, the authors extracted four cost categories, and nine benefit categories of using knowledge-based approaches in software documentation. Finally, the study points out important research gaps and suggests future research directions on the topic. One of the points they propose for further research focusses on using textual documentation for more than just capturing knowledge and to expand its purpose to other AKM aspects such as knowledge reuse, retrieval and sharing. While this study is similar in purpose and scope to the one presented in this thesis, the focus remains pointed at a small range of approaches in AKS related to documentation and knowledge capturing. Although the article mentions the quality of the analysed approaches with regards to knowledge sharing, a more detailed analysis pointed in this direction is missing and the study selection revolves solely on approaches related to textual documentation disregarding other types of knowledge sharing.

[28] presents results from an SLR focused on model-based AKM approaches and their support for variability management of SA documentation. Their results include a list of the main elements of SAKM models which, as the authors argue, can serve as a first step towards a commonly agreed approach for documenting architecture knowledge. Similar to [22], this study also focusses on documentation and knowledge capturing while only presenting little

information with regards to AKS.

The literature is also abundant in studies focussed on AKM approaches in general. [63] and [58] are such examples which analyse and compare different approaches and, respectively, tools for AKM activities. [63] analyses approaches based on the quality they present in performing the main AKM activities i.e., capturing, using, maintaining, sharing, and reuse of architectural knowledge and the quality of the evidence provided for each approach. [58] compare five different tools designed for helping architects with AKM activities with the help of an evaluation framework based on ten criteria(one of which refers to sharing knowledge and decision-making activities within distributed teams). Both studies present relevant findings with regards to AKM activities in general, but are different from the study on which this thesis is focused since they are applied to the more general scope of AKM and do not focus in great detail on AKS.

[59] present a systematic mapping study around software architectural decisions. The study reports on existing methods for capturing ADDs, the way in which ADDs are viewed with respect to other related SA entities such as requirements and quality attributes, provides a list of domains in which ADDs are researched the most and extract information related to how uncertainty in ADDs and sharing of group decisions are handled. The study gives a valuable overview of 10 years of research with focus on ADDs and helps researchers and stakeholders understand what existing papers propose and what future research might focus on. While this study presents a similar research methodology and protocol to the study presented in this thesis, they differ in the scope of research: [59] only focus on ADDs as SA entities and present results on every aspect concerning them and the currently described SMS focuses on AKS approaches and AKS related studies.

[41] also discuss he findings of a SMS which revolves around the practice of knowledge-based approaches. The study aims to identify, analyse the application of knowledge-based approaches such as knowledge recovery and knowledge capturing. The study reports that an increased interest in the application of knowledge-based approaches in software architecture is to be observed in recent years and that knowledge-based approaches, including knowledge capture and representation, reuse, sharing, recovery, and reasoning, have been employed in a spectrum of architecting activities and different domains.

A number of other systematic mapping studies are present in the literature which present results within the domain of software design and software testing: [47] and [24] describe their findings with regard to software product line

testing processes and [4] presents a study reporting on evidence in Object-Oriented software design.

While the study described in this thesis follows similar guidelines to the SLRs presented above and is related to the domain of AKM, all the approaches selected for the data extraction and synthesis phases of this study present functionalities for AKS in particular.

3 Systematic mapping study

In the context of the project "Towards a Pattern Language for Architectural Knowledge Sharing", a systematic mapping study(SMS) has been conducted. The main steps in the SMS can be defined as: search, selection, data extraction and data synthesis. My personal contribution revolved on the last three phases.

The following sections will describe the research methodology used for performing the SMS and outline the main steps for each phase(search, selection, data extraction, data synthesis) while detailing the research questions of main interest for the study.

3.1 Research Methodology

[32] define an SMS as "A broad review of primary studies in a specific topic area that aims to identify what evidence is available on the topic." In this article, they also identify three main reasons for performing such a study which can be shortly described as:

- Identification of evidence clusters and evidence deserts
- Directing the focus of future systematic reviews
- Identifying areas for more primary studies to be conducted

Three main types of studies can be observed, including an SMS. The two other types on this list complementary to SMSs are systematic literature reviews(similar to an SMS, but mostly conducted when the topic of research is very narrow or when a lot of evidence has been presented in that field) and tertiary reviews("a systematic review of systematic reviews, in order to answer wider research questions"[32]).

Given the nature of the topic of this thesis and the existing research focused on it, it can be stated that the number of SLRs conducted with respect to this topic does not justify the application of the domain on the performance of a tertiary review(see subsection 2.3). However, since SMS and SLR methodologies are often similar and the boundary separating one from the other is usually ambiguous, the differences between the two types of study have been taken into consideration in a more thorough manner. Both [5] and [32] provide a list of essential differences between the two approaches:

- how a research question is formed([32] states that Mapping studies generally have broader and more numerous research questions)

- how the publication corpus is explored(according to [32], search terms for SMSs are less focused and provide a larger number of publications for reviewing),

- what is the reviewing style(a broader data extraction process for an SMS than for an SLR, which tends towards categorization)

- what is the principal objective of review outcome(while an SMS focuses on summarizing the data analysed in the purpose of answering the research questions, the results will be more limited than those of an SLR[32])

In the context of these differences, both [5] and [32] agree that while an SMS methodology focuses on a very general topic and covers a broad collection of primary studies, aiming for classification of publications for a better understanding of the topic, an SLR produces more precise results and is conducted over a smaller number of publications.

Although the initial planning for this thesis was to perform an SLR study focused on extracting a pattern language for approaches in AKS, the amount of publications and their nature(many publications proposed new approaches with a relatively low level of evidence and very few of them compared or analysed the state of the art in general - see section 3.3 for a detailed explanation) demanded for a more general, broader type of study to be conducted in advance. For the reasons described above, an SMS proved to be more appropriate in this context.

The SMS described in this paper follows the guidelines introduced by [32].

3.1.1 Research questions

The following table presents the research questions marking the main interest points of the SMS conducted for this thesis:

RQ-1	What approaches for sharing AK have been proposed in the literature?
RQ-2	What is the intended project context(domain, system type, size) of the approach?
RQ-3	Which knowledge management strategy does the approach support?(personalization, codification, hybrid)
RQ-4	What are the mechanisms of the approach to support sharing of AK?
RQ-5	What type of architectural knowledge does the approach support? RQ-5.1. What type of AK is the approach intended for?(design, reasoning) RQ-5.2. Which architectural knowledge entities are captured by the approach?(ADDs, quality attributes, requirements, etc.)
RQ-6	How prescriptive or descriptive is the approach?
RQ-7	What level of evidence is provided for each approach?
RQ-8	How many studies focus on a specific approach and how many compare or present an overview of existing approaches?

Table 4: Research questions

Research question RQ-1 aims to collect a list of proposed approaches in the context of AK sharing. This list is useful for understanding what types of approaches have been proposed, whether one type of approach has been focussed upon more than others and for obtaining a general overview of the current state of the art.

Questions RQ-2 to RQ-7 attempt to further classify the approaches listed with the help of RQ-1 by taking into account the context for which the approaches were designed or certain characteristics.

RQ-2 aims at documenting the environmental specificities for which an approach has been proposed or tested(reported as industrial practice or documented as part of a case study). The project context related to a specific approach is documented using an abbreviated version of the model proposed by [37] which focuses only on domain and system type(agile, software intensive systems, etc.). The purpose of understanding the project context behind a certain approach is to be able to get an overview on how many approaches focus on specific types of projects, what approaches are more general and can be used for a variety of contexts and also what types of contexts have received little or no attention from researchers in the domain.

RQ-3 is used for classifying the approaches analysed with regard to the management strategy they cover. According to researchers in AKM ([29], [25]), there are two types of management strategies: personalization and codification. The codification strategy revolves around knowledge being carefully stored in a certain type of knowledge repository(e.g. databases, documents) to which company employees are granted access, while personalization enables knowledge sharing activities mainly through person-to-person contacts and knowledge is closely tied to the person who developed it([29]). Due to the importance that this strategy has on AKM and more specifically AKS ([25] consider a hybrid strategy to be the most efficient), this research question will help categorize the approaches accordingly.

Question RQ-4 aims to cover the way in which each approach functionally achieves its purpose for AK sharing(e.g. using a central knowledge repository, yellow pages linking people to specific sub-parts of the companies knowledge, tools for helping collaborating teams to share knowledge regarding design decisions or other AK elements).

Question RQ-5 with sub-questions RQ-5.1 and RQ-5.2 focus on the AK elements and the type of AK that are being captured or shared using the approach. The AK entities taken into consideration correspond to the ISO/IEC/IEEE 42010 standard for architecture descriptions with three extensions: the decision documentation framework proposed by [62], the technical debt documentation framework proposed by [42] and architectural assumptions ([68]). The reason for adding these extensions is the increasing attention they have received over the last years([42], [68], [38], [50]).

RQ-6 classifies approaches into prescriptive or descriptive. An approach with a prescriptive nature is used for capturing AK in a normative way, describing "how things should work", while a descriptive approach is used to document AK in a more flexible way, focusing only on "how things actually work"[6]. This question helps understand the preferences of architects regarding the nature of the approaches used for AKS and also helps obtain an overview on how many approaches of each type have been proposed.

As a continuation for question RQ-2, RQ-7 aims to document the type of empirical evidence given for an approach. The classification of obtained evidence is made according to [66] into five categories:

- no evidence,
- first or second party claim(any information published by anyone with a vested interest),

- third party claim(evidence published as an experience report, lessons learned, single company survey by someone without a vested interest),

- circumstantial evidence(well-documented controlled experiment by someone with a vested interest),

- evidence(well-documented controlled experiment by an independent researcher, series of case studies within a company),

- strong evidence(evidence from industrial practice).

Finally, RQ-8 aims to collect and synthesize all the information obtained from and classify the studies selected between three main types: studies that focus on, evaluate and/or present a specific approach; studies which compare two or more approaches according to a set of criteria or through case studies; studies which present an overview of the state of the art with regard to AKS practices and approaches. These classes of studies correspond to the three inclusion criteria used for the selection process(see section 3.2.2) and allow for a separation of studies which present approaches from studies which compare and evaluate the state of the art.

3.1.2 Quality assessment

In order to assess the quality of the finally selected studies, a quality assessment was performed on all studies. Table 6 presents the quality assessment form composed of nine questions, seven questions for which the answers can be determined by a numerical attribute between 0 and 1. QA-8 is a free text description of the study and finally, QA-9 helps better distinguish studies focusing on a particular approach from the others.

Id	Question	Answer type
QA-1	Is there a rationale for why the study was undertaken?	0 = no evidence; 1 = evidence from demonstration or toy examples; 2= evidence from expert opinions or observations; 3= evidence from academic studies (e.g., controlled lab experiments); 4= evidence from industrial studies (e.g., causal case studies); 5 = evidence from industrial practice.
QA-2	Is there an adequate description of the context (e.g. industry, laboratory setting, products used, etc.) in which the research was carried out?	0=no; 0.5=to some extent; 1=yes
QA-3	Is there a justification and description for the research design?	0=no; 0.5=to some extent; 1=yes
QA-4	Has the researcher explained how the study sample (participants or cases) were identified and selected, and what was the justification for such selection?	0=no; 0.5=to some extent; 1=yes
QA-5	Is it clear how the data was collected (e.g. through interviews, forms, observation, tools, etc.)?	0=no; 0.5=to some extent; 1=yes
QA-6	Did the researcher critically examine their own role, potential bias and influence during the formulation of research questions, sample recruitment, data collection, and analysis and selection of data for presentation? (limitations and threats to validity)	0=no; 0.5=to some extent; 1=yes
QA-7	Do the authors discuss the validity and limitations of their findings?	0=no; 0.5=to some extent; 1=yes
QA-8	Reason for inclusion (1 sentence about why the study was included, .e.g, how is it related to AKS)	free text
QA-9	Does the study presentor discuss an approach for AKS?	(yes/no/to some extent)

Table 6: Quality assessment form

3.1.3 Planning

During the period of time needed for conducting this study, several major modifications have been performed on the original planning due to relatively large differences between the initial expectations and the real effort needed.

The initial planning was designed for a period of six months and was divided between nine work packages(WP):

WP 1: Choosing between different proposed projects, discussing potential collaborations with project supervisors
WP 2: Defining project scope, structure and planning
WP 3: Writing the deliverable document
WP 4: Studying of existing literature in the domain area of interest
WP 5: Selection of relevant articles
WP 6: Extraction of relevant information from the studies selected in the previous step
WP 7: Organization of results and findings in the form of a pattern language for architectural knowledge sharing
WP 8: Data analysis and coding of a case study regarding a plug-in for architecture decision sharing
WP 9: Review of the written deliverable

Figure 1 shows a time line representing the initial planning constructed before the study was initiated.

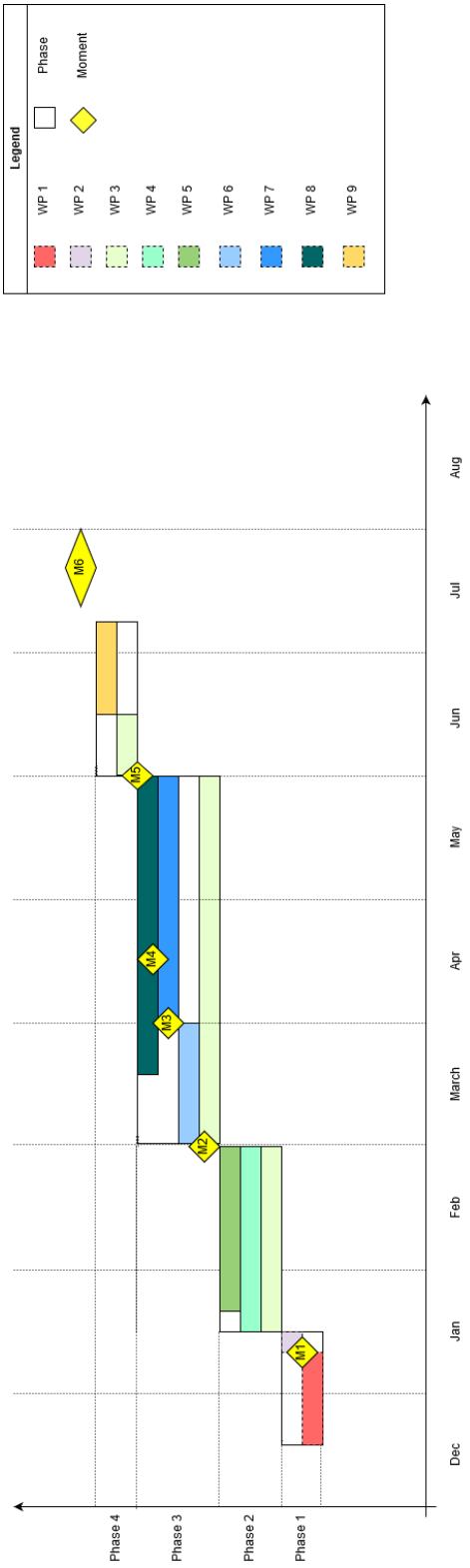

Figure 1: Initial planning diagram

The main modifications brought to this planning were:

- a one month extension brought to the final deadline
- a change in the type of the study performed from SLR(secondary literature review) to SMS(systematic mapping study) with a transformation of the data synthesis format
- removal of case study

These modifications were determined by several external and internal factors described in more detail in section 3.2.5 along with several measures of prevention which could improve the time effort of similar future studies.

