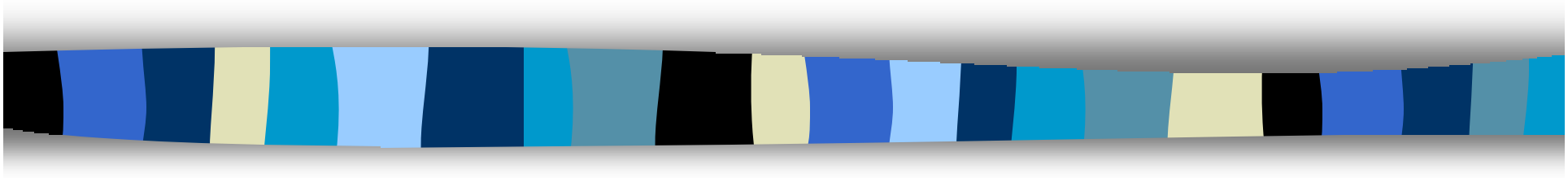
IFT3051 - Été 2004
Patrons de conception



Duc-Loc Huynh

Janice Ka-Yee Ng

à

Yann-Gaël Guéhéneuc

Houari Sahraoui

Département d'informatique et de recherche opérationnelle

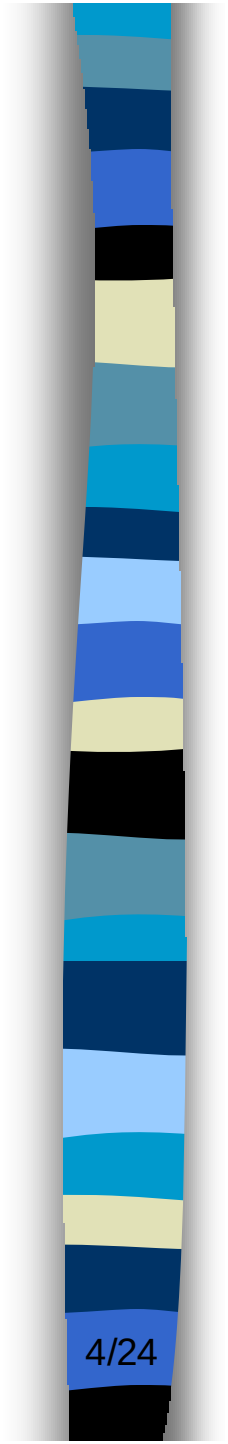
Université de Montréal

Pourquoi les patrons de conception ?

- La programmation orientée-objet a introduit une nouvelle perspective de concevoir les logiciels dans ce monde chaotique qu'est le codage
- En fait cette perspective a été introduite pour des raisons logistiques et économiques (maintenance)

Notre objectif….

- [À part d'obtenir un A+ pour notre cours]
- Identifier les patrons de conception que les programmeurs auraient éventuellement utiliser pour concevoir un logiciel
- Proposer des recettes « magiques »pour identifier les patrons



Notre aventure dons le monde merveilleux des Patrons de Conception ☺

Étape 1 - Méthodologie

- Initialement, notre vision de la « Bête » a été introduite par nos cours de génie logiciel. Donc, très abstraite…
- Ainsi au début nous nous sommes documentés de façon aléatoire sur l'Internet
- Évidemment, le livre « Design Patterns » du GoF a constitué principalement notre « Phare d'Alexandrie »

Étape 2 - Choix du programme

- Java Monopoly : logiciel de petite taille (environ 20 classes)
- Nous avons choisi ce logiciel afin de nous familiariser avec le concept de patron de conception. En fait, nous pensions que l'identification de patrons serait plus facile dans un programme de petite taille

Étape 2 - Choix du programme (suite)

- **Mapper XML** : logiciel de taille moyenne (environ 200 classes)
- Après notre échec avec le programme précédent, nous nous sommes dirigés vers un logiciel dont la structure est basée sur les patrons de conception
- En fait, la présence de patrons de conception était précisée dans la documentation

Étape 2 - Méthodologie

[Java Monopoly et Mapper XML]

- De façon indéterministe, nous avons commencé à l'ancienne !

