Pattern Writing

Live and Direct

Kevlin Henney
Curbralan Limited
kevlin@curbralan.com

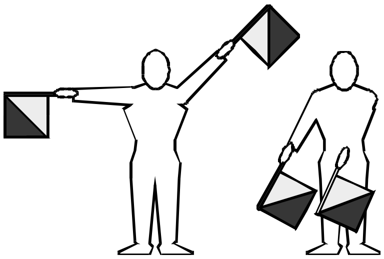
Frank Buschmann
Siemens AG
frank.buschmann@siemens.com

Agenda

- Intent
  - Learn about patterns by writing them
- Content
  - Essential patterns
  - Notes on the synthesis of form
  - Grouping patterns
  - Pattern languages
  - More on the synthesis of form

Essential Patterns

- Intent
  - Recap and dissect the pattern concept
- Content
  - What is a pattern?
  - Pattern anatomy
  - Beyond GOF
  - Kinds of patterns
  - Applying patterns



What is a Pattern?

- A pattern documents a recurring solution to a problem within a context
- Yes... but what does that mean?
  - Such a definition is not false... but not useful

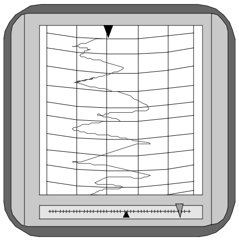
> ***A pattern is a piece of literature that describes a design problem and a general solution for the problem in a particular context.***
>
> **Jim Coplien**

A Pattern is about Design

- Design is a creational and intentional act
  - The conception and construction of a structure on purpose for a purpose
  - The act of design may be explicit or internal
- Patterns form a design vocabulary
  - Above and within the level of a function, class, component, subsystem, etc.
  - Patterns cut across the modular design elements

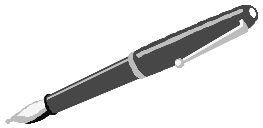
A Pattern is Context Sensitive

- Solution structure is sensitive to details of purpose and context
  - Therefore, a pattern is contingent not universal
- Context-free design is meaningless
  - No universal or independent model of design
  - Context can challenge and invalidate assumptions



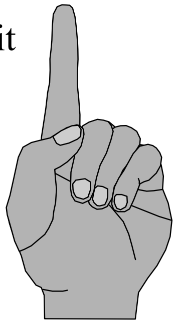
A Pattern is a Piece of Writing

- A pattern represents a thorough study of a problem–solution pairing
  - It may be either formal or informal, but it must be thorough
- A pattern tells a "successful software engineering story"
  - Albeit a short one



A Pattern is a Thing

- A pattern is architectural
  - A pattern represents configurational knowledge
  - A pattern is also the result of applying such knowledge, so you can 'point' to it
- From problem through to solution, including its consequences and rationale
  - It's not just the solution structure

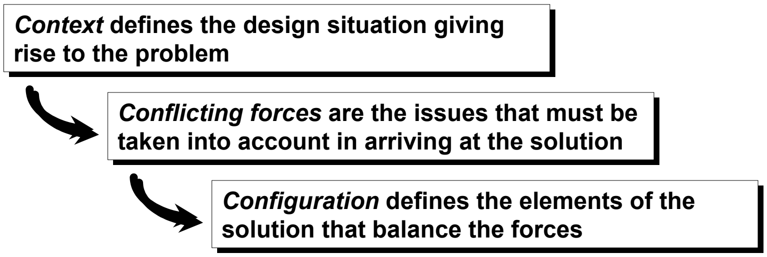


A Pattern is a Process

- However, a pattern is not just a static snapshot of design or design thinking
  - A pattern must describe what to build, why and, importantly, how
  - Each one is a highly specific mini-methodology
- The level of detail is set by the audience
  - Towns, buildings or construction?

Pattern Anatomy

- A pattern captures all the facets defining the design space for a problem–solution pairing
  - This needs to be reflected in the form



***Context* defines the design situation giving rise to the problem**

***Conflicting forces* are the issues that must be taken into account in arriving at the solution**

***Configuration* defines the elements of the solution that balance the forces**

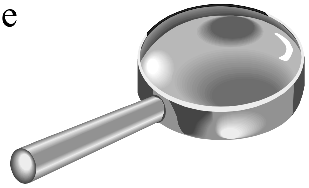
Beyond the Gang of Four

- *Design Patterns* has had a profound effect on what is considered to be design
  - Highly influential and a useful starting point
- But it has also restricted the expectations and design vocabulary of many

> ***The enormous success of design patterns is a testimonial to the commonality seen by object programmers. The success of the book*** **Design Patterns,** ***however, has stifled any diversity in expressing these patterns.***
>
> **Kent Beck**

