

# Design Patterns

M. Blay-Fornarino & S. Urli

# Motivations ... (toujours les mêmes)

- Faire une conception et un développement de qualité :
  - Extensibilité
  - Flexibilité
  - Maintenabilité
  - Réutilisabilité
  - Clarté

# Toujours pas de recette «magique»

... mais outre les bonnes pratiques :

- GRASP
- SOLID
- KIS : Keep It Simple
- DRY : Don't Repeat Yourself
- YAGNI : You Ain't Gonna Need It

**Patron de conception : un outil supplémentaire**



**Martin Fowler** ✓

@martinfowler



Suivre

The same solution can be a good pattern in some contexts and an antipattern in others. Its value depends on the context that you use it in

Voir la traduction

RETWEETS

**483**

J'AIME

**381**



13:12 - 8 juin 2016



483

381



Répondre à @martinfowler



**Richard Bishop** @bitsandhops · 8 juin

@martinfowler You mean I have to think? This is an outrage!



2

2



**Lee Hawkins** @therockertester · 9 juin

@martinfowler @charrett So you're suggesting there's no such thing as "best practices"? Tell that to those trying to standardize testing



2

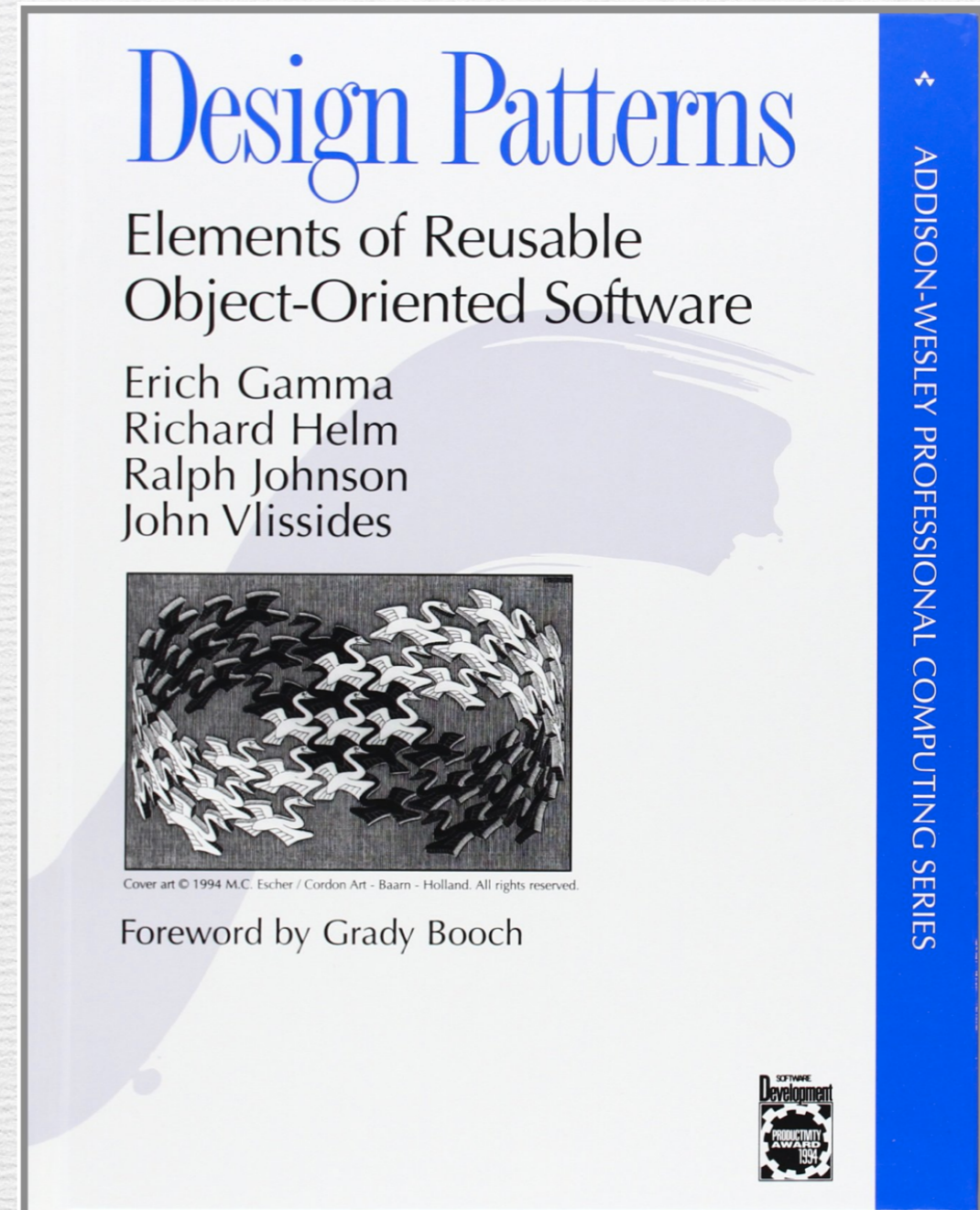


**The Guy** @Guygies · 8 juin

@martinfowler also known as the "don't use your Lamborghini to transport horse manure" rule.

# Patrons de conception?

- “ Généralisation d’une solution à un problème de conception récurrent par la description des classes et objets communicants ”



# Patrons de conception? Comment sont-ils définis?

- ➔ Identification d'un problème de conception récurrent
- ➔ Schématisation du problème de manière générique
- ➔ Description d'une solution sous la forme d'un patron

# Patrons de conception? Comment les utiliser?

- ➔ Identifier un problème dont le motif a fait l'objet d'une solution
- ➔ Rechercher le patron de conception adapté