3.2 Selection

This section outlines the methods used for the selection process, the sources used for the retrieval of publications relevant for the scope of the SLR, as well as the inclusion and exclusion criteria used. Finally, subsection 3.2.5 gives insight on some observations made during the performing of the selection phase which might be of use to researchers interested in the same type of study.

3.2.1 Sources selection

The scope of the search process of the SLR described in this thesis revolves around the two main types of search performed: manual and automated selection(see section 3.2.3 for a more detailed description of each search process).

The manual search has been performed on a list of venues. This list has been derived in concordance with the list presented by Zhang et al.([70]), which investigates venues mostly used for manual search processes in the context of secondary studies related to Software Engineering. Excepting several small additions to the list proposed by Zhang et al. which were considered for performing a more detailed search, table 8 shows the list of venues and journals presented by [70] which were used in the manual search process for this SLR.

The selection of the databases used for the automated search process was made according to the lists proposed by [70] and [7] which present them as commonly used for search processes in secondary studies related to software engineering. Table 10 presents a complete list of the databases used in this study.

Name	Abbreviation	Type	Database Catalogue
Journal of Systems and Software	JSS	journal	ScienceDirect
International Journal on Software Engineering and Knowledge Engineering	JSEKE	journal	WorldScientific
Working IEEE/IFIP Conference on Software Architecture	WICSA	conference	IEEE Xplore
International Conference on Software Engineering & Knowledge Engineering	SEKE	conference	DBLP
European Conference on Software Architecture	ECSA	conference	SpringerLink LNCS
CompArch	CompArch	conference	ACM
Information System Technology	IST	journal	ScienceDirect IEEE
International Conference Series on the Quality of Software Architectures	QoSA	conference	ACM

Table 8: Journals and venues selected for the manual search process

The study selection process was also validated against a time frame corresponding to the period between 2004 and July 2016. The rationale behind the choice of these boundaries is as follows:

- the lower boundary roughly corresponds to the year presented by [11] as the year of emergence of AKM as a research area, which gives reason to believe that the majority of studies related to AKS have been published after this moment in time and
- the upper boundary for reliability issues regarding the repeatability of the search process

The selected studies had to be written in English.

3.2.2 Inclusion and exclusion criteria

This section lists and describes the rationale behind the inclusion and exclusion criteria used for the selection of studies relevant to the domain of this SMS.

Name	**Type**	**Rationale for inclusion**
IEEE Xplore	Database	[7]
ACM Digital Library	Database	[7]
ScienceDirect	Database	[7]
SpringerLink	Database	[7]
Web of Science	Database	[70]
Wiley InterScience	Database	[70]

Table 10: Databases selected for the automated search process

IC-1	A study presents an approach for sharing architectural knowledge.
IC-2	A study evaluates/compares one or multiple approaches for sharing architectural knowledge.
IC-3	A study is about the state of the art of AK sharing or industrial practices with regard to AK sharing.

Table 12: Inclusion criteria

Table 12 presents the list of three inclusion criteria. These criteria helped categorise the selected studies between studies which present new approaches for AKS(IC1), studies which aim to validate the quality of existing approaches by comparing them to one another using a set of quality criteria(IC2) and studies which aim to gain an overview of the state of the art within the domain(IC3).

EC-1	A study is not written in English.
EC-2	A study is about knowledge sharing in general and not specifically about architectural knowledge.
EC-3	A study has been published before 2004 or after July 2016.
EC-4	The study is an editorial, position paper, abstract, keynote, opinion, tutorial summary, panel discussion, or technical report.
EC-5	A study is a duplicate of an already selected primary study.
EC-6	The full text of the study cannot be obtained.
EC-7	A study only focusses on capturing AK and does not discuss or support an AK Sharing process, e.g., the study only presents a specific architectural viewpoint.

Table 14: Exclusion criteria

Table 14 presents a list of seven exclusion criteria used in the selection process of the SMS. With the exception of EC-2 and EC-7, all exclusion criteria were designed to limit the search process to a scope range previously established(limits related to time span, language, duplicate studies, etc.) or to disregard studies for which the full text could not be obtained.

EC-2 aims to help exclude studies related to sharing of other types of knowledge not related to architectural knowledge.

Lastly, EC-7 was added at a later stage in order to decrease the large amount of studies selected by excluding the studies which specifically focused on capturing knowledge and presented no information related to sharing in particular.

3.2.3 Search process

The main methods used during the search process for the SMS conducted during this thesis consist of two types of search strategies: manual and automated search.

The manual search process consisted in several members of the team scanning the selected venues for relevant publications within a given timespan. This type of search process enables researchers to perform a high-quality selection of papers and ensures that the issues found are indeed relevant for the scope of the search. However, the manual search needs a high amount of time-effort, given that the amount of publications which need to be verified is significantly

larger than the number of publications which present a real interest for the research.

The second strategy used, the automated search helps speed up the search process and is conducted using specific search strings for every database selected in conformity with their sets of rules. A general search string containing a combination of terms specific to the scope of the research is first determined, and then it is tailored for each search engine. For a list of the search strings used for each database, see appendix(section A). While this type of search has the benefit of automatically sorting out most publications which are irrelevant for the scope of the SLR, its performance is bound to certain variable aspects such as the quality of search string, capability of search engine, and diversity of the subject([70]).

The process of search string construction followed the guidelines suggested by Petticrew and Roberts in [52]. The PICOC criteria([52]) consist of five different aspects which help define the main focus points of a search: population, intervention, comparison, outcome and context. Figure 2 gives an overview of these criteria applied to the scope of the search conducted for this SMS.

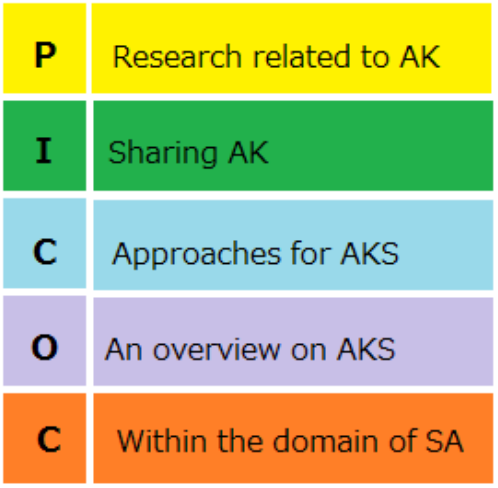

Figure 2: PICOC criteria with respect to the SMS search process conducted within the project "A pattern-language for AKS"

Table **??** presents the way in which the PICOC criteria were applied in order to obtain the terms constructing the search strings.

PICOC	**Derived Term**	**Synonyms and Alternatives**
Population	Architecture knowledge	Architectural knowledge, architecture design, architectural design, architectural design decision, architecture design decision, architecture decision, architectural decision, design rationale, design decision, architecture assumption, architectural assumption, technical debt,quality requirement, non-functional requirement, architecturally relevant requirement, architecture requirement, architectural requirement, architectural concern, architecture concern
Intervention	Share	Capture,Communicate, Document, Documentation, Exchange, Transfer, Understand
Comparison	Approach	Strategy,Mechanism, Tool, Process
Outcome	AKS body of knowledge	
Context	Software architecture	

Table 16: Search terms derived using PICOC criteria

The search string resulting from these terms was used for the automated search for all selected databases with only minor adjustments implied by the specificities of each database. In the case where a database would not allow

a search string composed by as many search terms, the search string was split between the optional terms(e.g. for a search string such as "architecture knowledge sharing OR architectural knowledge sharing", the search process would be performed two times, for "architecture knowledge sharing" and "architectural knowledge sharing" separately and then the results would be collected in the same repository).

The final search string used for the automated search process was the following:

```
1 (("architectural knowledge" OR "architecture knowledge" OR "
    architecture design" OR "architectural design" OR "
    architectural design decision" OR "architecture design
    decision" OR "architecture decision" OR "architectural
    decision" OR "design rationale" OR "design decision" OR "
    architecture assumption" OR "architectural assumption" OR "
    technical debt" OR "quality requirement" OR "non-functional
    requirement" OR "architecturally relevant requirement" OR "
    architecture requirement" OR "architectural requirement" OR "
    architectural concern" OR "architecture concern")
2 AND
3 ("share" OR "capture" OR "communicate" OR "document" OR "
    documentation" OR "exchange" OR "transfer" OR "understand")
4 AND
5 ("approach" OR "strategy" OR "mechanism" OR "tool" OR "process"
    )
6 AND
7 ("software architecture"))
```

If supported by the database, the search string was applied to the full text of the publications.

[70] suggest using a combination of both types of search for a better performance and also refer to the quasi-gold standard(QGS) concept as a means to assess the quality of the search queries. According to these instructions, the manual search process helps both refine the automated search process and establish the quasi-gold standard as a subset of primary studies relevant for the SMS. The QGS is defined for each database using only the venues indexed by that particular database. By verifying that the subset of articles which compose the QGS for a specific database are present in the results of the automated search process for that database, the quality of the search string is determined. This verification is measured using formula 1 based on the sensitivity and precision metrics.

$$Quasi-sensitivity = \frac{Numberofrelevantstudiesretrievedbyautomatedsearch}{NumberofarticlesinQuasi-goldStandard} * 100 \quad (1)$$

For the purpose of this SMS, sensitivity is defined as the ability of the search to find all relevant studies present in a database, while precision corresponds to the number of relevant studies found with regard to all found studies. [70] defines the threshold for quasi-sensitivity between 70% and 80%. The limits chosen for this study, as suggested by [21] for the optimum strategy are between 80-99% for sensitivity and between 20-25% for precision.

3.2.4 Selection process

Using the inclusion and exclusion criteria presented in section 3.2.2, the studies retrieved through the search process from each database or venue were selected or discarded for the data extraction phase during the selection process. This process was performed in two phases(1st round and 2nd round selection) by two different researchers in order to increase reliability.

The agreement between the two researchers was measured using the Cohen Kappa statistic suitable for nominal data having the advantage that it takes into consideration not only the number of studies on which the two researchers agreed or disagreed upon, but also the number of studies selected by chance([15]). Formulas 2 and 3 describe the mathematical model used for assessing this agreement:

$$k \equiv \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e} \quad (2)$$

$$p_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2} \quad (3)$$

The two-round selection process was performed as follows:

- 1st round selection: The list of studies retrieved through the manual and automated search processes was stored in different *.bib* files according to the database/venue through which they were found. The details of each study included a minimal set of attributes comprised by: title, authors, year, abstract and keywords, URL to fulltext.

  The selection at this point was based only on the review of the title, abstract and keywords, and not on the full text. In the case when

the abstract provided too little information and a verdict could not be drawn only from it, the conclusions were also consulted, according to the suggestions made by Brereton et al.([7]). The selection was performed in a rather inclusive manner, helping minimise the loss of potentially interesting studies.

- 2nd round selection: All studies selected by at least one reviewer in the first round were reevaluated during the second round. This reevaluation was based on the full text of the articles and was also performed by two different researchers. In the case of conflicts between the two researchers, the differences were discussed together with a third researcher until an agreement was defined.

3.2.5 Lessons learned

Several major modifications were made to the planning throughout the implementation of the project. This section presents the lessons learned from the project implementation, discusses why the planning modifications were necessary and points out suggestions on how they could have been avoided as advice for similar future research.

Existing guidelines
The SMS described in this thesis was performed following the guidelines presented by [7], [32] and [70]. These guidelines were applied for the following steps in performing the SMS:

- selecting the sources consisting in different venues and databases presented as most commonly used for secondary studies related to the domain of Software Engineering
- performing two types of search: manual and automated in combination with the use of QGS in order to verify the sensitivity and precision of the databases
- defining the main phases as the study as search, selection, data extraction and data synthesis
- defining the research questions, quality assessment form and data extraction form

Review of publications
The venues and databases used for the search process presented in 10 and 8 were selected according to [70]. A collection of publications from each database and venue was saved in a separate *.bib* file for easily tracing articles

back to the venue or database through which they were found. Each file was then analysed by two reviewers within two rounds for the selection process. Overall, a set of 13796 publications(13327 after the removal of duplicates within each file) were reviewed during the first round selection out of which 730 articles were approved for the second round. However, during the second round selection, it became more evident that many publications were present in two and sometimes even more files. This led to the same article being reviewed multiple times(many articles were present in three different files and were reviewed up to six times for the first round only) for the same selection round. Ultimately, the effect of this type of organisation of the publications produced a major time deficit which could have been prevented earlier on.

As a solution to this problem, all publications were collected within a single *.bib* file and with the help of the *Jabref* tool, duplicates were removed. A tracing mechanism was implemented helping to identify all venues and databases in which the article was initially found. Unfortunately, this solution was adopted later on in the selection phase, after the second round selection had already been completed for 5 venues and databases. Despite this fact, however, it allowed the removal of 200 duplicate studies which represented more than a quarter of all studies selected during the first round. The fact remains, however, that a lot of effort could have been spared by a more detailed examination of the files before starting the selection phase and by a better file management system.

Figures 3, 4 and 5 display Venn and chord diagrams which show the overlapping between the finally selected studies from all databases. As can be observed, multiple studies were found from the search performed on two or even three databases. If the selection process would have been finalised using the first approach, these articles, passing through both rounds of selection would have been reviewed 12 times(as opposed to four times in the ideal situation, two times by each reviewer in each round).

