

# GOTO AMSTERDAM 2023

#### Generic or specific? Making sensible software design decisions

#### **#GOTOams**



GOTO Guide LET US HELP YOU

# Ask questions through **the app**



Get IT ON Google Play \*\*\*\*\*
also remember to rate session

THANK YOU!

#GOTOams

#### Generic or specific?

#### Making sensible software design decisions

Bert Jan Schrijver bertjan@openvalue.eu



#### Let's meet Bert Jan Schrijver



# • OPENVALUE





#### What's next? Outline

Definitions •

Flexibility in software 🔸

Levels of • generic vs specific

Tools to help decide •



• The cost of generic

When & why to go generic

 Bonus: sharing code in an organization

Conclusion



#### What is software design?

### Design vs architecture

#### Specific solution (or design)

- Tailor made for use in a single place
- Tailored to a specific problem or scenario
- May not be easily adaptable to other situations

#### Generic solution (or design)

 More flexible and reusable solution Solution can be applied to a wide range of problems or scenarios Generified solution that can be used in more than 1 place

Background source: https://filmquarterly.org/2012/07/02/i-robot-what-do-robots-dream-of/

#### Hierarchical decomposition

- Breaking a system or problem into smaller parts that are easier to understand
- Example: Google search

#### Hierarchical decomposition



### Coupling

 Kind and degree of interdependence between building blocks of software Measure of how closely connected two components are Usually contrasted with cohesion (low coupling -> high cohesion)

#### Types of coupling

- Inheritance
- Messages or events
- Temporal
- Data types
- Data

Code / API (binary or source)

#### Be careful with coupling!

### Generic solution = coupling!

### The risk of DRY

#### Duplicated code doesn't hurt until you need to change it.

#### "Future proof" design

- Should we be prepared for future changes?
- Design should be structured to accommodate change
- Risk management: risk of wrong decision

#### About flexibility in software...

### "Highly specific code is often preferable to sophisticated configuration" - Stefan Tilkov

# When are we going to talk about generic vs specific?

Background source: https://7216-presscdn-0-76-pagely.netdna-ssl.com/wp-content/uploads/2011/12/confused-man-single-good-men.jpg

#### Generic vs specific: levels

- Code / class level
- Manually written vs generated code
- Library level
- Data level
- (Micro)service level
  Organisation level

### Generic or specific?

#### Tools to help decide

- Do we really need this now? (YAGNI)
- The 5 W's
- Time/effort for generic vs specific
- Myth of "first time right"
- Complexity and scope
- The rule of three

Future needs and evolution



Gunnar Morling 😚 @gunnarmorling

A fair share of software design is professional procrastination: keeping options on the table, doors open for as long as you can, in order to avoid locking yourself into a corner by committing too early to abstractions which turn out poor later on.

9:33 AM - Jan 21, 2023 - 45.1K Views

#### The rule of three

- When reusing code, copy it once, and only abstract the third time
  - Avoid writing the wrong abstraction
  - It's easier to make a good abstraction from duplicated code than to refactor the wrong abstraction
- "Three strikes and you refactor"

#### The rule of three

- First case: Just build it, don't genericise at all. Solve the problem in front of you (YAGNI)
- Second case: Duplicate the original, redesign and extract common behaviour while you change
- Third case: examine lessons from first two passes, design a generic solution that will make it easy to add your third case

#### Design heuristics

- Pass 1: YAGNI / rule of three: as simple and specific as possible
- Pass 2: based on solution domain knowledge: is a generic solution less work?
- Pass 3: based on problem domain knowledge: is the easiest solution actually correct?
- Pass 4: looking at customer behaviour or other non technical considerations, does this change your decision?

#### Strategic design

- Concept from Domain Driven Design
- Tool to help decide for generic vs specific
- But more about building yourself or not
- Subdomains:
  - Core domain
  - Supporting subdomain
  - Generic subdomain

#### Conway's law

- Organizations design systems that mirror their own communication structure
- Don't force a solution that goes against the organisation structure
- Be careful to go generic when teams don't want to work together

#### Conway's law in action

#### The cost of a generic solution

 Going generic may save time in the long run, but at which price? Another rule of three: building reusable components is 3x as difficult as single use The price you pay is coupling Both on code level and people/team level (communication overhead)

## What if you get it wrong?

#### The cost of abstractions

- There are no zero cost abstractions
- Efficiency gains of a generic solution are typically clear, but how about:
  - Onboarding new people
  - Readability
  - Coupling

#### The cost of abstractions

- Writing bad abstractions
  - Writing unnecessary reusable code
  - Introducing unnecessary coupling
- Maintaining bad abstractions
  - Hard to see
  - Hard to understand
  - Hard to extend

#### When / why to go generic

#### Bad reasons to go generic

- "We've always done it like this"
- "We don't want to depend on libraries"
- "We need to be future proof"
- Because the product owner wants it
   Because the product owner wants it
  - Because the architect wants it

#### Valid reasons to go generic

- Rule of three checks out
- You're pretty sure you're going to need it almost everywhere
- A library that lots of teams will use
- Complex logic or skills that only a couple of people have
  Gains are bigger than cost

#### Generic vs specific in different scopes

- Think back about the layers in hierarchical decomposition of a system
- Code vs component vs service
- Are the considerations for generic vs specific the same on every level?
- Risk when getting it wrong is higher when the level is higher
- Don't confuse generification with standardization!

