



NormandyJUG

POA

Programmation Orientée Aspects

enovea 
www.enovea.net



Présentation des intervenants

Guillaume Lours

<http://www.proxiad.com/>
<http://blog.proxiad.com/>



Sebastien Lecacheur

<http://blog.lecacheur.com>



Yann Petit

<http://blog.konkest.com>

NormandyJUG

Romain Verdier



<http://codingly.com>
<http://twitter.com/romainverdier>

Consultant FastConnect

<http://fastconnect.fr>





INTRODUCTION

- Qui a déjà **entendu parler** de l'AOP ?
- Qui a déjà **joué un peu** avec l'AOP ?
- Qui a une **idée vague** de ce que sont les principes et les mécanismes de l'AOP ?
- Qui veut prendre ma place ?



SOMMAIRE

1. La problématique
2. L'AOP kézako ?
3. Comment ça marche ?
4. En en Java ?
 1. Exemples
 2. Outillage
5. Bonus



1. Problématique

L'OOP permet aujourd'hui d'atteindre un **bon niveau d'abstraction**, et autorise une **modularité assez fine**.

Separation of Concerns (SoC)

Single Responsibility Principle (SRP)

L'OOP ne permet pas toujours d'
isoler certains « concerns »

Cross Cutting Concerns

« Considérations Entrecroisées »
ou
« Préoccupations Transversales »

Pooling Sécurité
Remoting Transactions
LoadBalancing Exceptions
Synchronisation Assertions
Globalization Caching
Profiling Persistence
Validation Logging



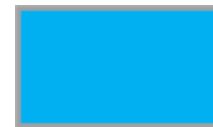
Méthode



Code métier



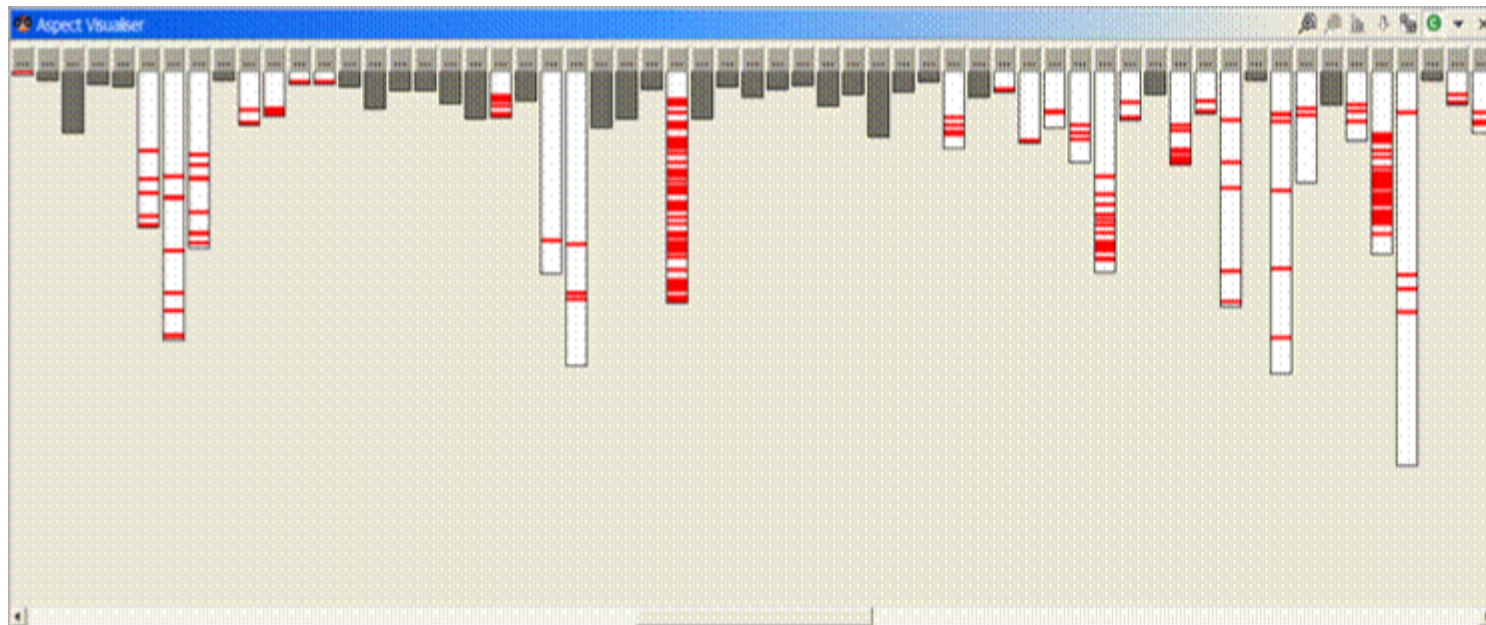
Transactions



Logging

Un exemple concret :

