



State of the Art Reports

An Overview of the State of the Art in Software Architecture

Dewayne E. Perry
Bell Laboratories
Murray Hill, NJ 07974
dep@research.bell-labs.com

Introduction

The problem of software architecture has long been a concern for those building and evolving large software systems. Aspects of software architecture are included in both system architecture and requirements specification. Typically, the software architecture is in place before the requirements are passed on to the system developers for implementation. The architecture functions as the overall structure within which the requirements are to be met.

More recently software architecture has become an important topic in software engineering research. This research ranges from investigating how to describe and analyze software architectures to codifying components, structures, styles and patterns as well as proposing novel ones.

In this mini-tutorial state of the art report on software architecture, I will cover the following topics:

- rationale for a separate discipline,
- models of software architecture and style,
- architectural specifications and their uses,
- codification of architectures, styles and elements,
- architectural processes, and
- where next?

Rationale for a separate discipline

Prior to the emphasis on architecture as a separate artifact in the software life-cycle, the consideration of architecture was found as part of the high level design.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee

ICSE 97 Boston MA USA

Copyright 1997 ACM 0-89791-914-9/97/05 ..\$3.50

Indeed, there are those who still think that that this is where it should be in the development process.

I will present what I consider to be the difference between architecture and design and discuss the rationale for software architecture as a separate artifact and a separate process entity.

Models of software architecture and styles

Underlying any discussion of software architecture is a model of what software architecture is. This model may be either explicit or implicit. To properly address the various issues of description, analysis and generation, we need to understand what these underlying models are. I will delineate some of the models that are being used in current software architecture work.

One defining difference among these models is whether an architectural specification is considered to be a description of an architecture or a prescription for an architecture. Each has different implications for both the architecture itself as well as for the architectural processes.

Architectural specifications and their uses

One of the important threads of current research is how to describe (or prescribe) software architectures. Part of how that problem is approached depends on what use the description (or prescription) is meant to support. The standard approach in practice is much the same for architecture as it is for all other development artifacts except code: informal prose (but possibly structured) documents.

I will discuss various architectural specification approaches and the kinds of uses that are intended or supportable with these approaches. Of interest are the kinds of analyses that can be performed on the architecture. Equally important are the uses of architectural descriptions for the generation or configuration of the

defined systems or system families.

Codification of architectures, styles and elements

One of the strengths of the research in software architectures, architectural styles, and design patterns is the attempt to standardize and codify various architectures and architectural elements. One goal of this work is the establishment of a handbook that characterizes the various elements, styles, patterns and architectures according to a number of important dimensions that must be considered by any practicing software architect in crafting an architecture appropriate to the business, project and user demands.

I will summarize the current state of this thread of research and practice.

Architectural processes

Without new processes to support and guide these new approaches of architectural-driven system construction and evolution, we will more likely than not be ineffective in realizing the potential inherent therein. I will discuss three related processes:

- architecture recovery,
- domain specific architectures, and
- product line architectures.

The discussion will cover the processes themselves as well as the technological and organizational structures that are needed to support them.

Where next?

I will close with a look at where the field is going in terms of both practice and research and what problems need to be solved in research for practice to move forward.

Viewgraphs available

Because of final paper deadline for the proceedings there was not enough time to write anything more than the overview printed here. The full paper and the viewgraphs will be found at my web site:

www.bell-labs.com/user/dep/work/swa/