Variability Support in Architecture Knowledge Management Approaches: A Systematic Literature Review

Iris Groher and Rainer Weinreich
Johannes Kepler University Linz, Austria
{iris.groher, rainer.weinreich}@jku.at

Abstract
Research on software architecture knowledge management (SAKM) within the last 10 years has focused on capturing and storing design decisions and their underlying rationales. Recent attempts have tried to combine SAKM with variability management to support the capture of knowledge for a set of related products. To assess the current state of the art of variability management in the context of SAKM, we conducted a systematic literature review to identify the characteristics of the existing approaches for combining variability and SAKM and the gaps related to their application in practice. We further identified the main elements of SAKM models related to variability management, which can serve as a first step towards a commonly agreed approach for documenting architecture knowledge in the context of software product line engineering.

1. Introduction

Early views on software architecture emphasized “the structure or structures of a software system and their externally visible properties” [4], and later views started to include the design decisions leading to these solution structures and their rationale as part of an extended view on software architecture [5]. This shift established software architecture knowledge management (SAKM) [1] as a still very active sub-discipline of software architecture research. SAKM deals with capturing, sharing, and reusing architectural knowledge [13]. Many different approaches with different purposes have been developed and documented to manage software architecture knowledge [29].

Capturing and managing architecture knowledge in the context of software product lines requires additional methodological and tooling support [7]. Product line architectures contain variability that allows derivation of concrete product architectures during product derivation [3]. In product lines, both design alternatives representing different architectural solutions and alternatives representing the differences between the product architectures need to be captured.

Recent research aims at investigating the role of SAKM in software product line engineering (SPLE) [7] and how variability management and SAKM can be integrated [8][25]. However, a systematic assessment of the state of the art of SAKM in the presence of variability is still missing. The goal of this paper is to fill this gap by systematically assessing existing approaches in this area. Also, as part of our current work we changed our own architecture management approach [33] to a service-based infrastructure. This has led us rethink the SAKM and variability management support of our approach [16][17][34]. We assume that we can learn from existing approaches that combine SAKM and variability management, especially from those which have been applied in an industrial setting.

The work presented in this paper is part of a systematic literature review which we performed on SAKM papers published in the last 10 years. The review focuses on model-based SAKM approaches in general and in this paper we concentrate on variability management support in existing SAKM approaches. The contribution of this paper is an analysis of how existing SAKM approaches support variability management and a classification of the provided support by identifying the different concepts related to variability management in the SAKM models. We explicitly assess the evidence provided by the different SAKM approaches and identify gaps in existing research on variability support in SAKM with respect to the provided evidence. The gaps serve as a first step towards a research agenda on integrating SAKM and variability management.

The remainder of the paper is structured as follows. In Section 2 we discuss important concepts related to SAKM and variability management. In Section 3 we provide information about the research method, i.e., about the individual steps of the systematic literature review performed as part of this study. In Section 4 we present the analysis results. This includes the list of identified approaches, the main variability elements of their SAKM models, and a discussion of the research questions. In Section 5 we discuss potential threats to validity. In Section 6 we present related work and conclude the paper in Section 7.
2. Architecture knowledge management and variability management

SAKM [1] is a still very active sub-discipline of software architecture research. It deals with capturing, sharing, and reusing architectural knowledge [13]. Capturing makes architecture knowledge explicit by documenting knowledge about design decisions in a dedicated form; knowledge maintenance refers to the activity of keeping the knowledge up to date; knowledge sharing aims at distributing the documented knowledge among different stakeholders and in different contexts; knowledge using refers to using the knowledge in different architecture-related activities such as architecture analysis and architecture review; knowledge reusing refers to using architecture knowledge from one project in another project.

In a product line context, decisions are not only made for single products but for a whole family of related products. Lytra et al. [25] distinguish between variability decisions and architectural decisions. Variability decisions refer to decisions made as part of variability management. They typically correspond to variation points and their associated variants and are commonly documented in variability models. Architectural decisions refer to decisions made during the design of the architecture of the product line or the individual products during product derivation. They are thus made on both product line and product level. Product line decisions affect the product line architecture (e.g., by selecting a specific component model). Product decisions affect a single product and are typically constrained by variability decisions (e.g., component A needs to be taken because variant B has been selected).

