Chapter 1

Introduction

1.1 Context

The ISO 42010 standard defines software architecture as the ‘fundamental concepts or properties of a system in its environment embodied in its elements, their relationships, and in the principles of its design and evolution’ (ISO/IEC/IEEE, 2011). The software architecture of a system is an abstraction, so the description or documentation of software architecture plays an important practical role. Elements of architecture documentation include descriptions of stakeholders (e.g. managers, business analysts, developers), and stakeholders’ concerns (e.g. performance, security), architectural decisions to satisfy the concerns, the rationales of the architectural decisions, and views that contain models (ISO/IEC/IEEE, 2011).

The software architecture discipline has evolved since its inception in the late 60’s. This evolution is marked by the following milestone ideas that have influenced decades of research and practice.

1. In the late 60’s – early 70’s, Dijkstra and Parnas proposed the concept of software architecture, emphasizing the importance of the structure of software systems.
2. In the mid 90’s, Shaw and Garlan emphasized components and connectors as fundamental software architecture concepts (Shaw and Garlan, 1996).
3. In the mid 00’s, an influential paper (Bosch, 2004) pushed for the perspective of software architecture as a result of architectural decisions, building on earlier ideas from (Perry and Wolf, 1992), that rationales of decisions are part of software architecture and architectural styles encapsulate decisions about architectural elements.

This thesis builds on the perspective of software architecture as a result of architectural decisions.
1. Introduction

Architectural decisions are a subset of design decisions that have a system-wide impact, involve trade-offs, are often constraining, are costly to change and hard to make (Zimmermann, 2011). Architectural decisions involve important choices on core components or connectors, and the overall software-intensive system, to satisfy and balance stakeholders’ concerns (Zimmermann, 2011). Architectural decisions have a key influence on the structure, behavior and quality attributes of software systems (Zimmermann, 2011). When making decisions, architects need to balance stakeholders’ concerns, such as quality attributes, business goals and functional requirements.

Examples of architectural decisions include selecting a development framework (e.g. J2EE, .NET), architectural patterns (e.g. client-server, layers), middleware for a distributed software system (e.g. an enterprise service bus), programming languages (e.g. PHP, Java), operating systems (e.g. Windows, Android, Linux), database systems (e.g. MySQL, Oracle), or how to decompose a system into modules.

1.2 Problem statement

Architectural decisions and their rationales are a critical part of the architectural knowledge of a system (Babar et al., 2009; Kruchten et al., 2006). In practice, such knowledge is often lost, or vaporized (Bosch, 2004; Jansen and Bosch, 2005).

The overall problem addressed in this thesis is the following:

**How can architectural knowledge vaporization be reduced?**

I focus on this overall problem because of its strong practical consequences:

- **Expensive maintenance and evolution**, since implementing new features in a software system requires understanding the previously made architectural decisions. Architects and developers need to spend substantial efforts to understand previous architectural decisions, to avoid introducing conflicts with them (Bosch, 2004) (Jansen et al., 2007).

- **Architectural drift** refers to the implementation drifting away from the initially intended architecture (Perry and Wolf, 1992), due to the lack of relevant and up-to-date architectural knowledge.
1.2 Problem statement

- Insufficient stakeholder communication by not sharing architectural knowledge (especially decisions), rationales and how the decisions satisfy stakeholders’ concerns (Jansen et al., 2007).
- Low reusability of previous decisions, through unawareness of reuse opportunities (Jansen et al., 2007).
- Poor traceability between architecture, implementation, and requirements (Harrison et al., 2007)

The following reasons contribute to architectural knowledge vaporization.
- Unawareness which refers to architects not realizing or not reflecting on their decision making (Harrison et al., 2007)
- Lack of training which refers to architects not knowing why and how to document decisions (Jansen et al., 2007)
- Difficulty which refers to the substantial effort that can be required by the documentation of architectural decisions (Harrison et al., 2007)
- Disruption which refers to architects postponing decision documentation to avoid disrupting their design flow (Harrison et al., 2007)
- Natural causes such as architects might change positions or retire (Jansen et al., 2007; Tang et al., 2006) or that architects might forget the rationales of decisions over time (Tang et al., 2006)

In this thesis, I focus on reducing the vaporization of architectural decisions, which are the most significant part of architectural knowledge (Babar et al., 2009; Kruchten et al., 2006). To reduce vaporization of architectural decisions, I propose the following high-level solution: improve architectural decision making processes, so that the processes contain explicit steps for documenting decisions.

