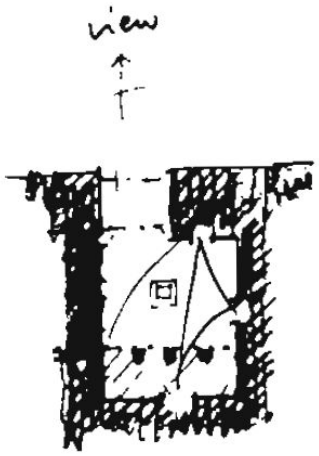
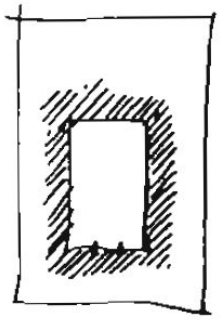
Design Patterns: HCI, UIs, Games



A living courtyard



Dead courtyard

Topics

- Motivation
- Design Patterns in UI Design
- Design Patterns in Game Design
- Closing Discussions

Design = Solutions

- Design is about finding solutions
- Unfortunately, designers often reinvent
  - Hard to know *how* things were done before
  - *Why* things were done a certain way
  - *How* to reuse solutions

How Can We Codify Design Knowledge?

- An effective and flexible design is difficult to get “right” the first time.
- Yet experienced designers do make good designs
  - New designers are usually overwhelmed by the all the design options available.
- Experienced designers evidently know something inexperienced ones don’t, what is it?

How Experienced Designers Solve a Problem

- Expert designers usually do ***not*** solve every problem from first principles, they ***reuse*** solutions that have worked for them in the past.
- When they find a good solution, they use it again and again.
- Such experience is part of what makes them experts.
- Such kind of experiences can be recorded as design patterns

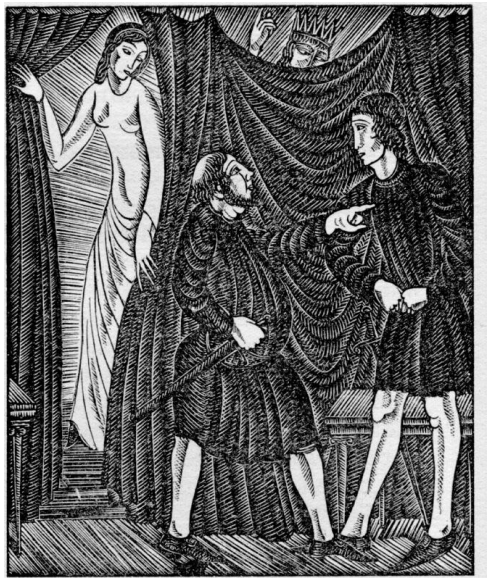
Good programmers code, great programmers steal

- “Good artists borrow (from other artists), but great artists steal !” **-Pablo Picasso**

- Compelling visual design takes practice and experience –a natural part of which is study and critique of other’s work

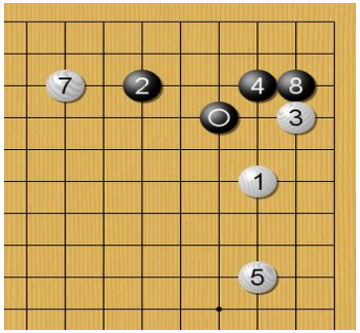
Novelists uses Patterns

- Novelists and playwrights rarely design their plots from scratch.
- They follow patterns like “Tragically Flawed Hero” (Macbeth, Hamlet, etc) or “The Romantic Novel” (countless romance novels)



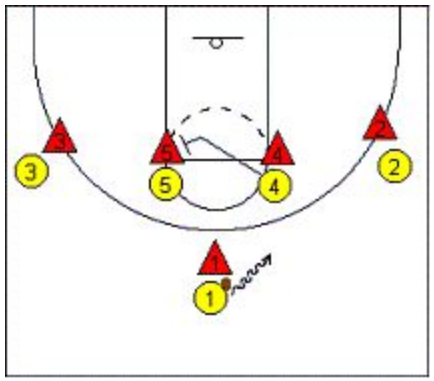
Game Players use Patterns

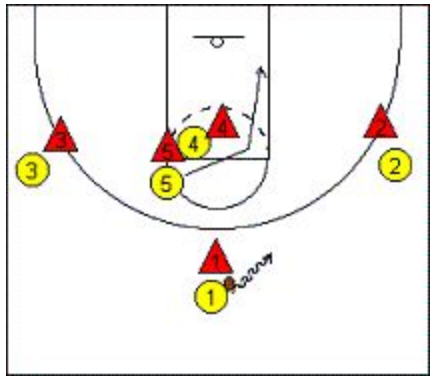
- Chess players, Go players, Basketball players all relay on “patterns”