During software product line engineering (SPLE) [27] both variability decisions and architectural decisions, including their relationships, need to be captured, shared, (re)used, and maintained. This study systematically investigates how existing SAKM approaches support variability management.

3. Research method

The systematic literature review was performed in accordance with Kitchenham's guidelines [20]. The main goal of our study was to identify the focus of research in terms of the different SAKM activities, how the activities are supported by tools and techniques, the evidence provided for the different activities, and what gaps exist in support of the different activities. The study was conducted from June 2013 to February 2014 and thus only includes papers published before June 2013.

3.1. Research questions

Even though the primary goal of our systematic literature survey was the analysis of support for the different SAKM activities, our search strategy (cf. Section 3.2) allowed us to answer additional research questions based on the identified primary studies since the primary studies include existing approaches for SAKM published in the last 10 years. In this study we analyze them to answer the following two research questions:

RQ1: How is variability management supported in existing SAKM approaches and what are the main elements related to variability management of the identified SAKM models?

RQ2: What are the gaps in existing research on variability support in SAKM with respect to the provided evidence?

In RQ1 we identify and analyze SAKM approaches that provide some kind of variability management support with the help of a systematic literature review. We also identify the main elements related to variability management of the SAKM models. In RQ2 we identify gaps and open issues in current research on variability management in the SAKM domain with respect to the provided evidence of the SAKM approaches. The identification of open issues is a first step towards a research agenda in this area.

3.2. Search strategy

We used an automated search strategy in four different scientific databases: IEEE, ACM, Springer, and Elsevier. As shown in Table 1, the search string we used is divided into three parts. The search string always contains the term “software architecture” because we are only interested in knowledge management approaches applicable in the area of software architecture. The second part consists of synonyms of “architectural knowledge” because AK is the context of our survey and the third part consists of synonyms of the terms “representation” or “management” because in our study we focus on approaches which provide some kind of knowledge representation or support for knowledge management with respect to SAKM activities. We automatically searched the four databases with all possible combinations of the search terms. To identify the synonyms commonly used in the software architecture community, we manually searched the titles, abstracts and keywords in different papers on SAKM.
Table 1. Search string for automatic search in scientific databases

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

To identify a suitable search string we conducted a pilot study of four venues (WICSA, ECSA, MODELS, QoSA) to verify and refine our initial key words. In particular, we manually examined the venues for relevant papers published in the last five years (since January 1, 2008). Table 1 shows the final search string that resulted from this analysis. The automatic search process in the four scientific databases yielded 440 publications.

3.3. Study selection process

We defined a number of inclusion and exclusion criteria for including or excluding a paper from the final review. Since this procedure leaves no room for subjective opinions, a single researcher (PhD student) conducted this part of the selection process. We defined the following inclusion and exclusion criteria in our study:

- **I1**: The study only includes publications available in electronic form.
- **I2**: The study only includes publications written in English.
- **I3**: The study only includes publications written since 2003.
- **I4**: The study only takes into account peer-reviewed publications appearing in journals, conferences, and workshops.
- **E1**: The study excludes introductions to special issues, workshops, tutorials, conferences, and conference tracks as well as editorials.
- **E2**: The study excludes PowerPoint presentations and short/extended abstract papers (publications with fewer than five pages).

Two hundred and eighty-six publications (about 65% of the initial set of papers obtained by the automatic search) were left after the exclusion of papers on the basis of the formal criteria. All publications meeting all inclusion criteria and not meeting any of the exclusion criteria were subjected to a voting stage. The voting stage involved four researchers (two senior researchers and two PhD students). Looking at the title and abstract of each publication the reviewers assessed whether the publication could contribute to answering the research questions or not. To determine the overall inter-rater agreement between the four reviewers we calculated Fleiss’s Kappa coefficient [14]. The value was 0.78 and according to Landis and Koch [23] this means substantial agreement. After the rating, publications in which the researchers strongly disagreed were discussed by at least two researchers with opposing opinions. After the voting stage 62 papers (about 22% of papers included after the application of the inclusion and exclusion criteria) were selected as potentially relevant for answering our research questions.