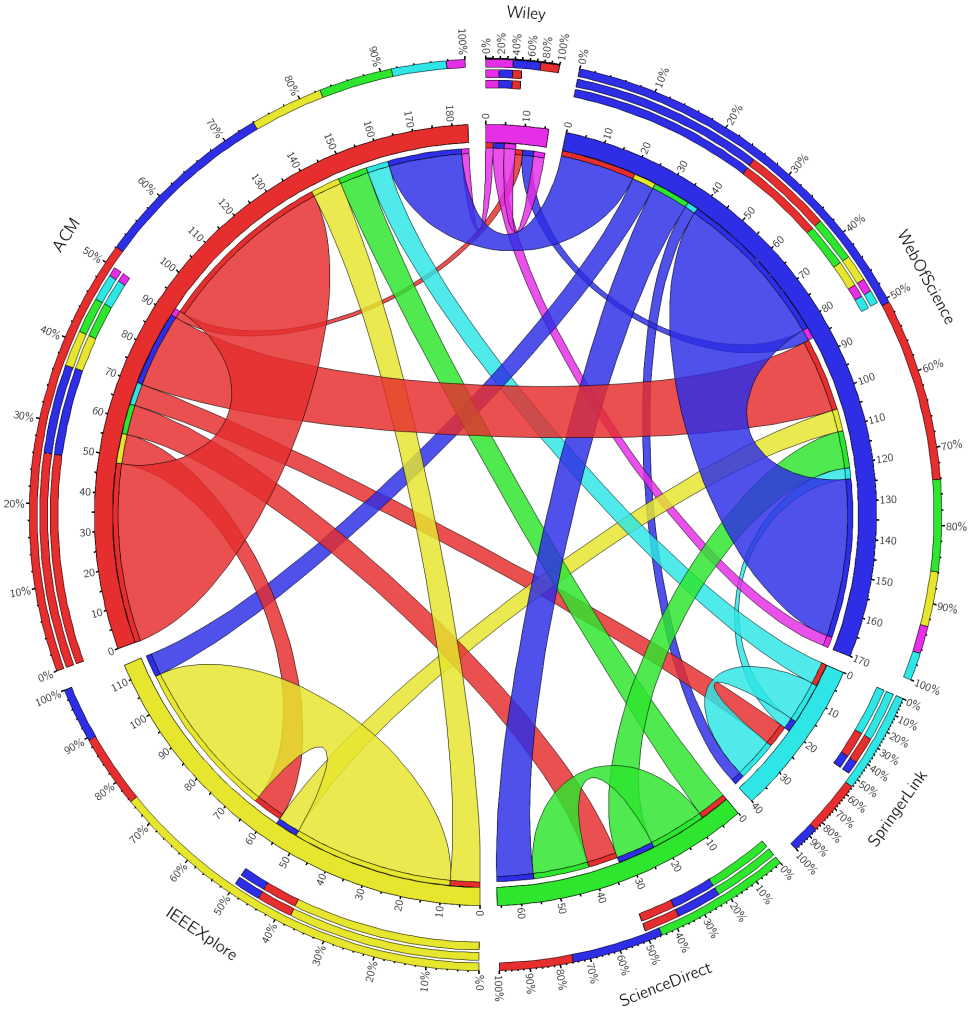

Figure 3: Chord diagram representing overlap between finally selected studies from all databases

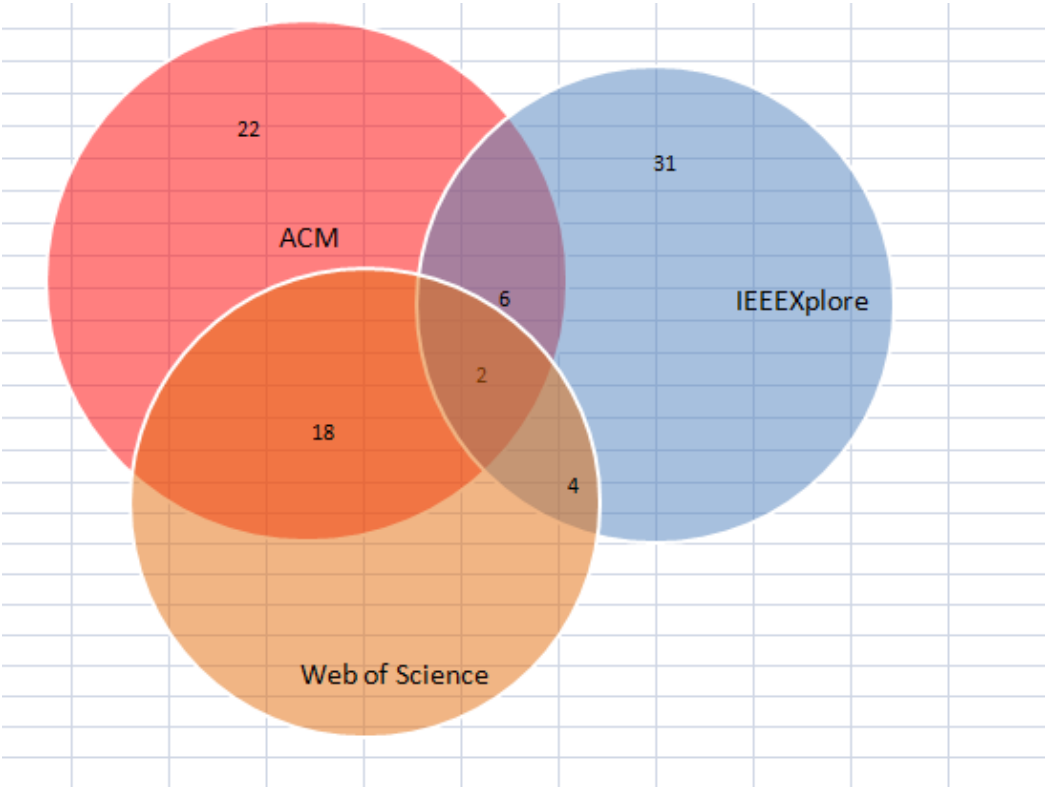

Figure 4: Venn diagram representing overlap between finally selected studies from ACM, IEEEXplore and Web of Science

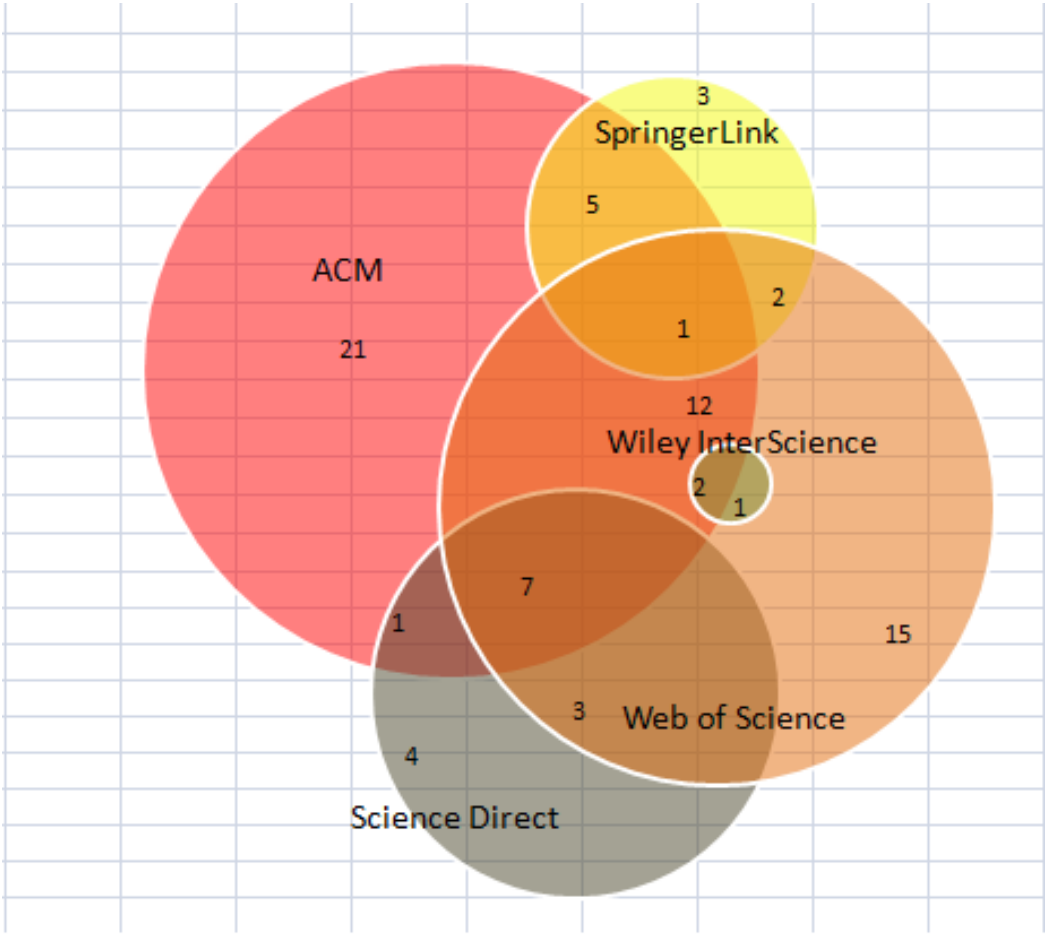

Figure 5: Venn diagram representing overlap between finally selected studies from ACM, ScienceDirect, Web of Science, Wiley and SpringerLink

Search strings

Another aspect which significantly slowed down the performance for the secondary study was present during the automated search process. As detailed in section 3.2.3, the search string was constructed before the beginning of the search process and then used for all databases without any major changes(the search terms were kept the same for all databases).

This tactic was useful in terms of database comparison(since all databases were searched using the same search string, comparisons can be made in order to assess the sensitivity and precision of the search string, etc.), however it had certain drawbacks related to the amount of effort spent on performing the search process. One such drawback was inflicted by the application of the QGS.

A lot of effort was spent for tailoring and iterating through different versions of the search string which would be able to satisfy the QGS for all selected databases. When a search for a certain database would fail to produce the subset of studies included in the QGS of that particular database, the search string would be tweaked in order to include all or most studies(within the limits of precision and sensitivity decided upon). These minor tweaks would then be tested again for all other databases in order to make sure that the changes did not affect the QGS.

This aspect along with the different specificities of each database(regarding number of allowed search terms, possibility of search throughout the full text, possibility of search using more than one term at a time, etc.) required for a lot of time and effort to be spent on the search process alone, which introduced a delay in the planning of the next phases as well.

3.3 Data extraction

This section will present the data extraction process. The first sub-section will describe the data extraction form and how each entry relates to one or more research questions as well as the way in which the process was performed. The second sub-section will describe preliminary findings and discuss preliminary observations which can be noted as immediate consequences of the data extraction process(e.g. number of studies analysed).

3.3.1 Extraction form

The data extraction process was performed using a data extraction form presented in table 18. The table shows the name of each entry column in the table, its type and the research question or questions to which it relates. The form was completed using the Jabref tool([1]) by setting up the specific preferences and then transferring the results to an excel file through a custom export for a better visualization.

Field	**Type**	**RQ**	**Notes**
Citekey			Identification of studies
Title	Free text		
Authors	Lastname1, Firstname1; Lastname2, Firstname2		
Year of publication	YYYY		
Source	Free text		Name of venue or journal
Abstract	Free text		
Search type	Multiple choice[MS; AS]		How was the study found?
Inclusion criteria	Multiple choice[IC1;IC2;IC3]	RQ8	
Quality score	Number between 0 and 7	RQ2	
Does the study present approaches for AKS?	Yes xor no	RQ1, RQ8	From QA question 9
Description of approach	Free text	RQ1	From QA question 8
KM strategy	Multiple choice[personalisation; codification;hybrid]	RQ3	
Sharing mechanisms	Free text	RQ4	
Type of AK	Multiple choice[design; reasoning]	RQ5.1	
AK entities	List of entities	RQ5.2	
Domain	Free text	RQ2	Domain for which the approach has been tested
System type	Free text	RQ2	Type of system for which the approach has been tested(agile, software intensive, etc.)
Pprescriptive Descriptive	Prescriptive xor descriptive	RQ6	
What type of evidence is presented?	0 = no evidence; 1 = evidence from demonstration or toy examples; 2= evidence from expert opinions or observations; 3= evidence from academic studies (e.g., controlled lab experiments); 4= evidence from industrial studies (e.g., causal case studies); 5 = evidence from industrial practice.	RQ7	From QA question 1

Table 18: Data extraction form

3.4 Preliminary results

The data extraction form was applied to the finally selected papers resulted from the two selection rounds. A list of 13796 papers was initially retrieved from the two types of selection applied to the set of venues and databases listed in section 3.2.1, out of which 469 papers were discarded as duplicates within the same database or venue. The resulting list of 13327 papers was further submitted to the first round of selection performed by two different researchers and resulted in a list of 730 papers(5,48%) which were further analysed in the second round. The second round selection, also performed by two different researchers produced a list of finally selected papers available in B containing 135 studies(234 with duplicates).

Table 6 presents an overview of the selection process per venue for the two selection rounds:

Venue	Search Type	Studies	Studies (w/o duplicates)	Cohen's Kappa 1	Included 1st	% Included 1st	Included 2nd	% Included 2nd
SHARK	Manual	70	70	0,60	39	0,56	0	0,00
JSS	Manual	2307	2307	0,79	26	0,01	7	0,00
JSEKE	Manual	712	712	0,16	23	0,03	0	0,00
SEKE	Manual	1653	1653	0,52	17	0,01	6	0,00
ECSA	Manual	169	169	0,69	28	0,17	17	0,10
IST	Manual	1462	1462	0,29	29	0,02	5	0,00
WICSA	Manual	450	450	0,53	67	0,15	36	0,08
QoSA	Manual	170	170	0,53	25	0,15	0	0,00
IEEE Xplore	Automated	1164	876	0,82	101	0,12	43	0,04
ACM Digital Library	Automated	264	257	0,54	146	0,57	48	0,19
ScienceDirect	Automated	118	118	0,33	58	0,49	15	0,13
SpringerLink	Automated	4517	4343	0,65	57	0,01	11	0,00
Web Of Science	Automated	90	90	0,67	54	0,60	43	0,48
Wiley InterScience	Automated	650	650	0,22	60	0,09	3	0,00
Total		13796	13327		730		234	

Figure 6: Overview of the selection process per venue

Figures 7 to 9 show an overview of the most "interesting" venues for the topic of this study with regards to the number of selected articles for each step(search process, 1st round selection, 2nd round selection).

