Génie logiciel pour la conception d'un Système d'Information

CSC4521

Voie d'Approfondissement
Intégration et Déploiement de Systèmes d'Information
(VAP DSI)

Design Patterns

http://jpaulgibson.synology.me/~jpaulgibson/TSP/Teaching/CSC4521/

.../CSC4521/CSC4521-DesignPatterns.pdf

Design Patterns - *Les patrons de conception*

http://3.bp.blogspot.com/-4aAqfjcB-zA/UdLDsWo7X3I/AAAAAAAABXk/FDui7etrEGo/s400/designpatterns02.jpg

Un design pattern -

- *décrit une structure commune et répétitive de composants en interaction (la solution) qui résout un problème récurrent de conception dans un contexte particulier*
- *est une technique avancée de programmation (Objet)*
- *est un outil (standard) dans la conception (orientée-objet)*

La Conception (Orientée-Objet)

- Un art difficile...

- Une conception
  réutilisable, extensible, adaptable, performante, …
  est extrêmement difficile

- Novice v concepteur expérimenté, typiquement:
  1. le novice hésite beaucoup entre différentes variantes
  2. l'expert trouve « tout de suite » la bonne solution

- Pourquoi la différence ? ….

  l'expérience

Design Patterns - *Les patrons de conception*

Expertise en Conception (Orientée-Objet)

- ne pas réinventer la roue
- réutiliser systématiquement des solutions qui ont fait leurs preuves
- pour une bonne conception (modulaire, élégante, adaptable …)

… mais comment ???...

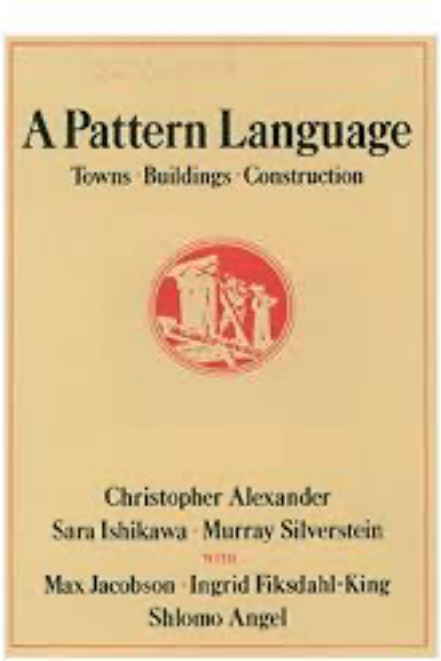
répétition de certains profils de classes ou collaboration d'objets: *design patterns, modèles de conceptions, patrons de conception, micro-architectures*

Design Patterns - *Les patrons de conception*

Origines

1970s - Travaux de l'architecte Christopher Alexander: a perfectionné la théorie des « ***Pattern languages*** » utilisée dans plusieurs domaines –

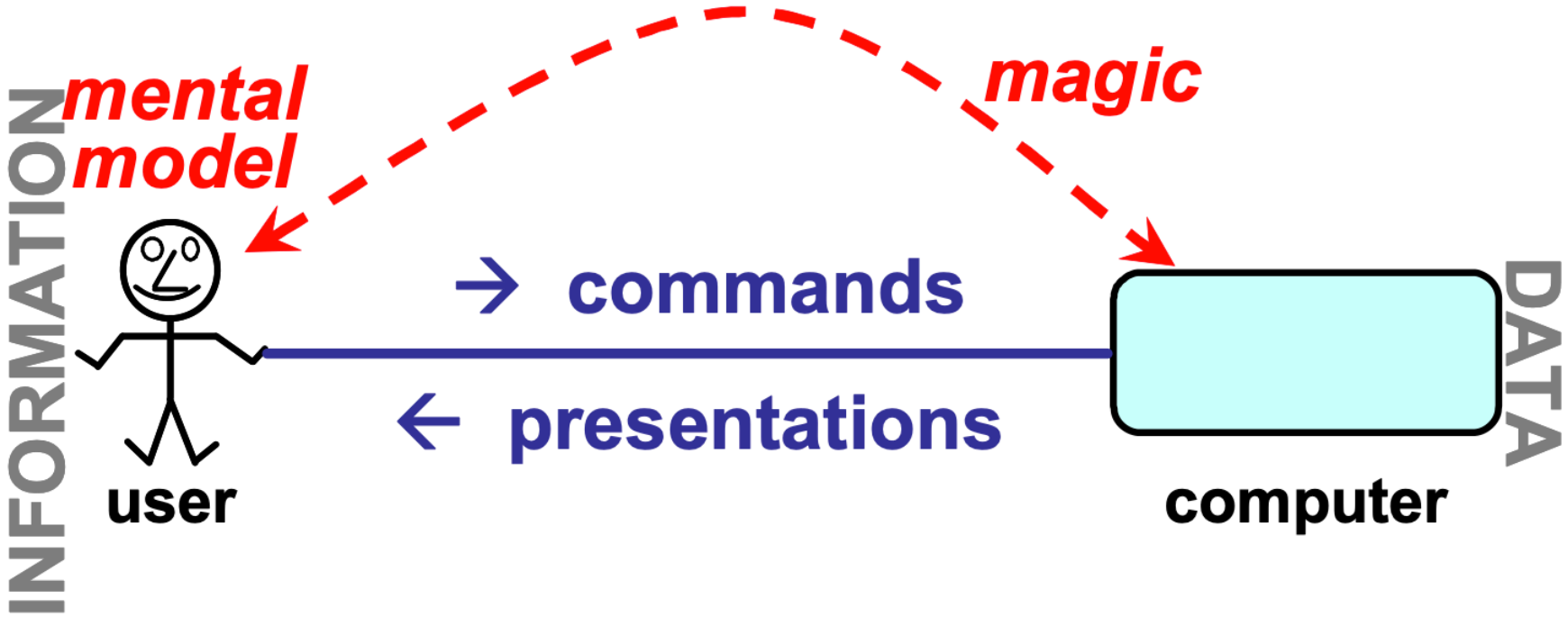
de l'anthropologie et de l'histoire de l'art, puis dans celui du design (les types architecturaux), ensuite en informatique