- ➔ Appliquer et adapter la solution proposée par le patron de conception

# Historique

- ❖ 1977/79 : Christopher Alexander - Patron de conception pour l'architecture des villes et bâtiments
- ❖ 1987 : Kent Beck et Ward Cunningham - Papier à OOPSLA sur l'utilisation de patrons de conception pour la programmation orientée objet
- ❖ 1994 : le «Gang of Four» (GoF) (Gamma, Helm, Johnson and Vlissides) publie Design Patterns: Elements of Reusable Object-Oriented Software - Présentation des 23 patterns fondamentaux



# Classification des patrons

## ❧ Patrons de **création** :

- ➔ dédiés à la création des objets.
- ➔ visent l'indépendance entre création et utilisation des objets.

## ❧ Patrons de **structure** :

- ➔ dédiés à la composition des objets.
- ➔ visent à conserver une bonne séparation des préoccupations.

## ❧ Patrons de **comportement** :

- ➔ dédiés à la communication entre les objets
- ➔ visent à conserver de la flexibilité dans les liens de communication.

# Classification

## ❧ 23 Patrons fondamentaux

### ➔ Patrons de création :

- ◉ *Abstract Factory*, Builder, Factory method,
- ◉ Prototype, Singleton

### ➔ Patrons de structure :

- ◉ Adapter, Bridge, *Composite*, Decorator, Facade, Flyweight, Proxy

### ➔ Patrons de comportement :

- ◉ Chain-of-responsibility, Command, Interpreter, Iterator, Mediator, Memento, *Observer*, State, Strategy, Template Method, Visitor

# Sources

- ❧ Les design patterns en Java - Steven John Metsker, William C. Wake
- ❧ Head first design patterns - Eric et Elisabeth Freeman
- ❧ The design pattern Smalltalk Companion
- ❧ Sherman R. Alpert, Kyle Brown, Bobby Woolf
- ❧ <http://blog.codinghorror.com>
- ❧ [http://en.wikipedia.org/wiki/Software\\_design\\_pattern](http://en.wikipedia.org/wiki/Software_design_pattern)