Architektur-Erkennung

14 Viewpoints in Software Architecture Reconstruction

Arie van Deursen
CWI & Delft Univ. of Technology, The Netherlands
Arie.van.Deursen@cwi.nl

Christine Hofmeister
Lehigh University, USA
hofmeister@cse.lehigh.edu

Rainer Koschke
University of Stuttgart, Germany
koschke@informatik.uni-stuttgart.de

Leon Moonen
Delft Univ. of Technology & CWI, The Netherlands
Leon.Moonen@computer.org

Claudio Riva
Nokia Research Center, P.O. Box 407, FIN-00045, Helsinki, Finland
claudio.riva@nokia.com

14.1 Introduction

Many software engineering tasks are hard to conduct without relevant architectural information (e.g., migrations, auditing, application integration, or impact analysis). Unfortunately, architectural information, if available at all, is often outdated, incorrect, or inappropriate.

Software architecture reconstruction is the process of obtaining a documented architecture for an existing system. Although such a reconstruction can make use of any possible resource (such as available documentation, stakeholder interviews, domain knowledge), the most reliable source of information is the system itself, either via its source code or observations on its execution.

It is widely accepted that architectures must be described by multiple views. A *view* is a representation of a whole system from the perspective of a related set of concerns [3]. Prominent views are the 4+1 views by Kruchten [4] or the Siemens views [2]. The recent book by Clements and colleagues [1] and the *IEEE Recommended Practice for Architectural Description of Software-intensive Systems* [3] give a larger catalog of architectural views.

Previous research in architecture reconstruction has focused on recovering a single architectural view or a few preselected views. The application of these techniques usually involves three steps: extract raw data from the source, apply the appropriate abstraction technique, and present or visualize the information obtained. These steps are specific to the views to be reconstructed.

Unfortunately, there is no set of "standard views"that fits all purposes of an architectural description, so that the applicability of these techniques is limited in scope. In recognition that views depend upon the specific purpose of an architectural description and these purposes may be very diverse in practice, an architecture reconstruction method should treat views as first-order elements.

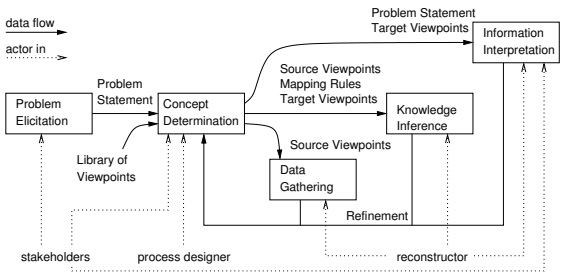

Figure 14.1: Interaction during reconstruction design.

Filling this gap, in this paper we describe Symphony, a process framework that has an explicit step for the discovery of the views that should be reconstructed in order to solve the problem at hand. Symphony is view-based

in recognition of the importance of multiple architectural views not only in presenting architecture but more fundamentally in defining the reconstruction activities. Symphony[1] is the result of a systematic analysis of our own experiences in software architecture reconstruction, cases conducted by close colleagues, and the various approaches that have been published in the literature. Symphony provides a conceptual framework that helps researchers by providing a unified approach to reconstruction, with consistent terminology and a basis for improving, refining, quantifying, and comparing reconstruction processes and case studies.

14.2 The Symphony Framework

Symphony consists of two stages. The first stage (Problem Elicitation and Concept Determination) produces a repeatable and reusable reconstruction strategy that creates the views necessary to address the original problem. This procedure may be useful beyond the scope of the current reconstruction: it can play a role in continuous architecture conformance checking and in future reconstructions. Although not an ultimate goal, the problem-dependent types of views created or refined in the Concept Determination phase are another reusable output of this stage.

The second stage of Symphony concerns the execution of the reconstruction strategy. Its outcome is the foundation for addressing the problem for which the particular reconstruction is carried out. A secondary outcome is the sequence of mappings from the source views (those extracted from the system's artifacts) to the target views (those that address the problem at hand). This sequence allows one to trace back the information in the views to the artifacts from which they were derived.

Typically the two stages are iterated: Reconstruction execution reveals new reconstruction opportunities, which lead to a refined understanding of the problem and a refined reconstruction design. The underlying types of source and target views and the mapping rules evolve throughout the process.