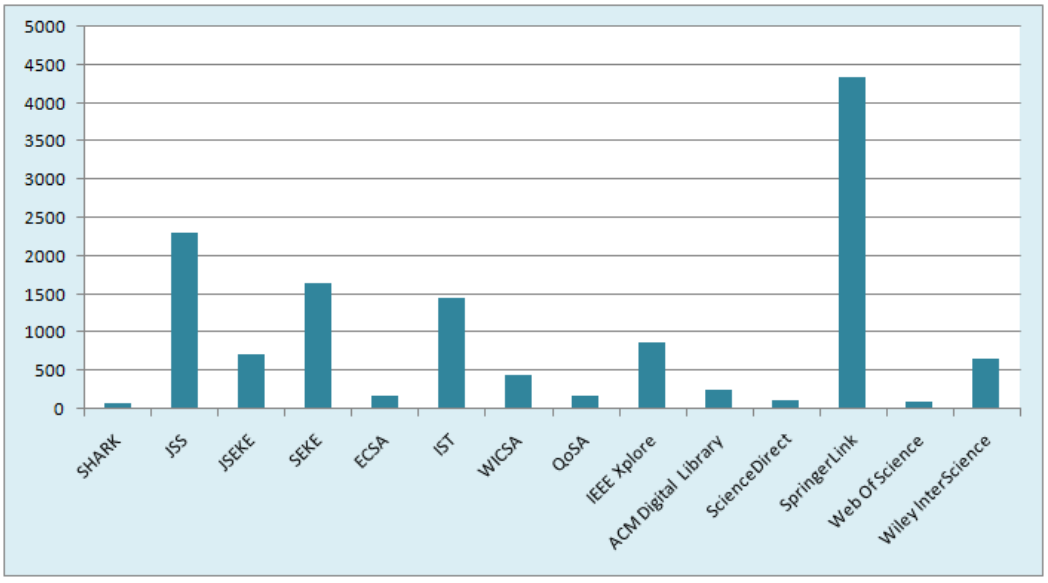

Figure 7: Most interesting venues search process

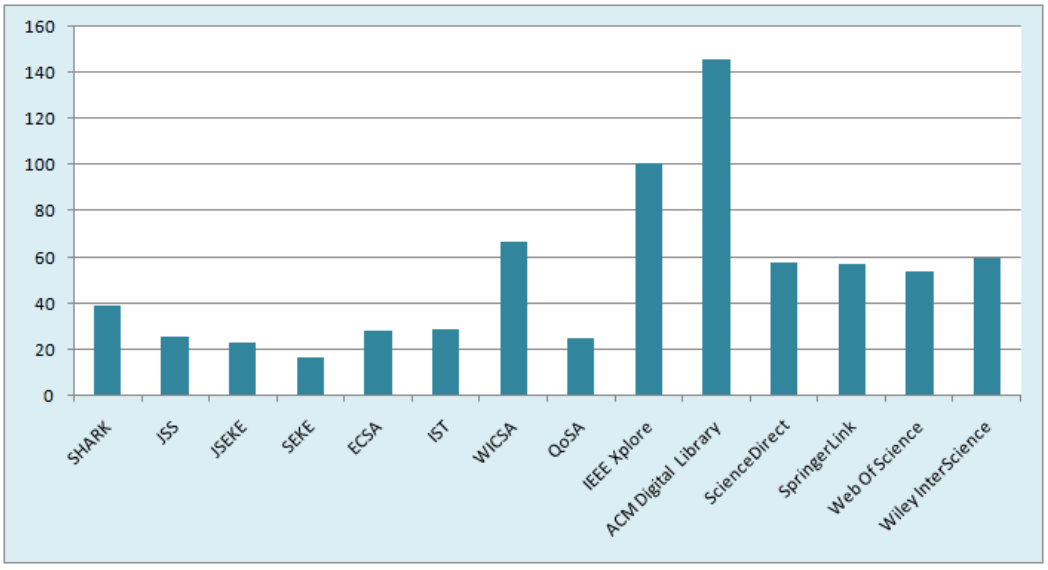

Figure 8: Most interesting venues 1st round selection

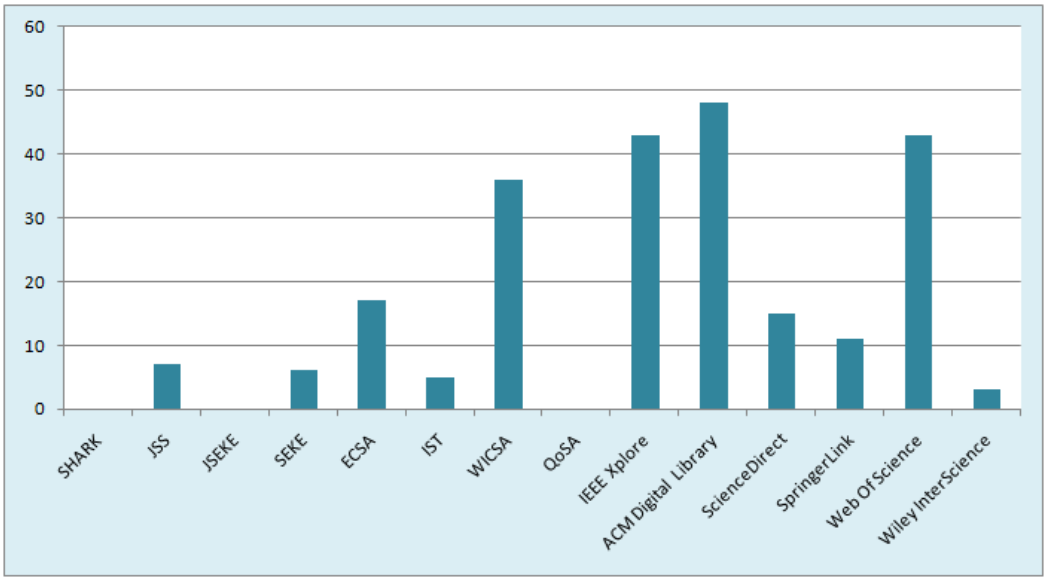

Figure 9: Most interesting venues 2nd round selection

As can be observed from figure 9, only four of the venues used for the SMS contributed to most of the finally selected studies: ACM Digital Library, IEEExplore, Web Of Science and WICSA. Also, from table 6, we can determine that ACM, ScienceDirect, ECSA and Web Of Science had the best precision among the selected venues and databases(more than 0,1% of selected studies made it to the final selection). By contrast, the databases and venues with the lowest precision were SpringerLink and JSS(see table 10 for a list of precision values for all sources with regards to the number of finally selected studies).

Venue/Database	Studies	Final selection	Precision
SHARK	70	0	0,000
JSS	2307	7	0,003
JSEKE	712	0	0,000
SEKE	1653	6	0,004
ECSA	169	17	0,101
IST	1462	5	0,003
WICSA	450	36	0,080
QoSA	170	0	0,000
IEEE Xplore	876	43	0,049
ACM Digital Library	257	48	0,187
ScienceDirect	118	15	0,127
SpringerLink	4343	11	0,003
Web Of Science	90	43	0,478
Wiley InterScience	650	3	0,005

Figure 10: Precision values for all sources calculated between initial and final number of selected studies

3.5 Data synthesis

RQ-1. What approaches for sharing AK have been proposed in the literature?
RQ-2. What is the intended project context(domain, system type, size) of the approach?

The list of finally selected studies describing the name and project context of each approach is made available in the appendix, section B.

RQ-3. Which knowledge management strategy does the approach support?(personalization, codification, hybrid)

Two main types of knowledge management strategies have been described by current research: codification and personalization with a third possibility being represented by a hybrid model which uses the two main types together([29], [25]). According to the type of strategy preferred by the stakeholders, the approaches used for AKM and AKS will differ as well, therefore it is important for the nature of this study to be able to assess how many approaches currently exist for each type and what they consist of.

After performing the data extraction process on the list of finally selected

studies which propose approaches for AKS, the following results were made available:

- 91 studies presented approaches related to the codification strategy
- 2 studies presented approaches related to personalization
- and only one approach supported a hybrid strategy

As can be observed from the results, the large majority of studies were oriented towards codification and only a very small number provided information related to a personalization strategy or to a hybrid approach.

Since current findings related to knowledge management strategies suggest that a hybrid strategy would bring more efficient results than any of the other two strategies used by themselves([25]), it would be interesting for future research to focus on AKS approaches which would support this type of strategy.

RQ-4. What are the mechanisms of the approach to support sharing of AK?

The studies proposing approaches selected during the final selection round used different sharing mechanisms for the support of AKS. A complete list of the approaches and the mechanisms they use can be found in B.

Table 20 shows a list of most commonly used types of mechanisms for a better visualization:

Type of sharing mechanism	**Number of approaches**
model-based frameworks and grids	54
ontologies and semantic wikis	11
knowledge repositories	7
architectural documents and adnotated textual information	6
patterns and reference architectures	4
architectural prototypes	2

Table 20: Sharing mechanisms used by the selected approaches

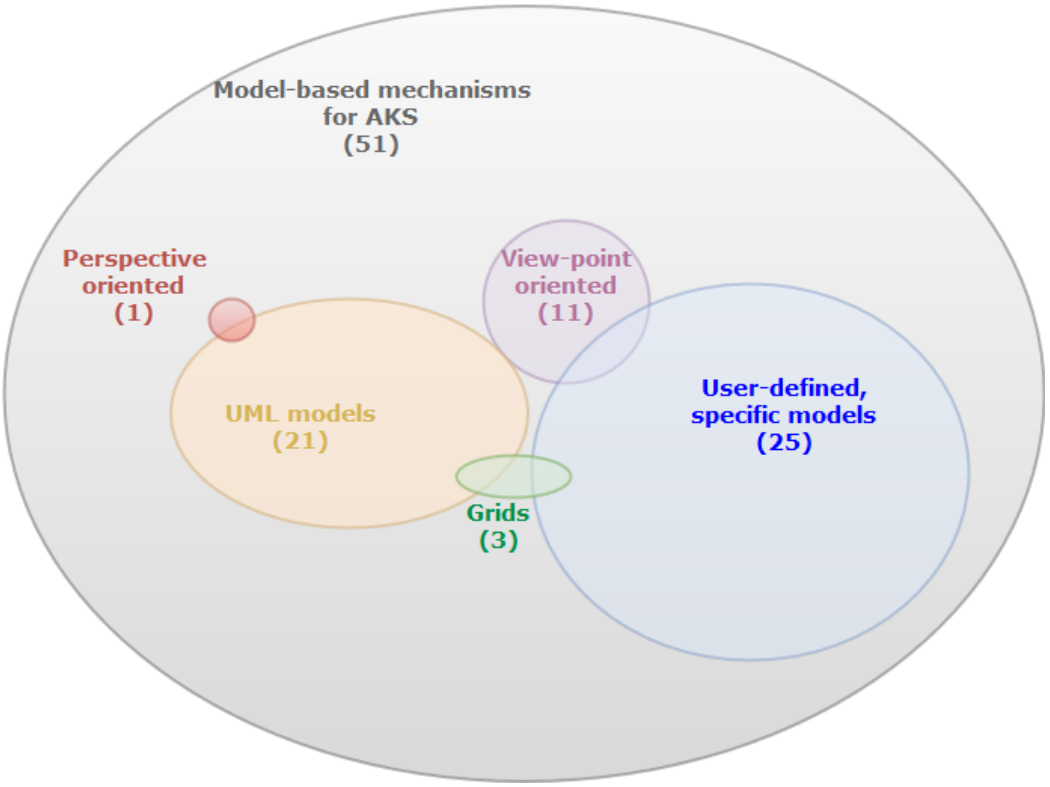

Figure 11: Model-based mechanisms for AKS

As can be observed from both table 20, the most popular types of AKS mechanisms are conceptual models and ontologies or semantic wikis. Model-based mechanisms for knowledge capturing and sharing can be further split into different categories as shown in figure 11.

RQ-5.1. What type of AK is the approach intended for?(design, rationale)

The finally selected studies which presented an approach for AKS were also analysed in terms of the type of knowledge for which they were intended(oriented towards design, rationale or both). The following Venn diagram displays the distribution of the finally selected studies corresponding to the first inclusion criteria between the two types of knowledge.

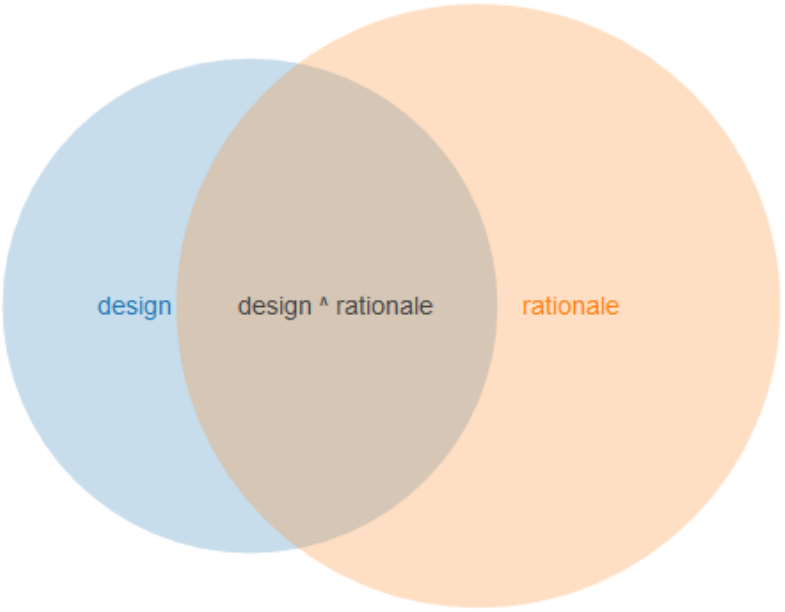

Figure 12: Venn diagram for approaches used in management of knowledge related to design or rationale

As can be observed from figure 12, both types of knowledge are well-represented by the existing approaches, although rationale-oriented knowledge seems to be of more interest.

Figure 13 helps gain more insight on the evolution of approaches with respect to the type of knowledge they were designed for by displaying the number of studies focussing on each knowledge type per year.

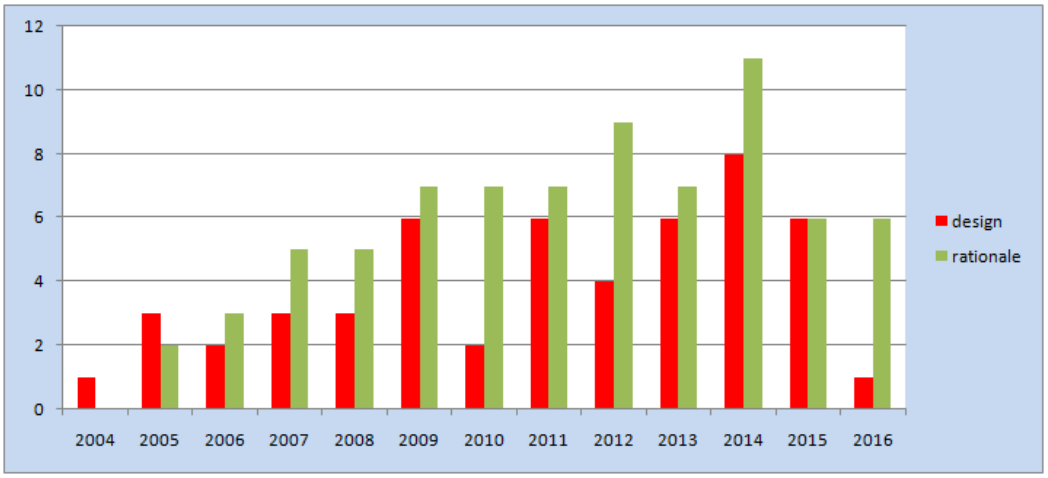

Figure 13: Number of approaches used in management of knowledge related to design or rationale per year

It can be observed that in early years of research(2004 and 2005), approaches were slightly more focussed on design knowledge, while starting from 2007, a continuously increasing number of approaches started focussing on rationale related knowledge. This shift can be attributed to the change in the general perception towards the job of a software architect which slowly moved from being solely design-oriented to a network of interconnected elements linked by rationale([11]).

