Rationale Visualization of Software Architectural Design Decision using Compendium

Mojtaba Shahin
Department of Computer Engineering and IT
Sheikh Bahaei University, Iran
mojtabashahin@gmail.com

Peng Liang
State Key Laboratory of Software Engineering (SKLSE)
Wuhan University, China
liangp@sklse.org

Mohammad Reza Khayyambashi
Faculty of Engineering, Department of Computer Engineering
University of Isfahan, Iran
m.r.khayyambashi@eng.ui.ac.ir

ABSTRACT

The justification for software architectural design decisions made throughout the architecting process is necessary for understanding, (re)using, communicating, and modifying an architecture design. Although there are many existing tools to capture, store, manage, and share the architectural design decisions explicitly, there still remains a need to visualize and explore the architectural design decisions and their underlying rationale. This paper investigates how Compendium tool can be employed to visualize architectural design decisions and their rationale, in order to improve the understandability and promote the communication of architectural design decisions.

Categories and Subject Descriptors
D.2.11 [Software Engineering]: Software Architecture – Languages;  I.6.8 [Computing Methodologies]: Simulation and Modeling – Visual

General Terms
Design

Keywords
Design rationale, architectural design decision, Compendium tool

1. INTRODUCTION

Software architecture plays an important role in managing complicated interactions between stakeholders of software-intensive systems to balance all kinds of constraints [1]. The architecting process can be considered as a decision making process, through which the appropriate decisions must be made. Traditional architectural approaches concentrate on components and connectors which cause problems, e.g., expensive system evolution, and limited reusability of architecture. These problems are partially due to the volatilization of design decisions and their rationale, which are implicit in people’s head mostly. Representation of design decision as a first-class entity in architecture design can alleviate these problems [3].

Practitioners and researchers have made great efforts to develop models and related tools to model, capture, manage, share, and (re)use architectural design decisions (ADDs) explicitly. In [4], the authors suggested five requirements for a decision view in architecture, including visual representation of design decisions. Although there are many existing tools to capture, store, manage and share ADDs explicitly, there still remains a need to visualize and explore ADDs and their underlying rationale in architecting process [5].

Visualizing and exploring ADD can assist the understanding of ADD and the reasoning behind the ADD (design rationale), especially in a collaborative and distributed development environment. There are a few tools providing visualization support of ADD, but none of them support explicit rationale visualization of ADD. Compendium is a semantic hypertext concept mapping tool which supports Issue-Based Information System (IBIS), an argumentation-based approach for design rationale representation. In this paper we employ Compendium to visualize and explore ADDs in a concrete ADD example based on Service-Oriented Architecture Decision (SOAD) model [2].

2. RATIONALE VISUALIZATION OF ADD

We adopt an ADD model on service-oriented architecture design, the SOAD model [2], to visualize and explore the ADDs based on this model. Figure 1 shows the concepts in SOAD: Architectural Decision (AD) is the central concept that expresses a single and concrete design issue, and has characteristics: e.g., Scope, Decision Drivers, etc. In this model, every AD has one or several AD alternatives with specification of Pros and Cons. ADkwik (Architectural Decision Knowledge Wiki) tool implements the SOAD model, and it is used to capture, store, manage, and share ADDs and their rationale in a text-based manner, but it does not provide any visualization capability. We employ Compendium to visualize the SOAD ADDs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC’10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03...$10.00.

Practitioners and researchers have made great efforts to develop models and related tools to model, capture, manage, share, and (re)use architectural design decisions (ADDs) explicitly. In [4], the authors suggested five requirements for a decision view in architecture, including visual representation of design decisions. Although there are many existing tools to capture, store, manage and share ADDs explicitly, there still remains a need to visualize and explore ADDs and their underlying rationale in architecting process [5]. Visualizing and exploring ADD can assist the understanding of ADD and the reasoning behind the ADD (design rationale), especially in a collaborative and distributed development environment. There are a few tools providing visualization support of ADD, but none of them support explicit rationale visualization of ADD. Compendium1 is a semantic hypertext concept mapping tool which supports Issue-Based Information System (IBIS), an argumentation-based approach for design rationale representation. In this paper we employ Compendium to visualize and explore ADDs in a concrete ADD example based on Service-Oriented Architecture Decision (SOAD) model [2].