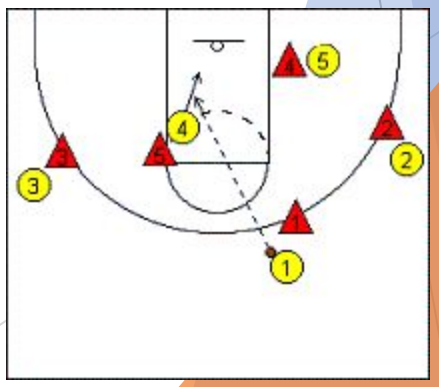
A star-point Joseki in Go

The Queen’s Gambit

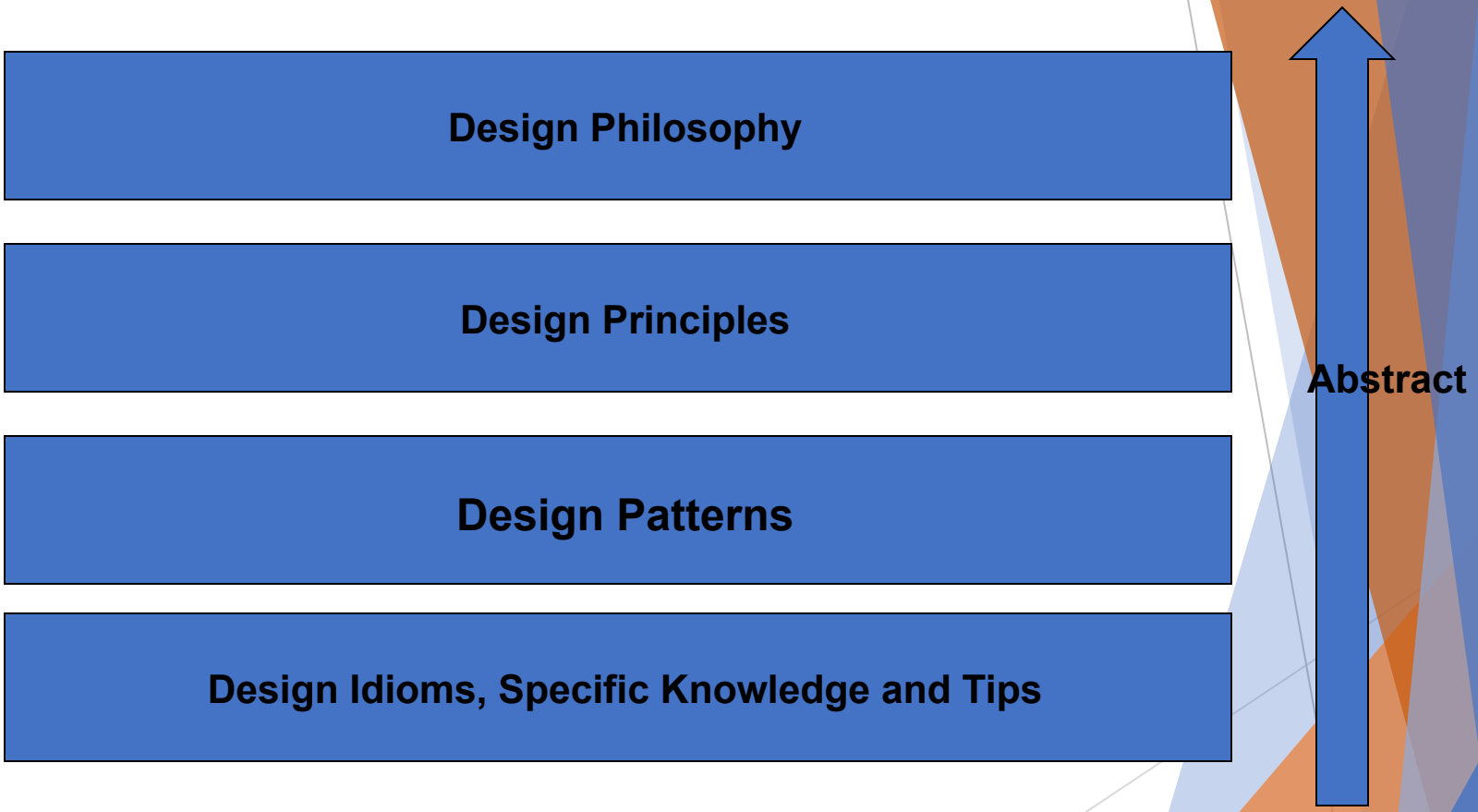




1-4 offense Pick-n-roll



The Design Methods Hierarchy



Design Patterns vs. Design Guidelines

- Design guidelines can be either abstract or concrete
  - Abstract guidelines usually do not suggest how to solve a problem
  - Concrete guidelines are usually too tailored to a specific interface
- Guidelines usually assume an absolute validity while design patterns emphasize the effectiveness in a particular context
- Guidelines are more useful for describing requirements where patterns are useful tools for those who need to translate requirements to specific software solutions.

Patterns vs. Idioms

- Not every design idea that uses the pattern syntax is a pattern.
- If an idea is too specific (e.g. programming language specific), then it is not a pattern.
- Specific ideas are called ***idioms***.
- Similarly, patterns cannot be too general.
- It must be clear how the pattern should be applied in a context.

Patterns vs. Lexicon/Vocabulary

- If some techniques appear so often in a domain, and everyone knows and use it in almost any situation, then most likely they are not patterns
  - E.g. in game design
    - Player
    - Stage
    - Enemy
    - Gun shots

What are Design Patterns…

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.

(from A Pattern Language, Alexander et al. 1977)

A pattern is the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts.

(from "*Understanding and Using Patterns in Software Development*", Dirk Riehle and Heinz Zullighoven)

"Patterns communicate insights into design problems, capturing the essence of the problems and their solutions in a compact form. They describe the problem in depth, the rationale for the solution, and some of the trade-offs in applying the solution.“

(from The Design of Sites, Van Duyne et al. 2007)

Design Patterns

A (very) brief history…

Christopher Alexander

b.1936 in Vienna. British-American architect, mathematician and design theorist
Professor Emeritus UC Berkeley

Ideas beyond architecture: urban design, software, sociology

In computer science:

- Design patterns movement (Gang of Four)
- Extreme programming movement
- The Wiki System (Wikipedia)
- Development of C++
- SimCity & Spore
- PageRank
- Agile software development

“Users are more sensitive to their needs than any architect could be.”
[1977 A Pattern Language]

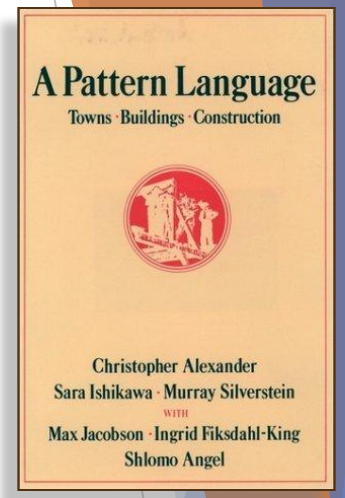
Christopher Alexander

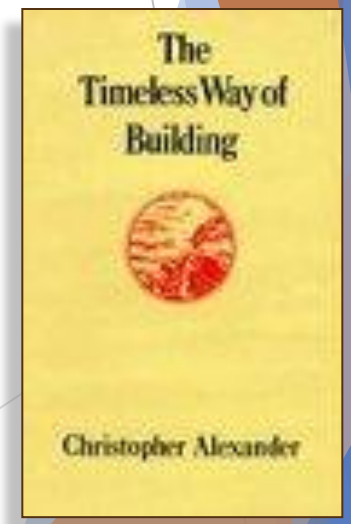
“Users are more sensitive to their needs than any architect could be.”

Christopher Alexander

- Patterns for Architecture
  - The Timeless Way of Building
  - A Pattern Language
  - The Oregon Experiment

- Towns, buildings, construction
  - Levels
  - Network
  - A language





Christopher Alexander

115 COURTYARDS WHICH LIVE**

❖

The courtyards built in modern buildings are very often dead. They are intended to be private open spaces for people to use – but they end up unused, full of gravel and abstract sculptures

There seem to be three distinct ways in which theses courtyards fail.