# Why specific is often faster

# Code golf



Advent of Code [About] [Events] [Shop] [Settings] [Log Out] Bert Jan Schrijver 20\* [year=>2022] [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

--- Day 5: Supply Stacks ---

The expedition can depart as soon as the final supplies have been unloaded from the ships. Supplies are stored in stacks of marked crates, but because the needed supplies are buried under many other crates, the crates need to be rearranged.

The ship has a giant cargo crane capable of moving crates between stacks. To ensure none of the crates get crushed or fall over, the crane operator will rearrange them in a series of carefully-planned steps. After the crates are rearranged, the desired crates will be at the top of each stack.

The Elves don't want to interrupt the crane operator during this delicate procedure, but they forgot to ask her which crate will end up where, and they want to be ready to unload them as soon as possible so they can embark.

They do, however, have a drawing of the starting stacks of crates and the rearrangement procedure (your puzzle input). For example:

[D] [N] [C] [Z] [M] [P] 1 2 3 move 1 from 2 to 1 move 3 from 1 to 3 move 2 from 2 to 1 move 1 from 1 to 2

[Q]	[]]							[H]
[G]	[S]	[Q]		[Z]				[P]
[P]	[F]	[M]		[F]		[F]		[S]
[R]	[R]	[P]	[F]	[V]		[D]		[L]
[L]	[W]	[W]	[D]	[W]	[S]	[V]		[G]
[0]	[H]	[H]	[T]	[D]	[L]	[M]	[B]	[B]
[T]	[0]	[B]	[5]	[1]	[0]	[B]	[]]	[N]
[F]	[N]	[F]	[V]	[0]	[Z]	[7]	[T]	[0]
1	2	3	4	5	6	7	8	.d.
-	2	5	-	5	0	'	0	5
move 1 from 8 to 1								
nove	e 1	1101	0 10					
move 1 from 6 to 1								
move 3 from 7 to 4								
move 3 from 2 to 9								
move 11 from 9 to 3								
move	e 1 1	from	6 to	o 9				
move 15 from 3 to 9								
move	e 5 1	from	2 to	3				
move	e 3 1	from	7 to	5				
move	e 6 1	from	9 +/	3				
110 00		6	1 +					
move	e 6 1	rrom	1 τα	0 6				
move	e 2 1	from	3 to	o 7				

```
Scanner scanner = new Scanner(input);
List<String> lines = new ArrayList<>();
List<String> instructions = new ArrayList<>();
```

```
// determine initial matrix width/height dimensions
int maxLineLength = 0, initialMatrixHeight = 0;
while (scanner.hasNextLine()) {
    String line = scanner.nextLine();
    lines.add(line);
    if (line.contains("[")) { initialMatrixHeight++; }
    if (line.endsWith("]") && line.length() > maxLineLength) { maxLineLength = line.length(); }
int initialMatrixWidth = (maxLineLength + 1) / 4;
Natrix matrix = new Matrix(initialMatrixWidth, initialMatrixHeight);
// init matrix and instruction
int \mathbf{v} = 0;
for (String line: lines) {
    if (line.contains("[")) { // matrix line
        <u>y++;</u>
        for (int x=1; x<initialMatrixWidth+1; x++) {</pre>
            matrix.put(x, y: initialMatrixHeight-y+1, line.charAt(4*(x-1)+1));
        3
    } else if (line.startsWith("move")) { instructions.add(line); }
}
```

```
ArrayList<String> stack1 = new ArrayList<>(Arrays.asList("F", "D","B", "Z","T", "J","R","N"));
ArrayList<String> stack2 = new ArrayList<>(Arrays.asList("R", "S", "N", "J","G", "Z","F", "Q"));
ArrayList<String> stack3 = new ArrayList<>(Arrays.asList("C", "R","N", "J","G", "Z","F", "Q"));
ArrayList<String> stack4 = new ArrayList<>(Arrays.asList("F", "V", "N", "G", "R", "T", "Q"));
ArrayList<String> stack5 = new ArrayList<>(Arrays.asList("L", "T", "Q", "F"));
ArrayList<String> stack6 = new ArrayList<>(Arrays.asList("Q", "C","W", "Z","B", "R","G", "N"));
ArrayList<String> stack7 = new ArrayList<>(Arrays.asList("F", "C","L", "S","N", "H","M"));
ArrayList<String> stack8 = new ArrayList<>(Arrays.asList("D", "N", "Q", "M","T", "J"));
ArrayList<String> stack8 = new ArrayList<>(Arrays.asList("D", "N", "Q", "M","T", "J"));
```

Planets Thi Columbus Robbs 100

#### Exception handling

```
@ControllerAdvice
public class ControllerExceptionHandler {
```

```
@ExceptionHandler(value = {ResourceNotFoundException.class, CertainException.class})
public ResponseEntity<ErrorMessage> resourceNotFoundException(ResourceNotFoundException ex, WebRequest request) {
    ErrorMessage message = new ErrorMessage(
        status,
        date,
        ex.getMessage(),
        description);
```

return new ResponseEntity<ErrorMessage>(message, HttpStatus.NOT\_FOUND);

OPENVALUE



BETTER CATS, FASTER

#### Website code re-use

#### Meet the driving force behind our team

Looking for their Java-developing owners? Visit openvalue.eu!



