

CHAPTER 4: PATTERNS AND STYLES IN SOFTWARE ARCHITECTURE

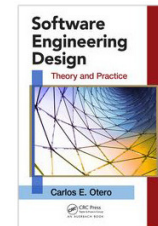
SESSION I: OVERVIEW AND HISTORY OF STYLES AND PATTERNS

Software Engineering Design: Theory and Practice

by Carlos E. Otero

Slides copyright © 2012 by Carlos E. Otero

For non-profit educational use only



May be reproduced only for student use when used in conjunction with *Software Engineering Design: Theory and Practice*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information must appear if these slides are posted on a website for student use.

SESSION'S AGENDA

- Overview of Architecture Styles and Patterns
- History of Architectural Styles and Patterns
 - ✓ Origin of styles and patterns
- Classification of Architectural Styles and Patterns
 - ✓ Data-Centered
 - ✓ Data flow
 - ✓ Distributed
 - ✓ Interactive
 - ✓ Hierarchical
- What's next...

OVERVIEW OF ARCHITECTURE STYLES AND PATTERNS

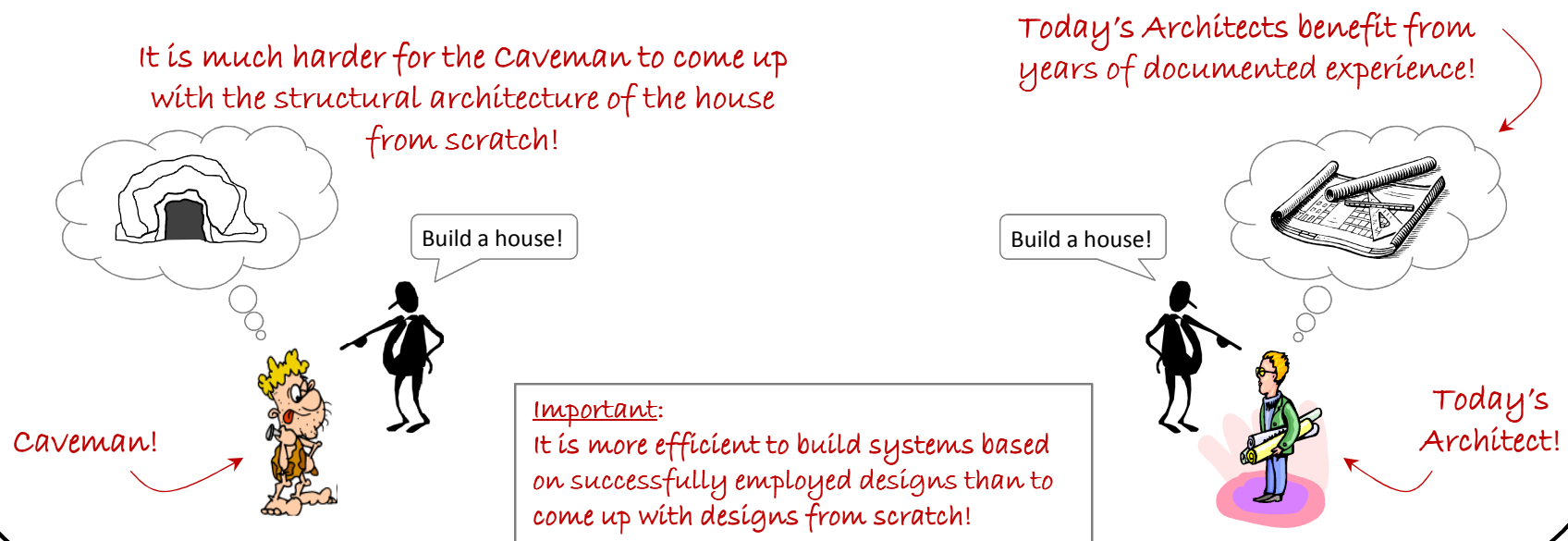
- In the previous module, it was established that software systems need to be carefully architected and evaluated from different perspectives.
 - ✓ This is necessary to address multiple concerns that shape the quality of the system from different stakeholders with different backgrounds.

- In this module, we pay special attention (mostly) to the *logical view* of software architecture.
 - ✓ That is, the perspective that deals with decomposing the software system into logical components that represent the structural integrity that supports functional and non-functional (quality) requirements.