1. There is too little ambiguity between indoors and outdoors…
2. There are not enough doors into the courtyard…
3. They are too enclosed…

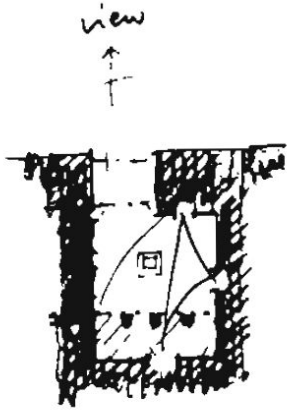
Therefore:

Place every courtyard in such a way that there is a view out of it to some larger open space; place it so that at least two or three doors open from the building into it and so that the natural paths which connect these doors pass across the courtyard. And at one edge, beside a door, make a roofed veranda or a porch, which is continuous with both the inside and the courtyard.

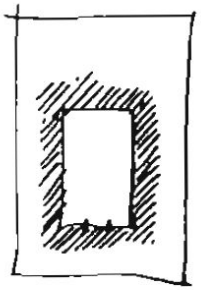
Build the porch according to the patterns for ARCADE (119), GALLERY SURROUND (166), and SIX-FOOT BALCONY (167)…

Christopher Alexander

But in a courtyard where the pattern of the opening and veranda and crossing paths is missing, there are forces which conflict in such a way that no one can resolve them for himself.



A living courtyard



Dead courtyard

Consider, for example, dead courtyard surrounded by walls on all sides, with no porch or halfway space between the indoors and the outdoors, and with no more than one path leading out to it.

The Nature of Design Patterns

- Not too general and not too specific
  - Use a solution “a million times over, without ever doing it the same way twice”
- Design patterns are a shared language
  - for “building and planning towns, neighborhoods, houses, gardens, & rooms.”
  - Ex. Beer hall is part of a center for public life…
  - Ex. Beer hall needs spaces for groups to be alone… ALCOVES

Pattern Format

1. Pattern Name
2. Context
3. Forces
4. Problem Statement
5. Solution
   - Solution Sketch
6. Other Patterns to Consider

Example - Alcoves

1. Pattern Title: Alcoves
2. Context:

   Collaborative and common areas in buildings.
3. Forces

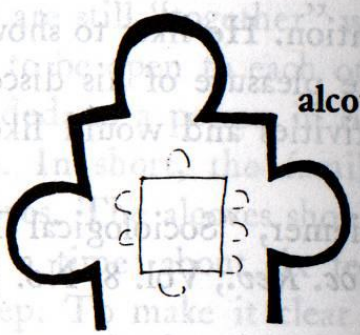
   Open spaces are inviting, but people want a sense of enclosure for private discussions.
4. Problem Statement

   Create an space that invites collaboration but also supports private discussion.

Example - Alcoves

5. Solution + sketch

Make small places at the edge of any common room, usually no more than 6 feet wide and 3 to 6 feet deep and possibly much smaller. These alcoves should be large enough for two people to sit, chat, or play and sometimes large enough to contain a desk or a table.



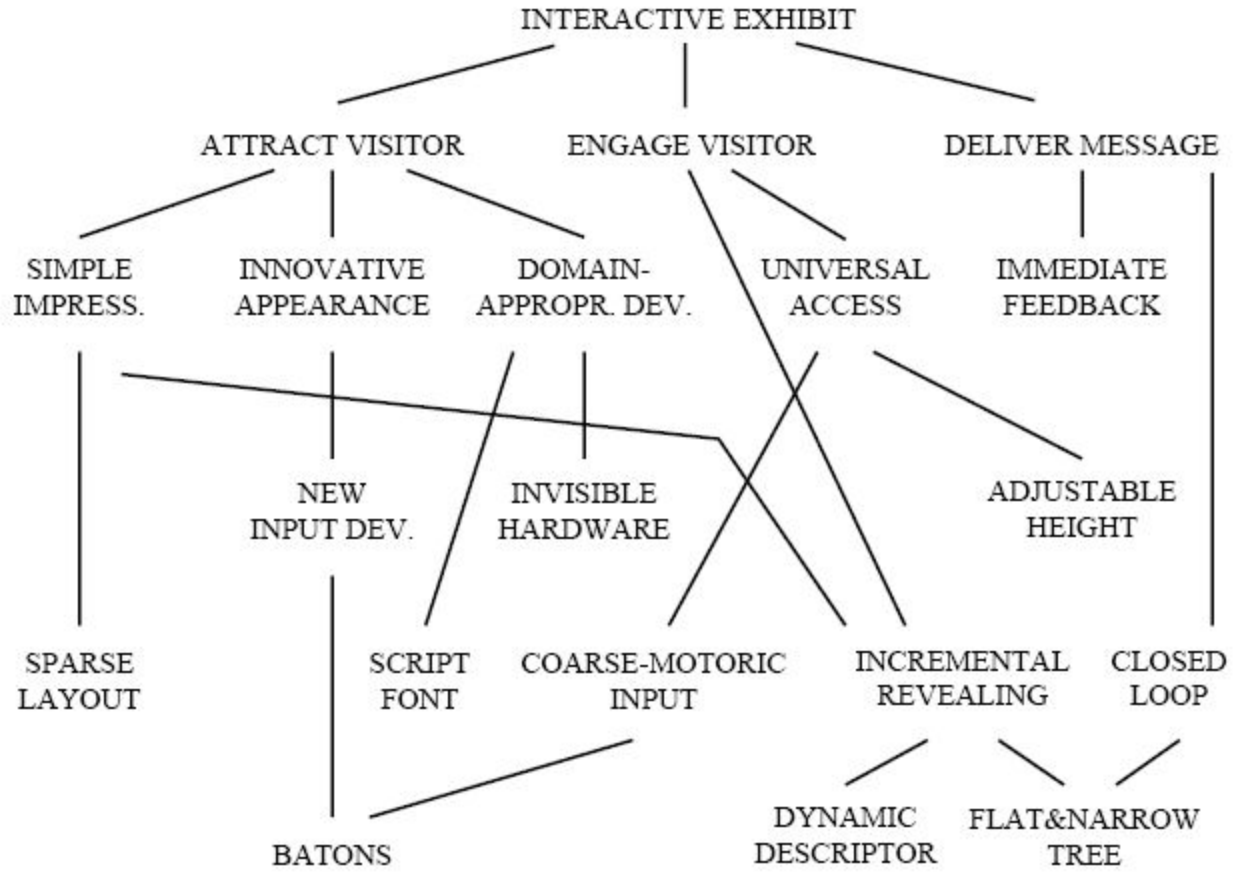
Many Patterns form a Pattern Language

- The patterns within a particular domain should be organized into a logical or naturally intuitive **structure**. Each pattern should indicate its **relationship** to other patterns and to the **pattern language** as a whole.
  - Noticing and naming the common problems in a field of interest
  - Describing the key characteristics of effective solutions for meeting some stated goal
  - Helping the designer move from problem to problem in a logical way
  - Allowing for many different paths through the design process

Pattern Languages

- Alexander emphasized the importance of pattern *languages* – more than just collections of patterns.
- Languages are sets of patterns that fill out a design space, and are chosen to complement each other.
- Forces in each pattern may explain the relations with other patterns.