Any proposal to improve such processes must consider the reasons for architectural knowledge vaporization, as follows:
- Unawareness - Improved processes should encourage architects to think more about their decisions, thus raising architects’ awareness about decisions
- Lack of training – Improved processes should have a low learning curve and provide sufficient advantages, so that architects are motivated to learn and use them
1. Introduction

- **Difficulty** – Improved processes should minimize documentation efforts and include steps to decompose the documentation task in small and easy to perform steps
- **Disruption** – Improved processes should encourage architects to focus on their decision making
- **Natural causes** – Improved processes should encourage immediate capturing of decisions rationales, to avoid the risk of architects forgetting them over time.

We revisit these five reasons and describe how the solution proposed in this thesis addresses them in Chapter 9 – Conclusions.

1.2.1 Research framework

Towards addressing the overall problem of this thesis, I used the iterative approach recommended by the design science framework (Wieringa, 2009), which is summarized in Figure 1.1. To solve practical problems from the environment, design science proposes iterations between practical problems and knowledge questions. The iterations start from a practical problem, which is decomposed into practical subproblems. In addition, knowledge questions help understand how to solve practical problems, as well as how to decompose the practical problems. In turn, answering knowledge questions may uncover further practical problems to be solved. Overall, Wieringa regards these iterations as a rational problem-solving process: analyze the environment and the goal, propose and evaluate changes towards the goal, apply changes and repeat iterations (Wieringa, 2009).

Wieringa defines a practical problem ‘as a difference between the way the world is experienced by stakeholders and the way they would like it to be’ and a knowledge question ‘as a difference between current knowledge of stakeholders about the world and what they would like to know’ (Wieringa, 2009). Practical problems aim at changing the environment (e.g. how to improve something? how to implement something?). Knowledge questions aim at changing knowledge about the environment (e.g. what are the facts? what are the effects?) (Wieringa, 2009).
1.3 Research questions

Figure 1.1. Design science framework, adapted from (Wieringa, 2009).

The iterations between practical problems and knowledge questions match the research efforts of this thesis, which starts from an overall practical problem (i.e. reducing architectural knowledge vaporization) of the software engineering industry. As part of my research efforts, I decomposed this practical problem into smaller, more targeted practical problems and knowledge questions. Typically, while working on a practical problem, new knowledge questions appeared, and the other way around. Given the limited resources, I had to focus my research efforts on the most promising practical problems and knowledge questions. Next, I present the practical problems and knowledge questions covered by this thesis.

1.3.1 Iteration A - Understand

In the first major iteration, my main goal was to understand the environment (i.e. the state of practice) in which architectural knowledge vaporization occurs, and the existing knowledge base (i.e. the state of research) that can help reduce architectural knowledge vaporization. Towards this, I investigated how architectural knowledge is managed in practice (item 1 in Figure 1.2), so that I could get a high-level overview of the situation in the industry. Since this
topic is very large, I focused mainly on identifying challenges and solutions for managing architectural knowledge.