a. Nous nous sommes fiés aux noms de classes pour définir leur appartenance à une structure de patron

b. Papier et crayon pour mettre en évidence les relations inter-classes

c. Établissement des correspondances entre les diagrammes obtenus et ceux du livre

d. Une fois un patron identifié, nous nous référons au diagramme des relations entre patrons de conception dans l'espoir de trouver d'autres bêtes

e. Mise à jour du XML pour des fins de comparaisons entre nous

Quelques illustrations…

- *Resultats.xml*

Étape 3 - Choix du programme

- **PMD** et **Nutch** : logiciels de taille moyenne (environ 200 classes)
- Encore plus fort, maintenant que nous nous sentons plus à l'aise envers les patrons, nous avons pris les deux premiers programmes de taille moyenne sans nous préoccuper de la certitude de la présence de patrons
- Une difficulté qui s'est ajoutée dans notre tâche d'identification réside en la présence de sous-classes

Étape 3 - Méthodologie

[PMD et Nutch]

a. Recherche des interfaces et classes abstraites en fonction des relations entre patrons
b. Établir les relations inter-classes à partir de ces dernières
c. À partir des noms de classes ainsi que de leurs méthodes, nous pouvons nous risquer à supposer l'appartenance de ces structures à un groupe de patrons

Étape 3 - Méthodologie (suite)

[PMD et Nutch]

d. Ensuite par la silhouette du diagramme, nous comparons les patrons du groupe afin de déterminer LA créature
e. Puis pour des fins de confirmation, nous observons les relations inter-méthodes
f. À ce point nous sommes assurés à 99% de l'exactitude du patron
g. Mise à jour du XML pour des fins de comparaisons entre nous

Quelques illustrations…

- Dans PMD…
  - Facade
  - Iterator
- Dans Nutch…
  - Iterator
  - Command

Étape 4 - Choix du programme

- **Azureus** : logiciel de taille ÉNORME (environ 1300 classes, sans compter les sous-classes)
- Nous avons voulu finir le projet avec un coup d’éclat ;-)

Étape 4 - Méthodologie

[Azureus]

- Nos armes : papier, crayon et Visio
- Comme à l'étape 3…

a. Recherche des interfaces et classes abstraites

b. Établir les relations inter-classes à partir de ces dernières

c. À partir des noms de classes ainsi que de leurs méthodes, nous pouvons nous risquer à supposer l'appartenance de ces structures à un groupe de patrons

Étape 4 - Méthodologie (suite)

[Azureus]

- Ensuite la Vocation !
- Plus besoin de scruter le livre autant qu'avant
- À force d'user les pages du livre, nous avons fusionné notre esprit avec celui des patrons
- Nous avons toujours besoin de dessiner les diagrammes de classes, mais les vas et viens entre les schémas obtenus et ceux du livre ont significativement diminué
- En fait, nous avons développé un esprit critique envers les structures des différents logiciels

Résultats

- Mapper XML
  - Abstract Factory (1)
  - Adapter (2)
  - Composite (1)
  - Facade (1)
  - Factory Method (1)
  - Observer (1)
  - Singleton (3)
  - Strategy (1)
  - Template Method (4)

Résultats (suite)

- Nutch
  - Singleton (1)
  - Bridge (2)
  - Command (2)
  - Memento (2)
  - Template Method (3)
  - Adapter (2)
  - Strategy (2)
  - Iterator (1)

Résultats (suite)

- PMD
  - Adapter (1)
  - Builder (2)
  - Composite (2)
  - Facade (1)
  - Factory Method (3)
  - Iterator (1)
  - Proxy (1)
  - Template Method (1)
  - Observer (2)
  - Visitor (1)

Résultats

- Azureus
  - Flyweight (1)
  - Interpreter (2)
  - Visitor (1)
  - State (1)
  - Factory Method (1)
  - Template Method (1)
  - Bridge (1)

Conclusion

- La procédure d'identification de patrons de conception n'est pas encore une tâche formelle
- Plus d'une fois, nous avons identifié des variantes de patrons. La structure divergeait de celle proposée par le GoF, mais leur comportement possédait les mêmes caractéristiques que celles du livre

Conclusion

- L’identification de patrons de conception, que nous avons effectué au cours de ce projet, contribuera à l’établissement et la confirmation de règles à propos des métriques d’un logiciel
- Ceci aurait pour but d’automatiser la tâche fastidieuse que nous avons dû effectuer manuellement

Conclusion

- Ainsi, comme en psychanalyse, cerner un patron de conception est une tâche abstraite, qui requiert une appréciation humaine, afin d’appeler l’esprit des Patrons

Pour ceux qui poursuivront …

Bonne chance!

et

Surtout n’essayez pas de nous contacter!