- When designing logical architectures, it is important to use past experience to discover overall strategies that have been used successfully in the development of software systems.
 - ✓ To this end, the concepts of *architectural styles* and *architectural patterns* have emerged as mainstream approach for achieving reuse at the architectural level.

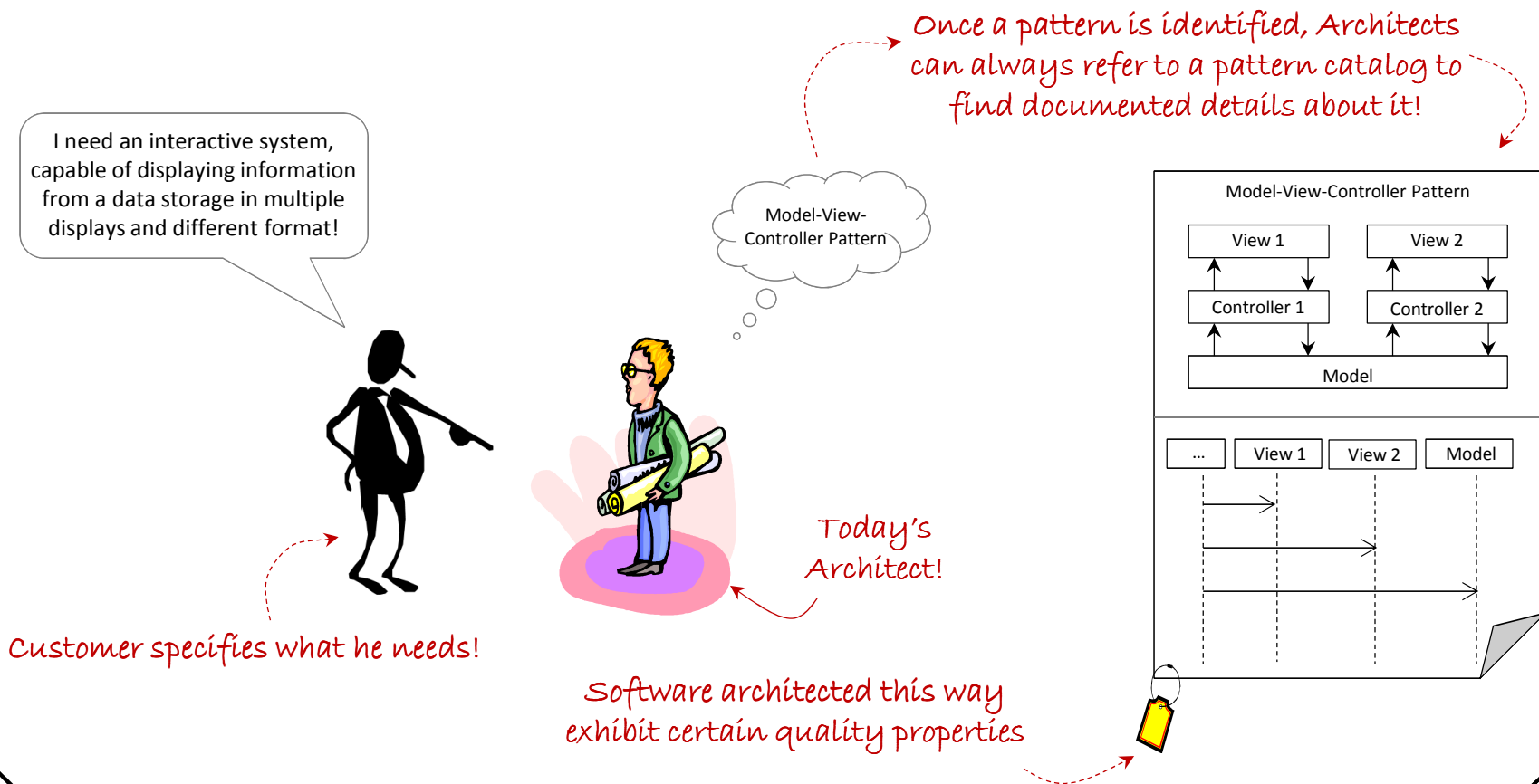
OVERVIEW OF ARCHITECTURE STYLES AND PATTERNS

- Architectural styles and architectural patterns provide generic, reusable solutions that can be easily understood.
 - ✓ These can be easily applied to new problems requiring similar architectural features.
- Decisions based on architectural styles and patterns benefit from years of documented experience that highlights
 - ✓ The solution approach to a given problem.
 - ✓ The benefits of these approaches.
 - ✓ The consequences of employing these approaches.