<table>
<thead>
<tr>
<th>1. Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>How can architectural knowledge vaporization be reduced?</td>
</tr>
<tr>
<td>1.1. What are potential solutions to the challenges for managing architectural knowledge?</td>
</tr>
<tr>
<td>1.1.1. What are the characteristics of architectural decisions?</td>
</tr>
<tr>
<td>1.1.2. What factors make architectural decisions difficult?</td>
</tr>
<tr>
<td>1.1.3. What are the differences between junior and senior software architects?</td>
</tr>
<tr>
<td>1.1.4. What are the differences between good and bad architectural decisions?</td>
</tr>
<tr>
<td>1.1.5. What are the papers on documenting architectural decisions?</td>
</tr>
<tr>
<td>1.1.6. Does current research on architectural decisions consider functional requirements and quality attributes?</td>
</tr>
<tr>
<td>1.1.7. What specific domains for architectural decisions are investigated?</td>
</tr>
<tr>
<td>1.1.8. What are the normative and descriptive papers?</td>
</tr>
<tr>
<td>1.1.9. What are the papers on addressing uncertainty in architectural decisions?</td>
</tr>
<tr>
<td>1.1.10. What are the papers on group architectural decisions?</td>
</tr>
<tr>
<td>1.1.11. What are the advantages and disadvantages of the Repertory Grid technique?</td>
</tr>
<tr>
<td>1.1.12. Does the Repertory Grid technique reduce architectural knowledge vaporization more than a template-based approach?</td>
</tr>
<tr>
<td>1.1.13. What are the advantages and disadvantages of REGAIN?</td>
</tr>
<tr>
<td>1.1.14. What are the improvement opportunities for REGAIN?</td>
</tr>
<tr>
<td>1.1.15. Which prioritization approach to use for REGAIN?</td>
</tr>
<tr>
<td>1.1.16. Is there a practical need for increasing consensus in group architectural decision making?</td>
</tr>
<tr>
<td>1.1.17. What are the effort and benefits offered by GADGET?</td>
</tr>
<tr>
<td>1.1.18. What are potential improvements to GADGET?</td>
</tr>
<tr>
<td>1.1.19. Compared to ADHOC, what is the impact of GADGET on increasing consensus among group architectural decision makers?</td>
</tr>
<tr>
<td>1.1.20. How do perceptions on GADGET vs. ADHOC differ among decision makers?</td>
</tr>
</tbody>
</table>

Figure 1.2. Summary of knowledge questions and practical problems addressed in this thesis for the three phases: understand (A), explore (B), and propose (C).
1.3. Research questions

Since architectural decisions are a significant part of architectural knowledge, in the next step, I focused on how architectural decisions are made in practice (item 2 in Figure 1.2). In this investigation, I answered four knowledge questions (items 2.1 – 2.4 in Figure 1.2) regarding real-world architectural decisions:

2.1. **What are the characteristics of architectural decisions?**
   Answering this question increases knowledge on real-world architectural decisions.

2.2. **What factors make architectural decisions difficult?** Since architectural decisions are difficult (i.e. hard to make) (Zimmermann, 2011), then it is important to understand the nature of the difficulties faced by practitioners.

2.3. **What are the differences between junior and senior software architects?** Junior and senior software architects might make architectural decisions with different characteristics and face different difficulties when making them.

2.4. **What are the differences between good and bad architectural decisions?** Understanding such differences helps improve architectural decisions.

Next, I investigated the state of research on architectural decisions (i.e. item 3 in Figure 1.2), by answering the following six knowledge questions.

3.1. **What are the papers on documenting architectural decisions?**
   Understanding existing work on documenting architectural decisions is a prerequisite for proposing improved approaches for reducing architectural knowledge vaporization.

3.2. **Does current research on architectural decisions consider functional requirements and quality attributes?** Satisfying functional requirements and quality attributes are critical activities of architects.

3.3. **What specific domains for architectural decisions are investigated?** Architects work and make decisions in many domains in the industry. Different domains may offer different challenges for making architectural decisions.
3.4. **What are the normative and descriptive papers?** This question refers to understanding which papers recommend approaches on architectural decisions (i.e. normative), and which papers present how architectural decisions are made in practice (i.e. descriptive).

3.5. **What are the papers on addressing uncertainty in architectural decisions?** Uncertainty plays an important role in making architectural decisions.

3.6. **What are the papers on group architectural decisions?** If an important part of architectural decisions are made in group (rather than individually), then it makes sense to propose approaches that help group architectural decision making.

At the end of the first major iteration, I understood five important points:

1. Architectural knowledge vaporization is a major challenge in the industry (see Chapter 2). This convinced me on the importance of the overall problem addressed in this thesis (i.e. reducing architectural knowledge).

