



A Safari for Deviating GoF Pattern Definitions and Examples on the Web

Apostolos Zarras and Panos Vassiliadis

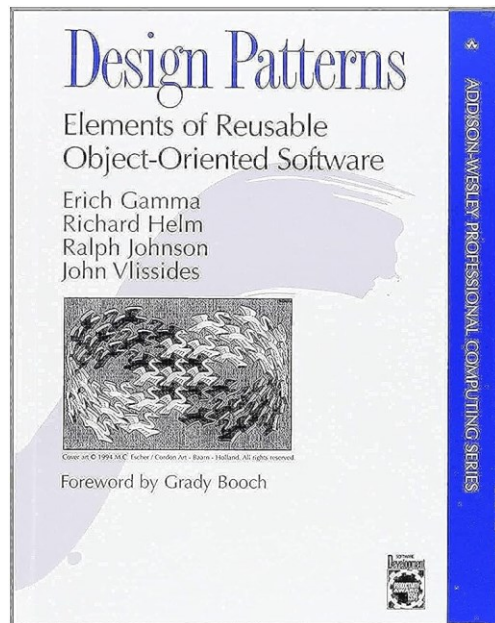
zarras@cs.uoi.gr, pvassil@cs.uoi.gr



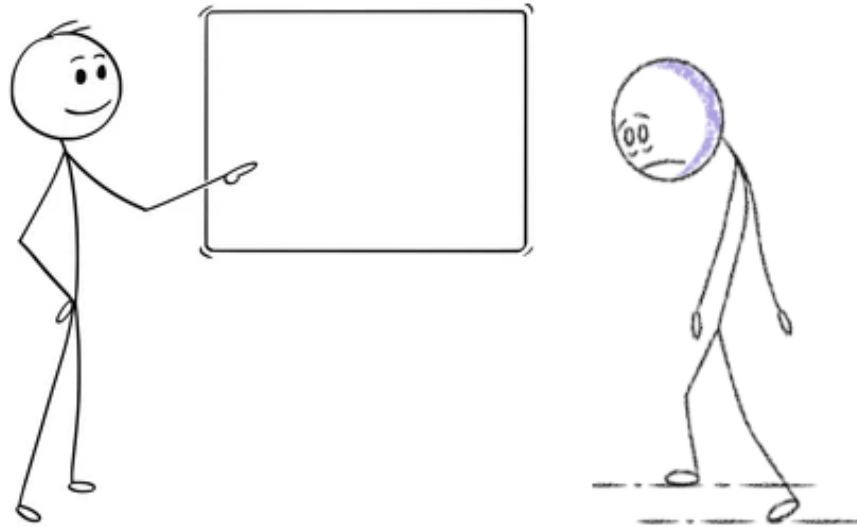
Dep. of Computer Science &
Engineering
University of Ioannina - Greece
www.cs.uoi.gr

GoF patterns

- ▶ **Design patterns** are descriptions of communicating **objects** and **classes** that are **customized** to solve a **general design problem** in a particular context.
- ▶ Various popular sources on the Web

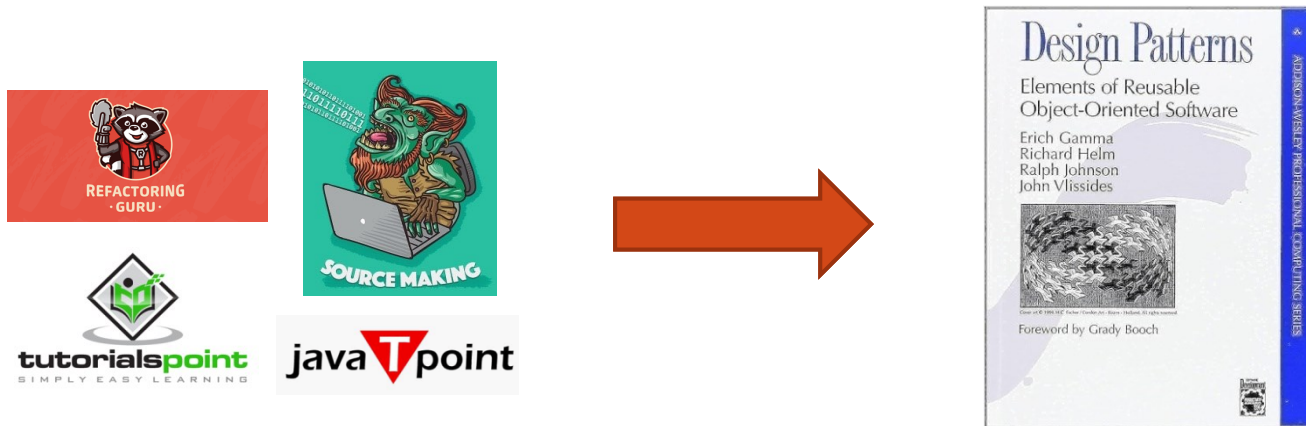


GoF patterns & the unfortunate student...