To ensure that no relevant publications had been forgotten, we conducted a snowball sampling [18] process. We went through all the references of the 62 selected primary studies and searched for potentially relevant publications. In total we found 1728 referenced publications. After removal of multiple entries 984 referenced publications remained. Forty publications were referenced by more than five publications.

From the 984 references we excluded all publications which did not meet the formal criteria (I1 to I3 and E1 to E2 introduced at the beginning of this section) or were not within the scope of AKM. After application of the formal criteria 72 potential publications were left of which 27 were already in the list of selected primary studies. The voting (as described above) for the remaining 42 publications resulted in 28 newly selected publications.

In total, 90 primary studies were selected through the search and selection process. We then performed an additional keyword search in the primary studies to obtain the studies which discuss variability management in the context of SAKM. We searched for “variability,” “variant,” “variants,” “product line,” and “product lines.” In total we identified 13 papers which discuss variability management in the context of
SAKM and can thus contribute to answering our research questions.

3.4. Data extraction

Each publication was read in detail to extract information about both the quality of the publication and specific information about the approaches or concepts, such as model, process or tool support for different SAKM activities, specific support for variability management and the evidence provided in the publication. In addition to general information about the selected study (such as authors, title, year of publication, publication venue, research context, relevance of approach) we also extracted information about the main elements of a potential SAKM meta-model and/or ontology and information about elements of the SAKM model related to variability management.

Each of the four reviewers extracted the data from about a quarter of the overall papers. The extracted data of each paper were additionally cross-checked by another reviewer. This means that each publication was read in detail by at least one reviewer and two reviewers checked the extracted data for each paper. The papers were assigned randomly to the four reviewers. Own publications were not self-extracted or checked. To obtain the same level of knowledge with respect to the data to be extracted, we performed a first round of extraction. Each reviewer extracted the data of one publication and all four reviewers discussed the data of these four publications. Afterwards the data extraction forms were adjusted according to the results of the discussions.

We explicitly rated the evaluation presented in the paper to assess which approaches demonstrated some usefulness through empirical studies or application in practice. The evidence was rated as no evidence (score 0.0); evidence obtained from demonstration examples (score 0.2); evidence obtained from expert opinions, surveys, observations and application examples in an industrial setting (score 0.4); evidence obtained from academic studies, case studies, or experiments (score 0.6); evidence obtained from industrial studies (score 0.8); and evidence obtained from industrial practice (score 1.0).

3.5. Data synthesis

After extracting the data from the primary studies we grouped the papers by approach. Typically, more than one paper was published for an approach, each having a different goal or focus. To be able to draw meaningful conclusions from our study regarding existing SAKM approaches and their variability in terms of management support we aggregated the results from the individual papers for each approach. From the 13 papers that discuss variability management in the context of SAKM we identified eight different approaches.

4. Results analysis

Table 2 presents the resulting eight SAKM approaches including their respective publications selected for analysis. The table lists each approach including a short description of the variability management support provided and the evidence with respect to variability management in the context of SAKM. In the next sections we analyze the approaches with regard to the research questions presented in Section 3.1.

4.1. RQ1: How is variability management supported in existing SAKM approaches and what are the main elements related to variability management of the identified SAKM models?

PAKME [2][7] is a web-based approach for SAKM that provides a central repository for managing architectural knowledge through a web interface. The repository supports the management of both generic (e.g., scenarios, patterns, design options) and project-specific knowledge (e.g., concrete scenarios, contextualized patterns, architecture decisions). In [2] the need for extending PAKME explicitly to support architecture design and evaluation in SPL was identified during the industrial evaluation of PAKME. The envisioned support involves identifying, modeling, and managing variability and variants in PAKME, including managing knowledge to support product derivation. The authors further plan to assess the utility of PAKME’s pattern repository for supporting pattern-based variability modeling and pattern-based description of design decisions for identifying design variants to realize different variation points captured in the variability model. Even though PAKME does not yet explicitly support SPL, in [7] the authors state that many artifacts in PAKME already support design and evaluation of product lines (such as general scenarios, design options, and analysis models). Required changes (such as explicit relationships between product-specific architectures and product line architectures) can easily be performed, as in PAKME modifications and tailoring are well supported [7].