2. Much work exists on documenting architectural decisions, as a means to reduce architectural knowledge vaporization (see Chapter 4). This suggested that although much work had been done towards reducing architectural knowledge, the approaches were not enough to solve the problem.

3. Existing work on documenting architectural decisions does little reuse of ideas from other fields (based on Chapter 4). This encouraged me to explore ideas from other fields to reuse proven approaches for reducing knowledge vaporization, in the second major iteration (B), detailed in 1.3.2.

4. Many architectural decisions are made in groups (see Chapter 3). This encouraged me to work towards improving group architectural decision making in the third major iteration (C), detailed in Section 1.3.3.

5. There are very few open-source tools to help architects make and document architectural decisions (based on Chapter 4). This encouraged me to offer practitioners open-source tool support for
making and documenting architectural decisions in the third major iteration (C), detailed in Section 1.3.3.

### 1.3.2 Iteration B - Explore

The points that I understood at the end of the first major iteration (A) motivated me to explore ideas from other fields for reducing architectural knowledge vaporization, improving group decisions, and offering open-source tool support. In particular, the knowledge engineering field was a particularly compelling source of potentially fruitful ideas for reducing architectural knowledge vaporization.

The rationale for looking at the knowledge engineering field was the following. The knowledge engineering field has decades of experience and a strong focus on capturing knowledge from experts in various domains. In contrast, in the software architecture field, the focus on capturing architectural knowledge started in 2004 (as found in Chapter 4). Because of the relatively recent (i.e. 2004) focus on capturing architectural knowledge, some shortcomings of proposed approaches are expectable. For example, proposed approaches might lack clear steps, validation, solid conceptual foundations, tool support, or significant functionality (e.g. no support for group decision making).

Still, reusing ideas from the knowledge engineering field has several risks. Some ideas might not be suitable for capturing architectural knowledge. Also, some ideas might be not specific enough for the software architecture field.

While aware of potential risks and benefits, in the second major iteration (item 4 in Figure 1.2), I explored ideas from the knowledge engineering field, to identify ideas that could help reduce architectural knowledge vaporization. In particular, I focused on the idea of using the Repertory Grid technique for making and capturing architectural decisions (item 4 in Figure 1.2). I started exploring this idea after personal discussions with Tim Menzies - a researcher with much experience in knowledge engineering. Starting from these initial discussions, I investigated advantages and disadvantages of the Repertory Grid technique (item 4.1), as well as if the Repertory Grid technique reduces architectural knowledge vaporization (item 4.2) better than a template based approach.
1.3.3 Iteration C - Propose

In the third iteration (C), I used the results of the previous two iterations to propose practical solutions to reducing architectural knowledge vaporization. First, the exploration in item 4 in Figure 1.2 indicated that the Repertory Grid technique had much potential for reducing architectural knowledge vaporization. Therefore, I proposed, evaluated, and refined a process (i.e. REGAIN, item 5 in Figure 1.2) based on the Repertory Grid technique to help make and document individual architectural decisions (i.e. decisions made by one architect). Next, I identified advantages and disadvantages of REGAIN (item 5.1), as well as improvement opportunities for REGAIN (item 5.2). An important improvement opportunity was to add and investigate a prioritization approach for REGAIN (item 5.3).

Second, the answers to items 2 and 3 in Figure 1.2 (i.e. on state of practice and research on architectural decisions) indicated a gap on group architectural decisions. This is a large topic, so I focused on a particular aspect: increasing consensus in group architectural decisions. First, I investigated the need for increasing consensus in group architectural decisions. Next, I proposed a process (i.e. GADGET) to increase consensus in group architectural decisions, and capture the group decision. Furthermore, I investigated costs, benefits, and potential improvements for GADGET (items 6.2 and 6.3). Finally, I compared GADGET with ADHOC (i.e. group decision making without using any prescribed approach). From the comparison, I identified the impact of GADGET on consensus and perceptions of the decision makers.

Third, I offered tool support for the REGAIN and GADGET processes (item 7 in Figure 1.2), since practitioners indicated the need for user-friendly, open source tool support.

