



Software Architecture in Action

Flavio Oquendo, Jair C Leite, Thais Batista



Chapter 12

Introduction

Learning outcomes of this chapter

- You will learn:
 - the concept of architectural style;
 - the introduction to four architectural styles that will be addressed in the next chapters:
 - pipe-filter,
 - client-server,
 - feedback control loop, and
 - blackboard.

The structure of this chapter

- What is an Architectural Style
- What is a Style-based Architecture
- Architectural Styles
- Summary



What is an Architectural Style

What is an Architectural Style

Conceptual Overview – 1/2

- An architectural style can be seen as a **collection of principles that shape the design of a software architecture** to achieve a set of related quality attributes.
- Operationally, an architectural style can be seen as a **collection of architectural design decisions and constraints on architectural decisions** to achieve beneficial qualities in the resulting software architecture.

What is an Architectural Style

Conceptual Overview – 2/2

- An architectural style is expressed by a **vocabulary of design elements** in terms of component types and connector types and a configuration type determining the allowed compositions of components and connectors.
- Note that an architectural style **has an associated semantic interpretation** supporting defined analyses of architectures designed in the style.

What is an Architectural Style

Example – 1/2

- In a pipeline architectural style a sequence of components is chained together by their standard streams, so that the output of each process feeds directly as input to the next one.
- Unix-like shells use the **pipeline architectural style** for composing commands, where the output of a command is piped to the input of the next command and so using standard output and input supporting the quality attribute of composability of commands.

What is an Architectural Style

Example – 1/2

- In the case of Unix-like shells these components are the called **commands** and the connectors are the **pipes**, represented in the shells by the ‘|’ character.
- For instance, to list files having “architecture” in their names in a current directory, we can use the command *ls* to output the list of all filenames then use the command *grep* to select only the lines of the *ls* output containing the string “**architecture**”, and use the command *less* to display the result in a scrolling page, which gives in the command line of a terminal: **ls -l | grep architecture | less.**



What is a Style-based Architecture

Style-based Architecture

Conceptual Overview

- The application of architecture styles in the design of software architectures **allows to achieve expected qualities more easily.**
- An architectural style **guides how to organize the components** of an architected system so that one can design the complete architecture and achieve the qualities intrinsic to the style.
- A style-based architecture is thereby defined as **an architecture description that has been designed to cope one or a combination of different architectural styles.**

Style-based Architecture

Example – 1/2

- In our running example of the RTC System, the architecture that was designed in the part I of the book **was designed without following a specific style.**
- In part II we introduced quality-based architectures, i.e. an **architecture that is designed to satisfy quality attributes using specific tactics** (see for instance Chapter 9 for tactics to achieve modifiability).

Style-based Architecture

Example – 2/2

- In this part III, we enhance the approach of designing quality-based architectures by the **one of style-based architectures**
 - which **allows to achieve different qualities** reusing architectural decisions that are embodied in a defined architectural style.
 - For instance, by designing architectures according to the pipeline style, **the quality attribute of modifiability is granted as pipelines imply modifiability.**



Architectural Styles

Architectural Styles

Pipe-Filter Overview – 1/2

- This Part III focuses on discussing four architectural styles: Pipe-Filter, Client-Server, Feedback Control Loop, and Blackboard.
- The ***Pipe-Filter style*** allows a sequential processing of data. Components are “filters” that read streams of data on its inputs and produces streams of data on its outputs. Connectors are “pipes” that transmits the output streams of one filter to inputs of another.

Architectural Styles

Pipe-Filter Overview – 2/2

■ In Chapter 13

- we discuss the structural and behavioral specification of the **Pipe-Filter style**.
- we present an **example of the Pipe-Filter style** in an architecture of the RTC System with distributed temperature sensors with no direct communication with the controller. Instead, they are connected with their adjacent sensors in a sequence.
- we also present the **Pipeline substyle** where all filters are sequentially connected.

Architectural Styles

Client-Server Overview

- The ***Client-Server style*** allows components, called “clients”, to send requests to a component, called “server”, and wait for a reply.
- In Chapter 14
 - we discuss **the structural and behavioral specification** of the Client-Server style.
 - we present **an example of the Client-Server in the RTC System** with a client that requests to the server (the controller) the average temperature.

Architectural Styles

Feedback Control Loop Overview – 1/2

- The ***Feedback Control Loop style*** allows a central component to control several actuators by analyzing information from sensors.
- In Chapter 15
 - we discuss the **structural and behavioral specification** of the Feedback Control Loop style.

Architectural Styles

Feedback Control Loop Overview – 1/2

- In Chapter 15 (continued)
 - we present **an example of the Feedback Control Loop style in an architecture of the RTC System** with three types of components (sensors, controller, actuators) and two types of connectors, characterizing a feedback loop where the controller uses sensors data from the environment to command actuators to act in the environment. This is a continuous loop where the actuators change the environment that will be then sensed by the sensors.

Architectural Styles

Blackboard Overview

- The ***Blackboard style*** defines a shared data structure (the blackboard) and multiple knowledge sources that interact with each other via the blackboard.
- In Chapter 16
 - we discuss the **structural and** behavioral specification of the Blackboard style.
 - we present an example of **the Blackboard style in an architecture of the RTC System** where the sensor monitor is a blackboard that receives data from sensors.

Summary

- In this chapter, you learnt
 - the Architectural Style concept
 - what is a Style-based Architecture
 - what are Architectural Styles

For Further Reading

- Clements, P., Bachmann, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R.: Documenting Software Architecture: Views and Beyond. SEI Series in Software Engineering (2003)
- Buschmann, F., Meunier, R., Rohnert, H. Sommerlad, P., Michael Stal, M.: Pattern-Oriented Software Architecture Volume 1: A System of Patterns, vol. 1. Wiley (1996)
- Vogel, O., Arnold, I., Chughtai, A., Kehrer, T.: Software Architecture: A Comprehensive Framework and Guide for Practitioners. Springer (2011)