Sometimes, the **information we find on the Web** about the GoF patterns **deviates** from the **original pattern definitions**

Research goal



Assess the **compliance** of **pattern definitions** and **examples** we find on the **Web** to the **original GoF pattern definitions**.

Setup of the study

		Source Making		Refactoring Guru		Tutorials Point		Java T Point		ALL	
Corpus Patterns' Definitions and Examples		Definition	Example(s)	Definition	Example(s)	Definition	Example(s)	Definition	Example(s)	Definitions	Examples
Creational	Abstract Factory	✓	7	✓	10	✓	1	✓	1	4	19
	Builder	✓	5	✓	10	✓	1	✓	1	4	17
	Factory Method	✓	6	✓	10	✓	1	✓	1	4	18
	Prototype	✓	7	✓	10	✓	1	✓	1	4	19
	Singleton	✓	4	✓	10	✓	1	✓	1	4	16
Structural	Adapter	✓	6	✓	10	✓	1	✓	1	4	18
	Bridge	✓	5	✓	10	✓	1	✓	1	4	17
	Composite	✓	9	✓	10	✓	1	✓	1	4	21
	Decorator	✓	8	✓	10	✓	1	✓	1	4	20
	Façade	✓	5	✓	10	✓	1	✓	1	4	17
	Flyweight	✓	7	✓	10	✓	1	✓	1	4	19
	Proxy	✓	6	✓	10	✓	1	✓	1	4	18
Behavioral	Chain of Responsibility	✓	6	✓	10	✓	1	✓	1	4	18
	Command	✓	8	✓	10	✓	1	✓	1	4	20
	Interpreter	✓	5	×	0	✓	1	✓	1	3	7
	Iterator	✓	6	✓	10	✓	1	✓	1	4	18
	Mediator	✓	6	✓	10	✓	1	✓	1	4	18
	Memento	✓	5	✓	10	✓	1	✓	1	4	17
	Observer	✓	8	✓	10	✓	1	✓	1	4	20
	State	✓	9	✓	10	✓	1	✓	1	4	21
	Strategy	✓	5	✓	10	✓	1	✓	1	4	17
	Template Method	✓	5	✓	10	✓	1	✓	1	4	17
Visitor	✓	6	✓	10	✓	1	×	0	3	17	
Total		23	144	22	220	23	23	22	22	90	409

Setup of the study

Objectives

- ▶ Identify **kinds of deviations**.
- ▶ **Quantify compliance**:
 - ▶ **# of deviations** in pattern definitions/examples.
 - ▶ **% of deviating definitions/examples**.
 - ▶ **Density of deviations** in pattern definitions/examples.
 - ▶ **# of deviations / cardinality of examined set**.

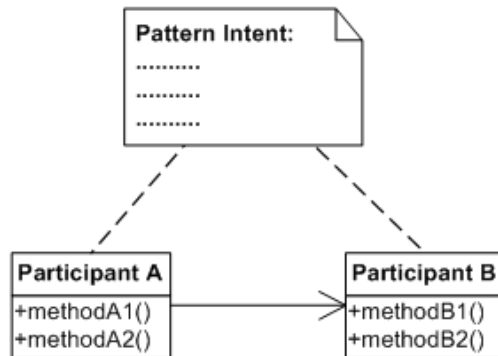
Comparison protocol

- ▶ **Intent** of pattern definitions/
- ▶ **Participants** in definitions and examples.
- ▶ **Methods** in definitions and examples.
- ▶ **Method implementations** in examples.

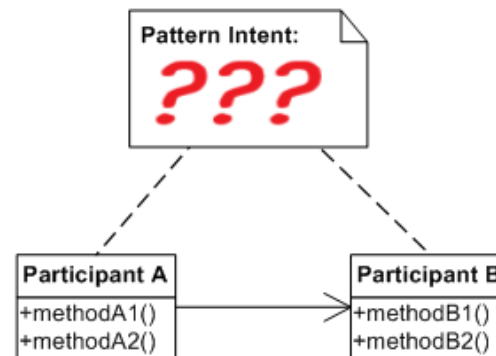
Kinds of deviations in pattern definitions & examples