Appels de méthodes de log dans les classes Tomcat



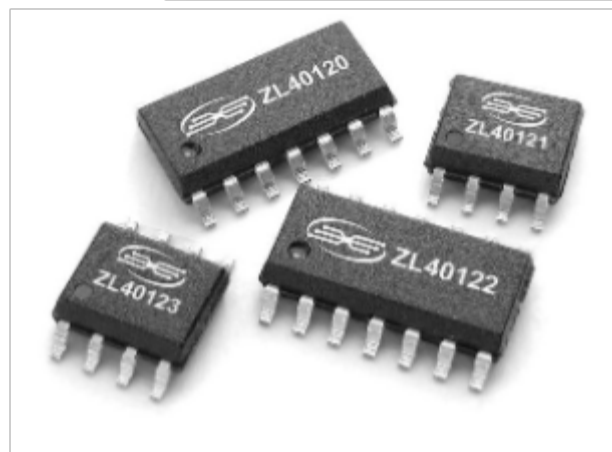


2. L'AOP Kézako ?





ACADEMY OF THE PACIFIC
A VILLAGE FOR LEARNING



aop^{uk}
association of
online publishers



AOPhub
The Association of Photographers

A.O.P.
i n f o r m a t i q u e

Advanced Observing Program
at Kitt Peak Visitor Center

Alpha Omega Publications®

On retiendra surtout :

Aspect Oriented Programming

On parle aussi d'AOSD : Aspect Oriented Software
Development.

On en français de POA : Programmation Orientée Aspect.

(Eliminer les **spaghettis** !)

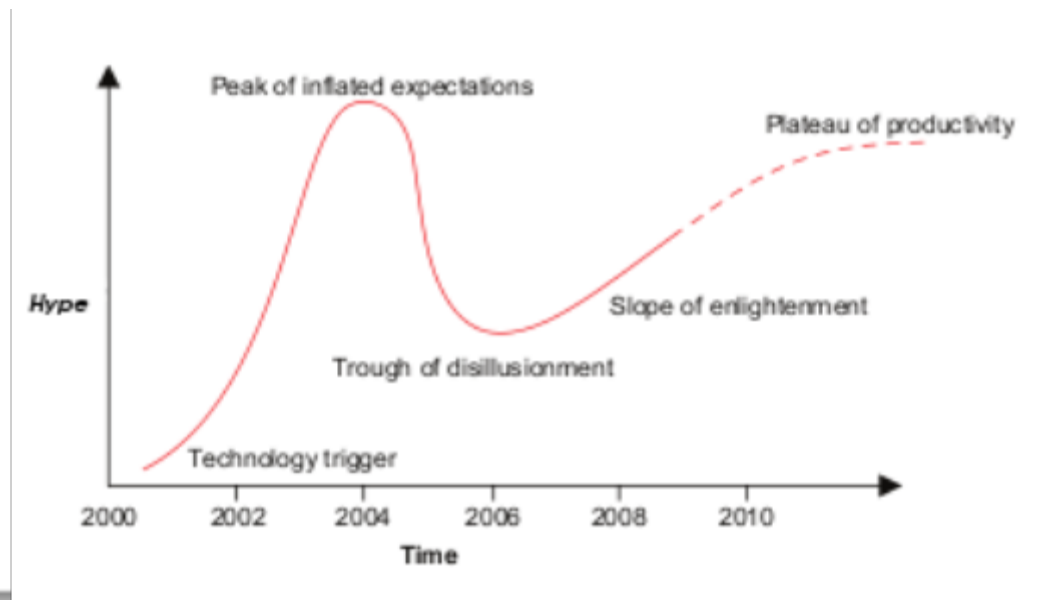
Permettre l'isolation des Cross Cutting Concerns, ces **aspects** transverses qu'il est impossible de factoriser convenablement grâce à l'OOP.