Kinds of Patterns

- General design patterns document common frameworks and application collaborations
  - Often independent of programming language
- Idioms are language, language model, or technology specific patterns
  - Common conventions of style and usage
- And many, many more



Patterns and Scale

- Patterns can be applied at different scales, not always specific to the pattern type
  - E.g. the recurrence of Proxy at all levels in a distributed system
- More useful to think of applying patterns strategically, tactically and logistically
  - Some patterns occupy only one scope, others many

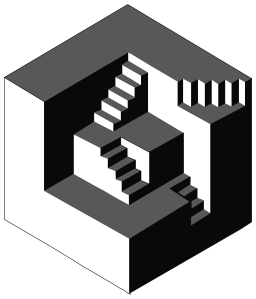
Application of Patterns

- Patterns are not rigid and automatic, and must be considered and adapted every time
  - They inform a design rather than dictate it
- Avoid BUFD with patterns
  - Neither their true nature nor true to their nature

> At the most elementary level, a building is a construction of physical elements or materials into a more or less stable form, as a result of which space is created which is distinct from the ambient space.
>
> **William Hillier**

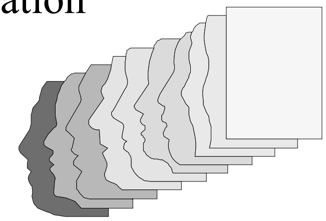
Notes on the Synthesis of Form

- Intent
  - Explore and describe some of what is involved in writing a pattern
- Content
  - Patterns are for humans
  - Pattern form
  - Motivating examples
  - Reference implementations?



Patterns are for Humans

- The audience for patterns is the human not the machine
  - Because a pattern represents the questions and reasoning required, not just the structure
- Therefore, pattern presentation is part of its substance



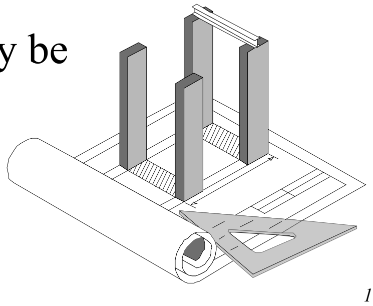
A Pattern as a Piece of Writing

- At heart, a pattern is a descriptive text
  - It has an author and an approach
  - It has a duty to name and document the problem, the solution and the rationale

> ***A man who has the knowledge but lacks the power to clearly express it is no better off than if he never had any ideas at all.***
>
> **Thucydides**

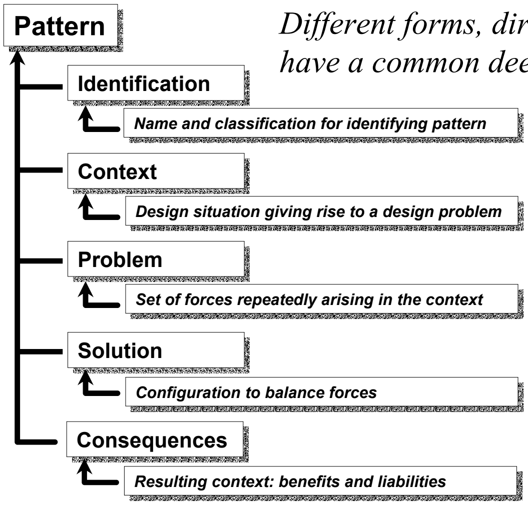
Form is Liberating

- Pattern form describes a schema for documenting patterns
  - Alexander, Portland, Gang of Four, POSA, Coplien, rule based, etc.
- The style of the form may be narrative and loose or highly structured



Essential Form Elements

Different forms, directly or otherwise, have a common deep structure



Pattern

Identification

Name and classification for identifying pattern

Context

Design situation giving rise to a design problem

Problem

Set of forces repeatedly arising in the context

Solution

Configuration to balance forces

Consequences

Resulting context: benefits and liabilities

An Example of Form

Name
- **The memorable name given to the pattern**

Problem
- **A short description of the problem being addressed by the pattern**

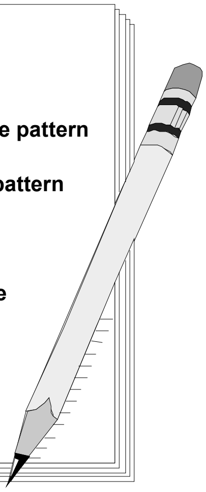
Solution
- **A short description of the solution recommended by the pattern**

Example
- **An example demonstrating the pattern in action**

Forces
- **The circumstances that make this pattern relevant and the principles that lie behind the solution**

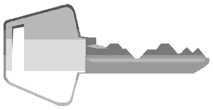
Consequences
- **The resulting context of applying the pattern**