Intent deviations

GoF definition



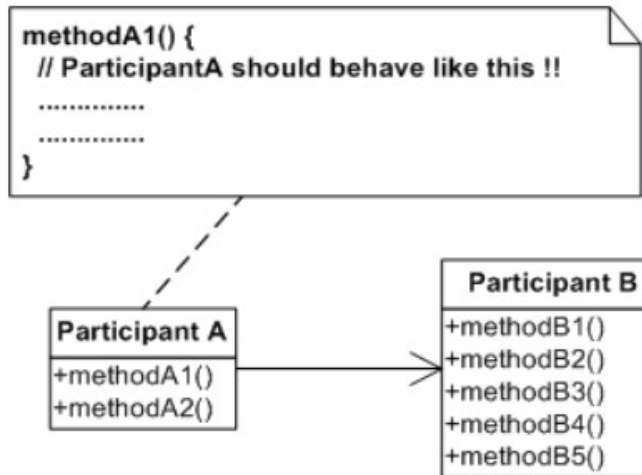
Site definition



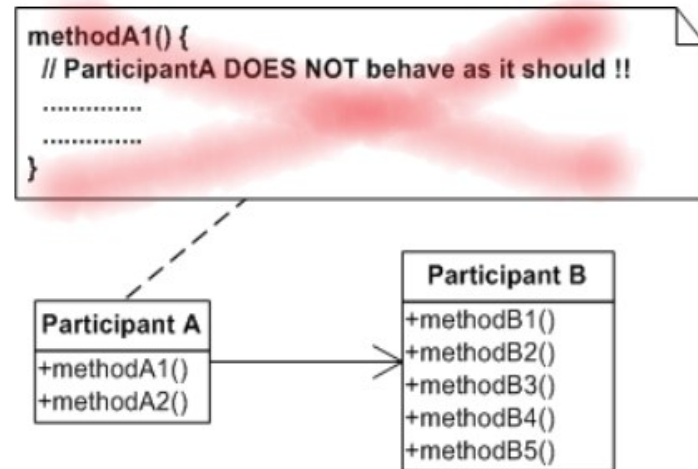
The **pattern definition** given in the Web site **does not reflect** the **original purpose** of the pattern....

Erroneous participants

GoF definition



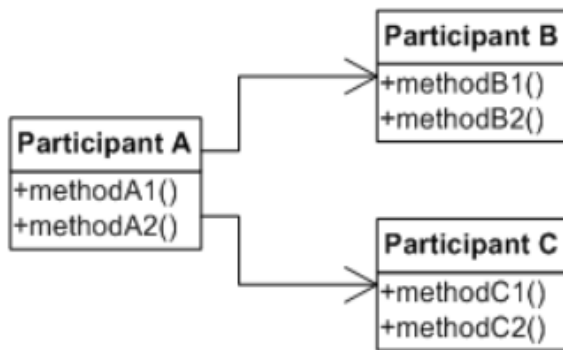
Site definition/example



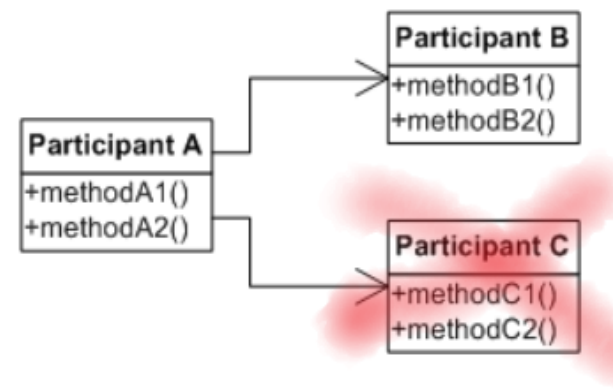
Participants **do not behave as they should** according to the original pattern definition.

Missing participants

GoF definition



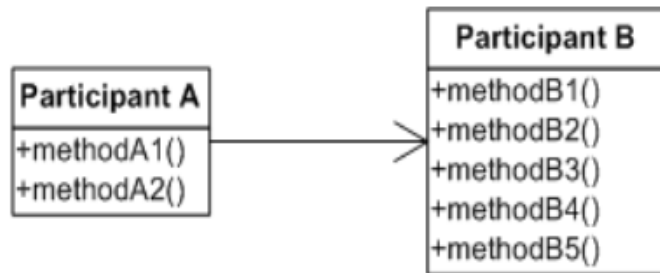
Site definition/example



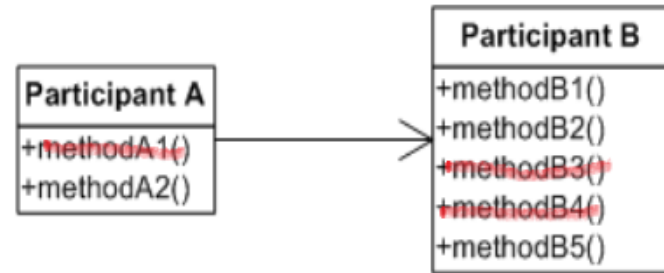
Definitions/examples **do not include all the participants** specified in the original pattern definition.

Incomplete participants

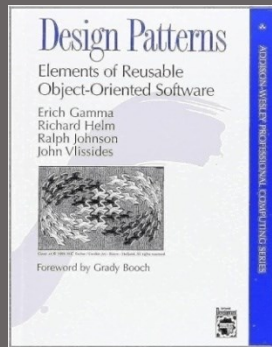
GoF definition



Site definition/example



Participants **do not provide** a **complete** and **exact set of methods** as specified in the original pattern definition.