2. RATIONALE VISUALIZATION OF ADD

We adopt an ADD model on service-oriented architecture design, the SOAD model [2], to visualize and explore the ADDs based on this model. Figure 1 shows the concepts in SOAD: Architectural Decision (AD) is the central concept that expresses a single and concrete design issue, and has characteristics: e.g., Scope, Decision Drivers, etc. In this model, every AD has one or several AD alternatives with specification of Pros and Cons. ADkwik (Architectural Decision Knowledge Wiki) tool2 implements the SOAD model, and it is used to capture, store, manage, and share ADDs and their rationale in a text-based manner, but it does not provide any visualization capability. We employ Compendium to visualize the SOAD ADDs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC’10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03...$10.00.

Practitioners and researchers have made great efforts to develop models and related tools to model, capture, manage, share, and (re)use architectural design decisions (ADDs) explicitly. In [4], the authors suggested five requirements for a decision view in architecture, including visual representation of design decisions. Although there are many existing tools to capture, store, manage and share ADDs explicitly, there still remains a need to visualize and explore ADDs and their underlying rationale in architecting process [5].

Visualizing and exploring ADD can assist the understanding of ADD and the reasoning behind the ADD (design rationale), especially in a collaborative and distributed development environment. There are a few tools providing visualization support of ADD, but none of them support explicit rationale visualization of ADD. Compendium1 is a semantic hypertext concept mapping tool which supports Issue-Based Information System (IBIS), an argumentation-based approach for design rationale representation. In this paper we employ Compendium to visualize and explore ADDs in a concrete ADD example based on Service-Oriented Architecture Decision (SOAD) model [2].

2. RATIONALE VISUALIZATION OF ADD

We adopt an ADD model on service-oriented architecture design, the SOAD model [2], to visualize and explore the ADDs based on this model. Figure 1 shows the concepts in SOAD: Architectural Decision (AD) is the central concept that expresses a single and concrete design issue, and has characteristics: e.g., Scope, Decision Drivers, etc. In this model, every AD has one or several AD alternatives with specification of Pros and Cons. ADkwik (Architectural Decision Knowledge Wiki) tool2 implements the SOAD model, and it is used to capture, store, manage, and share ADDs and their rationale in a text-based manner, but it does not provide any visualization capability. We employ Compendium to visualize the SOAD ADDs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC’10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03...$10.00.

Practitioners and researchers have made great efforts to develop models and related tools to model, capture, manage, share, and (re)use architectural design decisions (ADDs) explicitly. In [4], the authors suggested five requirements for a decision view in architecture, including visual representation of design decisions. Although there are many existing tools to capture, store, manage and share ADDs explicitly, there still remains a need to visualize and explore ADDs and their underlying rationale in architecting process [5]. Visualizing and exploring ADD can assist the understanding of ADD and the reasoning behind the ADD (design rationale), especially in a collaborative and distributed development environment. There are a few tools providing visualization support of ADD, but none of them support explicit rationale visualization of ADD. Compendium1 is a semantic hypertext concept mapping tool which supports Issue-Based Information System (IBIS), an argumentation-based approach for design rationale representation. In this paper we employ Compendium to visualize and explore ADDs in a concrete ADD example based on Service-Oriented Architecture Decision (SOAD) model [2].