Origines

1980s –

Exemple «historique» du modèle **MVC** (Model-View-Controller) de Smalltalk (Trygve Reenskaug [Xerox Parc]& Krasner, Pope)



Design Patterns - *Les patrons de conception*

Origines

1990s - Erich Gamma, Richard Helm, Ralph Johnson
et John Vlissides - Gang of Four (**GoF**) *Design Patterns: Elements of Reusable Object-Oriented Software*

Design Patterns - *Les patrons de conception*

Plus récement (depuis 2010)

UML and patterns

Web 2.0: Design Patterns and Business Models

Agile software development: principles, patterns, and practices

Design Patterns and Aspects

Design Patterns for Concurrent programming

Design Patterns for Services and Web Applications

Formal Specification of Design Patterns

Software factories/product lines with patterns

Design patterns for IOT

Big data patterns…

AI design patterns

10 Trending Design Patterns in IoT Solutions and Architectures

Publisher/Subscriber
Digital Twin
Sensor Aggregation
Gateway (Hub & Spoke Architecture)
Edge/ Fog
Mesh Architecture
Event-Driven Architecture
Data Lake
Time-Series Data Architecture
Command and Control

https://www.linkedin.com/pulse/10-trending-design-patterns-iot-solutions-vishal-bhardwaj/

« Définitions »

Alexander :

As an element in the world, each pattern is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves.

As an element of language, a pattern is an instruction, which shows how this spatial configuration can be used, over and over again, to resolve the given system of forces, wherever the context makes it relevant.

Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you ***can use this solution a million times over, without ever doing it the same way twice.***

« Définitions »

Pree and Sikora :

Design Patterns describe [software] framework construction on ***a level higher than the underlying programming language.***

Because patterns have become the vogue in the software engineering community, the term now is used wherever possible, adorning even project management or organizational work. So the genericity of the term pattern might be the reason that patterns are found everywhere, a fact which is regarded as a clear indication of a hype.

"Design Patterns for Object-Oriented Software Development", 1997

Erich Gamma (GoF):

Patterns provide you with tools that help you with design problems. They do so not by giving a pat solution but by explaining trade-offs. Even though patterns are abstracted from concrete uses, they also provide you valuable implementation hints. From my perspective it is the fact that patterns are implementable that makes them so valuable.

Patterns are distilled from the experiences of experts. They enable you to repeat a successful design done by someone else. However, since patterns enable many implementation variations you still have to keep the brain turned on.

Since patterns provide you with names for design building blocks they provide you with a vocabulary to describe and discuss a particular design.

I think patterns as a whole can help people learn object-oriented thinking: how you can leverage polymorphism, design for composition, delegation, balance responsibilities, and provide pluggable behavior.

« Définitions »

Wikipedia: **http://fr.wikipedia.org/wiki/Patron_de_conception**

« Les patrons de conception décrivent des solutions standard pour répondre à des problèmes d'architecture et de conception des logiciels. À la différence d'un algorithme qui s'attache à décrire d'une manière formelle comment résoudre un problème particulier, les patrons de conception décrivent des procédés de conception généraux. On peut considérer un patron de conception comme une formalisation de bonnes pratiques, ce qui signifie qu'on privilégie les solutions éprouvées.
Il ne s'agit pas de fragments de code, puisque les patrons de conception sont le plus souvent indépendants du langage de programmation, mais d'une méthode de conception, c'est-à-dire d'une manière standardisée de résoudre un problème qui s'est déjà posé par le passé. Le concept de patron de conception a donc une grande influence sur l'architecture logicielle d'un système.