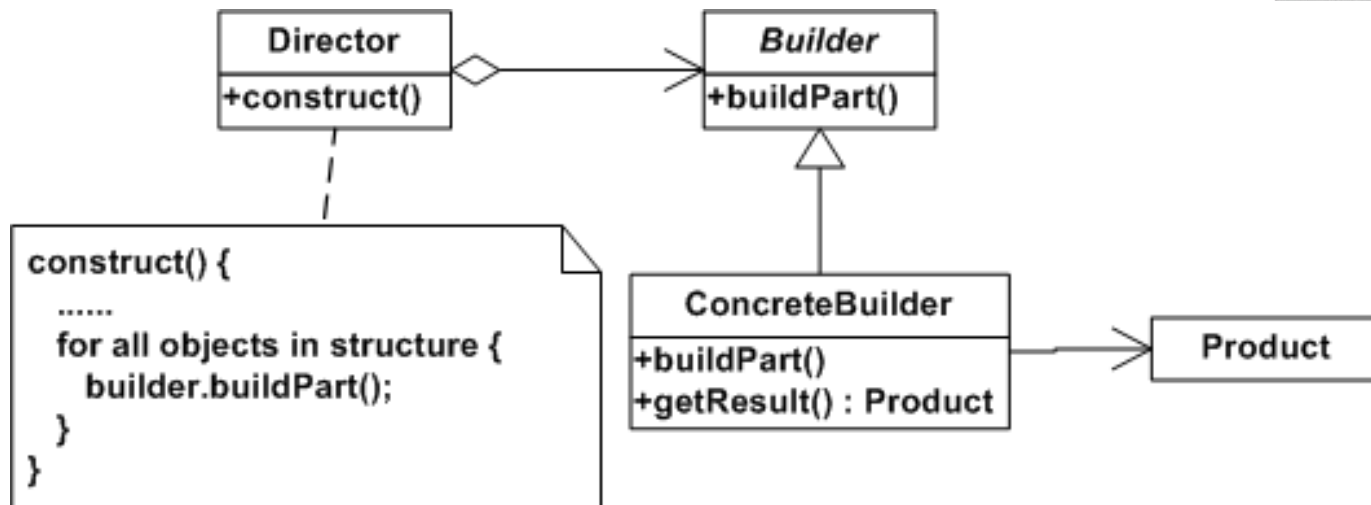
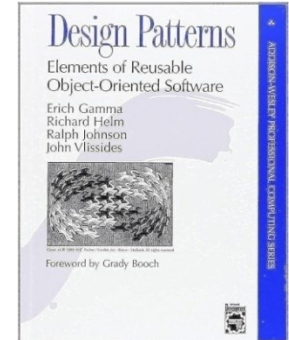
VS.



Builder Pattern

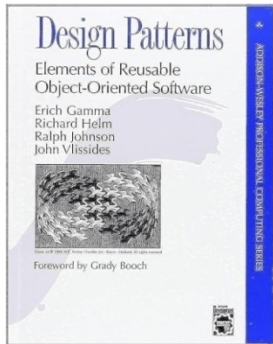
GoF Builder

“Separate the construction of a complex object from its representation so that the same construction process can create different representations.”



Intent Deviation:

Representation independence is gone !!!



"*Separate the **construction** of a **complex object** from its **representation** so that the same construction process can create different representations.*"

"*Builder pattern **builds** a **complex object** using **simple objects** and using a **step by step** approach. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object. A Builder class builds the final object step by step. This **builder** is **independent** of other **objects**.*"



Erroneous Participants: Builders the do not build!!

Director realizes **3 (clone) construction processes** which **depend** on the **internal representation** of the **constructed objects** !!!