Figure 1.2 shows that the practical problems (with the exception of offering tool support) and knowledge questions are further decomposed into more focused knowledge questions or practical problems. Each of the knowledge questions from 1 to 4, and practical problems 5, 6, and 7 are also high-level research questions. Furthermore, knowledge questions and practical problems 1 to 6 are further decomposed into more concrete research questions (e.g. 1.1, 5.3).
1.4 Research methods

Related work and rationales for each research question are presented in their corresponding chapters of this thesis. The corresponding chapters are presented in the next section.

1.4 Research methods

To answer the seven high-level research questions and their corresponding 21 research questions, this thesis used empirical research methods that provide evidence to substantiate the answers. Next, I describe briefly each empirical research method, and their typical usages.

1. **Experiment** – this (mostly) quantitative empirical research method is used to confirm or reject predefined hypotheses about the relation between factors, by applying different treatments in a controlled environment and measuring the effect of the treatments (Wohlin et al., 2012). Experiments are typically used to establish cause-effect relationships among predefined factors (Wohlin et al., 2003).

2. **Survey** – there are two types of surveys. First, quantitative surveys are used to ‘describe, compare or explain knowledge, attitudes and behavior’ (Pfleeger and Kitchenham, 2001), using questionnaires filled out by participants from carefully selected samples. Second, qualitative surveys (or interview studies) are used to collect data and impressions from participants about something (e.g. a process) (Hove and Anda, 2005; Myers and Newman, 2007; Seaman, 2008). Depending on the goal of the study, qualitative surveys can be structured (i.e. with a narrow, confirmatory focus and closed questions), unstructured (i.e. with a broad, exploratory focus and open-ended questions), or semi-structured (i.e. mix of the two) (Seaman, 2008).

3. **Case study** – this qualitative empirical research method is used to study contemporary phenomena in their natural contexts (Runeson and Höst, 2009; Yin, 2003). Case studies are particularly useful for descriptive and exploratory studies, and less so for establishing causal relationships (Runeson and Höst, 2009).

4. **Systematic mapping study** – this empirical research method is used to identify, evaluate and interpret the research on a particular topic (Kitchenham and Charters, 2007). Systematic mapping studies are similar to systematic literature reviews. However, systematic mapping...
studies cover more papers and focus on a broader literature analysis, compared to systematic literature reviews (Kitchenham and Charters, 2007; Petersen et al., 2008).

Table 1.1 shows the mapping of the above four research methods to the research questions in Figure 1.2, and where in this thesis the research questions are presented. Table 1.1 shows that this thesis presents nine empirical studies: two experiments, five surveys (one survey per chapter in Chapters 1, 2, 5, and two surveys in Chapter 4), a case study, and a systematic mapping study. Overall, 83 practitioners and 177 students participated in nine empirical studies and provided evidence to answer the research questions in Figure 1.2.
1.5 Publications overview

<table>
<thead>
<tr>
<th>ID</th>
<th>High-level research question</th>
<th>Empirical method</th>
<th>Participants</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>How is architectural knowledge managed in practice?</td>
<td>Survey</td>
<td>11 practitioners</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>How are architectural decisions made in practice?</td>
<td>Survey</td>
<td>43 practitioners</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>What is the state of research on architectural decisions?</td>
<td>Systematic mapping study</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Can the Repertory Grid technique reduce architectural knowledge vaporization?</td>
<td>Survey</td>
<td>27 students</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>How to support making and documenting individual architectural decisions?</td>
<td>Survey, Experiment</td>
<td>16 practitioners, 30 students</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>How to support making and documenting group architectural decisions?</td>
<td>Case study, Experiment</td>
<td>13 practitioners, 120 students</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>What tool can support REGAIN and GADGET?</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
</tbody>
</table>

1.5. Publications overview

This thesis is based on a collection of published papers. My contributions included planning the research, collecting data, interpreting data, reporting research results, revising and rewriting manuscripts to ensure the quality level required for publication, Table 1.2 indicates the publications corresponding to the chapters of this thesis.
## Table 1.2. Mapping of thesis chapters to publications.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Under review at the <em>Information and Software Technology</em> journal.</td>
</tr>
</tbody>
</table>