OPENVALUE



#### Meet the driving force behind our team

Looking for their Java-developing owners? Visit openvalue.eu!



She's like a whirlwind in the house. Very clingy and will meow if you walk away. Steals your socks and might bite when not petted correctly.



Despite being a terrific singer, cannot meow. Probably because he grew up with dogs?



Expects to be involved in all daily operations in the house. Yes, all of them.

meowing. Loves to sleep outside if the weather is nice. tries hard when he thinks he's hungry, even when he just ate. Keyboards are great places to walk. where he wants. He decides when he gets attention and will loudly protest if you sleep in. Will lick your nose.





Does not like strangers, will protect herself and hide. Otherwise meows at you until she has your undivided attention.



Highly energetic good-weather cat. Prefers weight loss over food that is not up to his standards. Sleeps in your lap and will meow back in conversation. Favorite hobby? Jumping on your back.



Enjoys sleeping on a very specific chair in the kitchen. He does not appreciate it when people need said chair. Will annoy them until it's made available again. Is otherwise kind, as long as he sleeps.



Frummel is everyone's friend but loves to taunt his brother Max. When Max gets fed up he lashes out, which only encourages Frummel. Imagine when he grows up.



Elegant feline, lets youngsters Max and Tijgetje do their thing. Lets her caretakers think they are in control. Expects to be picked up and brought to her sleeping accommodations.



Likes to play with potatoes, and dangerous walkways. Likes to be an alarm clock in the morning and recently discovered she has no balls.



Curious and attention-hungry. Fails at hunting birds and butterflies but never gives up! Her name means snow. She hates snow.



Bit of a civa, loves attention and makes sure you know it. Will join you on a walk but sits down and meows when you go too far from home.



Loves mainly 2 things: breakfast and dinner. Affection towards anyone is proportionate to when she's had her last meal. Regularly disappears for a day. OpenValue. High quality cats. 15 seasoned full snack experts.

Find us online:



#### Generic solutions on organization level

#### Sharing code within an organization

- Sharing code efficiently at scale is hard
- Sharing code at scale means:
  - Multiple modules that share code
  - Multiple team members
  - High rate of change
  - Little to no loss of individual productivity

#### Sharing code within an organization

- Challenges:
  - Refactoring
  - Versioning
  - Reviewing
  - Builds and codebase size monorepo?

Source: https://medium.com/@jeffwhelpley/the-problem-with-shared-code-124a20fc3d3b

#### Monorepositories

- Monorepo: 1 large repository for a group of projects (possible all projects)
- Good: easy to make changes across projects
- Bad: dependencies & build times

#### Considerations on sharing code in an org

- Discovery: what code / libraries exist?
- Distribution: binary or source dependency?
- Import: well defined API's or chaos?
- Versioning, upgrades and lifecycle

management

- Who maintains it?
- Possible approach: inner source culture

Source: https://medium.com/@jeffwhelpley/the-problem-with-shared-code-124a20fc3d3b, background source: Klara Kulikova on Unsplash



#### Generic or specific?

- Consider:
  - YAGNI / Rule of three / 5 W's
  - Cost of generic

  - Scope / levelConway's law
  - Organization

Generic or specific? It depends.

A program that works perfectly but is impossible to change will become useless. A program that does not work but is easy to change will become and remain useful continuously.

#### Write. simple. code.

the same of the local day is the second second second second second

Source: https://images.unsplash.com/photo-1515611926865-4fcb1c2ce28d?ixlib=rb-4.0.3&dl=kelly-sikkema-kxtB2TFBF2g-unsplash.jpg&q=80&fm=jpg&crop=entropy&cs=tinysrgb

# THAT'S IT. Now go kick some assi

Source: https://cdn2.vox-cdn.com/thumbor/J9OqPYS7FgI9fjGhnF7AFh8foVY=/148x0:1768x1080/1280x854/cdn0.vox-cdn.com/uploads/chorus\_image/image/46147742/cute-success-kid-1920x1080.0.0.jpg

#### Questions?

Min



#### Thanks for your time. Got feedback? Tweet it!



3.4.4



All pictures belong to their respective authors





GOTO Guide

#### \*\*\*\*

# Remember to rate this session

THANK YOU!



Get IT ON Google Play

#GOTOams