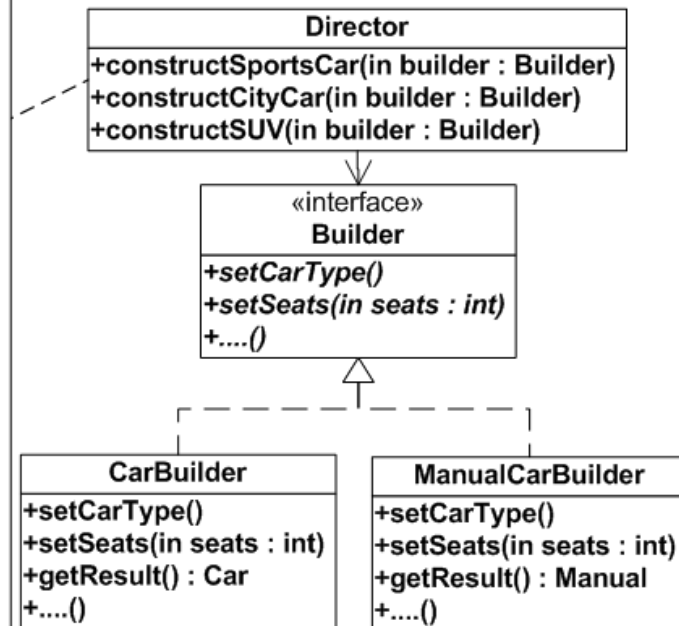


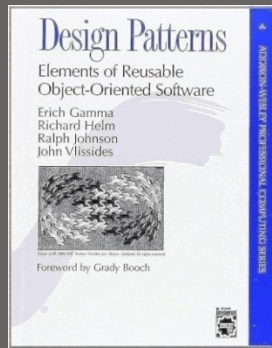
REFACTORING
· GURU ·

```
public void constructSportsCar(Builder builder) {
    builder.setCarType(CarType.SPORTS_CAR);
    builder.setSeats(2);
    builder.setEngine(new Engine(3.0, 0));
    builder.setTransmission(Transmission.SEMI_AUTOMATIC);
    builder.setTripComputer(new TripComputer());
    builder.setGPSNavigator(new GPSNavigator());
}

public void constructCityCar(Builder builder) {
    builder.setCarType(CarType.CITY_CAR);
    builder.setSeats(2);
    builder.setEngine(new Engine(1.2, 0));
    builder.setTransmission(Transmission.AUTOMATIC);
    builder.setTripComputer(new TripComputer());
    builder.setGPSNavigator(new GPSNavigator());
}

public void constructSUV(Builder builder) {
    builder.setCarType(CarType.SUV);
    builder.setSeats(4);
    builder.setEngine(new Engine(2.5, 0));
    builder.setTransmission(Transmission.MANUAL);
    builder.setGPSNavigator(new GPSNavigator());
}
```





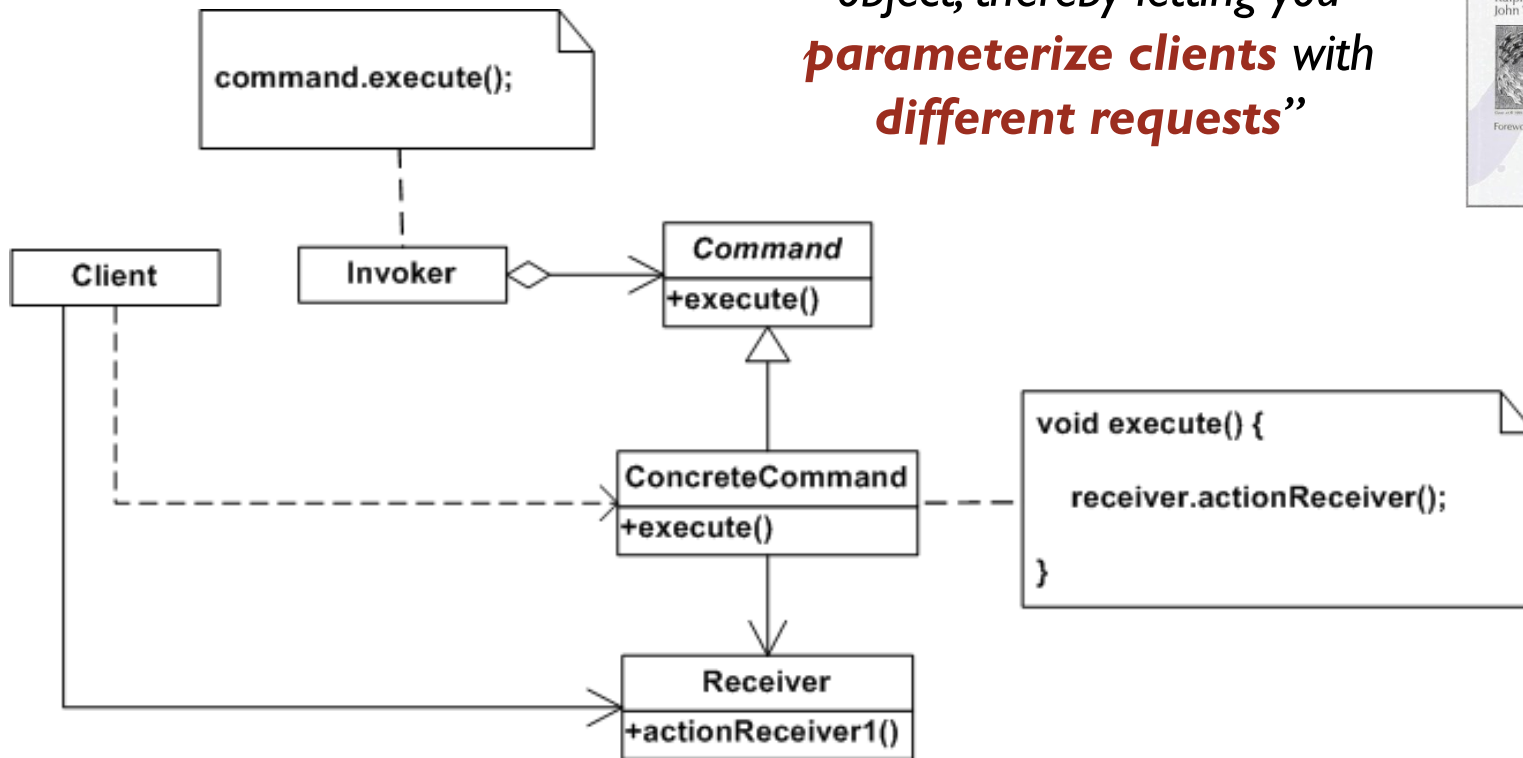
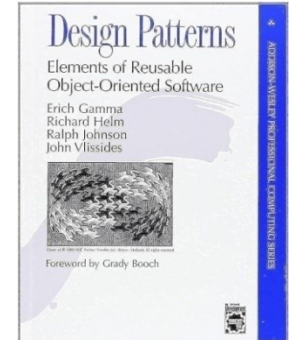
VS.