ADDSS [6][7][9] is a web-based tool for managing architectural design decisions. It provides a basic dependency model for relating decisions and can
establish traceability between requirements, decisions, and architectures. Additionally, the status of decisions and the time they were made can be captured. ADDSS does not yet explicitly support variability management. To support SPLs, the data model of ADDSS can be extended with variation points, variants, and their relationships [7]. Such an extension would support association of design decisions with their respective variation points and variants. Relating parts of the variability model to a subset of the architecture in its current form is not possible because ADDSS cannot relate a set of design decisions to individual architectural parts. In [7] a core data model has been presented which contains the common SAKM entities supported by both PAKME and ADDSS. This core model has been extended to support variability management. Variation points can be captured and include a constraint rule which defines the logical relationship between the variants. Categories support the classification of variation points. Variation points are attached to architectural design decisions, which explain and motivate the definition of variation points. Variants are also linked to decisions to justify the selection of a particular variant in the architecture. In addition, the binding time of a variation point or variant can be captured. Binding times can also be related to decisions to indicate why a certain binding time has been chosen. In general, design decisions, requirements, and architecture descriptions can be product-specific or belong to the entire product line. This is indicated through attributes in the respective elements of its SAKM model.

The Decision View approach [10] introduces a new architectural view (in addition to the “4+1” views proposed by Kruchten [21]) to capture architecture knowledge. The view supports the capturing of design decisions, their rationale, applied patterns, related use cases, and design process iterations. The approach integrates variation points as specific types of decisions into its SAKM model. Other supported types of decisions are styles and patterns, but its SAKM model can be extended to cater for other types of decisions. Constraints and dependencies between decisions are also considered in the model. Queries support reasoning about decisions. Design decisions can be represented as a tree or a network.

The LISA [16][33][34] approach uses a single architecture-description model throughout the whole architecture lifecycle. This model is composed of a number of integrated sub-models describing different aspects of a software architecture. In addition to both low-level and high-level structures for describing the software architecture itself, the LISA model provides support for SAKM by including requirements and design decisions as first-class elements in the architecture model [34]. LISA provides explicit support for variability management by integrating the orthogonal variability modeling (OVM) approach of Pohl et al. [27] into the architecture model [16][17]. The model provides concepts like variation points, variants, and variability dependencies to capture the variability between a set of systems. A variability dependency is an association between a variation point and a variant. Variability dependencies can be optional, mandatory, or alternative. An optional variability dependency states that a variant related to a variation point can be part of a product but it does not need to be part of it. A mandatory dependency states that a variant must be part of a product if the associated variation point is part of it. Alternative dependencies define the number of optional variants which need to be selected for the associated variation point. Constraint dependencies describe restrictions (requires or excludes relationships) between variants, between variation points, or between variants and variation points. To document how the variability is realized in the system, traceability links can be defined between variation points or variants and development artifacts. Any element of the LISA model (e.g., requirements, decisions, components, classes) can be linked to variation points and variants. The integration of the variability model into the architecture model supports full traceability of variability from requirements and design decisions to architecture and code.

ADDRA [19] supports the recovery and documentation of architectural decisions. The recovery is based on the analysis of architectural deltas, which represent changes made to the architecture. The approach supports different ways to recover decisions from the architectural deltas. One way is the analysis of reference frameworks and platforms because they often include variability for design decisions. The deltas can be analyzed to see whether they include such decisions. ADDRA does not yet integrate variability management support directly in its SAKM model.

