Introduction to Patterns

Baris Aktemur

CS 534| Ozyegin University

Software Patterns

- Automobile designers don't design cars using laws of physics.
  - They reuse standard designs with successful track records.
  - Starting from scratch typically isn't worth the cost.
- Software patterns: attempts to describe successful solutions to common software problems.

[Schmidt, Fayad and Johnson, Software Patterns, CACM, Oct. 1996/Vol. 39, No. 10]

Patterns

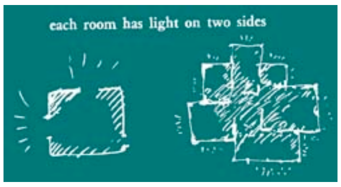
- Patterns teach useful techniques
- Patterns help people communicate better
- Patterns help people reason about what they do and why

[Schmidt, Fayad and Johnson, Software Patterns, CACM, Oct. 1996/Vol. 39, No. 10]

Design Pattern

- A general solution to a recurring problem
  - Christopher Alexander: "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."

LIGHT ON TWO SIDES OF EVERY ROOM

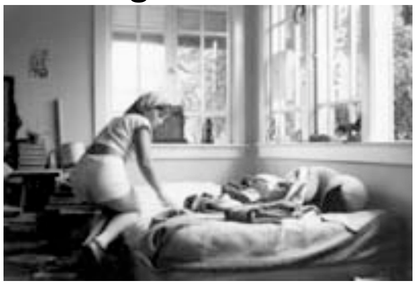


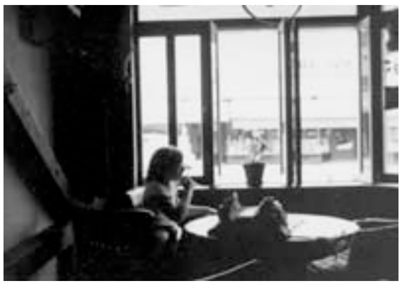
- When they have a choice, people will always gravitate to those rooms which have light on two sides, and leave the rooms which are lit only from one side unused and empty.
- Locate each room so that it has outdoor space outside it on at least two sides, and then place windows in these outdoor walls so that natural light falls into every room from more than one direction.

[C. Alexander, A Pattern Language, Oxford University Press, 1977]

LIGHT ON TWO SIDES OF EVERY ROOM

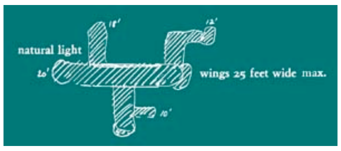
- With light on two sides ..... and without





[C. Alexander, A Pattern Language, Oxford University Press, 1977]

WINGS OF LIGHT



- Modern buildings are often shaped with no concern for natural light - they depend almost entirely on artificial light. But buildings which displace natural light as the major source of illumination are not fit places to spend the day.
- Arrange each building so that it breaks down into wings which correspond, approximately, to the most important natural social groups within the building. Make each wing long and as narrow as you can - never more than 25 feet wide.

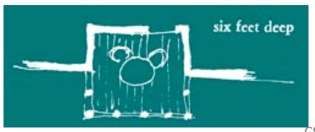
[C. Alexander, A Pattern Language, Oxford University Press, 1977]

WINGS OF LIGHT

- *A monster building - no concern for daylight inside.*

[C. Alexander, A Pattern Language, Oxford University Press, 1977]

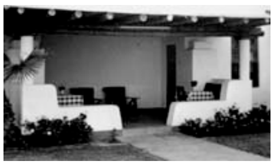
SIX-FOOT BALCONY



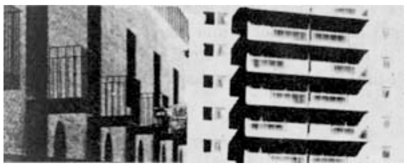
- Balconies and porches which are less than six feet deep are hardly ever used.
- Whenever you build a balcony, a porch, a gallery, or a terrace always make it at least six feet deep. If possible, recess at least a part of it into the building so that it is not cantilevered out and separated from the building by a simple line, and enclose it partially.