Command Pattern

GoF Command

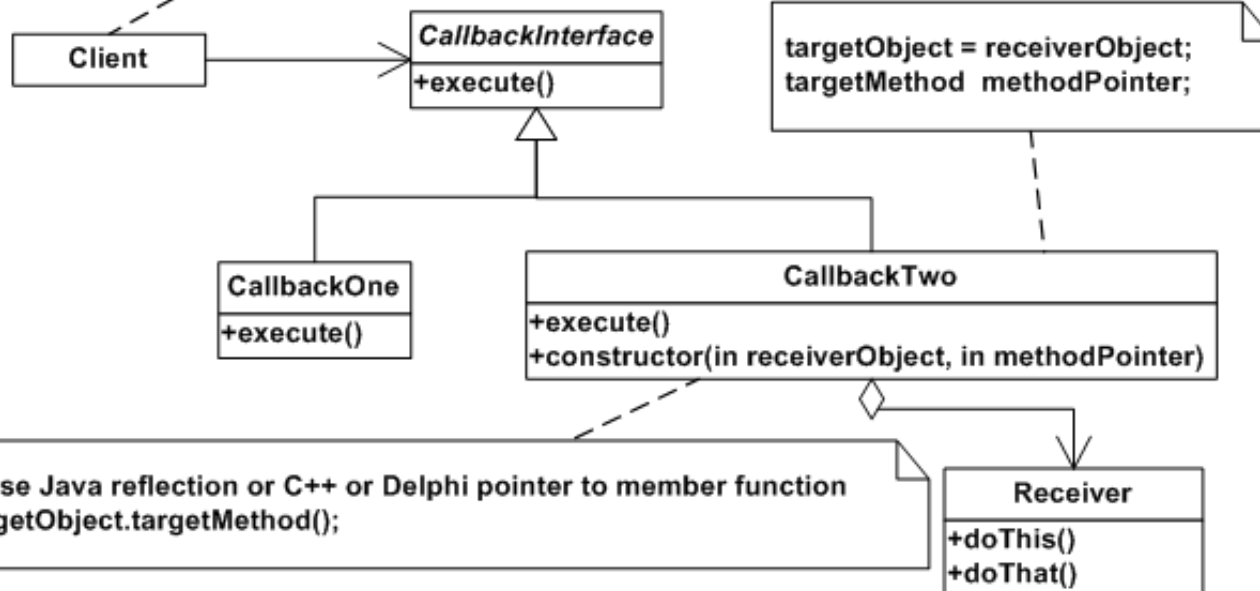
“encapsulate a request as an object, thereby letting you parameterize clients with different requests”



Missing Participant: Parameterization is Gone !!!

MISSING Invoker !

```
CallbackInterface token = new CallbackTwo(new Receiver(), "dothis");  
// the token object is passed to another object and that object calls  
token.execute();
```