Oui, mais comment ?

En mettant la notion d'aspect transverse au premier plan, et en offrant un **complément à l'OOP.**

Mais l'AOP c'est nouveau ou quoi ?

- Des antécédants : le design pattern Visiteur.
- 1997 Gregor Kiczales publie a propos de l'AOP.
- 2001 Sortie de AspectJ par Xerox PARC (trigger)
- 2004 (peak) IBM working on AJDT, AspectJ In Action 1
- 2006 (desillusion) Pas d'adoption de masse
- ... (enlightment) Fusion AspectJ/AspectWerkz, SpringSource prend le relai d'IBM, AspectJ in Action 2



En AOP, comment définir un aspect ?

Quoi ?

Où ?

Comment ?

Aspect = Advice(s) + Pointcut(s)

Advice (Greffon)

Pièce de code constituant
tout ou une partie de la
logique de l'aspect.

What.

Join Point (Point de jonction)

Point valide de branchement dans le programme.

Appel ou exécution de méthodes,
Instanciation d'objets,
Execution de constructeurs,
Accès à des champs

...

Partout ou quelque-chose est appelé/exécuté.
Mais pas sur un if ou un for par exemple.

Category	Exposed join point	Code it represents
Method	Execution	Method body
Method	Call	Method invocation
Constructor	Execution	Execution of an object's creation logic
Constructor	Call	Invocation of an object's creation logic
Field access	Read access	Access to read an object's or a class's field
Field access	Write access	Access to write an object's or a class's field
Exception processing	Handler	Catch block to handle an exception
Initialization	Class initialization	Class loading
Initialization	Object initialization	Object initialization in a constructor
Initialization	Object pre-initialization	Object pre-initialization in a constructor
Advice	Execution	Advice execution



Pointcut (Point d'action)

Choix de **Join Points** pour
l'application des **Advices**.

Where. (When.)

En résumé, l'AOP consiste à :

- Identifier les « **cross cutting concerns** »
- Ecrire un ou plusieurs greffons – **les Advices** – qui représentent l'aspect et qui permettent d'encapsuler la « préoccupation transverse ».
- Choisir où appliquer ces greffons – **les Pointcuts...** à partir des **Join Points**.

... choisir où appliquer les greffons ?

Justement, **comment** les
appliquer ?