A Pattern Language for Interactive Exhibits



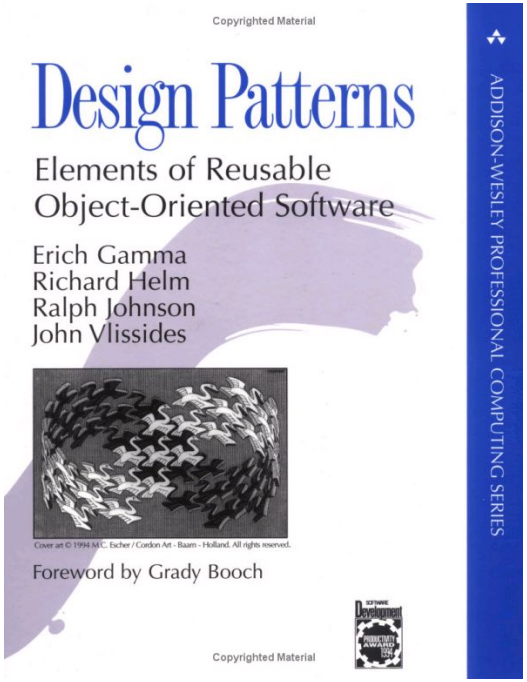
From [Borchers 2001]

Discussion

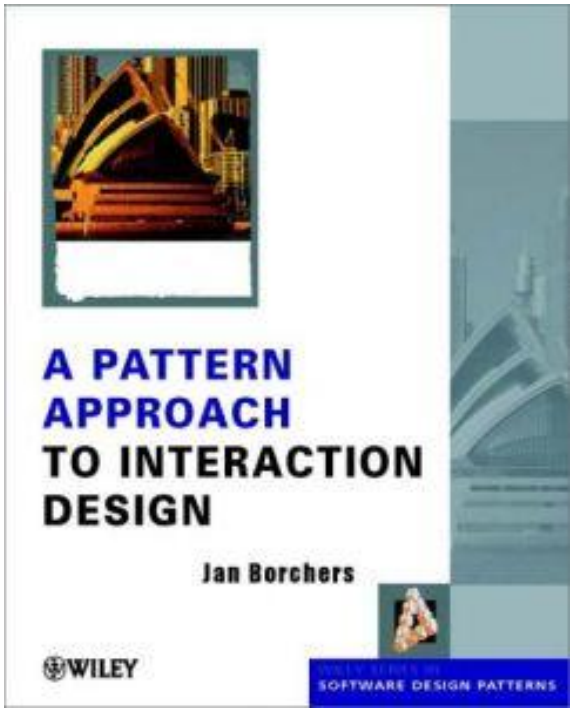
- What's the difference between design patterns and some experts' tips in design?
- Why the context is required in documenting a pattern?
- Can design patterns replace hand-on practices in design?

From Architecture to Computer Science

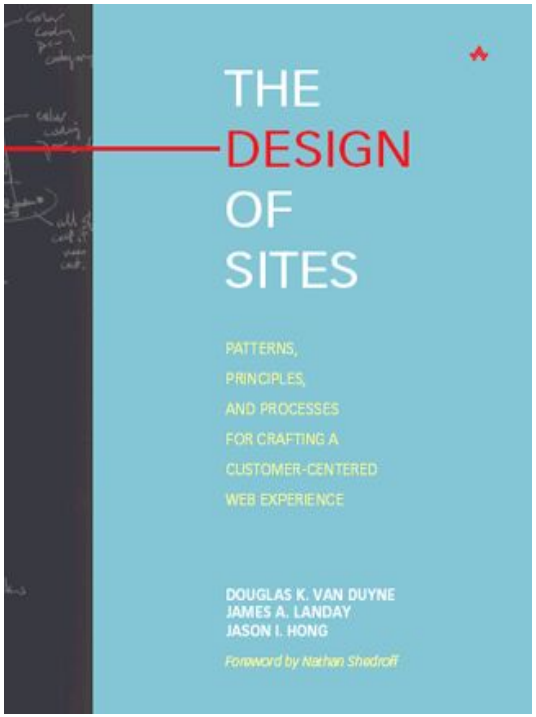
- In early 90s, Gamma, et. al borrowed the idea from architecture and used it in software engineering
  - Communicate OO software design problems & solutions



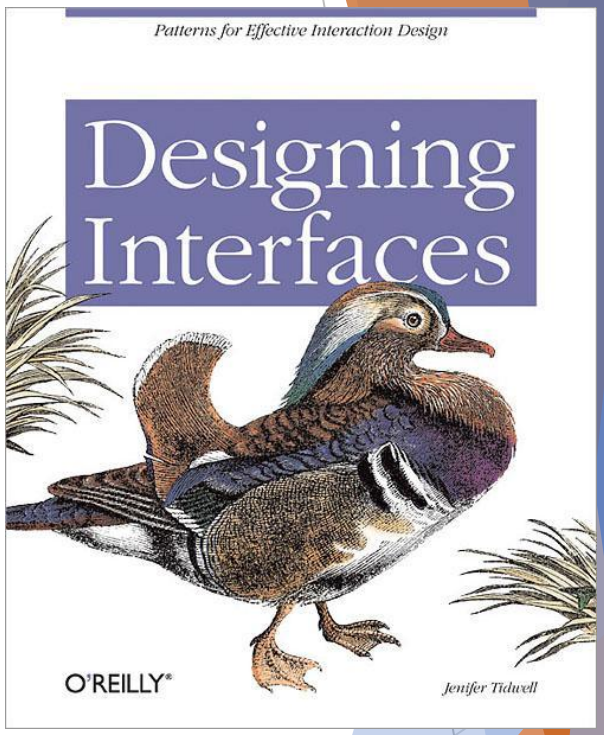
Patterns in HCI/UI Design



[Borchers 2008]



[Duyne et al 2007]