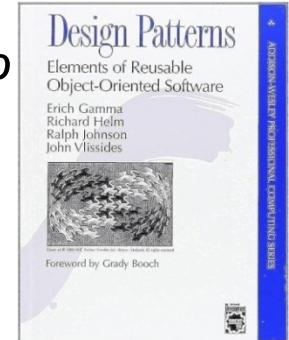
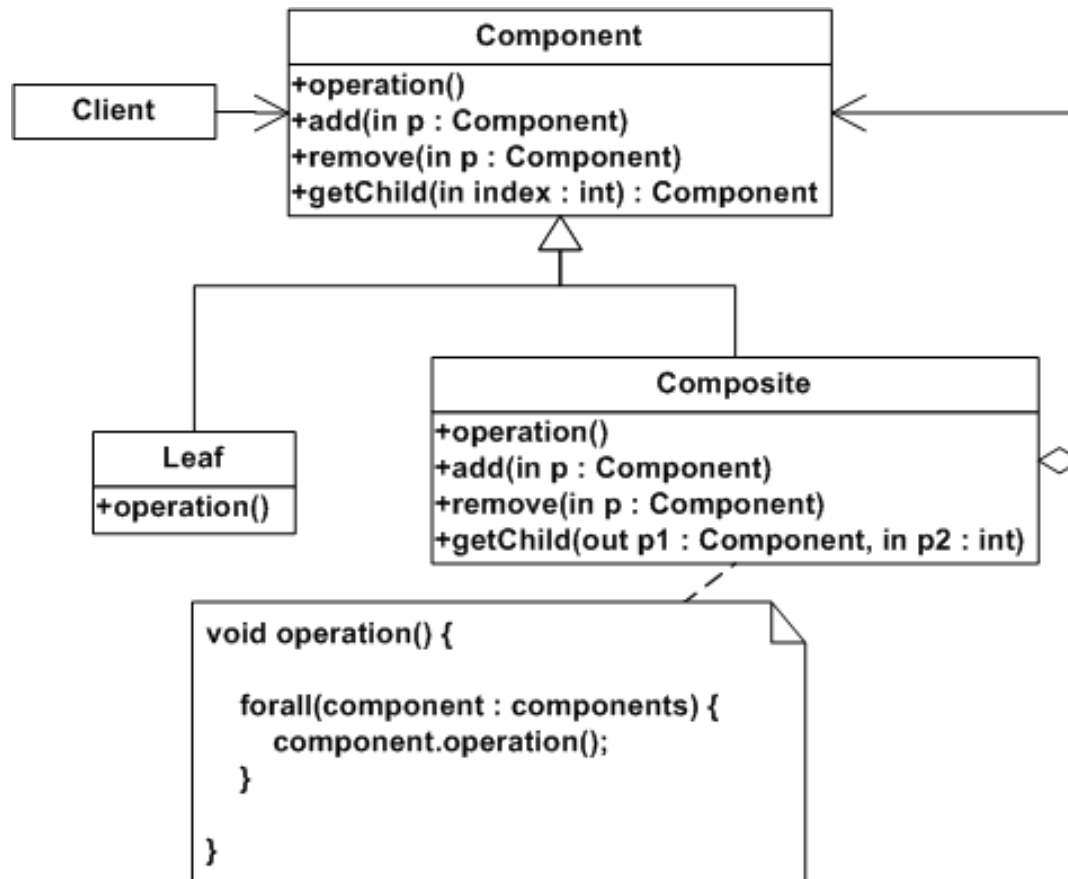
VS.



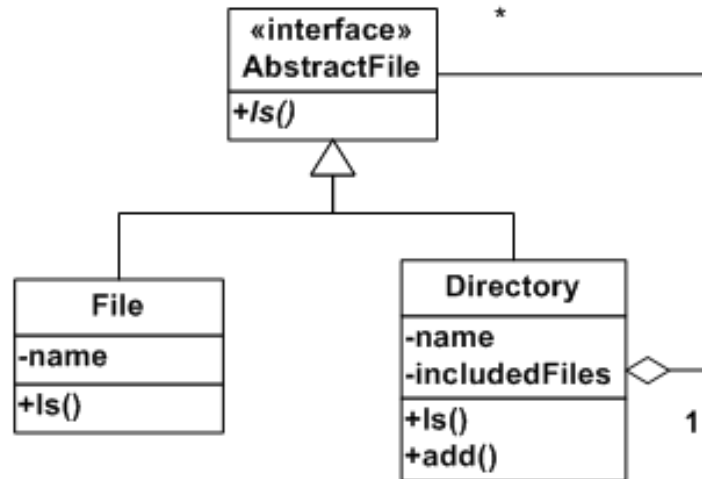
Composite Pattern

GoF Composite

“**compose objects** into tree structures to represent **part-whole hierarchies**”



Incomplete participant: Retrieving/removing parts not possible !!!



MISSING
remove(), getChild() !

```
public void ls() {
    .....
    .....
    for (Object includedFile : includedFiles) {
        // Leverage the "lowest common denominator"
        AbstractFile obj = (AbstractFile) includedFile;
        obj.ls();
    }
    .....
    .....
}
```

Compliance of pattern definitions & examples

Compliance of definitions

Deviations in pattern definitions				
Patterns	Source Making	Refactoring Guru	Tutorials Point	Java T Point
Sum	8	7	3	1
Density of deviations in pattern definitions	0.35	0.32	0.13	0.05
% Patterns with deviating definitions	30.43%	27.27%	13.04%	9.09%

- **The majority of the definitions adhere** to the original pattern definitions.
- **Patterns with deviating definitions are not frequent.**
- The **density** of deviations in the definitions is **low**.

Compliance of definitions

Kinds of deviations in pattern definitions

	Source Making		Refactoring Guru		Tutorials Point		Java T Point	
	# deviations	density	# deviations	density	# deviations	density	# deviations	density
Intent Deviations	0	0.00	0	0.00	3	0.13	1	0.05
Missing Participant	7	0.30	4	0.18				
Incomplete Participant	1	0.04	3	0.14				

- The **density** of **intent deviations** is **low**.
- The **density** of **missing participants** is **higher** than the **density** of **incomplete participants**.
- **3/11** missing participants → **incorrect definitions**.