[C. Alexander, A Pattern Language, Oxford University Press, 1977]

SIX-FOOT BALCONY



Six-feet deep



Narrow balconies are useless

[C. Alexander, A Pattern Language, Oxford University Press, 1977]

Design Pattern

- A general solution to a recurring problem
  - Christopher Alexander: "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."
- Think about chess

"Early Bird" pattern

- Problem: I don't like wasting time waiting in a line.
- Solution: Move early before the line gets long.

trt.net.tr

dailymail.co.uk

Assignment: Your Patterns

- Any patterns from daily life that you can identify?
- Your pattern should have
  - a name
  - a description of a recurring problem
  - multiple examples of contexts in which the problem occurs
  - a solution to the problem

Design Patterns in OOP

- "... simple and elegant solutions to specific problems in OOSD."
- " ... capture solutions that have developed and evolved over time."
- "... for greater reuse and flexibility in software."
- Not for beginners
  - "Aha!" vs. "Huh?"

[Gamma, Helm, Johnson, and Vlissides. Design Patterns. Addison-Wesley, 1994*]*

Design Patterns in OOP

- Design is specific to a problem, but should also be general enough to apply to future problems.
- "Avoid or minimize redesign."
- "Find a good solution, use it again and again."
- *Isn't this what engineering is about?*

[Gamma, Helm, Johnson, and Vlissides. Design Patterns. Addison-Wesley, 1994*]*

Design Patterns

- So, the more patterns I have, the better my software design...
  - No. Number of design patterns used is **not** a quality metric.
- Top-down approach (novice programmer mistake):
  - "How can I fit this pattern into my design?"
  - Create artificial problems just to use patterns
- Bottom-up approach (expert programmer):
  - Identify the problem first
  - Then apply an appropriate pattern to solve the problem

Kinds of Patterns

- Architectural patterns
  - High level, involves subsystems
  - E.g. Layers
- Design Patterns
  - Medium level, involves classes and objects
  - E.g. Façade
- Idioms
  - Language dependent
  - E.g. Counted Pointers

GOF Book

- Gamma, Helm, Johnson, Vlissides
- A **catalog** of design patterns
- 23 patterns
  - "descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context."
- There are many, many other patterns
  - Parallel programming, object-oriented databases, distributed computing, ...
- The choice of the programming language

Some history

- http://c2.com/cgi/wiki?HistoryOfPatterns
- OOPSLA 87:
  - Kent Beck and Ward Cunningham. "*Using Pattern Languages for Object-Oriented Programs*", Workshop on the *Specification and Design for Object-Oriented Programming*
- 1990-1991:
  - Eric Gamma working on his PhD on ET++
  - Gamma and Helm meet at OOPSLA/ECOOP
- Hillside group: August 1993
  - Kent Beck and Grady Booch sponsored a meeting in Colorado
  - Ward Cunningham, Ralph Johnson, Ken Auer, Hal Hildebrand, Grady Booch, Kent Beck and Jim Coplien were there
- OOPSLA 94: GOF Book is out
- PLoP 95: Proceedings

Elements of a Pattern

- Name
  - A handle to describe the problem and the solution
  - A vocabulary of patterns
  - Short, descriptive **nouns** for effective communication
- Problem
  - Description of when to apply the pattern
  - Context
- Solution
  - The elements of the design, their relations and collaborators
  - Abstract
- Consequences
  - Trade-offs (e.g. space and time) in using the pattern

[Gamma, Helm, Johnson, and Vlissides. Design Patterns. Addison-Wesley, 1994*]*

Format of a pattern

- GOF format
- Siemens format
- Fowler format
- ...

Classification

- Purpose
  - Creational
    - concern the process of object creation.
  - Structural
    - composition of classes or objects.
  - Behavioral
    - interaction of classes or objects and distribution of responsibility.
- Scope
  - Class
    - Static (compile-time) relationships (i.e. inheritance)
  - Object
    - Dynamic, but involves static relations, too.

[Gamma, Helm, Johnson, and Vlissides. Design Patterns. Addison-Wesley, 1994*]*