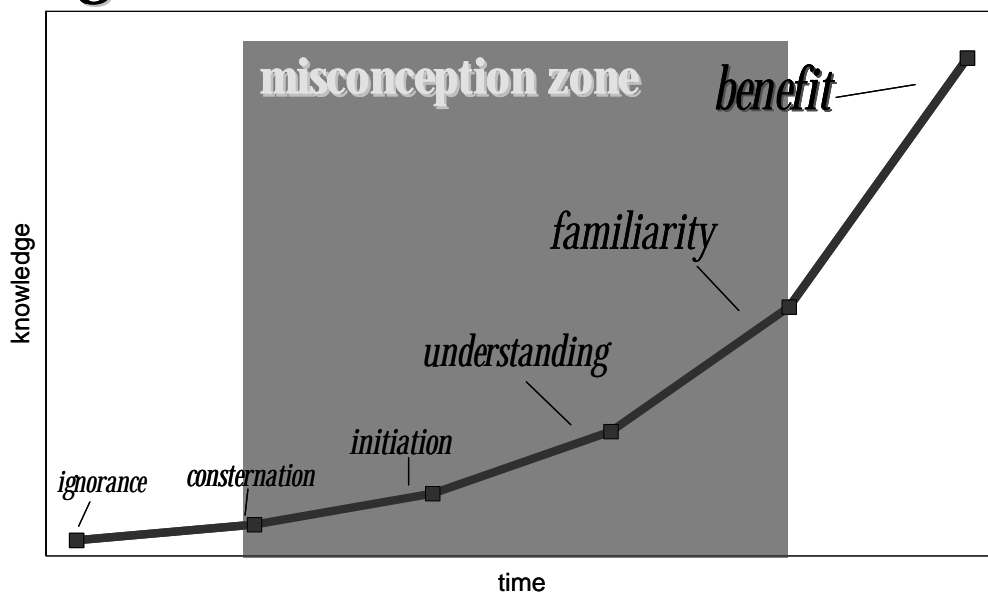


Patterns: The Top Ten Misconceptions

John Vlissides
IBM Research

Stages of Pattern Awareness



Three Categories of Misconceptions

- I. Misconceptions about what patterns *are***
- II. Misconceptions about what patterns *can do***
- III. Misconceptions about *the community that promotes them***

Misconception 1:

**"A pattern is a
solution to a problem in a context."**

One of Christopher Alexander's definitions

⇒ heresy?

Misconception 1:

"A pattern is a solution to a problem in a context."

A simple counterexample:

Problem: **How do I redeem my winning lottery ticket before it expires?**

Context: **The dog ate the ticket an hour before the**

Solution: **Cut the dog open, fish out the ticket, and run to the nearest redemption station.**

Misconception 1:

"A pattern is a solution to a problem in a context."

What's missing?

I. *Recurrence:* Makes solution relevant elsewhere.

***Teaching:* Lets you tailor solution to your problem.**

***Name:* A handle for referring to the pattern.**

A Real Pattern

Name: Generation Gap

**Context: A tool that generates code automatically.
Programmers modify the code to add or change behavior.**

**How do you prevent subsequent regeneration
from clobbering a programmer's modifications?**

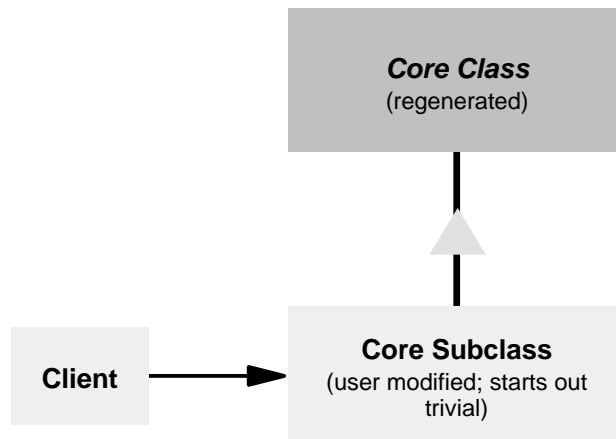
A Real Pattern (cont'd)

Forces:

- **need controlled access to generated internals**
- **comments-based approach ugly, unchecked**
- **auto-diff-and-merge too iffy**

A Real Pattern (cont'd)

Solution: Use inheritance to separate generated code from modifications.



A Real Pattern (cont'd)

Consequences:

- + modifications are decoupled from generated code
- + modifications can have privileged access
- + subsequent regeneration doesn't mean reapplying changes

- double the number of classes
- complicates integration into existing class hierarchies

Misconception 2:

**"Patterns are just jargon, rules,
programming tricks, data structures...."**

The "Belittling Dismissal"

⇒ **superficial familiarity, with dash of cynicism**

Misconception 2:

**"Patterns are just jargon, rules,
programming tricks, data structures...."**

Jargon?

- **e.g., "pattern," "forces," "the QWAN"**
("Quality Without A Name")
- **comparatively spare**
- *an effective pattern writer eschews pattern-specific jargon*

Misconception 2:

**"Patterns are just jargon, rules,
programming tricks, data structures...."**

Rules?

- you don't apply patterns mindlessly

Programming tricks? Data structures?

- overemphasis on solution
- *"So's your ol' man!"*

Misconception 3:

"Seen one, seen them all."

Beware the broadbrush!

