Visualizing Software Architectural Design Decisions

Larix Lee and Philippe Kruchten

University of British Columbia
{llee,pbk}@ece.ubc.ca

Abstract. Software architecture can be represented as a set of design decisions. Exploring and analyzing architectural design decisions are difficult due to how the decisions are represented and displayed. We describe four visualization aspects that apply to architectural design decision exploration and analysis: 1) tabular listing; 2) decision structure visualization; 3) decision chronology visualization; and 4) decision impact visualization. These aspects address some situations where visualization helps people understand and utilize decisions.

1 Introduction

Designing software is the process of making many design decisions that defines and guides the development of a software system. Some of these decisions are architectural design decisions, as they govern the overarching goals and characteristics of the system being developed. For example, the decision to support which target platforms or the decision to use a particular communication topology are architectural design decisions. These decisions intertwine with many other decisions such that changing one of these decisions may significantly affect other aspects of the design [3], leading to the view that these decisions are a fundamental part of software architecture [1, 2].

Keeping track of architectural design decisions is important in the later stages of software development and maintenance, when the original software architects are no longer available to answer various questions on the design intents and what decisions they made or had previously rejected. We addressed the issues of decision capture in a previous work [6]; however, it can be difficult to explore and analyze the captured decisions effectively. We propose that visualizing the decisions with different aspects may help with architectural design decision exploration and analysis by highlighting certain decision information that will assist the software architect, designer, or developer in performing their tasks. We describe these aspects and how they could contribute to architectural design decision exploration and analysis. We also present conceptual examples using screenshots of a decision visualization tool [7] that we developed.

2 Architectural Decision Exploration and Visualization

Exploration and analysis of architectural design decisions depend on how we structure and represent the information of each decision. Various design decision representation models have been proposed and these models focus on different aspects or attributes.

R. Morrison, D. Balasubramaniam, and K. Falkner (Eds.): ECSA 2008, LNCS 5292, pp. 359–362, 2008.

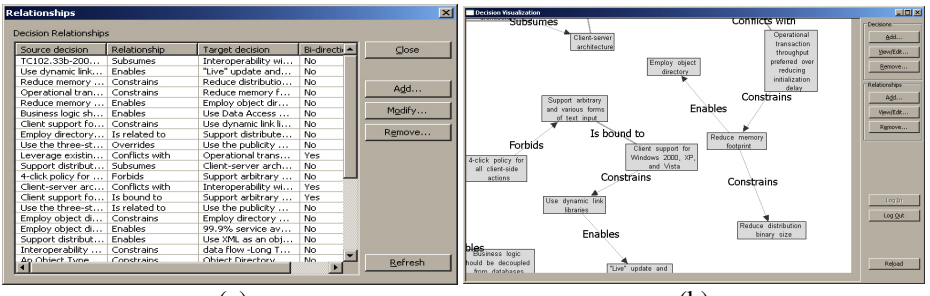
The argumentative design rationale approaches like the Issue-Based Information Systems [5] focus on the background information, context, and available options at the time the decision was made, while other models focus more on the decisions' attributes and interconnectivities by representing them as first-class entities, such as the architecture decision description template [8] and our decision ontology model [4]. We chose our decision model as it includes aspects of development processes like decision states (e.g., "idea", "tentative", "decided" or "approved") and it makes decision relationships explicit. Relationships include how decisions constrain, forbid, enable, subsume, override, comprise, bind, conflict with, or are alternatives to other decisions.

Exploring and analyzing design decisions also depend on the way the structured decision information is conveyed. Visualization is often used to explore large networks or to understand complex systems like in program comprehension. Visualization is also an identified requirement for the decision view of software architecture [2]. Decision visualization helps people build mental models of the decision space when the decision set gets large (several hundred architectural design decisions) and reveals information hidden in a set of design decisions by supporting associations, groupings, and layout (where we can identify patterns). Since different tasks and situations focus on different decision attributes, different visualization views into the decision model can help prioritize or filter out information that is not directly relevant to a particular exploratory or analytical task. We propose four visualization aspects to visualize software architectural design decisions: tabular listing, decision structure visualization, decision chronology visualization, and decision impact visualization.

Tabular Listing. The purpose of this visualization aspect is to provide a quick and effective way to browse and retrieve information from design decisions. The textual tabular representation facilitates decision querying and simple decision entry because the data representation can be easily parsed on a computer screen or on a paper printout. Although tables provide efficient textual display of decision information, it is difficult to quickly trace and assess decision structures, relationships, and properties when the decision sets get large or change relationships frequently. Fig. 1a shows a list of design decisions for a project. Relationship lists are not shown in this figure.

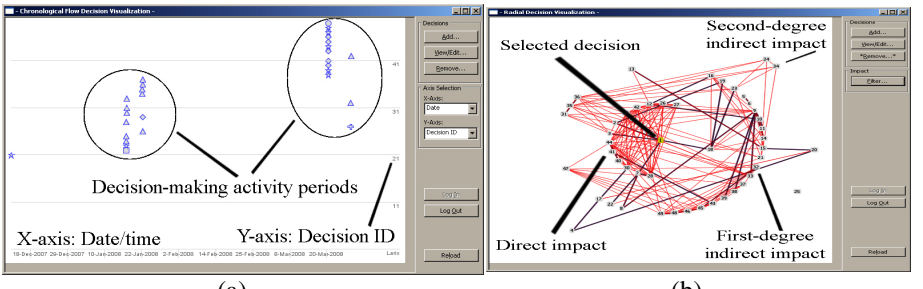
Decision Structure Visualization. The goal of this aspect is to increase understanding of the architecture's decision structure. The decision structure guides the capture, perusal, and manipulation of decisions and their relationships without sacrificing comprehension of the architecture the decision represents and the decision interconnectivities. An effective way to sort and analyze decision information is to represent the decisions visually using graphs. Decisions are represented as nodes and the relationships are directed edges, as shown in Fig. 1b. Viewing design as the result of applying a set of design decisions, the visualization may display decisions, their attributes, and their relationships separate from the architectural components, that is, a "decision-only" view of the software architecture.