RQ-5.2. Which architectural knowledge entities are captured by the approach?(ADDs, quality attributes, requirements, etc.)

Table 22 presents a list of architectural knowledge entities which were the focus of at least one approach presented in the finally selected articles.

AK entity name	Number of approaches
architectural design decisions	57
requirements	7
tactics	3
architectural design rules	2
architectural solutions	2
assumptions	1
forces	1
quality attributes	1
technical debt	1
not specified	26

Table 22: AK entities

As can be observed from table 22, the AK entity on which most of the selected approaches focussed on is represented by architectural design decisions. This finding confirms existing statements which suggest design decisions to be the most important elements in the field of software architecture and that they represent the core part of a system's architecture with all other elements depending on them([12], [10], [8]). While these statements are backed up by several studies in the field, most studies also stress on the fact that all software entities are highly correlated and present numerous interdependencies which also need to be shared and maintained([8], [10]).

Although most approaches follow the guidelines presented by findings which present ADDs as being the most important part of the architecture, not all take into consideration the reverse side of the matter related to their dependencies and relationships with other elements and fail to mention ways in which these aspects will be shared or maintained. Out of the 57 approaches which focused on sharing knowledge related to ADDs only 11 mentioned sharing and maintaining knowledge related to other entities and to their relationships.

RQ-6. How prescriptive or descriptive is the approach?

Using the data extraction form, the finally selected studies corresponding to the first inclusion criteria(which proposed an approach for AKS) were divided into two categories: prescriptive and descriptive(see section 3.1.1).

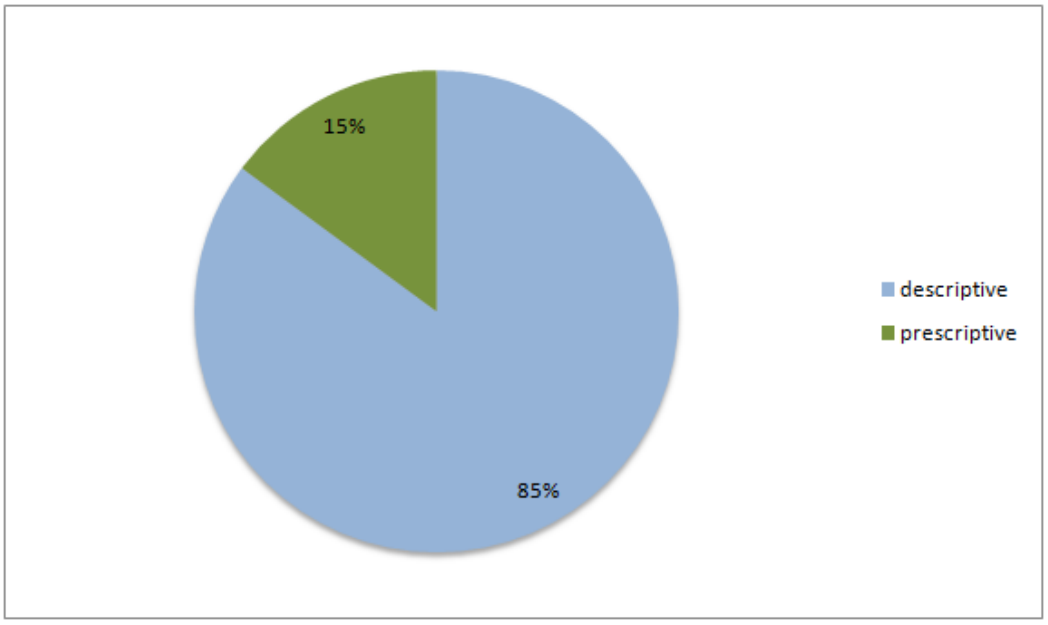

Figure 14: Distribution of finally selected approaches for AKS between prescriptive and descriptive

As can be observed from figure 14, this distribution of proposed approaches is highly uneven, with 85% descriptive and only 15% prescriptive approaches.

RQ-7. What level of evidence is provided for each approach?

In order to assess the level of evidence provided in the finally selected studies, six categories of evidence were applied in conformity with the list proposed in [66](see section 3.1.1 for the detailed list and explanations). The levels range from 0 to 5, with 0 representing no evidence at all and 5 representing the highest degree of evidence(evidence from industrial practice).

Figure 15 displays a pie chart representing the distribution of studies with regards to these levels of evidence.

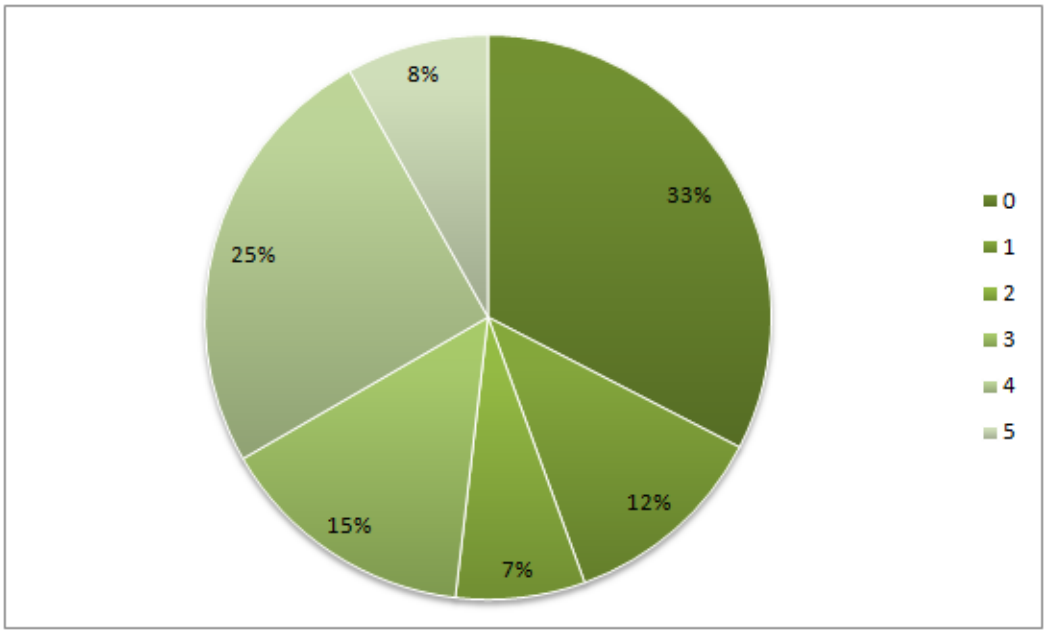

Figure 15: Distribution of finally selected studies with regards to the evidence level

As can be observed from figure 15, approximately one third of the finally selected studies presented no evidence for the approaches they proposed, while only 8% of the studies were attributed to the maximum evidence level.

Although some of the approaches were later validated in a different study and evidence was given to support their quality(28 studies), the number of studies which remain without any type of evidence which may support their findings is still relatively large. In addition, if we take into consideration the cumulation of studies with a very low level of evidence(0 or 1), we can observe that they amount to 45% of all selected studies, representing almost one half.

Considering these numbers, it would appear that research regarding approaches for AKS is still in an initial phase and that a large amount of new research should be focussed on evaluating and confirming or disproving currently existing findings without a high level of evidence.

RQ-8. How many studies focus on a specific approach and how many compare or present an overview of existing approaches?

A list of three inclusion criteria was used for the selection process of the SMS(see section 3.2.2). These criteria helped group the studies selected as

relevant for the purpose of the SMS into three main categories:

- studies that introduce new approaches for AKS
- comparison studies between two or more previously proposed approaches for AKS
- studies which assess the state of the art for the domain related to AKS(surveys, qualitative studies, etc.)

Figure 16 presents a pie chart distribution of the finally selected articles with respect to the above mentioned categories.

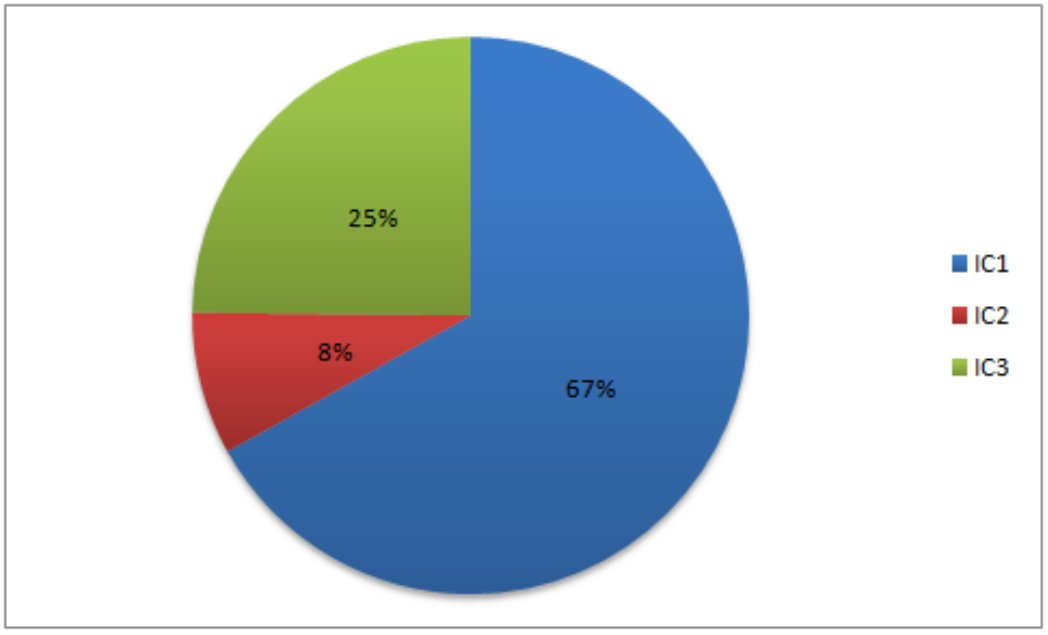

Figure 16: Distribution of finally selected studies with regards to the inclusion criteria

By analysing the amount of currently available research in all three groups, several interesting aspects can be observed. The first aspect relates to the highly uneven distribution of the studies: 67% of the studies correspond to the first inclusion criteria(IC1), while only 8% are attributed to the second inclusion criteria(IC2). The high amount of studies corresponding to the first inclusion criteria(almost two times more than the other two criteria combined) suggests that the increased interest in the domain of AKS has pushed researchers into building many solutions for different AKS problems. However, the lack of studies which compare and evaluate these solutions

against each other suggests that the field of AKS has not yet reached a mature phase and is in need of research that aims to confirm and validate the quality of present findings.

Figure 17 analyses the distribution of the selected studies grouped into each category over time for a better visualization of the evolution of research in the field of AKS.

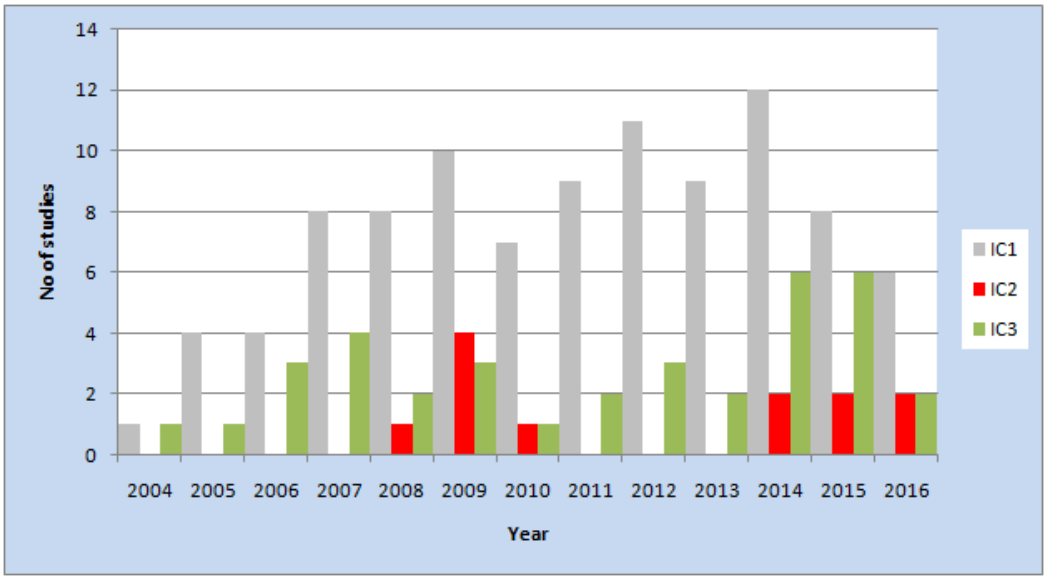

Figure 17: Distribution of finally selected studies with regards to the inclusion criteria

As can be observed, studies corresponding to the first inclusion criteria have been published in slowly increasing numbers from 2004 until 2014, at which point they started to decrease slowly. Studies related to the state of the art or industrial practice(IC3) have also steadily increased in numbers and have reached their peak in the last two years which shows an increasing interest in understanding the state of research in this field. Lastly, studies corresponding to the second inclusion criteria can be seen forming two distinct patches in the graph: one between 2008 and 2012 and one in the past three years. Although the popularity of such studies appears to have dropped for a period of three years in 2010, it seems to have recently regained interest and remains at a steady value.

The trend of research seems therefore to tend towards a maturity of the field due to a slight but steady decrease in the number of proposed approaches and an increase in the number of studies which compare them or analyse the

state of the art of AKS. However, the gap between the two types of studies is still significant and future research is still needed for a better understanding of the field.

4 Threats to validity

Secondary studies are often subject to validity threats due to the nature of the research. The following subsections analyse and describe the measures taken in order to maximise four types of validity important for the SMS described in this paper: construct, internal, external and conclusion validity). The four validity types are analysed conforming to the definitions proposed in [23], [67].