Lytra et al. [26] support staged decision-making to support multiple levels of design decisions such as high-level, technology-, domain-, and application-dependent design decisions. The different levels of decisions are typically performed by different stakeholders or groups of stakeholders. Multiple stages of decisions are also supported. At each stage the decision space derives from decisions taken in previous stages, which means that fewer possible decisions are left in later stages because of decisions taken in earlier stages. The approach is similar to staged configuration of feature models [11] and can thus well be used in a product line context during product derivation. Variability management concepts have not yet been added to its SAKM model.
Table 2. Selected SAKM approaches

<table>
<thead>
<tr>
<th>Approach</th>
<th>Variability Management Support</th>
<th>Variability Management Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 PAKME</td>
<td>Need for variability management support identified during industrial evaluation of PAKME, many artifacts can be used as-is for SPL, additional modifications required but easily possible</td>
<td>0.0 (note: example mentioned but not presented)</td>
</tr>
<tr>
<td>A10 ADDSS</td>
<td>Extension of its SAKM model to capture variation points and variants necessary but easily possible; relationships between variability model and architecture not supported</td>
<td>0.0 (note: example mentioned but not presented)</td>
</tr>
<tr>
<td>A11 Decision View</td>
<td>Variation points are specific types of design decisions in its SAKM model and can be captured along with their constraints and relationships</td>
<td>0.0</td>
</tr>
<tr>
<td>A13 LISA</td>
<td>Variability management support directly integrated into decision model, traceability between architecture and variability model</td>
<td>0.2</td>
</tr>
<tr>
<td>A30 ADDRA</td>
<td>Supports recovery of domain knowledge from reference platforms and frameworks including variability for design decisions provided by the platform/framework</td>
<td>0.2</td>
</tr>
<tr>
<td>A34 Lytra et al.</td>
<td>Multiple decision levels and decision stages could support the decision-making process in a product line context during product derivation</td>
<td>0.4</td>
</tr>
<tr>
<td>A38 Trujillo et al.</td>
<td>Extensions (variations) of a base architecture can have associated design decisions which could be used to form the final system</td>
<td>0.0</td>
</tr>
<tr>
<td>A43 Lago et al.</td>
<td>Invariability is explicitly modeled and related to features in a feature model</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Trujillo et al. [32] present an approach which provides concepts to extend architectures on the basis of design decisions. Design decisions are defined for the base architecture. The base architecture can be refined and extended at designated extension points (similar to variation points) which make the architecture extensible for new requirements. Extensions can have associated design decisions and can be used to model variability in the architecture along with the associated design decisions. Product architectures are generated by composing the base architecture and the extensions along with their related design decisions.

Lago et al. [22][28] present an approach for modeling both features and invariability (assumptions) in a product line context. Assumptions refer to things in the systems and their environment which will not change. Assumptions are explicitly modeled and related to features in a feature model to document their impact on the features. Features are realized by structural elements such as packages, interfaces, and modules. Assumptions can be categorized to support their grouping.

**Discussion:** Table 2 summarizes how variability management is supported by the approaches discussed above. Analysis of existing SAKM approaches shows that it is mostly limited to capturing information which is typically part of a variability model such as variation points, variants, and features in addition to design decisions (cf. Table 3). Out of the eight SAKM approaches contained in this study, four integrate support for variability management in their decision models. Two of the remaining approaches (PAKME and ADDSS) present a common core model with an extension for variability management. Two approaches (ADDRA, Lytra et al.) do not integrate variability management concepts in their decision models.

In analyzing the elements related to variability management of the different SAKM models we found that the provided concepts are quite similar. Most models integrate the concept of variation points and associated variants (core model of PAKME and ADDSS, Decision View, LISA, Trujillo et al.). The
captured variation points and variants can typically be linked to design decisions. The links can serve as justification for the introduced flexibility. This aligns with the aims of the OVM approach whereby a central variability model is related to the different artifacts such as requirements, architecture, code, and documentation.