OVERVIEW OF ARCHITECTURE STYLES AND PATTERNS

- Similar to the previous example, today's software architect can benefit from numerous documented styles and patterns for software architecture.



HISTORY OF ARCHITECTURE STYLES AND PATTERNS

- Before we move on, it is important to discuss the history of styles and patterns. This will help us eliminate some of the confusion between the terms *styles* and *patterns*.
- In 1977, Christopher Alexander presented a language intended to help individuals, or teams, design quality structures of different sizes, shapes, and complexities [1]. According to Alexander et al.:
 - ✓ *“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”*
- Alexander’s work resulted in a catalogue of 253 patterns, each describing in detail the essential information required for documenting the patterns, including [1]:
 - ✓ Picture of the pattern
 - ✓ Context of the pattern
 - ✓ Problem that the pattern attempts to solve
 - ✓ Evidence for its validity
 - ✓ Related patterns

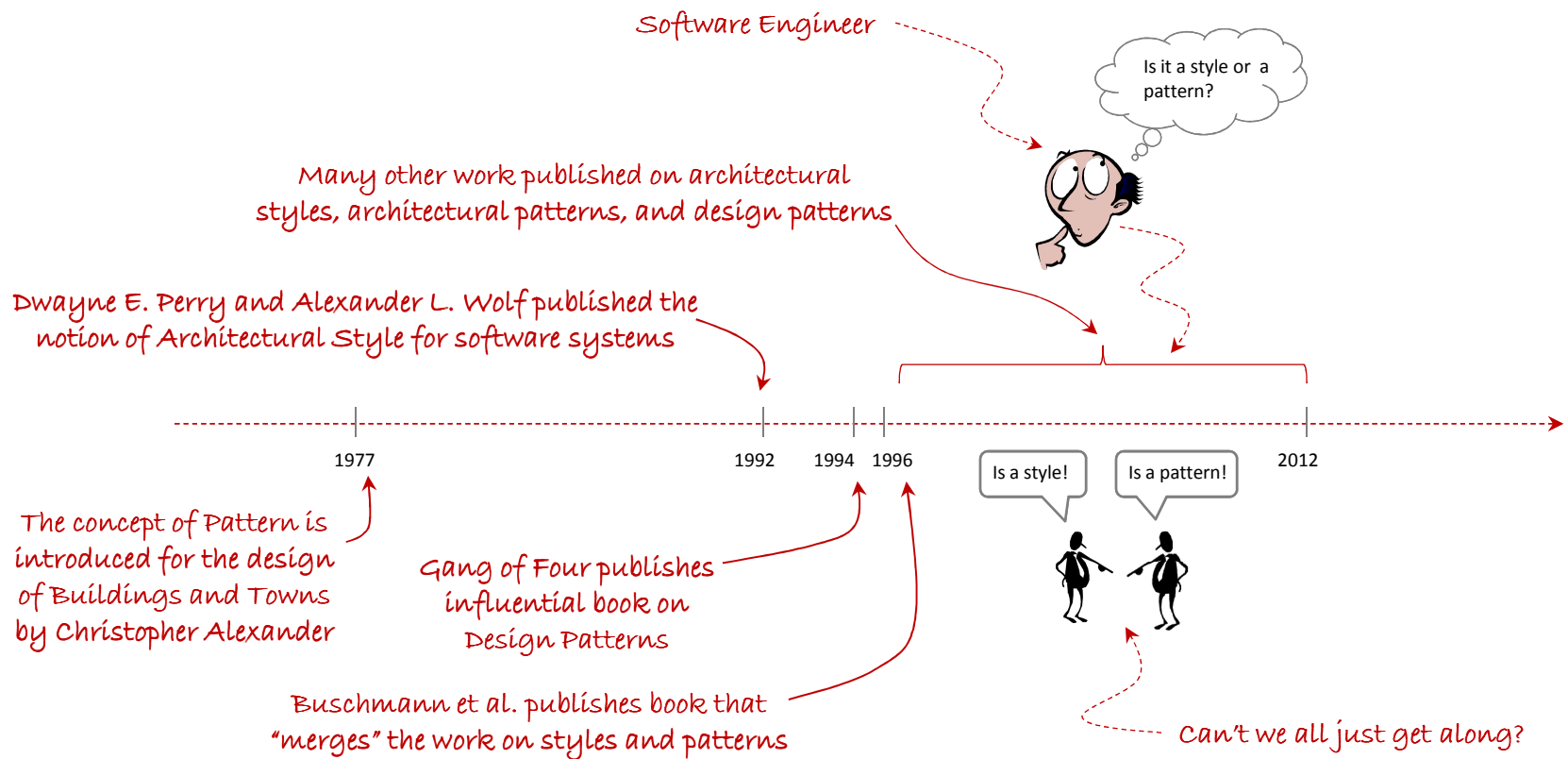
HISTORY OF ARCHITECTURE STYLES AND PATTERNS

- Although Alexander's work on patterns appears relevant to the software engineering profession, it actually referred to patterns found in the design of buildings and towns.
 - ✓ This work, however, significantly impacted the field of software engineering.
- In the 1990s, the software engineering community began researching and finding recurring high-level problem solutions in terms of specific elements and their relationships; these were originally referred to as **architectural styles** [2].
 - ✓ Similar to Alexander's work, Architectural Styles provided the means for software architects to reuse design solutions in different projects; that is, to use a "solution a million times over, without ever doing it the same way twice." [1]