4.1 Construct validity

Construct validity refers to the degree to which the operationalization of the measures in a study actually represents the constructs in the real world([23]). In order to ensure or maximise this type of validity, several measures were taken:

- the search terms used in the construction of the search strings were derived systematically using the PICOC criteria(population, intervention, comparison, outcome and context - section 3.2.3)
- in the construction of the search strings, different terms which could represent the same aspect related to AKS were taken into consideration and added to the string using the boolean operator "OR"(e.g. "approach" OR "strategy" OR "mechanism" OR "tool" OR "process")
- the search process was conducted in two main phases: manual and automated. The resulting sets of studies obtained through the manual search process represented the QGS against which the results obtained from the automated search were verified(see section 3.2.3)

4.2 Internal validity

Internal validity refers to the extent to which the treatment or independent variable(s) were actually responsible for the effects seen to the dependent variable([23], [67]).

The type of statistics used for conducting the SMS described in this paper was basic, descriptive, with a minimal risk of damaging the internal validity of the results produced.

4.3 External validity

External validity refers to the degree to which the findings of the study can be generalized to other participant populations or settings([23]).

The threats to external validity for the SMS described in this paper have been minimized with the help of a thorough description of the protocol used(with details related to the venues and databases used for the search process - section 3.2.3, a set of inclusion and exclusion criteria - section 3.2.2, a description of the data extraction form - section 3.3) which can be easily reconstructed and extended by other researchers. Extensions which can be added to the protocol may include a larger time frame, a different set of venues and databases used in the search process, etc.

However, the SMS conducted for the purpose of this study revolves around the topic of approaches related to AKS. All papers taken into consideration for this study were selected or discarded based on their relevance for this domain. With this in mind, the results obtained from the study and presented in this paper are strictly dependant to the domain of AKS and cannot be extended to a larger scope.

4.4 Conclusion validity

Conclusion validity refers to whether the conclusions reached in a study are correct and is directly related to the application of statistical tests to the data in the case of controlled experiments([23], [67]).

The conclusions presented in this paper with regards to the SMS have been reached using a set of 135 studies, selected through a broad search range(both time range and set of sources used for the search process were relatively large) and can be expected to provide general results. Also, two different researchers were involved in the selection process of each paper for each round and maintained an inclusive rather than exclusive approach which minimized the overlooking of important papers.

5 Conclusions

This thesis revolves around the conducting of a Systematic Mapping Study for analysing approaches in Architectural Knowledge Sharing. This type of secondary study helps understand the state of the art and obtain a general overview of existing research in a particular domain.

Through the search process of the SMS, 14 venues and databases were searched, resulting in an initial list of 13796 studies. After two rounds of selection, 135 studies were finally selected based and analysed during the data extraction phase.

After performing the data synthesis on the finally selected studies, several conclusions emerged:

- A constantly increasing number of studies related to AKS have been performed starting with the year 2004 until the present day
- There has been an increase in studies related to the state of the art and industrial practices and a slight decrease in studies presenting new approaches over the last three years which would suggest that the research in the field is tending towards a more mature stage aiming to analyse and validate current approaches
- The level of evidence presented for existing approaches is still very low, with more than half of the approaches having a level of evidence less than or equal to 2 and one third of approaches presenting no level of evidence
- Studies tend to focus on descriptive approaches rather than prescriptive ones
- Approaches are highly oriented towards the management of architectural design decisions and tend to overlook other AK entities and the relationships between them
- Studies seem to be slightly shifting their orientation from AKM and AKS of design knowledge to rationale
- Approaches are strongly oriented towards a codification KM approach and neglect suggested benefits of hybrid approaches

With these results in mind, the suggestions for future research would consist of:

- performing more studies aiming to validate and analyse existing approaches with a higher level of evidence
- including a larger diversity of AK entities in the approaches or focusing on a singular entity but keeping in mind the relationships between that entity and other elements
- focussing more on hybrid approaches rather than approaches which solely reside on codification

Appendices

A Search strings

A.1 IEEEXplore:

```
("Abstract":"share" OR "capture" OR "communicate" OR "
 documentation" OR "document" OR "exchange" OR "transfer" OR "
 understand" AND "Abstract":"approach" OR "strategy" OR "
 mechanism" OR "tool" OR "process" AND "software architecture"
 ) AND "architectural knowledge" OR "architecture knowledge"
 OR "architecture design" OR "architectural design" OR "
 technical debt" OR "architecture decision" OR "architectural
 decision" OR "design rationale" OR "design decision" OR "
 architecture assumption" OR "architectural assumption" OR "
 architecture requirement" OR "quality requirement" OR "non-
 functional requirement" OR "architecturally relevant
 requirement" OR "architectural requirement" OR "architectural
  concern" OR "architecture requirement"

```

A.2 ACM Digital Library:

```
"(recordAbstract:(""architectural knowledge"" , ""architecture
 knowledge"" , ""architecture design"" , ""architectural
 design"" , ""architectural design decision"" , ""architecture
  design decision"" , ""architecture decision"" , ""
 architectural decision"" , ""design rationale"" , ""design
 decision"" , ""architecture assumption"" , ""architectural
 assumption"" , ""technical debt"" , ""architecture
 requirement"" , ""architectural requirement"" ""architectural
  concern"", ""architecture concern"", ""quality requirement""
 , ""non-functional requirement"", ""architecturally relevant
 requirement"")
OR acmdlTitle:(""architectural knowledge"" , ""architecture
 knowledge"" ,
""architecture design"" , ""architectural design"" , ""
 architectural design decision"" , ""architecture design
 decision"" , ""architecture decision"" ,
""architectural decision"" , ""design rationale"" , ""design
 decision"" , ""architecture assumption"" , ""architectural
 assumption"" , ""technical debt"" ,
```

```
5  ""architecture requirement"" , ""architectural concern"", ""
     architecture concern"", ""quality requirements"", ""non-
     functional requirement"", ""architecturally relevant
     requirement"", ""architectural requirement"")) AND (
     recordAbstract:(""share"" , ""capture"" , ""communicate"" , "
     "document"" , ""documentation"" , ""exchange"" , ""transfer""
     , ""understand""))
6  AND (recordAbstract:(""approach"" , ""strategy"" , ""mechanism"
     " , ""tool"" , ""process""))
7  AND (acmdlTitle:(""architectural knowledge"" , ""architecture
     knowledge"" , ""architecture design"" , ""architectural
     design"" ,
8  ""architectural design decision"" , ""architecture design
     decision"" , ""architecture decision"" , ""architectural
     decision"" , ""design rationale"" , ""design decision"" , ""
     architecture assumption"" , ""architectural assumption"" , ""
     technical debt"" , ""architecture requirement"" , ""
     architectural concern"", ""architecture concern"", ""quality
     requirement"", ""non-functional requirement"", ""
     architecturally relevant requirement"", ""architectural
     requirement"") OR acmdlTitle:(""share"" , ""capture"" , ""
     communicate"" , ""document"" , ""documentation"" , ""exchange
     "" , ""transfer"", ""understand"")
9  OR acmdlTitle:(""approach"" , ""strategy"" , ""mechanism"" , ""
     tool"" , ""process""))
10 AND (""software architecture"")"
11
```

A.3 ScienceDirect:

```
1  "pub-date > 2003 and pub-date < 2017 and tak(("architectural
     knowledge" OR "architecture knowledge" OR "architecture
     design" OR "architectural design" OR "architectural design
     decision" OR "technical debt" OR "architecture design
     decision" OR "architecture decision" OR "architectural
     decision" OR "design rationale" OR "design decision" OR "
     architecture assumption" OR "architectural assumption" OR "
     architecture concern" OR "architectural concern" OR "quality
     requirements" OR "non-functional requirements" OR "
     architecturally relevant requirements" OR "architecture
     requirements" OR "architectural requirements") AND ("share"
     OR "capture" OR "communicate" OR "documentation" OR "document
     " OR "exchange" OR "transfer" OR "understand") AND ("approach
     " OR "strategy" OR "mechanism" OR "tool" OR "process")) AND (
     "software architecture") "
2
```

A.4 SpringerLink:

```
"((""architectural knowledge"" OR ""architecture knowledge"" OR
  ""architecture design"" OR ""architectural design"" OR ""
  architectural design decision"" OR ""architecture design
  decision"" OR ""architecture decision"" OR ""architectural
  decision"" OR ""design rationale"" OR ""design decision"" OR
  ""architecture assumption"" OR ""architectural assumption""
  OR ""technical debt"" OR ""quality requirement"" OR ""non-
  functional requirement"" OR ""architecturally relevant
  requirement"" OR ""architecture requirement"" OR ""
  architectural requirement"" OR ""architectural concern"" OR "
  "architecture concern"") AND (""share"" OR ""capture"" OR ""
  communicate"" OR ""document"" OR ""documentation"" OR ""
  exchange"" OR ""transfer"" OR ""understand"") AND (""approach
  "" OR ""strategy"" OR ""mechanism"" OR ""tool"" OR ""process"
  ") AND (""software architecture""))
within English   2004 - 2016"

```

A.5 Web Of Science:

```
TOPIC: (("architectural knowledge" OR "architecture knowledge"
  OR "architecture design" OR "architectural design" OR "
  architectural design decision" OR "architecture design
  decision" OR "architecture decision" OR "architectural
  decision" OR "design rationale" OR "design decision" OR "
  architecture assumption" OR "architectural assumption" OR "
  technical debt" OR "quality requirement" OR "non-functional
  requirement" OR "architecturally relevant requirement" OR "
  architecture requirement" OR "architectural requirement" OR "
  architectural concern" OR "architecture concern") AND ("share
  " OR "capture" OR "communicate" OR "document" OR "
  documentation" OR "exchange" OR "transfer" OR "understand")
  AND ("approach" OR "strategy" OR "mechanism" OR "tool" OR "
  process") AND ("software architecture")) AND YEAR PUBLISHED:
  (2004-2016)

```

A.6 Wiley InterScience:

```
("architectural knowledge" OR "architecture knowledge" OR "
  architecture design" OR "architectural design" OR "
  architectural design decision" OR "architecture design
  decision" OR "architecture decision" OR "architectural
  decision" OR "design rationale" OR "design decision" OR "
  architecture assumption" OR "architectural assumption" OR "
  technical debt" OR "quality requirement" OR "non-functional
  requirement" OR "architecturally relevant requirement" OR "
  architecture requirement" OR "architectural requirement" OR "
  architectural concern" OR "architecture concern") in All
  Fields AND ("share" OR "capture" OR "communicate" OR "
  document" OR "documentation" OR "exchange" OR "transfer" OR "
  understand") in All Fields AND ("approach" OR "strategy" OR "
  mechanism" OR "tool" OR "process") in All Fields AND ("
  software architecture") in All Fields between years 2004 and
  2016
```

B Studies in final selection

Title	Authors	Year	Rationale	Approach name	Domain	System type	Related Approaches
capturing and using software architecture knowledge for architecture-based software development	ali babar, muhammad and gorton, ian and jeffery, ross	2005	ic1	PAKME			built on top of Hipergate (groupware platform)

pakme: a tool for capturing and using architecture design knowledge	babar, muhammad ali and wang, xiaowen and gorton, ian	2005	ic1	Process-based Architecture Knowledge Management Environment (PAKME)			
capturing and using quality attributes knowledge in software architecture evaluation process	babar, muhammad ali and capilla, rafael	2008	ic1	PAKME	security	large scale	
architecture description leveraging model driven engineering and semantic wikis	baroni, alessandro and muccini, henry and malavolta, ivano and woods, eoin	2014	ic1				
crafting a global teaming model for architectural knowledge	beecham, sarah and noll, john and richardson, ita and ali, nour	2010	ic1	Global Teaming Model (GTM)			

ontology-driven visualization of architectural design decisions	de boer, remco c. and lago, patricia and telea, alexandru and van vliet, hans	2009	ic1				
experiences with semantic wikis for architectural knowledge management	boer, remco c. de and vliet, hans van	2011	ic1		e-government	distributed, large scale	
embedded design rationale in software architecture	capilla, rafael	2009	ic2				
intelligent analysis of software architecture rationale for collaborative software design	chanda, nagaprashanth and liu, xiaoqing	2015	ic1	ISARCS (intelligent software architecture rationale capture system)			

scenario-based architectural design decisions documentation and evolution	che, meiru and perry, dewayne e.	2011	ic1	TVM (triple-view model)	monitoring	large-scale, complex	
towards software assets origin selection supported by a knowledge repository	cicchetti, antonio and borg, markus and sentilles, severine and wnuk, krzysztof and carlson, jan and papatheocharous, efi	2016	ic1	ORION			
the usefulness of architectural knowledge management practices in gsd	clerc, viktor and lago, patricia and vliet, hans van	2009	ic3				

systematic architectural decision management, a process-based approach	dragomir, ana and lichter, horst and budau, tiberiu	2014	ic1				
topdocs: using software architecture knowledge base for generating topical documents	eloranta, v. p. and hylli, o. and vepsalainen, t. and koskimies, k.	2012	ic1	TopDocs	education	small size system	the approach was tested on top of Polarion (application lifecycle management platform)
prerequisites for successful architectural knowledge sharing	farenhorst, rik and lago, patricia and vliet, hans van	2007	ic3				
a just-in-time architectural knowledge sharing portal	farenhorst, rik and izaks, ronald and lago, patricia and vliet, hans van	2008	ic2, ic3				

dpmtool: a tool for decisions management in distributed software projects	garrido, pedro jose and vizcaino, aurora and andrada, juan and monasor, miguel j. and piattini, mario	2012	ic1	DPMTool (distributed project management tool)			
ontology-based software architecture documentation	graaf, k. a. de and tang, a. and liang, p. and vliet, h. van	2012	ic1	ArchiMind	management	large scale	implementatio of previously discussed Lightweight Software Ontology, uses OntoWiki tool
open source based tools for sharing and reuse of software architectural knowledge	henttonen, katja and matinlassi, mari	2009	ic2				

supporting the collaborative development of requirements and architecture documentation	hesse, tom-michael and paech, barbara	2013	ic1				
decdoc: a tool for documenting design decisions collaboratively and incrementally	hesse, tom-michael and kuehlwein, arthur and roehm, tobias	2016	ic1	DecDoc	software	large scale	
tool support for architectural decisions	jansen, anton and der ven, jan and avgeriou, paris and hammer, dieter	2007	ic1	Archium	communication	small-scale	
integrating decision management with uml modeling concepts and tools	konemann, patrick	2009	ic1	IBM Rational Software Modeler and Architectural Decision Knowledge Wiki			

a study of architectural decision practices	latoza, thomas d. and shabani, evelina and van der hoek, andre	2013	ic3				
customizing the capture of software architectural design decisions	lee, larix and kruchten, philippe	2008	ic1		entertainment	small scale	
capturing and maintaining architectural knowledge using context information	miesbauer, cornelia and weinreich, rainer	2012	ic1		extension for LISA approach		