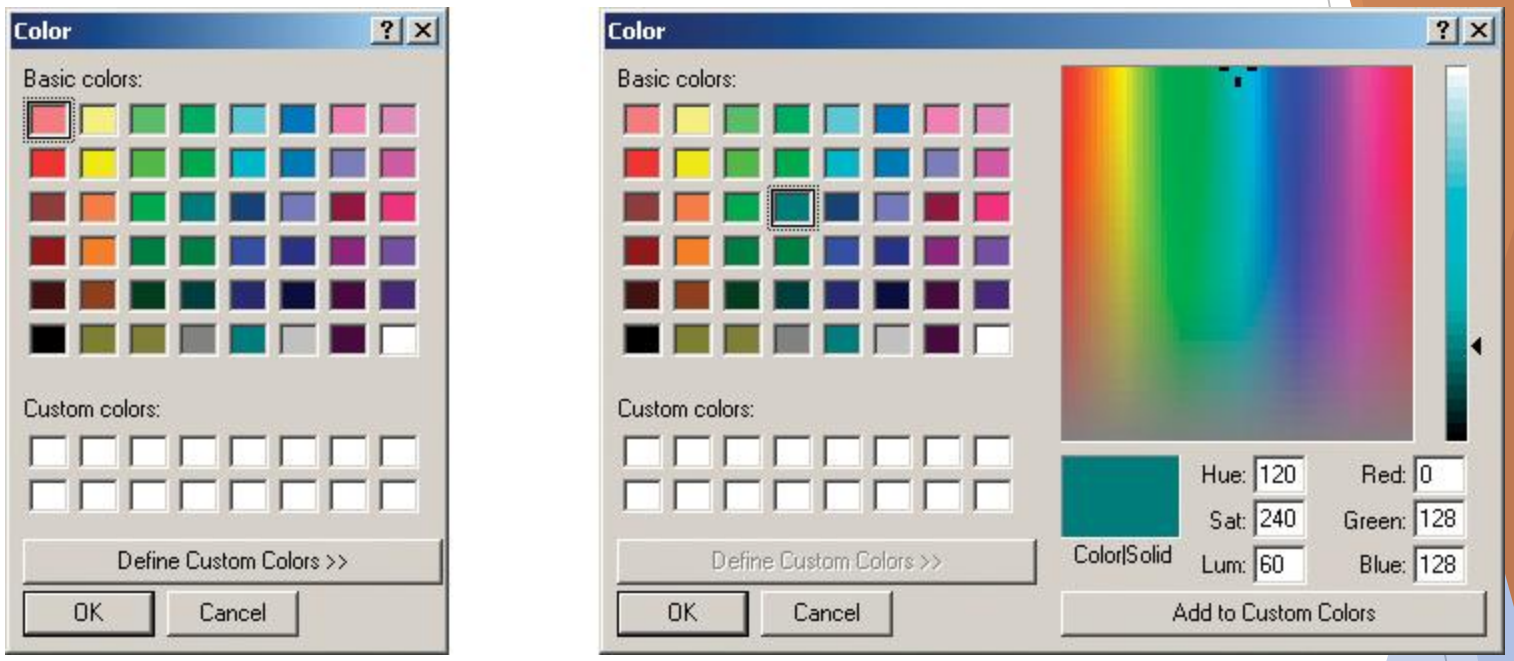
[Tidwell 2010]

- Some patterns from designing interfaces here

UI Patterns Categorized by Tidwell 2005

- Information architecture and application structure
- Navigation, signposts and wayfinding
- Layout of page elements
- Actions and commands
- Visualization patterns
- Forms and controls
- Builders and editors
- Visual style and aesthetics

Extras on Demand



- Description: Show the most important content up front, but hide the rest. Let the user reach it via a single, simple gesture

Extras on Demand

- Context
  - There's too much stuff to be shown on the page, but some of it isn't very important. You'd rather have a simpler UI, but you have to put all this content somewhere
- Solution
  - Ruthlessly prune the UI down to its most commonly used, most important items. Put the remainder into their own page or section. Hide that section by default; on the newly simplified UI, put a clearly marked button or link to the remainder, such as "More Options." Many UIs use arrows or chevrons, ">>", as part of the link or button label. Others use "…", especially if the button launches a new window or page.

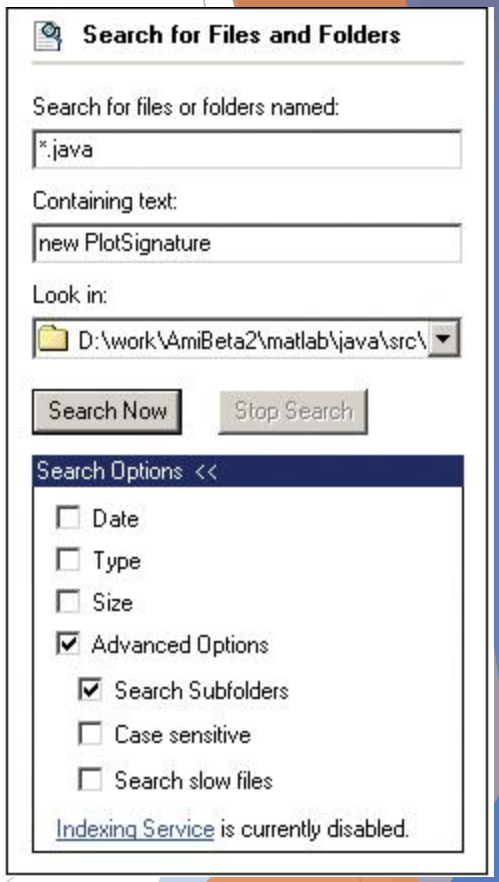
Extras on Demand

- Examples

Experts warn of possible Web attack

Seeing a rise in hacker activity that could be a prelude to a broad Internet attack, security experts urged computer users to protect their machines by installing a free patch. Internet security firms issued similar warnings, saying they've seen increased chatter in hacker discussion groups and chat rooms. "We are expecting something sooner rather than later," said Dan Ingevaldson, engineering director at Internet Security Systems in Atlanta.