On peut donc considérer les patrons de conception comme un outil de capitalisation de l'expérience appliqué à la conception logicielle. »

Les éléments d'un patron de conception

1. Nom:
   identification d'un concept +…
2. Problème:
   situations dans lesquelles le patron s'applique +…
3. Solution:
   éléments du modèle de conception +…
4. Conséquences:
   effets de l'application du modèle sur la conception +…

Le catalogue GoF patterns (3 catégories)

Création

Un patron de création permet de résoudre les problèmes liés à la création et la configuration d'objets.

Structure

Un patron de structure permet de résoudre les problèmes liés à la structuration des classes et leur interface en particulier.

Comportement

Un patron de comportement permet de résoudre les problèmes liés aux interactions entre les classes.

Design Patterns - *Les patrons de conception*

Le catalogue GoF (23 exemples) - **those used in previous projects**

Création

* **Fabrique abstraite (Abstract Factory)** * Monteur (Builder)
* Fabrique (Factory Method) * Prototype (Prototype)
* **Singleton (Singleton)**

Structure

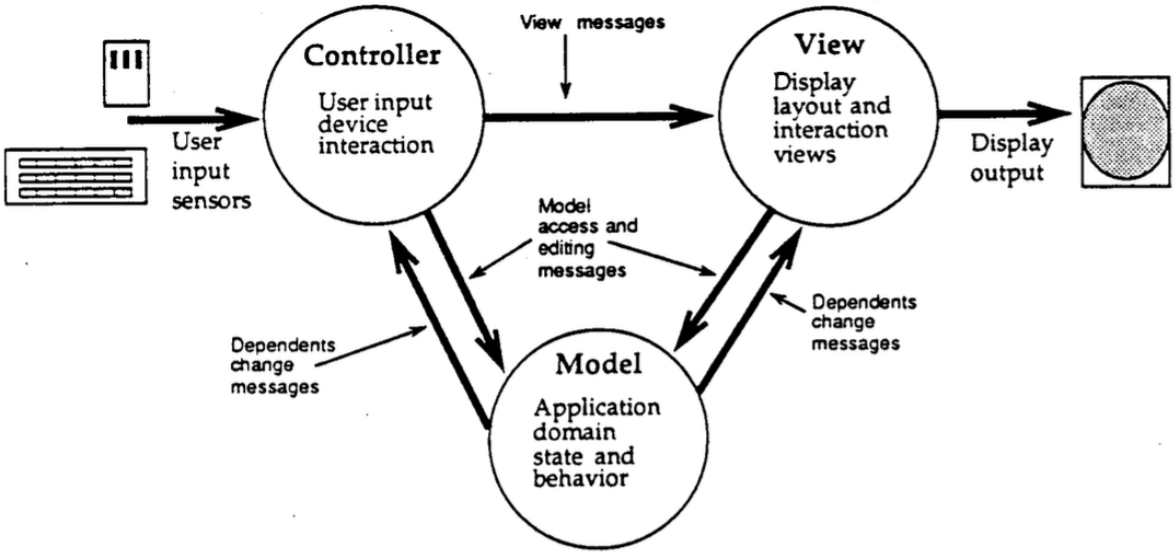
* Adaptateur (Adapter) * Pont (Bridge) * Objet composite (Composite)
* **Décorateur (Decorator)** * **Façade (Facade)**
* Poids-mouche ou poids-plume (Flyweight) * **Proxy (Proxy)**

Comportement

* **Chaîne de responsabilité (Chain of responsibility)**
* Commande (Command) * Interpréteur (Interpreter) * **Itérateur (Iterator)**
* Médiateur (Mediator) * Memento (Memento) * **Observateur (Observer)**
* État (State) * **Stratégie (Strategy)**
* Patron de méthode (Template Method) * **Visiteur (Visitor)**

Pattern Composition

Exemple: Le patron Modèle-Vue-Contrôleur (**MVC**) est souvent implementé en utilisant une « combinaison » des patrons *Observateur*, *Stratégie* et *Composite*



A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80, Krasner and Pope, 1988.

Stratégie: controller-view

Composite: view-view

Observateur: model-view, model-controller

Design Patterns - *Les patrons de conception*

The work to come:

The best way to learn about patterns (and pattern compositions) is to look at examples.

We shall do these in Java, but any (OO) language can be used to implement/re-use a pattern.

Today, we will look at 2 patterns useful in your projects:

- Singleton
- MVC

We will look at other patterns as we/you need them