architecturally significant requirements, reference architecture, and metamodel for knowledge management in information technology services	miksovic, christoph and zimmermann, olaf	2011	ic1				
weaving a network of architectural knowledge	navarro, elena and cuesta, carlos e. and perry, dewayne e.	2009	ic1				extension for ATRIUM
toward a collaborative method for knowledge management of software architectural decisions based on trust	nejad, marzie samghani and moaven, shahrouz and habibi, jafar and alidousti, razie	2015	ic1				

analysis of design meetings for understanding software architecture decisions	pedraza-garcia, gilberto and astudillo, hernan and correal, dario	2014	ic1	Design Verbal Interventions Analysis (DVIA)	aerospatial	large scale, complex	
how to avoid taking three lefts when you can go right: making the architectural perspective count	savio, deepti and surya-narayana, girish	2012	ic3				
architectural design decision: existing models and tools	shahin, mojtaba and liang, peng and khayyambashi, mohammad reza	2009	ic2				
a spatial hypertext wiki for architectural knowledge management	solis, carlos and ali, nour and babar, muhammad ali	2009	ic1	ShyWiki			

distributed requirements elicitation using a spatial hypertext wiki	solis, carlos and ali, nour	2010	ic1	Spatial Hypertext Wiki			
capturing architecture documentation navigation trails for content chunking and sharing	su, moon ting and hosking, john and grundy, john	2011	ic1	KaitoroCap			
knowledge reuse of software architecture design decisions and rationale within the enterprise	sundaravadivel, subhashree and vaidyanathan, aparajithan and ramaswamy, srini	2014	ic3				
architecting in networked organizations	tamburri, damian a. and lago, patricia and dorn, christoph and hilliard, rich	2014	ic1			distributed team	

the architect's role in community shepherding	tamburri, damian a. and kazman, rick and fahimi, hamed	2016	ic3				
software architecture documentation: the road ahead	tang, antony and liang, peng and vliet, hans van	2011	ic1				
a survey on knowledge management in software engineering	vasanthapriyan, shanmuganathan and tian, jing and xiang, jianwen	2015	ic3				
the architect's role in practice: from decision maker to knowledge manager?	weinreich, rainer and groher, iris	2016	ic3				

combining pattern languages and reusable architectural decision models into a comprehensive and comprehensible design method	zimmermann, olaf and zdun, uwe and gschwind, thomas and leymann, frank	2008	ic1				
architectural decision guidance across projects - problem space modeling, decision backlog management and cloud computing knowledge	zimmermann, olaf and wegmann, lukas and koziolek, heiko and goldschmidt, thomas	2015	ic1			ADMentor	add-in for Sparx Enterprise Architect

a stakeholder-centric optimization strategy for architectural documentation	andres diaz-pace, j. and nicoletti, matias and schiaffino, silvia and villavicencio, christian and emiliano sanchez, luis	2013	ic1	V&B(views and beyond)			
the value of architecturally significant information extracted from patterns for architecture evaluation: a controlled experiment	babar, muhammad ali and kitchenham, barbara and maheshwari, piyush	2006	ic1	ASIP			
architectural prototyping: an approach for grounding architectural design and learning	.e. bardram , h.b. christensen , k.m. hansen	2004	ic1				

extending software architecting processes with decision-making activities	capilla, rafael and nava, francisco	2008	ic3				
on the role of architectural design decisions in software product line engineering	capilla, rafael and babar, muhammad ali	2008	ic1				implements product-line decision support in ADDSS, PAKME
viability for codifying and documenting architectural design decisions with tool support	capilla, rafael and duenas, juan c. and nava, francisco	2010	ic1	ADDSS (Architecture Design Decision Support System)			
a model to represent architectural design rationale	carignano, m. c. and gonnet, s. and leone, h.	2009	ic1		software	small size system	

an approach to documenting and evolving architectural design decisions	che, meiru	2013	ic1	Triple View Model			
architectural prototyping in industrial practice	christensen, henrik baerbak and hansen, klaus marius	2008	ic1	architectural prototypes	finance, software, transportation	large scale, distributed teams	
the architect's mindset	clerc, viktor and lago, patricia and van vliet, hans	2007	ic3				
architecture design for the large-scale software-intensive systems: a decision-oriented approach and the experience	cui, xiaofeng and sun, yanchun and xiao, sai and mei, hong	2009	ic1	ABC/DD	communication	large scale, software intensive	

software architecture decision-making practices and challenges: an industrial case study	dasanayake, sandun and markkula, jouni and aaramaa, sanja and oivo, markku	2015	ic3				
an exploratory study on ontology engineering for software architecture documentation	de graaf, k. a. and liang, p. and tang, a. and van hage, w. r. and van vliet, h.	2014	ic1		printing industry	large scale	
stream-add - supporting the documentation of architectural design decisions in an architecture derivation process	dermeval, diego and pimentel, joao and silva, carla and castro, jaelson and santos, emanuel and guedes, gabriela and lucena, marcia and finkelstein, anthony	2012	ic1	STREAM-ADD	transportations	small size system	an extension for STREAM, which does not support documentation of ADDs and their rationale

architectural knowledge in product line engineering: an industrial case stu	dhungana, deepak and rabiser, rick and grunbacher, paul and prahofer, herbert and federspiel, christian and lehner, klaus	2006	ic3				
producing just enough documentation: the next sad version problem	diaz-pace, j. andres and nicoletti, matias and schiaffino, silvia and vidal, santiago	2014	ic1	NSVP(next software architecture document version problem)			extension of regular SAD (software architecture document)
knowledge-based approaches in software documentation: a systematic literature review	ding, wei and liang, peng and tang, antony and van vliet, hans	2014	ic2				

the value of design rationale information	falessi, davide and briand, lionel c. and cantone, giovanni and capilla, rafael and kruchten, philippe	2013	ic1				
what's in constructing a domain model for sharing architectural knowledge?	farenhorst, rik and boer, remco c. de and deckers, robert and lago, patricia and vliet, hans van	2006	ic1				
effective tool support for architectural knowledge sharing	farenhorst, rik and lago, patricia and vliet, hans van	2007	ic3, ic1				

knowledge transfer, translation and transformation in the work of information technology architects	figueiredo, mayara costa and de souza, cleidson r. b. and pereira, marcelo zilio and prikladnicki, rafael and audy, jorge luis nicolas	2014	ic3				
architech: tool support for nfr-guided architectural decision-making	franch, xavier	2012	ic1	ArchiTech			
empirical study of architectural knowledge management practices	galster, matthias and babar, muhammad ali	2014	ic3				

design guidelines for software processes knowledge repository development	garcia, javier and amescua, antonio and sanchez, maria-isabel and bermon, leonardo	2011	ic1	Process Asset Libraries (PAL)	agile, large scale		
combining architectural design decisions and legacy system evolution	gerdes, sebastian and lehnert, steffen and riebisch, matthias	2014	ic1				
how organisation of architecture documentation affects architectural knowledge retrieval	de graaf, k.a. and liang, p. and tang, a. and van vliet, h.	2016	ic1, ic2	ArchiMind			

variability support in architecture knowledge management approaches: a systematic literature review	groher, iris and weinreich, rainer	2015	ic2				
ontobrowse: a semantic wiki for sharing knowledge about software architectures	happel, hans-jorg and seedorf, stefan	2007	ic1	Ontobrowse			
exploring software architecture context	harper, k. eric and zheng, jiang	2015	ic1		transportation	small size system	
forces on architecture decisions - a viewpoint	heesch, u. van and avgeriou, p. and hilliard, r.	2012	ic1			small scale	an extension for a framework introduced in previous work(Heesch2 - A documentation framework for architecture decisions)

the lonesome architect	johan f. hoorn and rik farenhorst and patricia lago and hans van vliet	2011	ic3				
software architecture as a set of architectural design decisions	jansen, a. and bosch, j.	2005	ic1	Archium	education	small size system	
documenting after the fact: recovering architectural design decisions	jansen, anton and bosch, jan and avgeriou, paris	2008	ic1	ADDRA(Architectural Design Decision Recovery Approach)	software	small scale	
enriching software architecture documentation	jansen, anton and avgeriou, paris and van der ven, jan salvador	2009	ic1		large scale, complex		
reusing security solutions: a repository for architectural decision support	jasser, stefanie and riebisch, matthias	2016	ic1				

the essential components of software architecture design and analysis	rick kazman and len bass and mark klein	2006	ic3, ic1	APTIA (Analytic Principles and Tools for the Improvement of Architectures)			uses combination of ATAM, QAW, ADD and CBAM
linking design decisions to design models in model-based software development	konemann, patrick and zimmermann, olaf	2010	ic1				
documentation of software architecture from a knowledge management perspective – design representation	kruchten, philippe	2009	ic3				
architecture-centric modeling of design decisions for validation and traceability	kuster, martin	2013	ic1			small scale system	

capturing software architectural design decisions	lee, larix and kruchten, philippe	2007	ic1				
visualizing software architectural design decisions	lee, larix and kruchten, philippe	2008	ic1				
sharing architecture knowledge through models: quality and cost	liang, peng and jansen, anton and avgeriou, paris	2009	ic2				
advanced quality prediction model for software architectural knowledge sharing	peng liang and anton jansen and paris avgeriou and antony tang and lai xu	2011	ic1	AMQPM			is an advanced version of MQPM (mapping quality prediction model), which brings enhancements to SMQPM (simple MQPM) and RMQPM (random MQPM)

software architecture rationale capture through intelligent argumentation	liu, xiaoqing frank and chanda, nagaprashanth and barnes, eric christopher	2014	ic1				
visualization and comparison of architecture rationale with semantic web technologies	claudia lopez and pablo inostroza and luiz marcio cysneiros and hernan astudillo	2009	ic1		culture		uses a combination of Softgoal Interdependency Graphs and ontologies
bridging the gap between software architecture rationale formalisms and actual architecture documents: an ontology-driven approach	claudia lopez and victor codocedo and hernan astudillo and luiz marcio cysneiros	2012	ic1	TREx			

architectural decision making for service-based platform integration: a qualitative multi-method study	lytra, ioanna and sobernig, stefan and zdun, uwe	2012	ic1, ic3			large scale	
supporting consistency between architectural design decisions and component models through reusable architectural knowledge transformations	lytra, ioanna and tran, huy and zdun, uwe	2013	ic1		industry automation	large scale	based on ADvISE a tool for assisting architectural decision making for reusable ADDs, and VbMF2 a tool for describing architectural view models and performing model-driven code generation

reusable architectural decision models for quality-driven decision support: a case study from a smart cities software ecosystem	lytra, ioanna and engelbrecht, gerhard and schall, daniel and zdun, uwe	2015	ic1	COCOADVISE	software	large scale, complex	extension of QOC(question options and criteria) approach
industrial implementation of a documentation framework for architectural decisions	manteuffel, christian and tofan, dan and koziolek, heiko and goldschmidt, thomas and avgeriou, paris	2014	ic1		large scale	add-in for Enterprise Architect	
linking model-driven development and software architecture: a case study	mattsson, anders and lundell, bjoern and lings, brian and fitzgerald, brian	2009	ic1	model-driven development	communication	large size, distributed teams	

an approach for modeling architectural design rules in uml and its application to embedded software	mattsson, anders and fitzgerald, brian and lundell, bjorn and lings, brian	2012	ic1				
old theories, new technologies: understanding knowledge sharing and learning in brazilian software development companies	menolli, andre and cunha, maria alexandra and reinehr, sheila and malucelli, andreia	2015	ic2, ic3				
on the social dimensions of architectural decisions	muccini, henry and tamburri, damian a. and rekha, v. smrithi	2015	ic3				

real world influences on software architecture - interviews with industrial system experts	mustapic, g. and wall, a. and norstrom, c. and crnkovic, i. and sandstrom, k. and froberg, j. and andersson, j.	2004	ic3				
processes for creating and exploiting architectural design decisions with tool support	nava, francisco and capilla, rafael and duenas, juan c.	2007	ic1	ADDSS 2.0			new version of ADDSS1.0
team situational awareness and architectural decision making with the software architecture warehouse	nowak, marcin and pautasso, cesare	2013	ic1				

dvia: understanding how software architects make decisions in design meetings	pedraza-garcia, gilberto and astudillo, hernan and correal, dario	2015	ic1	Design Verbal Interventions Analysis (DVIA)			follow-up on Pedraza-Garcia2014
requirements communication and balancing in large-scale software-intensive product development	pernstaal, j. and gorschek, t. and feldt, r. and floren, d.	2015	ic1	BRASS framework	transportation	large scale, software intensive	
successful architectural knowledge sharing: beware of emotions	poort, eltjo r. and pramono, agung and perdeck, michiel and clerc, viktor and vliet, hans	2009	ic3				
the boomeranged software architect	premraj, r. and nauta, g. and tang, a. and vliet, h. v	2011	ic3				

after the scrum: twenty years of working without documentation	ratanotayanon, sukanya and kotak, jigar and sim, susan elliott	2006	ic3, ic1			agile	
knowledge representation of the software architecture design process based on situation calculus	roldan, maria luciana and gonnet, silvio and leone, horacio	2013	ic1				based on a previously designed tool for modelling design evolution using situation calculus
operation-based approach for documenting software architecture knowledge	roldan, maria luciana and gonnet, silvio and leone, horacio	2016	ic1				implemented using TracED tool
assessing the architectonics of large, software-intensive systems using a knowledge-based approach	rosso, c. del and maccari, a.	2007	ic1, ic3		communications	large size, software intensive	

setting up architect team	sarang, p.	2007	ic1		IT		
communicating architectural knowledge: requirements for software architecture knowledge management tools	schwittek, widura and eicker, stefan	2010	ic1, ic3				
model-centric approach to software design and stakeholder-specific architecture views in scope of a financial institution	senti, patrick	2009	ic1	TSDOC	finance	large-scale systems, globally distributed teams	
rationale visualization of software architectural design decision using compendium	shahin, mojtaba and liang, peng and khayyambashi, mohammad reza	2010	ic1	Compendium			uses SOAD model

suitability of software architecture decision making methods for group decisions	v, smrithi rekha and muccini, henry	2014	ic2				
modeling the interactions between decisions within software architecture knowledge	soliman, mohamed and riebisch, matthias	2014	ic1				
enriching architecture knowledge with technology design decisions	soliman, mohamed and riebisch, matthias and zdun, uwe	2015	ic3				
a study of architectural information foraging in software architecture documents	su, moon ting and tempero, ewan and hosking, john and grundy, john	2012	ic3				

