

ADDSS: Architecture Design Decision Support System Tool

Rafael Capilla¹, Francisco Nava¹, Jesús Montes¹, Carlos Carrillo²

¹*Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Madrid, Spain*

²*Departamento de Ingeniería y Arquitecturas Telemáticas, Universidad Politécnica de Madrid*
{rafael.capilla, francisco.nava@urjc.es, jmontes22@gmail.com, ccarrillo@diatel.upm.es}

Abstract

This paper describes the ADDSS tool which enables capturing and documenting architectural design decisions in order to avoid knowledge vaporization.

1. Introduction

Recently, the software architecture research community has faced the need to document the design decisions that led to any software architecture [2]. Design decisions can be defined as “*the information that explains the reasons by which a design choice is made, often based on a set of requirements or user’s needs*”. In addition, design rationale describes the reasons and the assumptions for making a particular decision. Compared to other artifacts like code, design decisions can explain the reasons for selecting a particular architecture during architecture evaluation. One of the major benefits for capturing such knowledge is to reconstruct past decisions and avoid knowledge vaporization, as well as to reduce the maintenance effort spent in, for instance, recovering architecture descriptions. Because capturing knowledge is hard and time consuming, two main strategies can be used: *codification* aims to store the key relevant information of a given domain or area of expertise; *personalization* is more focused in capturing and producing knowledge that can be tailored to specific user’s needs. Our ADDSS approach mainly uses a codification strategy but some kind of personalization is allowed on behalf of the use of optional attributes [4] that can be selected by the user. In general, knowledge management tools can be used to store architectural knowledge but certain features specific to architecting processes has led to the development of a set of research prototype tools, like Archium [7], AREL [9], and PAKME [1]). Other tools like Compendium, ArchStudio, and SoftArch provide less support for capturing design decisions (e.g.: only

Compendium support the concept of design decision [6]). To advance in the state of the art, we describe the Architecture Design Decision Support System (ADDSS) [3], a web-based tool for capturing and documenting architectural design decisions. This tool relies on the ideas described in “decision view” [5], which considers the information of design decisions as a new architecture view which crosscuts the other traditional architecture views [8].

2. ADDSS Approach

Recording, using, managing, and documenting architectural design decisions are, among others, new complementary activities that should be carried out in parallel to typical architecture modeling tasks. The challenge to transfer the expertise of software architects, which often remains tacit, into explicit documented knowledge motivated the construction of ADDSS, which includes the following features.

(i) *Capturing design rationale*: Design decisions are captured in ADDSS in a reusable to be used afterwards if needed. Our tool provides a specific template to capture the items that describe a design decision with its rationale. A second template is used to support the personalization approach and captures additional items that can be selected by the user as optional information according to its preferences. This twofold approach makes more flexible and agile the effort spent in recording the design decisions.

(ii) *Existing architectural knowledge*: Architectural knowledge encompasses design decisions plus designs, but also well-known patterns and architectural styles may fall in this category. ADDSS provides a repository of patterns as proven design solutions.

(iii) *Explicit relationships to other products of the software lifecycle and between decisions*: During the capturing of the decision, the user can select the requirements that motivated the decisions, and explicit

links from decisions to requirements are established. After a set of decisions are made, the user can upload an architecture which is the result of those decisions, and ADDSS defines also an explicit link between those decisions and the architecture uploaded. All these links can be used in maintenance for trace purposes, in order to, for instance, know the root causes of changes or to estimate change impact analysis when requirements change. In addition, users can establish dependencies between decisions, which may adopt the form of a network. These links are documented explicitly in ADDSS and they can be used to estimate the impact of adding, modifying, or removing a decision.

(iv) *Iterative construction process*: Because software architects build their architectures as a set of successive refinements, ADDSS simulates the same activity to show the evolution of architectures. In addition, we support different architecture views.

(v) *Explicit documented knowledge*: ADDSS documents the decisions and architectures as PDF reports that are generated automatically.