Méthode



OOP

VS

Code métier



Logging



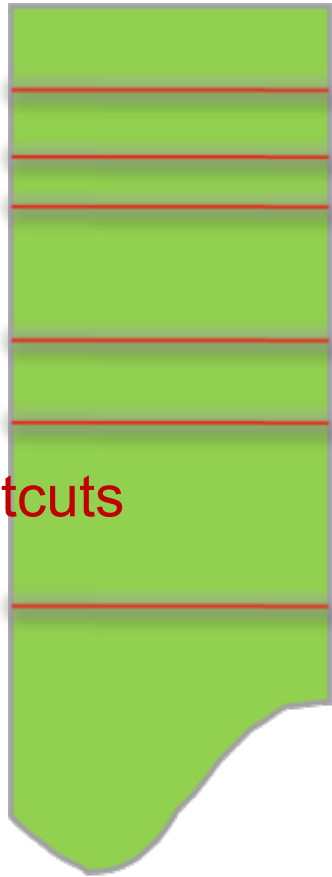
Transactions

OOP + **AOP**



Code métier

Pointcuts

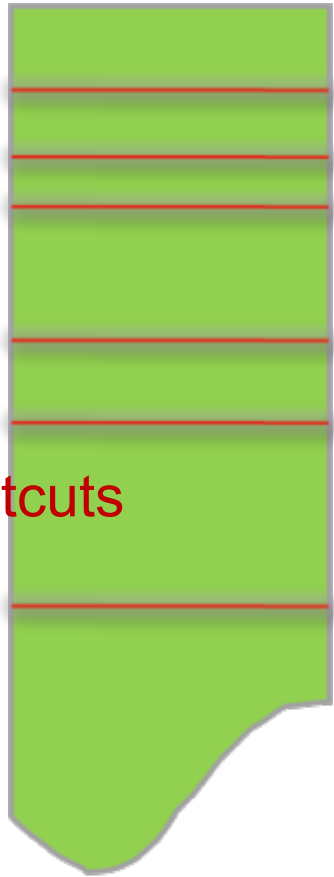


Aspects

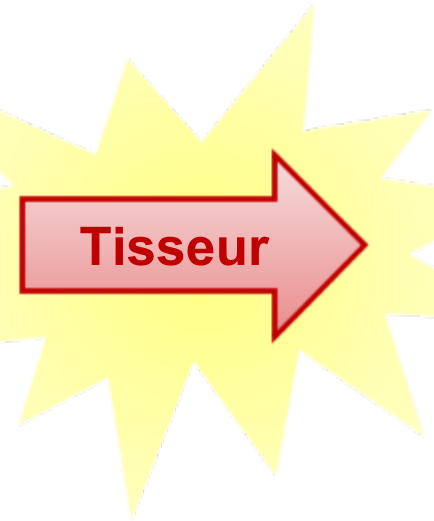


OOP + AOP

Code métier



Aspects



OOP + AOP



3. Comment ça marche ?

Tissage (Weaving) :

Insertion des greffons dans le programme à instrumenter, aux points d'action définis.

Il faut un moyen de décrire les aspects : **greffons** et **points d'action**.



Différentes méthodes selon les implémentations :

- AspectJ : historiquement par mots clefs (aspect, pointcut...) puis annotations.
- AspectWerkz : annotations.
- SpringAOP : fichiers de configuration XML
- Jboss AOP : configuration XML et annotations.

2 types de tissage

- **Compile Time Weaving**
tissage à la compilation
 - Tissage de sources
 - Tissage de bytecode
- **Load Time Weaving**
tissage au chargement

Tissage statique.

Le tissage à lieu **avant** la compilation, **durant** la compilation, ou juste **après** la compilation.

Mais avant l'exécution du programme.

Tissage dynamique.

Le tissage à lieu **durant l'exécution** du programme.



4. Et donc en JAVA ?



Quelles technologies ?

- **AspectJ**

- Extension du langage Java
- Utilisation de mots clés ou d'annotations pour le tissage

- **AspectWerkz**

- API AOP
- Utilisation d'un fichier de configuration ou des annotations pour le tissage

- **JBoss AOP**

- Idem AspectWerkz

- **Spring AOP**

- Spring AOP est une version allégée de la programmation par aspect qui ne permet de mettre des advices que sur l'exécution de méthode.
- Spring AOP peut être utilisé avec de la configuration xml, les annotations AspectJ ou bien un mix.



Quel Outillage

- AspectJ Development Tool plugin pour Eclipse.
- Spring IDE basé sur Eclipse
- IntelliJ IDEA plugins for AspectJ or AspectWerkz





Quelques Exemples :

- Sébastien :
 - Un logger avec AspectJ (Eclipse 3.5 + AJDT 1.6)
 - Un exemple de débogage
 - Illustration du tissage / la perspective AspectJ d'Eclipse
 - Un profiler avec JBossAOP sur des EJB3 (JBoss 5.1 + JBoss AOP 2.0)
 - Un exemple de développement spécifique sur un logiciel standard

- Guillaume :
 - Un logger sur chacun des niveaux d'interceptions de Spring AOP (Around, Before, After, ...)
 - 1 aspects interceptant l'exécution des méthodes annotées @Transactional





Bonus

Liens Web :

- AspectJ : <http://www.eclipse.org/aspectj/>
- Guide du programmeur AspectJ : <http://www.eclipse.org/aspectj/doc/released/progguide/>
- JBoss AOP : <http://jboss.org/jbossaop/>
- Guide d'utilisateur JBoss AOP : <http://jboss.org/jbossaop/docs/2.0.0.GA/docs/aspect-framework/userguide/en/html/index.html>
- Livre de référence sur Spring dont une partie sur Spring AOP : Spring par la pratique (<http://www.springparlapratique.org/>)
- Documentation Officielle Spring AOP : <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/ch07.html>

Autres liens :

- Une implémentation en Javascript (proof-of-concept) : <http://zer0.free.fr/aspectjs/>
- <http://blog.lecacheur.com>

Bibliographie :

- AspectJ In Action 2nd Edition by Ramnivas Laddad