2. RATIONALE VISUALIZATION OF ADD

We adopt an ADD model on service-oriented architecture design, the SOAD model [2], to visualize and explore the ADDs based on this model. Figure 1 shows the concepts in SOAD: Architectural Decision (AD) is the central concept that expresses a single and concrete design issue, and has characteristics: e.g., Scope, Decision Drivers, etc. In this model, every AD has one or several AD alternatives with specification of Pros and Cons. ADkwik (Architectural Decision Knowledge Wiki) tool2 implements the SOAD model, and it is used to capture, store, manage, and share ADDs and their rationale in a text-based manner, but it does not provide any visualization capability. We employ Compendium to visualize the SOAD ADDs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC’10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03...$10.00.

1 http://compendium.open.ac.uk/institute/
2 http://www.alphaworks.ibm.com/tech/adkwik
2.1 Mapping from SOAD to Compendium

In this section, we present how SOAD concepts are mapped to Compendium nodes, which are visualization elements in Compendium. Table 1 presents the mappings between Compendium nodes and SOAD concepts. Decision node in Compendium can be mapped to the AD concept in SOAD model. Note node in Compendium is used to provide extra and useful information about a node; therefore we can map the Note node to Role, Scope, EditorialInfo, Decision Drivers and Recommendation concepts in SOAD. Question node in Compendium can represent Problem Statement concept in SOAD. Answer node in Compendium represents alternative solution addressing to Question with Pro and Con Arguments, which support and object to this alternative, and therefore Answer node is used to represent ADAlternative concept in SOAD. Pro and Con Argument can be directly mapped to Pros and Cons in SOAD.

Table 1. Mapping SOAD concepts in Compendium nodes.

<table>
<thead>
<tr>
<th>SOAD Concepts</th>
<th>Compendium Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural Decision (AD)</td>
<td>Decision</td>
</tr>
<tr>
<td>Role, Scope, EditorialInfo,</td>
<td></td>
</tr>
<tr>
<td>Decision Drivers, Recommendation</td>
<td>Note</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Question</td>
</tr>
<tr>
<td>ADAlternative</td>
<td>Answer</td>
</tr>
<tr>
<td>Pros</td>
<td>Pro Argument</td>
</tr>
<tr>
<td>Cons</td>
<td>Con Argument</td>
</tr>
</tbody>
</table>

2.2 SOAD ADD Visualization Example

We use an example from SoadSample3 provided by ADwik to visualize the ADDs in SOAD model. Figure 2 presents the visualization of a SOAD ADD and its related entities in Compendium. The design problem (rieben) to be addressed is: “Should a Service Composition Layer (SCL) be introduced into the solution architecture?” The “ServiceCompositionParadigm” decision (rieben) gets selected with five alternatives (rieben): “Integration layer”, “SCL and workflow pattern”, “Programming language component model”, “Portal or mashup”, and “human user invoking atomic business functions”. Every alternative has positive and negative effects to the design problem, which are represented by Pro (rieben) and Con (rieben) Argument”. The rationale of an ADD can be easily visualized using Compendium in a straightforward way. For example, the positive effect (rieben) of alternative (rieben) “Programming language component model” is “Mature technique”, while the negative effect (rieben) of this alternative is “Not leveraging SOA benefits”. Users can easily follow the visualization link between rationale entities through an ADD in Compendium, and update their decision when design context changes (e.g., “when a component model is employed instead of SOA”, then the negative effect “Not leveraging SOA benefits” can be ignored/removed). The rationale visualization of ADD can improve the understandability of ADDs.

3. CONCLUSIONS

Software architects usually are interested in understanding why the software architecture looks the way it does. The architectural design decisions and their rationale can answer this question if they can be captured, managed, shared and (re)used explicitly. Although many ADD models and related tools have been proposed for these objectives, there is still a need to visualize and explore of ADDs and underlying rationale. In this paper we employ Compendium tool, a general design rationale visualization tool, to visualize ADDs and the underlying rationale. The visualization of ADD with related entities using Compendium can assist architects in understanding ADD and its rationale, and communicating architecture design effectively.

4. ACKNOWLEDGMENTS

This research project has been partially supported by the National Natural Science Foundation of China (NSFC) under Grant No.60903034.

5. REFERENCES