- In 1994, Gamma, Helm, Johnson, and Vlissides—better known as the Gang of Four (GoF)—published their influential work that focused on a finer-grained set of object-oriented detailed design solutions that could be used in different problems "a million times over, without ever doing it the same way twice."
 - ✓ Influenced by Alexander's work, they called these **Design Patterns**.
 - ✓ Their work resulted in the creation of a catalogue of 23 (detailed design) patterns.
 - ✓ Each pattern was described in detail, using a specific pattern specification format.

HISTORY OF ARCHITECTURE STYLES AND PATTERNS

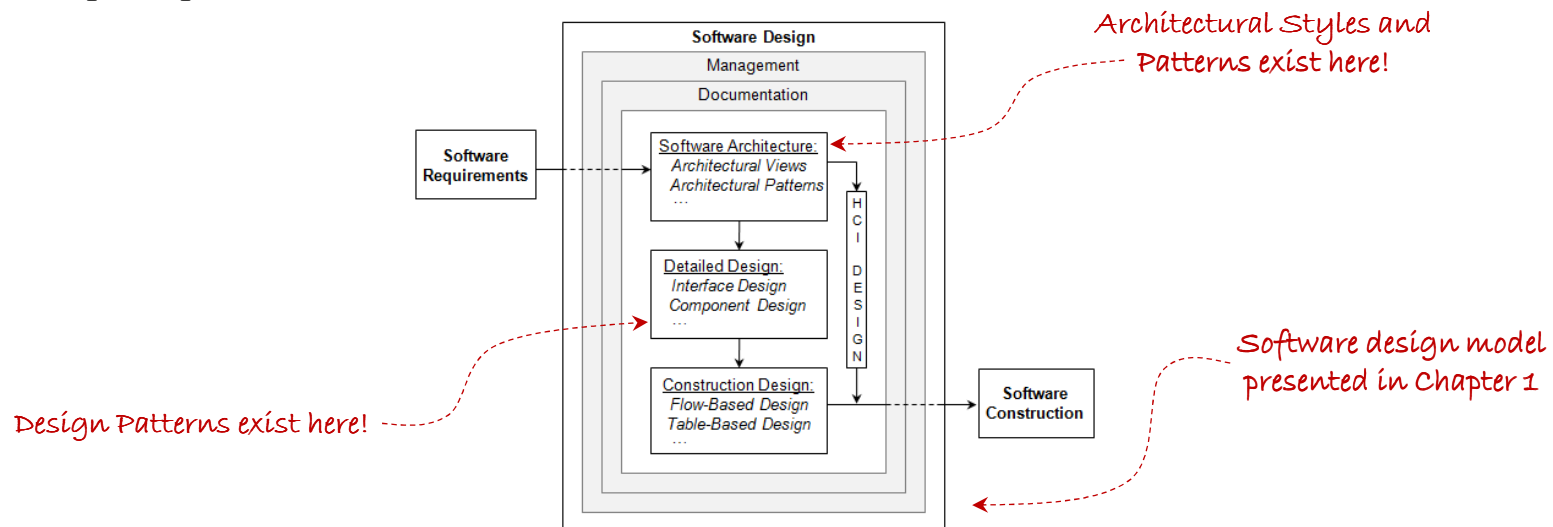
- In 1996, the work of Buschmann, Meunier, Rohnert, Sommerland, and Stal [3], integrated the work of architectural styles and design patterns by providing a set of well-known architectural styles using a pattern-like approach [2].
 - ✓ They referred to these as **Architectural Patterns**.
- In their work, Buschmann et al. provided their views about architectural patterns vs. design patterns[3]:
 - ✓ “*Architectural patterns ... specify the system-wide structural properties of an application. They specify the system-wide structural properties of an application, and have an impact on the architecture of its subsystems.*”
 - ✓ “Design patterns are medium-scale patterns.”
 - ✓ “*The application of a design pattern has no effect on the fundamental structure of a software system, but may have a strong influence on the architecture of a subsystem.*”
- Hopefully, this helps clear the air between the concepts of *architectural patterns* vs. *design patterns*, but what about *Architectural Styles* vs. *Architectural Patterns*?
 - ✓ There **are** documented differences between architectural styles and patterns!
 - ✓ However, perhaps Buschmann et al. [3] put it best by stating that “*Architectural styles are very similar to our architectural patterns. In fact every architectural style can be described as an architectural pattern.*”