Table 3. Variability elements supported by the analyzed approaches

<table>
<thead>
<tr>
<th>Approach</th>
<th>Variability Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAKME and ADDSS</td>
<td>variation point, variant, constraint rule, category, binding time</td>
</tr>
<tr>
<td>Decision View</td>
<td>variation point</td>
</tr>
<tr>
<td>LISA</td>
<td>variation point, variant, variability dependency, constraint dependency</td>
</tr>
<tr>
<td>ADDRA</td>
<td>N/A</td>
</tr>
<tr>
<td>Lytra et al.</td>
<td>N/A</td>
</tr>
<tr>
<td>Trujillo et al.</td>
<td>extension point</td>
</tr>
<tr>
<td>Lago et al.</td>
<td>feature</td>
</tr>
</tbody>
</table>

Apart from relating variability elements to design decisions, it is important to link variability and decision elements to the architecture representation. This is necessary to analyze how variability has been addressed in the design of a software system. Tool support for linking a variability model to both design decisions and parts of the architecture description is only provided by the LISA approach. However, the LISA approach also lacks empirical evidence for variability management.

In summary, the common aim of the approaches is support for capturing what varies in the system (variation points) and how it varies (variants). The approaches do not support complex constraints between variation points and between variants. At the most only requires and excludes constraints between variation points or variants are supported. Also, there is no clear distinction between decisions at product line level and decisions at product level in the approaches we analyzed.

As support for capturing variability clearly dominates, existing approaches could thus be used during domain engineering, where the product line architecture is defined. One approach (ADTRA) provides support for decision recovery, which could potentially be used during mining variability from existing systems or platforms. This is especially useful for establishing product line architectures from existing product architectures. Lytra et al. present an approach that could be valuable during application engineering where concrete products are derived from the product line. However, product derivation by binding variants through decision-making is poorly supported.

In summary, none of the approaches provides process support for explicitly capturing variability during SAKM activities. In [12] a process for documenting variability design rationale is presented but lacks empirical evidence.

Systematically using the information captured in the variability and decision models is also currently not supported. For example, support for architecture evaluation and review in a product-line context would be valuable. The need for consistency checking between decision and variability models has also been identified in [7].

4.2. RQ2: What are the gaps in existing research on variability support in SAKM with respect to the provided evidence?

Analysis of existing SAKM approaches shows that a broad application of SAKM in the context of variability and software product lines is still missing. As presented in Table 2, four approaches out of eight do not provide any evidence (evidence level 0.0). The core model of PAKME and ADDSS, the Decision View approach, and the approach of Trujillo et al. propose variability management extensions to their SAKM models but do not provide any examples. Three approaches (LISA, ADDRA, Lago et al.) provide demonstrations examples (evidence level 0.2). Lytra et al. provide evidence obtained from expert opinions (evidence level 0.4). None of the approaches have been applied in case studies or in industrial projects.

In the literature survey used as the basis for this study we assessed the evidence levels of 47 different SAKM approaches.\(^1\) In total, knowledge-capturing approaches had the highest rated evidence, followed by knowledge-using approaches. Knowledge reuse and sharing approaches provide some evidence but approaches related to knowledge maintenance do not provide any evidence. Compared with this, the total evidence for variability management in the context of SAKM is very low.

Even though SPLE has proven successful in industrial practice, explicit variability management has not yet found its way into mainstream SAKM approaches. Of the 47 SAKM approaches identified during our study only eight provide support for variability management. Even though the provided concepts are similar, the approaches are quite different in nature. PAKME and ADDSS integrate support for

\(^1\) A list of all the approaches and papers we identified in the systematic literature review can be found at http://www.se.jku.at/akm_slr.zip
variability management into a common core model but lack tool support and evidence. The decision view approach provides variation points as specific decision types but lacks support for more advanced concepts and evidence. LISA fully integrates OVM in its architecture management infrastructure, but the approach has only been applied in an application example. ADDRA focuses on variability mining. Lytra et al. and Trujillo et al. support the derivation of products during application engineering. This diversity clearly shows that a general SAKM approach with support for variability management, with adequate tool and process support, and with sufficient evidence for its applicability in practice is still a gap that has to be filled. This is a prerequisite for industrial adoption.

5. Assessment of review

Similarly to [15] we evaluate our systematic review against the four quality questions proposed by Kitchenham et al. [20]:

QA1: Are the review’s inclusion and exclusion criteria described and appropriate? This quality criterion is met as we explicitly define and discuss the inclusion and exclusion criteria we used in Section 3.3. The criteria are similar to the criteria used in other systematic literature reviews (e.g., [15][24][31]).