knowledge reuse of software architecture design decisions and rationale within the enterprise	sundaravadivelu, subhashree and vaidyanathan, aparajithan and ramaswamy, srini	2014	ic1, ic3				
when software architecture leads to social debt	tamburri, d. a. and nitto, e. d.	2015	ic1	Dahlia			
a survey of the use and documentation of architecture design rationale	tang, a. and babar, m. a. and gorton, i. and han, jun	2005	ic3				
using bayesian belief networks for change impact analysis in architecture design	antony tang and ann nicholson and yan jin and jun han	2007	ic1				enhancement brought to ARM (Architecture Rationalisation Method) using a combination of AREL and Bayesian Belief Networks

a comparative study of architecture knowledge management tools	tang, antony and avgeriou, paris and jansen, anton and capilla, rafael and babar, muhammad ali	2010	ic2				
feature-based rationale management system for supporting software architecture adaptation	tekinerdogan, bedir and sozer, hasan and aksit, mehmet	2012	ic1		leisure	large size, open-source	implemented using the ArchiRationale tool, a design rationale management system
capturing tacit architectural knowledge using the repertory grid technique (nier track)	tofan, dan and galster, matthias and avgeriou, paris	2011	ic1				

improving architectural knowledge management in public sector organizations - an interview study (s)	tofan, dan and galster, matthias and avgeriou, paris	2013	ic3				
capturing and making architectural decisions: an open source online tool	tofan, dan and galster, matthias	2014	ic1				
past and future of software architectural decisions – a systematic mapping study	dan tofan and matthias galster and paris avgeriou and wes schuitema	2014	ic3				

empirical evaluation of a process to increase consensus in group architectural decision making	tofan, dan and galster, matthias and lytra, ioanna and avgeriou, paris and zdun, uwe and fouche, mark-anthony and de boer, remco and solms, fritz	2016	ic1	GADGET			extends previous work using the Repertory Grid Technique
making the right decision: supporting architects with design decision data	ven, jan salvador van der and bosch, jan	2013	ic1				
knowledge transfer between senior and novice software engineers: a qualitative analysis	viana, davi and conte, tayana and souza, cleidson r. b. de	2014	ic3				

integrating requirements and design decisions in architecture representation	weinreich, rainer and buchgeher, georg	2010	ic1		medicine	small-scale	based on the LISA meta-model and LISA toolkit
an expert survey on kinds, influence factors and documentation of design decisions in practice	weinreich, rainer and groher, iris and miesbauer, cornelia	2015	ic3				
software architecture knowledge management approaches and their support for knowledge management activities: a systematic literature review	weinreich, rainer and groher, iris	2016	ic2				

using architectural perspectives	woods, e. and rozanski, n.	2005	ic1	architectural perspectives	finance	enterprise	similar to viewpoint frameworks
diagrammatic modeling of architectural decisions	zalewski, andrzej and ludzia, marcin	2008	ic1	MAD(Maps of Architectural Decisions)			
capturing architecture evolution with maps of architectural decisions 2.0	zalewski, andrzej and kijas, szymon and sokolowska, dorota	2011	ic1	MAD 2.0	communications	large scale system	extension for MAD(Maps of Architectural Decisions)

References

[1] Jabref tool. `http://www.jabref.org/`. Accessed: 2017-08-10.

[2] Muhammad Ali Babar and Ian Gorton. A tool for managing software architecture knowledge. In *Sharing and Reusing Architectural Knowledge-Architecture, Rationale, and Design Intent, 2007. SHARK/ADI'07: ICSE Workshops 2007. Second Workshop on*, pages 11–11. IEEE, 2007.

[3] Muhammad Ali Babar, Ian Gorton, and Ross Jeffery. Capturing and using software architecture knowledge for architecture-based software development. In *Quality Software, 2005.(QSIC 2005). Fifth International Conference on*, pages 169–176. IEEE, 2005.

[4] John Bailey, David Budgen, Mark Turner, Barbara Kitchenham, Pearl Brereton, and Stephen Linkman. Evidence relating to object-oriented software design: A survey. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, pages 482–484. IEEE, 2007.

[5] Balbir Barn, Souvik Barat, and Tony Clark. Conducting systematic literature reviews and systematic mapping studies. In *Proceedings of the 10th Innovations in Software Engineering Conference*, ISEC '17, pages 212–213, New York, NY, USA, 2017. ACM.

[6] David E Bell and Howard Raiffa. *Decision making: Descriptive, normative, and prescriptive interactions*. Cambridge University Press, 1988.

[7] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4):571–583, 2007.

[8] Wanfeng Bu, Antony Tang, and Jun Han. An analysis of decision-centric architectural design approaches. In *Sharing and Reusing Architectural Knowledge, 2009. SHARK'09. ICSE Workshop on*, pages 33–40. IEEE, 2009.

[9] F. Buschmann. *PatternOriented-Software-Architecture-A-System-of-Patterns-Volume-1*. Number vol. 1. Wiley, 1996.

[10] Rafael Capilla and Muhammad Ali Babar. On the role of architectural design decisions in software product line engineering. In *European Conference on Software Architecture*, pages 241–255. Springer, 2008.

[11] Rafael Capilla, Anton Jansen, Antony Tang, Paris Avgeriou, and Muhammad Ali Babar. 10 years of software architecture knowledge management practice and future. 2015.

[12] Rafael Capilla, Francisco Nava, Sandra Pérez, and Juan C Dueñas. A web-based tool for managing architectural design decisions. *ACM SIGSOFT software engineering notes*, 31(5):4, 2006.

[13] Henrik Baerbak Christensen and Klaus Marius Hansen. An empirical investigation of architectural prototyping. 2010.

[14] Robert Cloutier, Gerrit Muller, Dinesh Verma, Roshanak Nilchiani, Eirik Hole, and Mary Bone. The concept of reference architectures. *Systems Engineering*, 13(1):14–27, 2010.

[15] Lionel Cohen and Mary Jean Scott. Fractionation procedures in radiation therapy: a computerised approach to evaluation. *The British journal of radiology*, 41(487):529–533, 1968.

[16] Harry Collins. *Tacit and explicit knowledge*. University of Chicago Press, 2010.

[17] E Jeffrey Conklin and KC Yakemovic. A process-oriented approach to design rationale. *Human-Computer Interaction*, 6(3):357–391, 1991.

[18] Jeff Conklin and Michael L. Begeman. gibis: A hypertext tool for exploratory policy discussion. In *Proceedings of the 1988 ACM Conference on Computer-supported Cooperative Work*, CSCW '88, pages 140–152, New York, NY, USA, 1988. ACM.

[19] R.C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A.J. Jansen. *Architectural Knowledge: Getting to the Core*, pages 197–214. Springer Verlag, LNCS 4880, 2008. core08.

[20] Klaas Andries de Graaf, Peng Liang, Antony Tang, Willem Robert van Hage, and Hans van Vliet. An exploratory study on ontology engineering for software architecture documentation. *Computers in Industry*, 65(7):1053–1064, 2014.

[21] Oscar Dieste and Anna Griman Padua. Developing search strategies for detecting relevant experiments for systematic reviews. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, pages 215–224. IEEE, 2007.

[22] Wei Ding, Peng Liang, Antony Tang, and Hans Van Vliet. Knowledge-based approaches in software documentation: A systematic literature review. *Information and Software Technology*, 56(6):545–567, 2014.

[23] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering*, pages 285–311, 2008.

[24] Emelie Engström and Per Runeson. Software product line testing–a systematic mapping study. *Information and Software Technology*, 53(1):2–13, 2011.

[25] Rik Farenhorst and Remco C de Boer. Knowledge management in software architecture: State of the art. In *Software Architecture Knowledge Management*, pages 21–38. Springer, 2009.

[26] David Garlan and Mary Shaw. An introduction to software architecture. *Advances in software engineering and knowledge engineering*, 1(3.4), 1993.

[27] Davydd J Greenwood. Organizational learning ii: Theory, method, and practice. *Industrial & Labor Relations Review*, 50(4):701, 1997.

[28] Iris Groher and Rainer Weinreich. Variability support in architecture knowledge management approaches: a systematic literature review. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 5393–5402. IEEE, 2015.

[29] MT Hansen, Nitin Nohria, and Thomas Tierney. What's your strategy for managing knowledge? response. *Harvard Business Review*, 77(3):196–196, 1999.

[30] Neil B Harrison and Paris Avgeriou. Leveraging architecture patterns to satisfy quality attributes. In *European Conference on Software Architecture*, pages 263–270. Springer, 2007.

[31] Anton Jansen, Tjaard de Vries, Paris Avgeriou, and Martijn van Veelen. Sharing the architectural knowledge of quantitative analysis. In *International Conference on the Quality of Software Architectures*, pages 220–234. Springer, 2008.

[32] B. Kitchenham and S Charters. Guidelines for performing systematic literature reviews in software engineering, 2007.

[33] Barbara Kitchenham, Rialette Pretorius, David Budgen, Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen G. Linkman. Systematic literature reviews in software engineering - A tertiary study. *Information & Software Technology*, 52(8):792–805, 2010.

[34] Josef Kolbitsch and Hermann A Maurer. The transformation of the web: How emerging communities shape the information we consume. *J. UCS*, 12(2):187–213, 2006.

[35] Philippe Kruchten. Architectural blueprints—the "4+ 1" view model of software architecture. *Tutorial Proceedings of Tri-Ada*, 95:540–555, 1995.

[36] Philippe Kruchten. What do software architects really do? *Journal of Systems and Software*, 81(12):2413–2416, 2008.

[37] Philippe Kruchten. Contextualizing agile software development. *Journal of Software: Evolution and Process*, 25(4):351–361, 2013.

[38] Philippe Kruchten, Robert L Nord, and Ipek Ozkaya. Technical debt: From metaphor to theory and practice. *Ieee software*, 29(6):18–21, 2012.

[39] Patricia Lago and Paris Avgeriou. First workshop on sharing and reusing architectural knowledge. *SIGSOFT Softw. Eng. Notes*, 31(5):32–36, September 2006.

[40] Jintae Lee and Kum-Yew Lai. What's in design rationale? *Human–Computer Interaction*, 6(3-4):251–280, 1991.

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

[42] Zengyang Li, Peng Liang, and Paris Avgeriou. Architecture viewpoints for documenting architectural technical debt. 2016.

[43] Allan MacLean, Richard M Young, Victoria ME Bellotti, and Thomas P Moran. Questions, options, and criteria: Elements of design space analysis. *Human–computer interaction*, 6(3-4):201–250, 1991.

[44] André Menolli, Maria Alexandra Cunha, Sheila Reinehr, and Andreia Malucelli. "old" theories,"new" technologies: Understanding knowledge sharing and learning in brazilian software development companies. *Information and Software Technology*, 58:289–303, 2015.

[45] Andre Menolli, Sheila Reinehr, and Andreia Malucelli. Organizational learning applied to software engineering: a systematic review. *International Journal of Software Engineering and Knowledge Engineering*, 23(08):1153–1175, 2013.

[46] Francisco Nava, Rafael Capilla, and Juan C Dueñas. Processes for creating and exploiting architectural design decisions with tool support. In *European Conference on Software Architecture*, pages 321–324. Springer, 2007.

[47] Paulo Anselmo da Mota Silveira Neto, Ivan do Carmo Machado, John D McGregor, Eduardo Santana De Almeida, and Silvio Romero de Lemos Meira. A systematic mapping study of software product lines testing. *Information and Software Technology*, 53(5):407–423, 2011.

[48] Ikujiro Nonaka and Noboru Konno. The concept of" ba": Building a foundation for knowledge creation. *California management review*, 40(3):40–54, 1998.

[49] Ikujiro Nonaka and Hirotaka Takeuchi. *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press, 1995.

[50] Robert L Nord, Ipek Ozkaya, Philippe Kruchten, and Marco Gonzalez-Rojas. In search of a metric for managing architectural technical debt. In *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on*, pages 91–100. IEEE, 2012.

[51] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *EASE*, volume 8, pages 68–77, 2008.

[52] Mark Petticrew and Helen Roberts. *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.

[53] Michael Polanyi. *The tacit dimension*. Routledge, London, 1966.

[54] Smrithi Rekha and Henry Muccini. Suitability of software architecture decision making methods for group decisions. In *European Conference on Software Architecture*, pages 17–32. Springer, 2014.

[55] Sharon Ryan and Rory V. O'Connor. Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, 55(9):1614–1624, 2013.

[56] Mojtaba Shahin, Zengyang Li, and Peng Liang. Recovering software architectural knowledge from documentation using conceptual model. pages 556–561, 2013.

[57] Mikael Svahnberg, Jilles Van Gurp, and Jan Bosch. A taxonomy of variability realization techniques. *Software: Practice and experience*, 35(8):705–754, 2005.

[58] Antony Tang, Paris Avgeriou, Anton Jansen, Rafael Capilla, and Muhammad Ali Babar. A comparative study of architecture knowledge management tools. *Journal of Systems and Software*, 83(3):352–370, 2010.

[59] Dan Tofan, Matthias Galster, Paris Avgeriou, and Wes Schuitema. Past and future of software architectural decisions – a systematic mapping study. 2014.

[60] Jeff Tyree and Art Akerman. Architecture decisions: Demystifying architecture. *IEEE software*, 22(2):19–27, 2005.

[61] Hataichanok Unphon and Yvonne Dittrich. Software architecture awareness in long-term software product evolution. *Journal of Systems and Software*, 83(11):2211–2226, 2010.

[62] Uwe Van Heesch, Paris Avgeriou, and Rich Hilliard. Forces on architecture decisions-a viewpoint. In *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on*, pages 101–110. IEEE, 2012.

[63] Rainer Weinreich and Iris Groher. Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review. *Information and Software Technology*, 80:265–286, 2016.

[64] Rainer Weinreich, Iris Groher, and Cornelia Miesbauer. An expert survey on kinds, influence factors and documentation of design decisions in practice. *Future Generation Computer Systems*, 47:145–160, 2015.

[65] Etienne Wenger. *Communities of practice: Learning, meaning, and identity*. Cambridge university press, 1998.

[66] Claes Wohlin. An evidence profile for software engineering research and practice. In *Perspectives on the Future of Software Engineering*, pages 145–157. Springer, 2013.

[67] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[68] Chen Yang, Peng Liang, and Paris Avgeriou. A survey on software architectural assumptions. *Journal of Systems and Software*, 113:362–380, 2016.

[69] Andrzej Zalewski and Marcin Ludzia. Diagrammatic modeling of architectural decisions. In *European Conference on Software Architecture*, pages 350–353. Springer, 2008.

[70] He Zhang, Muhammad Ali Babar, and Paolo Tell. Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637, 2011.