HISTORY OF ARCHITECTURE STYLES AND PATTERNS



HISTORY OF ARCHITECTURE STYLES AND PATTERNS

- Today, the terms *architectural styles* and *architectural patterns* are used to convey fundamental structural and architectural organization for software systems.
 - ✓ Throughout the rest of the course, these terms are used interchangeably to denote reusable design solutions that occur during the *software architecture activity* of the design process.
 - ✓ The term *Design Patterns*, as seen later on in the course, is used to denote reusable design solutions that occur during the *detailed design activity* of the design process.
- Architectural styles and architectural patterns do not describe the detailed design of systems
 - ✓ They are used as basis for system decomposition and for analyzing the structure of systems in principled manner.



ARCHITECTURAL PATTERN CLASSIFICATION

- The choice of applying architectural patterns depend on the type of system, requirements, and desired quality attributes.
 - ✓ These characteristics help guide the choice of selecting one particular pattern over another.
- In some cases, more than one architectural pattern can be used in combination to collectively provide the appropriate architectural solution.
- Architectural patterns can be classified depending on the type of system as shown below:

Type	Description
Data-Centered	Systems that serve as a centralized repository for data, while allowing clients to access and perform work on the data.
Data Flow	Systems oriented around the transport and transformation of a stream of data.
Distributed	Systems primarily involve interaction between several independent processing units connected via a network.
Interactive	Systems that serve users or user-centric systems.
Hierarchical	Systems where components can be structured as a hierarchy (vertically and horizontally) to reflect different levels of abstraction and responsibility.

WHAT'S NEXT...

- In this session, we presented fundamental concepts of architectural styles and patterns, including:
 - ✓ Overview of Architectural Styles and Patterns
 - ✓ History of Architectural Styles and Patterns
 - Origin of styles and patterns
 - ✓ Classification of styles and patterns
 - ✓ Types of systems for classifying styles and patterns:
 - Data-Centered
 - Data flow
 - Interactive
 - Hierarchical

- In the next session, we will discuss two types of systems: Data-centered and Data Flow, together with essential architectural patterns for these systems, including:
 - ✓ Blackboard
 - ✓ Pipes-and-Filters

REFERENCES

- [1] Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977.
- [2] Clements, Paul, Rick Kazman, and Mark Klein. *Evaluating Software Architectures*. Santa Clara, CA: Addison Wesley, 2001.