Figure 1 shows the deployment diagram describing the main parts and technologies used in ADDSS. We used a client-server approach to deploy ADDSS. The client side uses AJAX and XML technologies to improve the visualization of the architectures. In the server side, PHP dynamic pages access a MySQL database to store the information requirements, architectures and decisions. The *gd2.dll* library is used to generate the thumbnail images of the architectures and the *fpdf* library is used to produce the on-line PDF reports from the database. In addition, MD5 encryption functions secure user passwords, while multilingual support is managed by PHP session variables.

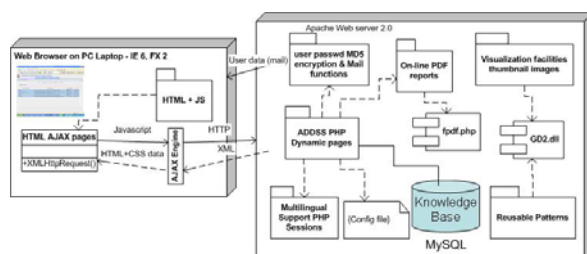


Figure 1. ADDSS architecture and technologies used.

3. Conclusion

At present, we have carried out several case studies to prove the capabilities of ADDSS for capturing and documenting architectural design decisions. Two experiences were enacted in the Rey Juan Carlos University (one with 22 students and the other with

two software architects [3]) for recording the decisions of the architecture of a virtual reality system. A third experience was done in conjunction with researchers from the Fraunhofer IESE (Kaiserslautern, Germany) during June 2007, to store the decisions of a set of architectures recovered with a reverse engineering tool (i.e.: SAVE). ADDSS was proven to be useful for capturing such decisions but people from the Fraunhofer observed that to scale-up ADDSS to industrial applications, they recommended to extend the multi-user management features and a way for resolving overlapping or incompatible decisions. Our initial results seem promising and we expect savings in further maintenance and evolution activities.

4. References

- [1] Ali- Babar, M. A. and Gorton, I. A Tool for Managing Software Architecture Knowledge. Proceedings of the 2nd Workshop on Sharing and Reusing Architectural Knowledge, ICSE Workshops, IEEE DL (2007).
- [2] Bosch, J. Software Architecture: The Next Step, Proceedings of the 1st European Workshop on Software Architecture (EWSA 2004), Springer-Verlag, LNCS 3047, pp. 194-199 (2004).
- [3] Capilla, R., Nava, F., Pérez, S. and Dueñas, J.C. A Web-based Tool for Managing Architectural Design Decisions, Proceedings of the 1st Workshop on Sharing and Reusing Architectural Knowledge, ACM Digital Library, Software Engineering Notes 31 (5) (2006).
- [4] Capilla, R., Nava, and Dueñas, J.C. Modeling and Documenting the Evolution of Architectural Design Decisions, Proceedings of the 2nd Workshop on Sharing and Reusing Architectural Knowledge, ICSE Workshops, IEEE DL (2007).
- [5] Dueñas, J.C. and Capilla, R. The Decision View of Software Architecture, Proceedings of the 2nd European Workshop on Software Architecture (EWSA 2005), Springer-Verlag, LNCS 3047, pp. 222-230 (2005).
- [6] Jansen, A., Boshj, J. Evaluation of Tool Support for Architectural Evolution, ASE 2004, 375-378, (2004).
- [7] Jansen, A., van der Ven, J., Avgeriou, P. and Hammer, D.K. Tool Support for Architectural Decisions, 6th Working IEEE / IFIP Conference on Software Architecture (WICSA 2007), pp. 4, (2007).
- [8] Kruchten, P., Architectural Blueprints. The “4+1” View Model of Software Architecture, IEEE Software 12 (6), pp.42-50 (1995).
- [9] Tang, A., Jin, Y. and Han, J. A Rationale-based Model for Design Traceability and Reasoning, Journal of Systems and Software 80, pp. 908-934, Elsevier (2007).