FULL STORY



Breadcrumbs

From http://java.sun.com

- Description: On each page in a hierarchy, show a map of all the parent pages, up to the main page

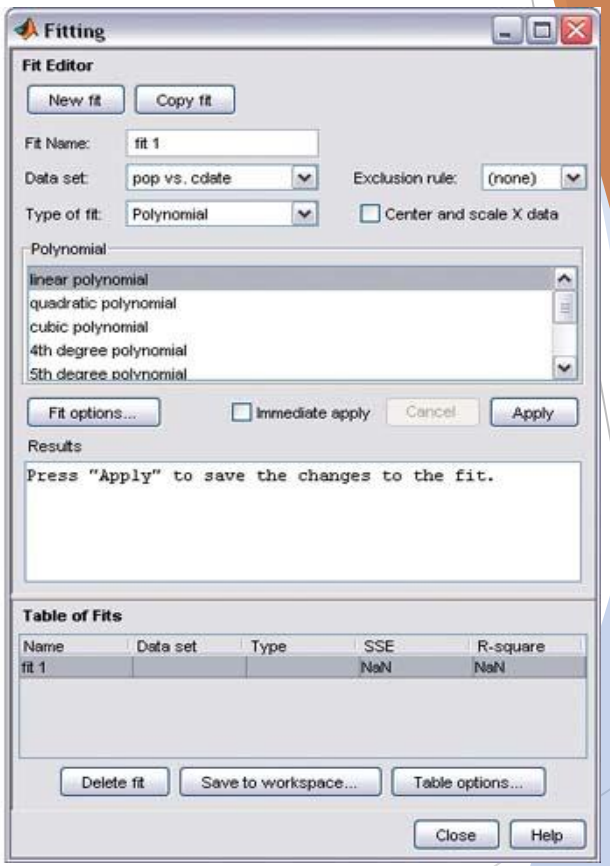
Breadcrumbs

- Context
  - The application or site has a straightforward tree structure, without much interlinking among the tree elements. Users work their way up and down this tree, either via direct navigation or searching.
- Solution
  - Near the top of the page, put a line of text or icons indicating the current level of hierarchy. Start with the top level; to its right, put the next level; and so on down to the current page. Between the levels, put a graphic or text character—usually a right-pointing arrow—to indicate movement from one level to the next.

Breadcrumbs

- Examples

Responsive Enabling

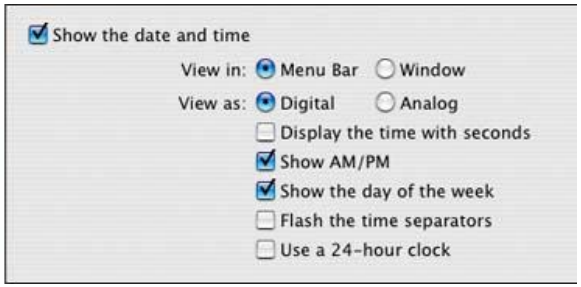


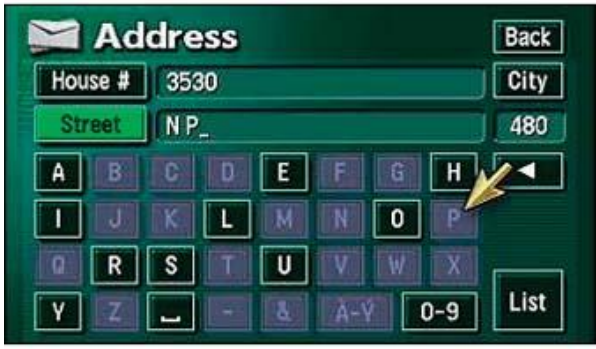
Responsive Enabling

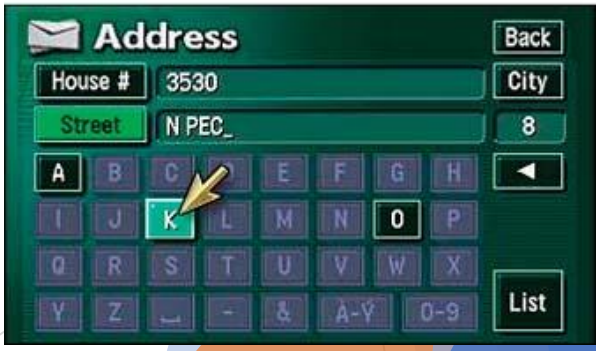
- Context
  - The UI walks the through complex step-by-step, perhaps because he is computer-naive, or because the task is rarely done (as in a Wizard). But you don't want to force the user to go page by page at each step—you'd like to keep the whole interface page. Furthermore, you want to keep the interface stable
- Solution
  - In some applications, most actions on the UI start off disabled—only the actions relevant to the user's first step are available. As the user makes choices and performs actions, more disabled items should be enabled and brought into play

Responsive Enabling

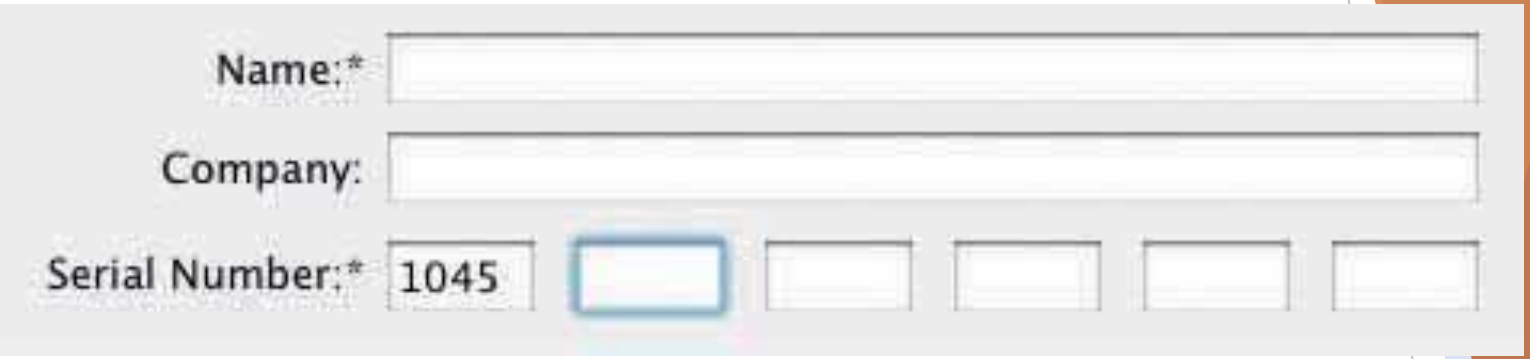
- Examples







Structured Format



- Description: Your interface requests a specific kind of text input from the user, formatted in a certain way.

Structured Format

- Context
  - The input format expected by your program is familiar and well-defined, and your program don't expect any users to need to deviate from the format you expect. Examples include credit card information, local telephone numbers, and license strings or numbers
- Solution
  - Design a set of text fields that reflect the format being asked for. Keep the text fields short, as clues to the length of the input. Once the user has typed all the digits or characters in the first text field, confirm it for her by automatically moving the input focus to the next field. She can still go back and re-edit the first one, of course, but now she knows how many digits are required there.