Relationships
- **How the pattern relates to or interacts with others**



Identification

- As with variables, naming is important
  - Should be based on something from the solution structure
  - Should be memorable, usable and distinct
- Additionally, a pattern may have reference or classification information
  - E.g. an intent, a thumbnail, a scope

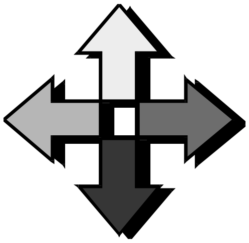


The Context

- Where does the design problem arise — what is its situation?
  - For idioms, the context is strongly weighted by the technology being used
- The context of application must be clear, either implicitly or explicitly
  - The commonest misapplication of patterns comes from using them outside their context

The Problem

- There are commonly two parts to the presentation of the problem
  - The problem statement, which summarises
  - The listing of forces, which dissects
- Forces are options in conflict and create the design tension
  - Why is the problem really a problem?



The Solution

- The description of the solution may consist of many parts
  - A statement describing the proposed arrangement that resolves the problem
  - One or more diagrams or code fragments outlining the essential features of the solution
  - Details relating to roles and rationale of elements in the proposed configuration

The Consequences

- No design decision is free of consequences
  - And no design decision is free
  - What are the benefits and liabilities of applying the pattern?
- Forces should be balanced
  - Every action has an equal and opposite reaction
  - Allows meaningful evaluation of the pattern

Cause and Effect

- The problem can be tricky to nail down
  - We are not always conscious of where we started solving a problem
  - We often find the solution easier to articulate
- Forces are hard... very hard
  - Therefore, consider looking at the consequences before enumerating the forces

Motivating Examples

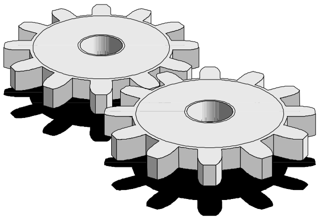
- One (or more) examples are often associated with a pattern
  - An example often illustrates better than a model that speaks only in the most abstract terms
- And the examples must be motivating
  - It is not sufficient that they are real
- The level and breadth of examples helps determine the audience for a pattern

Reference Implementations?

- However, a motivating example is not a reference implementation for the pattern
  - The distilled essence outlined as the general solution is also not
- Reference implementations are not possible
  - A sketch by definition loose rather than rigid, suggestive rather than prescriptive
  - There is always one more variation

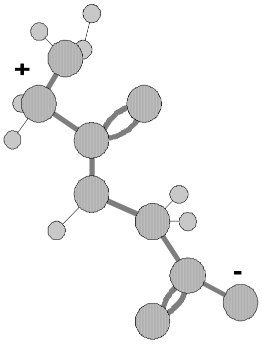
Generic and Generative

- Patterns are generic and generative
  - But not in the sense of generic programming and generative programming
- A pattern does not enumerate the solution space, only the forces and consequences
  - Patterns are intentionally incomplete



Grouping Patterns

- Intent
  - Extend the scope from individual patterns to groups of patterns
- Content
  - Pattern communities
  - Pattern catalogues
  - Complementary patterns
  - Compound patterns

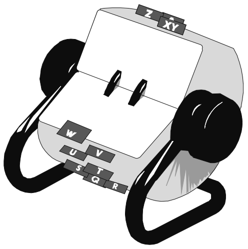


Pattern Communities

- Patterns can be used in isolation with some degree of success
  - Represent foci for discussion or point solutions
- However, patterns are in truth gregarious
  - They like the company of other patterns
  - By definition any system comprises multiple parts and multiple aspects

Pattern Catalogues

- A pattern catalogue is a collection of individual patterns
  - Not necessarily interconnected
- A catalogue is normally organised around a theme or domain
  - And substructured in terms of features, intent or architectural characteristic

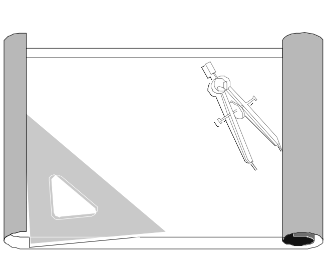


Complementary Patterns

- Patterns often have one or more partners with whom they might be associated
  - They deepen the design dialogue
- Such partners are opposites...
  - Either representing alternative solutions to similar problems
  - Or representing the completion of a natural symmetry

Compound Patterns

- A compound pattern is a named commonly occurring combination of other patterns
  - Mutually supportive community of patterns
  - Tend to appear together
  - Constituent patterns have high visibility