QA2: Is the literature search likely to have covered all relevant studies? According to Kitchenham et al. [10] this criterion is met if four or more digital libraries have been searched and additional search strategies have been included. This quality criterion is met as we performed an automatic search in four scientific databases (IEEE, ACM, Springer, and Elsevier) and additionally conducted a snowball sampling process. To ensure that our search terms were accurate, we conducted a pilot study to verify and refine the initial key words.

QA3: Did the reviewers assess the quality/validity of the included studies? This quality criterion is met as we explicitly assessed the quality of each primary study according to defined quality criteria. In total we defined eight criteria (cf. Section 3.4) to rate the quality of the publications. During the quality assessment process we also rated the evidence level of the approach presented in the publication. To reduce the personal bias in study assessment, the extracted data were checked by at least two reviewers. Also, own publications were not self-extracted or checked and a pilot extraction was performed. Furthermore, we looked at related studies (such as [24]) and analyzed how papers contained in both our study and the related study were rated. We found a strong agreement between the quality assessment results in most cases.

QA4: Were the basic data/studies adequately described? We consider this quality criterion as met as we used a detailed data collection form for each study. To ensure a common understanding of the data to be extracted from the studies we performed a pilot extraction. Each reviewer extracted the data of one publication and four reviewers discussed the data of these four publications. The data extraction forms were adjusted according to the results of the discussions.

6. Related work

In the last few years some studies have been published which assess the current state of the art in SAKM according to different criteria. We discuss those studies below. To the best of our knowledge there exists no study which systematically assesses variability management support in existing SAKM approaches.

Li et al. [24] conducted a systematic mapping study to assess the state of the art in how knowledge-based approaches are applied in software architecture. Although they added search terms related to variability management to their search string they did not perform an analysis of approaches which provide support for variability management in the context of SAKM.

Tofan et al. [31] present a systematic mapping study to assess the state of research on architectural decisions. They look at aspects like support for quality attributes and group decision-making. Although they also conclude that empirical evidence of research in SAKM is rather scarce, they did not specifically analyze approaches related to variability management.

Tang et al. [30] compare five SAKM tools and the support they provide in the architecture life cycle. Albeit in their study they point out that some approaches provide concepts related to variability management, support for variability documentation is not part of their tool comparison framework. The study analyzes which concepts of the IEEE 1471-2000 standard are supported but in contrast to our study does not compare the concepts related to variability management.

We [35] analyze the goals and aims of existing SAKM approaches with respect to different knowledge management activities. We also compare the important elements of the related SAKM models to identify core model elements. However, the analysis of variability management support is not part of our study.

Galster et al. [15] have conducted an SLR to assess the current state of the art in variability management research. They look at variability management methods, related activities in the software development process, and evidence provided for the different
approaches. They do not discuss variability in the context of design decisions and architecture knowledge management. They found that the development activities most affected by variability are architecture and design. This supports our initial motivation for conducting this study on variability management in the context of SAKM.

7. Conclusion

Analysis of SAKM approaches published in the last 10 years, which provide variability management support shows that approaches for capturing variation points and variants dominate. Of the approaches integrating these concepts into their SAKM models some also support capturing relationships between variability and design decisions. Integrating variability management concepts directly into the SAKM models ensures tight integration of variability modeling and decision modeling. In a software product-line context it is necessary that decision modeling is supported in both domain and application engineering. As pointed out in [25] architectural decisions affect both the product-line architecture and the different product architectures. The capturing and maintenance of the different decisions in the different phases need to be reflected in the development process. Tool support for capturing, maintaining, and using variability and design decisions together is also in the very early stages. For adequate support it is necessary to integrate variability modeling, capturing of related design decisions, and architectural solutions in one tool environment.

Another main finding is that there is still very weak evidence for variability in the context of SAKM. The published approaches have not yet been applied in case studies or industrial projects. More work in this direction is an important prerequisite for applying SAKM approaches in SPLE practice.

8. References