Structured Format

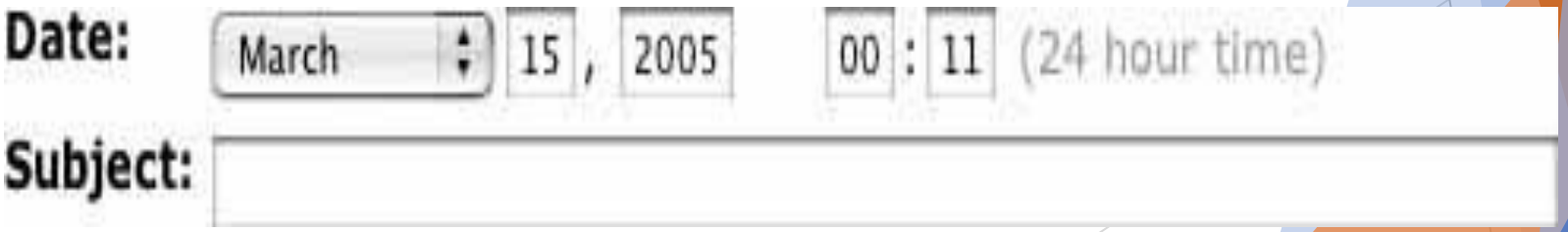
- Examples

Telephone Number: (504) 555 - 1212

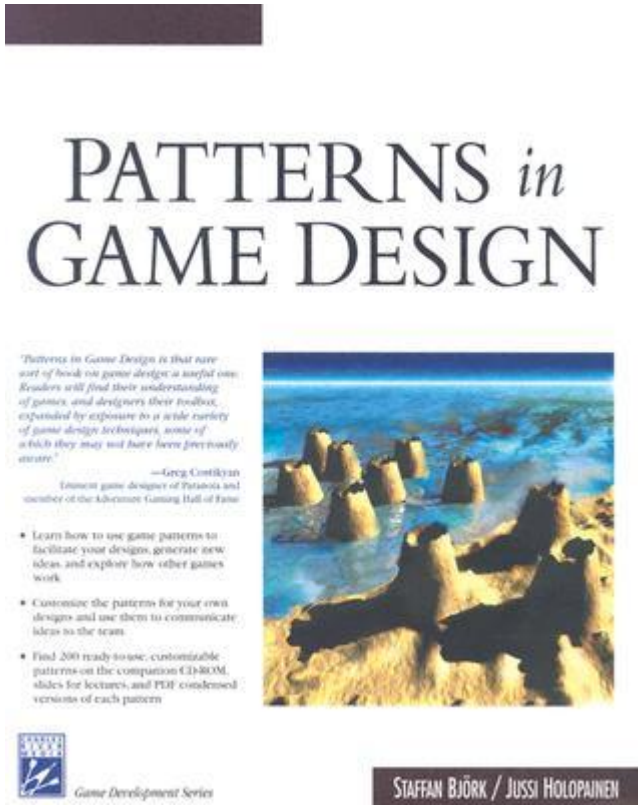
Credit Card Number: 1021 1234 5678 0000

Date: 12 / 25 / 2004

ISBN Number 0 - 1950 - 1919 - 9



Patterns in Game Design



[Bjork 2004]

Game Patterns Categorized by Bjork 2004

- Game elements
- Resource and resource management
- Information, communication and presentation
- Actions and event patterns
- Narrative structures, predictability and immersion patterns
- Social interaction
- Goals
- Goal structures
- Game Sessions
- Game mastery and balancing
- Meta games, replayability and learning curves

Producer-Consumer

Description: The production of resource by one game element that is consumed by another game element or game event

Producer-Consumer

Consequences

- Concrete, and very common pattern
- Can regulate the flow of the game
- Can increase the complexity of the game, especially if the players can control the producer-consumer elements
- Can increase the feeling of player control

Producer-Consumer

- Using the pattern
  - Production regulation
    - Based on: time or turn, player actions, game events, element configuration, continuous production vs. one-time etc.
    - Effects: what is produced, indicators to players, play mode changes etc.
  - Consumption regulation
    - Based on and effects as in production
  - Use of *Factory*, *Accumulator* and other related patterns

High Score Lists

- Description: give players the chance to rank themselves against other players who have previously played the game

High Score Lists

- What's the Context?
- What's the Solution?

Power-Ups

- Description: Power-Ups are game elements that give time-limited advantages to the player that picks them up

Power-Ups

- What's the context and solution of Power-Ups?
- Show me how can you leverage Power-Ups in your class project

Paper-Rock-Scissors

- Description: Sets of three or more actions form cycles where every action has an advantage over another action

Paper-Rock-Scissors

- Context
  - Game designers want to avoid a general winning strategy in the game play, thus encourage players to observe the activities of the in-game opponents thus promote **tension** and **randomness** of the game
- Solution
  - It's your turn now, show me how can you design such an element in your class project

What to Expect from Design Patterns?

- A common design vocabulary
- A documentation and learning aid
- An adjunct to existing methods
- A target for refining existing design

Summary

- Motivation for patterns
- What is design patterns, what is a pattern language
- A brief history of design patterns
- Examples of UI design patterns and game design patterns