A significant benefit for graphical structure visualization is its cognitive assistance, in the context of helping people create a mental map of the decisions. Other benefits include the ability to detect missing or orphaned decisions that may denote design incompleteness and the preservation of decision contexts in relation to one another.



(a) (b)

Fig. 1. Screenshots of our visualization tool implementing the tabular listing and graphical structure visualization: (a) Decision list showing the current set of design decisions, (b) structure visualization of a set of design decisions and their relationships as a directed graph



(a) (b)

Fig. 2. Screenshots of our visualization tool implementing the chronological and decision impact visualization aspects: (a) Chronological view showing two decision activity periods within a five-month interval, (b) decision impact view of a design decision (center node of the concentric node circles). Nodes represent decisions and edges represent decision impact-relationships.

Decision Chronology Visualization. The goal of this aspect is to increase understanding of the architecture's dynamic nature. Software design changes over time, so the design decisions made will also change. The visualization should handle the evolution of the design decisions by supporting versioning and decision states. Links to previous decision versions show design progression and traceability.

Keeping track of the history of the changes would better explain the architectural story and reasons behind the design. Moreover, a timeline view would display decisions that were created or modified during a specified time interval. This would be beneficial in periodic design reviews by determining what has changed since the last review or by identifying which areas are still being actively designed. Fig. 2a shows how the visualization highlights decision-making sessions via decision clusters.

Decision Impact Visualization. The goal of this aspect is to increase the understanding of the architecture's dependencies on its set of design decisions. This visualization helps to visually identify the impact decisions have on each other using decision relationships and properties. More concisely, the impact visualization aspect utilizes the traceability

provided by the decision attributes represented in the structural aspect to create a potential impact matrix upon which software architects, designers, and developers draw conclusions. Fig 2b visualizes this matrix by showing how one decision (in the center) can directly impact (innermost concentric node circle) or indirectly impact decisions to the n^{th}-degree (outer concentric circles).

3 Conclusions

The four proposed aspects to visualize software architectural design decisions provide a means to explore and analyze large sets of decisions. Visualization helps people create mental maps of the decisions and it highlights information that is not directly visible. Moreover, each visualization aspect can highlight or reveal different decision attributes to support specific tasks, such as risk analysis, system cleanup, or other uses of architectural decisions [9]. Although we identified four aspects, these aspects do not cover all situations and there may be other aspects that could reveal more information provided by a set of decisions or reveal them more effectively. Using various decision representation models or discovering other visualization aspects should be investigated to enhance support for architectural decision exploration and analysis.

References

1. Bosch, J.: Software Architecture: The Next Step. In: Oquendo, F., Warboys, B.C., Morrison, R. (eds.) EWSA 2004. LNCS, vol. 3047, pp. 194–199. Springer, Heidelberg (2004)
2. Duenas, J.C., Capilla, R.: The Decision View of Software Architecture. In: Morrison, R., Oquendo, F. (eds.) EWSA 2005. LNCS, vol. 3527, pp. 222–230. Springer, Heidelberg (2005)
3. Jansen, A., Bosch, J.: Software Architecture as a Set of Architectural Design Decisions. In: Proc. 5th Working IEEE/IFIP Conference on Software Architecture (WICSA 2005), pp. 109–120. IEEE Computer Society, Pittsburgh (2005)
4. Kruchten, P.: An Ontology of Architectural Design Decisions. In: Bosch, J. (ed.) Proc. 2nd Groningen Workshop on Software Variability Management, pp. 55–62. Rijksuniversiteit Groningen, Groningen (2004)
5. Kunz, W., Rittel, H.W.J.: Issues as Elements of Information Systems, Working Paper 131. Institute of Urban and Regional Development. The University of California at Berkeley, Berkeley (1970)
6. Lee, L., Kruchten, P.: Customizing the Capture of Software Architectural Design Decisions. In: Proc. 21st Canadian Conference on Electrical and Computer Engineering, pp. 693–698. IEEE, Niagara Falls (2008)
7. Lee, L., Kruchten, P.: A Tool to Visualize Architectural Design Decisions. In: Becker, S., Plasil, F. (eds.) QoSA 2008. LNCS, vol. 5281, pp. 43–54. Springer, Heidelberg (2008)
8. Tyree, J., Ackerman, A.: Architecture Decisions: Demystifying Architecture. IEEE Software 22(2), 19–27 (2005)
9. van der Ven, J.S., Jansen, A.G.J., Avgeriou, P., Hammer, D.K.: Using Architectural Decisions. In: Hofmeister, C., Crnković, I., Reussner, R. (eds.) QoSA 2006. LNCS, vol. 4214, pp. 1–10. Springer, Heidelberg (2006)