Wide range of styles, scope, detail, maturity—and quality

Major Pattern Styles

Alexandrian

- Predecessor Patterns
- Problem
- Discussion
- Solution
- Successor Patterns

Coplienian

- Problem
- Context
- Forces
- Solution
- Resulting Context

Gang of Four (GOF)

- Intent
- AKA
- Motivation
- Applicability
- Structure
- Participants
- Collaborations
- Consequences
- Implementation
- Sample Code
- Known Uses
- Related Patterns

Pattern Writing Takes Time

Design Patterns took four years for 23 patterns

- archeology
- identifying and characterizing recurrences
-
- feedback from reviewers, Internet community
- iteration: each rewritten 10-20 times

Quality is directly proportional to pain!

Misconception 4:

"Patterns need tool or methodological support to be effective."

Patterns are primarily *food for the brain*.

Main benefits:

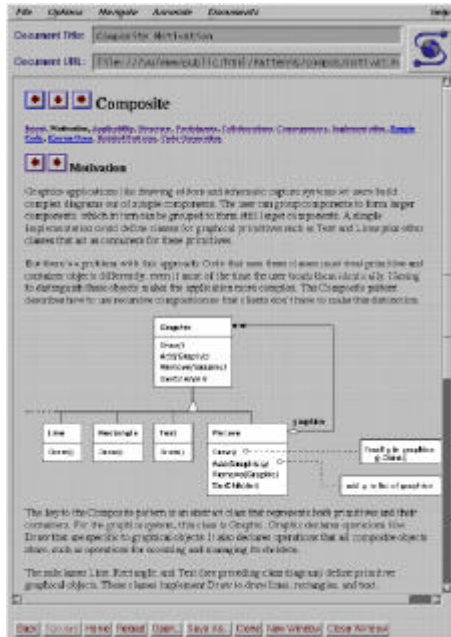
- I. Capturing expertise**
- II. Communication**
- III. Documentation**
- IV. Restructuring**

Some tools are useful...



Hypertext rendition of GOF

Some tools are useful...



Click to view and navigate the entire book



Automatic code generation

...others, less so.



Specify design tradeoffs

...others, less so.



Spit out the code

Flashy, but

- saves mainly typing
- brain remains the bottleneck
- most useful for pedagogy

Summary

Patterns *are not*

- 1. a solution to a problem in a context
jargon, rules, or data structures**

dependent on tools or methodology

Misconceptions about what patterns *can do*

Misconception 5:

**"Patterns guarantee reusable software,
higher productivity, world peace, etc."**

Patterns don't guarantee *anything!*

They merely empower—potentially.

Under-promise, over-deliver

Misconception 6:

"Patterns 'generate' whole architectures."

**Like unto the former misconception,
but higher-falutin'**

What is "generativity"?

No consensus

My take:

- **a pattern's ability to create "emergent behavior"**
- **makes patterns more than just pat solutions**
- **lies primarily in the pattern's**

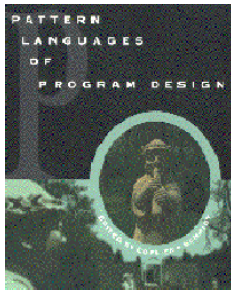
*Patterns don't **generate** anything—people do!*

Misconception 7:

"Patterns are for (object-oriented) design or implementation."

GOF's a lot to blame for this one

Pattern Languages of Program Design 1



- I. Frameworks & Components**
- II. Systems & Distributed Processing**

Process & Organization
Design Patterns & Catalogs
Architecture & Communication
Object Usage & Style
Events & Event Handlers

Pattern Languages of Program Design 2



I. Language-Specific Patterns & Idioms

General-Purpose Patterns

Special-Purpose Patterns

Architectural Patterns

Process & Organization

**Concurrent Programming/
Distributed Systems**

Pattern Languages of Program Design 3



Lots of new domains:

✓ **Software Evolution**

✓ **Software Testing**

✓ **Diagnostic Logging**

✓ **Transport Systems**

✓ **Pattern Writing**

✓ **Fire Alarm Systems**

✓ **Basket Weaving (he's kidding)**

Misconception 8:

**"There's no evidence
that patterns help anybody."**

Used to hold water, but not anymore

Benefits reported in

- **journals (CACM, SP&E)**
- **conferences (OOPSLA, ECOOP, ICSE)**
- `patterns-discussion@cs.uiuc.edu`

Caveat: **evidence is mostly qualitative**

Summary

Patterns cannot

- 5. guarantee higher quality, reuse, or anything else
generate entire architectures
be limited to OOD
be dismissed as unproven**

Misconceptions about the *pattern community*

Misconception 9:

**"The pattern community
is a clique of elites."**

elite, *n.* 1. the choice or distinguished part; the best
2. a size of typewriter type equivalent to
10-point printing type. There are 12 elite characters to
adj. distinguished.

clique, *n.* a small, exclusive set or snobbish group of
people within a larger group.

The PLoP Community

A microcosm of the community at large

Diversity

- **international**
- **from large multinationals and small startups**
- **analysts, designers, implementers**
- **students and professors**
- **big-name authors and fledglings**

The PLoP Community

Growth & fluidity

- attendance doubled in three years
- proliferation of conferences
 - Euro-, Chili-, & TelePLoPs; UP

Emphasis on experience

- more practitioners than academics
- early proponents all from industry

A grassroots movement

Misconception 10:

**"The pattern community
is self-serving, even conspiratorial."**

Poppycock!

Rot!

Balderdash!

Malarkey!

Humbug!

Drivel!

Horse hockey!