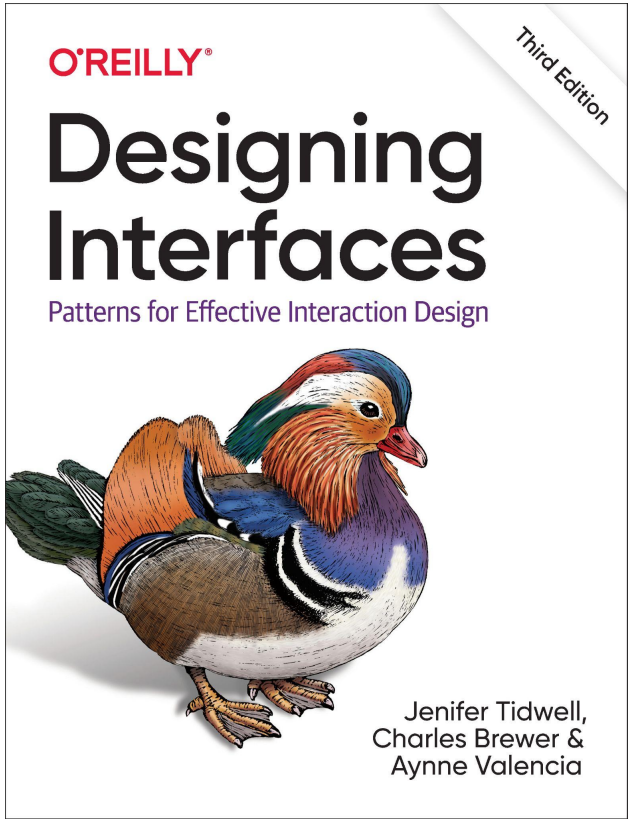
A Design Language

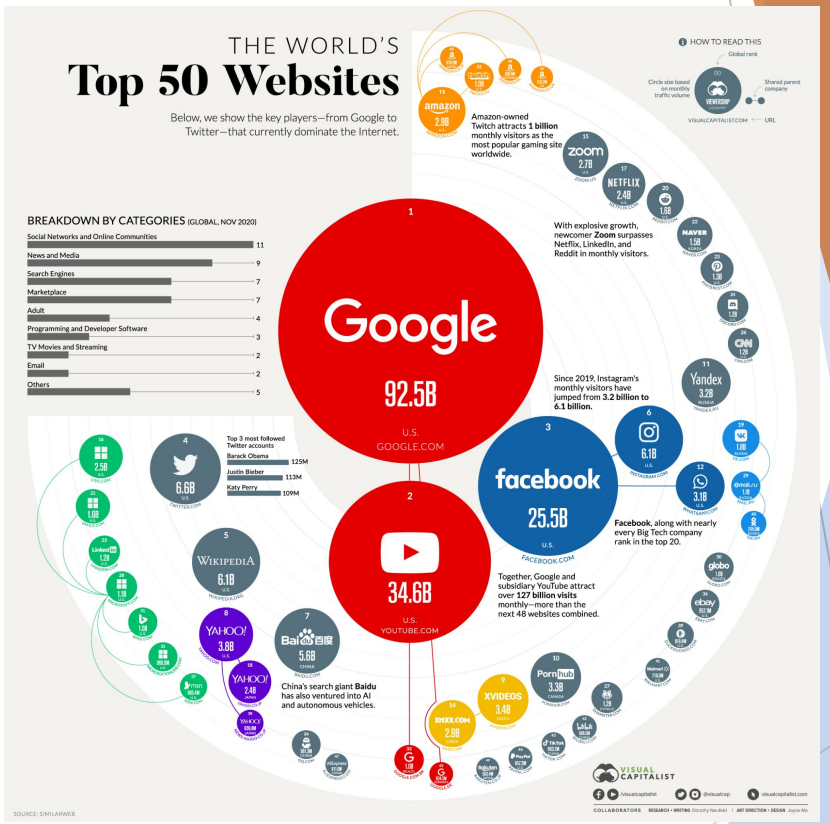
- The best designs will use many design patterns that dovetail and intertwine to produce a greater whole.

It is possible to make buildings by stringing together patterns, in a rather loose way. A building made like this, is **an assembly of patterns**. It is not dense. It is not profound. But it is also possible to put patterns together in such a way that many patterns overlap in the same physical space: the building is very dense; it has many meanings captured in a small space; and through this density, it becomes profound.

(from A Pattern Language, Alexander et al. 1977)

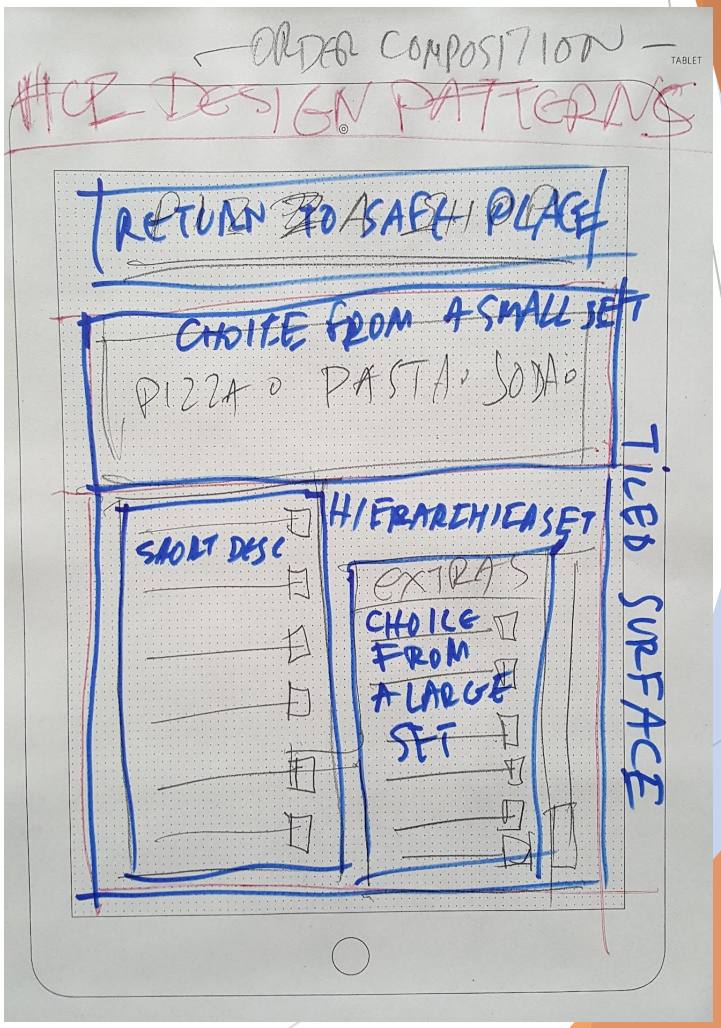
Analyze (find the patterns)





- Jenifer Tidwell, Charles Brewer & Aynne Valencia (2020) Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly Media; 3rd edition

Design (build using patterns)



Βιβλιογραφία

- Alexander, C. (1977). *A pattern language: towns, buildings, construction*. Oxford university press.
- Riehle, D., & Züllighoven, H. (1996). Understanding and using patterns in software development. *Tapos*, *2*(1), 3-13.
- Van Duyne, D. K., Landay, J. A., & Hong, J. I. (2007). *The design of sites: Patterns for creating winning web sites*. Prentice Hall Professional.
- Borchers, J. O. (2008). A pattern approach to interaction design. In *Cognition, Communication and Interaction* (pp. 114-131). Springer, London.
- Gamma, E. et al. (1995). *Design patterns: elements of reusable object-oriented software*. Pearson Education India.
- Tidwell, J. (2010). *Designing interfaces: Patterns for effective interaction design*. " O'Reilly Media, Inc.".
- Bjork, S., & Holopainen, J. (2004). *Patterns in game design (game development series)*. Charles River Media, Inc..
- Jenifer Tidwell, Charles Brewer & Aynne Valencia (2020) Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly Media; 3rd edition