Reconstruction Design The Reconstruction Design is devided in two steps: *Problem Elicitation* analyzes the problem triggering the reconstruction and involves all stakeholders. Once the problem is understood, the Concept Determination step is used to determine the architectural information needed to solve the problem and the way to derive this information. In this step, the architect is a process designer, defining the architectural reconstruction that will take place in the Reconstruction Execution.

The reconstruction activities are defined in terms of the views they deal with: A *source view* is a view of a system that can be extracted from artifacts of that system, such as source code, build files, configuration information, documentation, or traces. A *target view* is a view of a software system that describes the as-implemented architecture and contains the information needed to solve the problem for which the reconstruction process was carried out.

Views are specified by so-called viewpoints. In IEEE 1471, a *viewpoint* describes the rules and conventions used to create, depict, and analyze a view based on this viewpoint [3]. A view conforms to a viewpoint. While a view describes a particular system, a viewpoint specifies the kind of information that can be put in a view and is independent of any particular system.

In the Concept Determination activity, the viewpoints for the target and source views are selected or defined, and the mapping rules from source to target views are designed. The mapping rules are ideally a formal description of how to derive a target view from a source view. Realistically, parts will often be in the form of heuristics, guidelines, or other informal approaches. If a mapping can be completely formalized, the reconstruction can be fully automated. This is not typically possible for software architecture, thus we expect the mapping to contain both formal and informal parts.

Reconstruction Execution The *Reconstruction Execution* stage (cf. Figure 14.2) operates only at the level of views constrained by the viewpoints created before.

The goal of the *Data Gathering* step is to collect the data that is required to recover selected architectural concepts from a system's artifacts through static or dynamic analyses. *In Knowledge Inference*, the reconstructor applies the mapping rules to populate the target views by condensing the low-level details of the source view and abstracting them into architectural information. The mapping rules and domain knowledge are used to define a map between the source and target view.

In the *Information Interpretation*, conclusions are drawn from the reconstructed views. These conclusions then lead to measures to be taken to remedy the problem. To this end, the target views need to be made accessible both physically and mentally to all stakeholders.

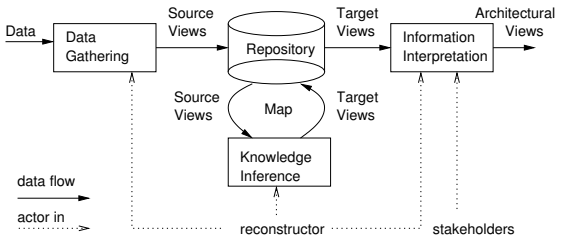

Figure 14.2: Reconstruction execution interactions.

14.3 Concluding Remarks

Symphony incorporates the state of the practice, where reconstruction is problem-driven and uses a rich set of architecture views. It has been applied by the authors in academic and industrial case studies and unifies other existing reconstruction techniques and methods.

Viewpoint selection and definition is an important part

[1] The name Symphony reflects that a successful reconstruction is the result of the interplay of many different instruments. Moreover, the authors' collaboration in the area of software architecture reconstruction started in the music room of Castle Dagstuhl in Germany.

of the Symphony process. Using viewpoints to specify the input and output of an activity allows us to decompose the reconstruction process systematically and to review the outcome of each activity. In addition, we can reuse an activity as a building block to compose new reconstruction processes.

In addition, Symphony provides a common reference framework that can be used when classifying and comparing various techniques described in the literature. It helps us to find and demarcate research problems in software architecture reconstruction. For example, Symphony's viewpoint emphasis calls for a catalog of reconstruction methods, techniques, and experiences organized by viewpoints. Moreover, it raises the question what reconstruction-specific viewpoints exist. Symphony's inclusion of mappings between source and target views suggests finding a systematic way to discover and describe such mappings as a key research question. Problems like these are hard to tackle. Symphony makes it possible to address them on a case-by-case basis, offering its process model as a way to classify and compare results.

Bibliography

[1] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, 2002.

[2] C. Hofmeister, R. Nord, and D. Soni. *Applied Software Architecture*. Object Technology Series. Addison Wesley, 2000.

[3] IEEE P1471-2000. IEEE recommended practice for architectural description of software-intensive systems, 2000.

[4] Phillipe Kruchten. The 4+1 view model of architecture. *IEEE Software*, 12(6):42–50, November 1995.