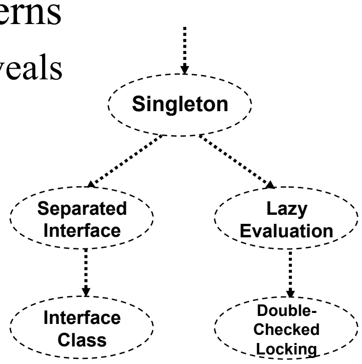
Pattern Stories

- Some patterns may be applied in a sequence from one to another
  - Development of a single system can be considered a single narrative example
- Progressive and exemplary application of patterns to create a system

> ***History rarely happens in the right order or at the right time, but the job of a historian is to make it appear as if it did.***
> **James Burke**

Pattern Deconstruction

- Many common patterns actually hide a small community of patterns
  - Close inspection often reveals another level of design consideration
- This perspective makes many patterns clearer



Singleton

Separated Interface

Lazy Evaluation

Interface Class

Double-Checked Locking

Pattern Languages

- Intent
  - Explore the concept of and motivation for pattern languages
- Content
  - What is a pattern language?
  - Pattern completion
  - A simple pattern language
  - Gradual stiffening

What is a Pattern Language?

- A pattern language connects patterns
  - Consequences of one pattern may generate forces that must be resolved by another
- Generates a particular kind of system

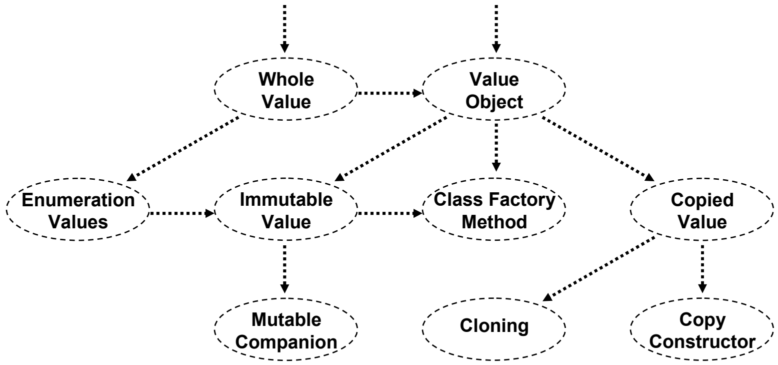
> ***A pattern language is a collection of patterns that build on each other to generate a system. A pattern in isolation solves an isolated design problem; a pattern language builds a system. It is through pattern languages that patterns achieve their fullest power.***
>
> **Jim Coplien**

Pattern Completion

- Solution structure of one pattern may be described in terms of other patterns
  - One pattern 'completes' another
- A compound pattern is, to some extent, a mini pattern language or a fragment of one

> ***"I wish life was not so short," he thought. "Languages take such a time and so do all the things one wants to know about."***
> **J R R Tolkien**

A Simple Pattern Language



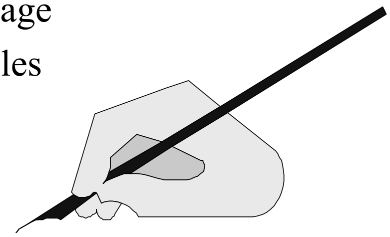
Patterns of Value
A pattern language for value-based programming in Java

Gradual Stiffening

> ***The fundamental philosophy behind the use of pattern languages is that buildings should be uniquely adaptable to individual needs and sites; and that plans of buildings should be rather loose and fluid, in order to accommodate these subtleties....***
>
> ***Recognize that you are not assembling a building from components like an erector set, but that you are instead weaving a structure which starts out globally complete, but flimsy; then gradually making it stiffer but still rather flimsy; and only finally making it completely stiff and strong....***
>
> **Christopher Alexander**

More on the Synthesis of Form

- Intent
  - Consider some of what is involved in writing a pattern language
- Content
  - Framing a pattern language
  - Pattern language examples

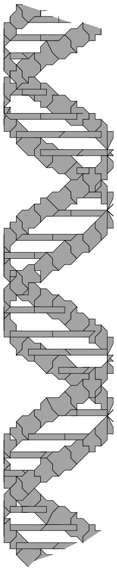


Framing a Pattern Language

- Often the overall context and common forces are established up front
  - I.e. before the pattern content
- Consistency of form, brevity of form and spatial organisation become more important

> ***Neither can embellishment of language be found without arrangement and expression of thoughts, nor can thoughts be made to shine without the light of language.***
>
> **Marcus Tullius Cicero**

Pattern Language Examples



- What example(s) should be used to illustrate a pattern language?
  - One or more running examples across all or many of the patterns
  - An associated fully worked story, up front or at the end
- Again, the nature of the example and depth of form dictate the audience

Summary

- Patterns distil successful design experience
  - Pattern writing is ultimately the articulation of this experience
  - Both form and content matter
- No pattern is an island
  - Patterns engage the designer, and therefore the reader, in a dialogue