Compliance of examples

Deviations in pattern examples								
	Source Making		Refactoring Guru		Tutorials Point		Java T Point	
Patterns	# deviations	# examples	# deviations	# examples	# deviations	# examples	# deviations	# examples
Sum	154	143	104	220	21	23	24	22
Density of deviations in pattern examples	1.08		0.47		0.91		1.09	
% Patterns with deviating examples	86.36%		52.17%		50.00%		63.64%	

- **Patterns with deviating examples** are quite **frequent**.
- The **density** of **deviating examples** is **medium high**.
- In **all sites** the **% of patterns** with **deviating examples** is **higher** than the **% of patterns** with **deviating definitions**.

Compliance of examples

Kinds of deviations in pattern examples								
	Source Making		Refactoring Guru		Tutorials Point		Java T Point	
	# deviations	density	# deviations	density	# deviations	density	# deviations	density
Missing Participant	108	0.76	49	0.22	14	0.61	15	0.68
Incomplete Participant	28	0.20	45	0.20	4	0.17	5	0.23
Erroneous Participant	17	0.12	10	0.05	3	0.13	6	0.27

- **Missing participants** occur **more often**, then we have **incomplete** and **erroneous** participants.
- The **number** of **missing participants** is **high**.
- The **numbers** of **incomplete** and **erroneous participants** are **low**.
- **60/186** of **missing participants** → **incorrect examples**.
- **20/82** of **incomplete participants** → **incorrect examples**.

Takeaway messages for the developers

Takeaway messages



- **Pattern definitions/examples** you find on the Web may **deviate** from the **original**.
- Watch out for **different kinds of deviations: intent** deviations, **missing**, **incomplete**, and **erroneous** participants.
- Their **impact varies**:
 - **Intent** deviations and **erroneous** participants result to **incorrect definitions/examples**.
 - **Missing** and **incomplete participants** may result to **incorrect** or **incomplete** definitions/examples.
- Be **more concerned** about deviating **examples** than **definitions**.
- The **choice** of the **site** is **important**, depending on what you are looking for.



Auxiliary slides

Deviations in pattern definitions				
Patterns	Source Making	Refactoring Guru	Tutorials Point	Java T Point
Abstract Factory	0	0	1	0
Builder	0	0	1	0
Factory Method	0	0	0	0
Prototype	0	0	0	0
Singleton	0	0	0	0
Adapter	0	0	0	0
Bridge	0	0	0	0
Composite	1	1	0	0
Decorator	0	0	0	0
Façade	0	0	0	0
Flyweight	1	2	0	0
Proxy	0	0	0	0
Chain of Resp	1	0	0	0
Command	1	0	0	0
Interpreter	0		0	0
Iterator	0	2	0	0
Mediator	2	1	0	0
Memento	0	0	0	0
Observer	1	1	0	1
State	0	0	0	0
Strategy	0	0	0	0
Template Method	0	0	0	0
Visitor	1	0	1	
Sum	8	7	3	1
Density of deviations in pattern definitions	0.35	0.32	0.13	0.05
% Patterns with deviating definitions	30.43%	27.27%	13.04%	9.09%

Deviations in pattern examples								
	Source Making		Refactoring Guru		Tutorials Point		Java T Point	
Patterns	# deviations	# examples	# deviations	# examples	# deviations	# examples	# deviations	# examples
Abstract Factory	12	7	0	10	2	1	3	1
Builder	1	5	1	10	3	1	3	1
Factory Method	5	6	1	10	1	1	1	1
Prototype	5	7	7	10	0	1	1	1
Singleton	1	4	0	10	0	1	0	1
Adapter	8	6	7	10	1	1	1	1
Bridge	0	5	0	10	0	1	0	1
Composite	19	9	20	10	4	1	3	1
Decorator	2	8	1	10	0	1	0	1
Façade	0	5	0	10	0	1	0	1
Flyweight	13	7	20	10	1	1	0	1
Proxy	3	6	0	10	0	1	0	1
Chain of Resp	10	6	0	10	1	1	1	1
Command	9	7	1	10	0	1	0	1
Interpreter	8	5			0	1	3	1
Iterator	17	6	23	10	2	1	2	1
Mediator	15	6	0	10	3	1	2	1
Memento	3	5	11	10	1	1	1	1
Observer	16	8	12	10	2	1	1	1
State	4	9	0	10	0	1	2	1
Strategy	3	5	0	10	0	1	0	1
Template Method	0	5	0	10	0	1	0	1
Visitor	0	6	0	10	0	1		
Sum	154	143	104	220	21	23	24	22
Density of deviations in pattern examples	1.08		0.47		0.91		1.09	
% Patterns with deviating examples	86.36%		52.17%		50.00%		63.64%	