

Patterns and Software Architecture (PASA)

Lecture II: Design Patterns

Ulrik Pagh Schultz, 14/2/2001

Where did patterns come from?

Christopher Alexander

- Observation 1: current architecture fails to meet real demands and to improve human condition.
- Observation 2: Some buildings possess a “quality without a name.”

Idea: capture this QWAN in patterns, and describe how to use these patterns.

I: The Timeless Way of Building (1975)

- Three-part solution:
 - “The quality:” essence of living and useful things; imparts freedom, wholeness, comfort, adaptability, . . . ; gives satisfaction; improves human condition.
 - “The gate:” mechanism to reach the quality; manifested as living common pattern language.
 - “The way:” how to apply patterns to obtain quality; ordered sequence of piecemeal growth.
- (Follow the way to pass through the gate to reach the quality.)
- Patterns used by architects, builders, and future inhabitants.

II: A Pattern Language (1977)

- Generative pattern language consisting of 253 patterns
- Covers regions, towns, transportation, homes, offices, rooms, lightning, gardens, building materials, . . .
- Examples: mosaic of subcultures, ring roads, pedestrian street, light on two sides of every room, window place, bed alcove, box columns, . . .

Example: “Pedestrian Street”

Pedestrian street ([APL-100])

Context: The simple social intercourse created when people rub shoulders in public is one of the most essential kinds of social “glue” in society.

Problem: This glue is largely missing, in part because much of the process of movement is taking place in indoor corridors and lobbies.

Solution: Pedestrian street. Arrange buildings so that they form pedestrian streets with many entrances and open stairs directly from the upper stories to the street, so that even movement between rooms is outdoors, not just movement between buildings.

What is a pattern?

1. A named nugget of structured information, which captures a proven solution to a recurring problem within a certain context amidst competing concerns.
2. Three-part rule, which expresses a relation between a context, a system of forces occurring in this context, and a certain configuration which allows these forces to resolve themselves.
3. Three-part rule, which expresses a relation between a context, a problem, and a solution; a pattern is both a relationship and an instruction to solve the problem; it is both a process and a thing.

What are the elements of a pattern?

Context: A situation giving rise to a problem.

Problem (forces): The recurring problem arising in the context; requirements, constraints, desirable properties.

Solution (consequences): A proven solution to the problem; how to balance the forces.

...

How do patterns work?

- How experts work:
 - Rely on experience from a large number of concrete problems
 - Recall a similar problem
 - Reuse the essence of its solution to solve the new problem
- How patterns work:
 - Document well-proven experience
 - Help novices to act as if they were experts
 - Support experts in the design of large-scale projects

What characterizes a pattern?

- An abstraction from a recurring concrete form.
- Solves a problem, is a proven concept, isn't an obvious solution, describes a relationship, has a significant human component.
- When is something a pattern? Must be recurring and both a process and a thing; a pattern supports itself and argues for why it's good.
- A pattern is where theory and practice meet.

But what about software?

Patterns in software

1987 Cunningham and Beck: “Using Pattern Languages for Object-Oriented Programming”

1990-1993 OOPSLA workshops on architecture

1991 Coplien: “Advanced C++ Programming Styles and Idioms”

1993 The Hillside Group: Beck, Cunningham, Johnson, Booch, Coplien

1994 Start of **PLoP** conferences, the **GoF** book

1995 First PLoP book

1996 Pattern-Oriented Software Architecture (**POSA**) [Buschmann et al]

1997 Analysis patterns

1998 Anti-patterns

What do patterns do for software?

Patterns do...

- ... document existing, well-proven design experience
- ... identify and specify higher level abstractions
- ... provide a common vocabulary and understanding
- ... help to document software architectures
- ... explicitly address non-functional properties

Patterns do not...

- ... provide an exact solution
- ... provide complete solutions
- ... solve all design problems

What are the components of a design pattern?

There are different pattern formats, but a pattern should contain the following components (in some form):

- Name
- Problem
- Context
- Forces
- Solution
- Examples
- Resulting context
- Rationale
- Related patterns
- Known uses

In addition, an abstract/short overview is advantageous.

What other design patterns?

GoF	POSA	...
abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, decorator, facade, flyweight, proxy	chain of responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor	whole-part, master-slave, proxy, command processor, view handler, forwarder-receiver, client-dispatcher- server, publisher-subscriber
		...

(7) Alexander and design patterns

1. How do Alexander's patterns compare to design patterns?
2. What happened to Alexander's work, and what are the implications for design patterns?

Alexander vs. design patterns [1/2]

Similarities

- Structure of the description:
 - context
 - problem (forces)
 - solution (consequences)
- Experience in action
- Recurring solutions to recurring problems
- Common vocabulary

Alexander vs. design patterns [2/2]

Differences

Alexander's patterns	Design patterns
Direct user involvement Vocabulary for users and designers "Beautiful places to live"	Indirect user involvement Vocabulary for designers "Beautiful program designs" Documentation

Evolution of Alexander's work

- Alexander's work has met with some challenges;
 - his ideas were never accepted in his community
 - experiments with his ideas only had limited success
- Primary reason: whole building culture must match
- Fortunately, software development is different from architecture!

Continuation: movie with C. Alexander at OOPSLA'96

