NetAspect

Framework AOP pour .Net

Sommaire

- AOP ?
- Présentation de NetAspect
- Exemple
- La suite

AOP

- Traiter séparément des préoccupations transverses
- Exemple :
  - Logging
  - Authentification
  - Transactions
  - Etc…

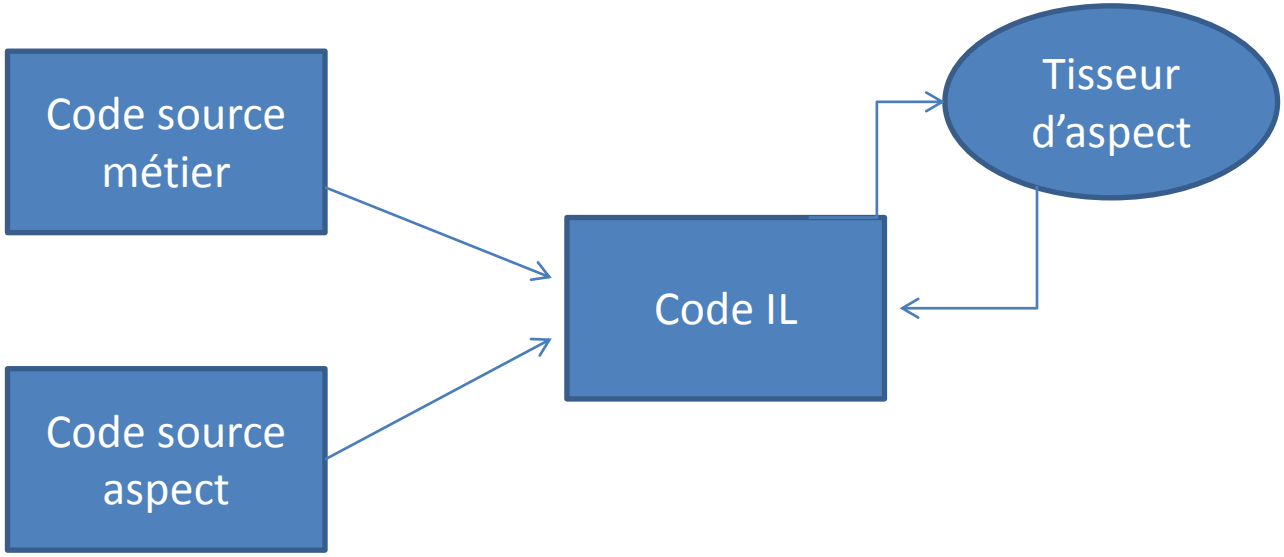
AOP

- Problématique :

```
public void CreatePerson(string name)
{
    if (name == null)
        throw new Exception ("name must not be null");
    Logger.Log("Create a person : " + name);
    if (!currentUser.IsAdmin())
        throw new Exception ("must be admin");
    var transaction = session.CreateTransaction();
    try
    {
        personDao.Create(new Person(name));
        transation.Commit();
    }
    catch
    {
        transaction.RollBack();
        throw;
    }
}
```

Mécanique

- Développement de l’aspect
- Développement du code métier
- Définitions des points de jonctions
- Après compilation (ou à l’éxécution) tissage de l’aspect avec le code métier


Code source métier
Code source aspect
Code IL
Tisseur d'aspect

AOP

- Un vocabulaire complexe :
  - Aspect
  - Greffon
  - Pointcut
  - Joinpoint
  - Tissage

NetAspect

Présentation

- Projet Open-Source
- C# .Net et Compact Framework
- 500 tests
- 32000 Lignes de code
- 10 mois
- 97% couverture de code

Fonctionnalités

- Tisse
  - Les méthodes
  - Les instructions
  - Les paramètres
- Intercepte :
  - Avant la méthode
  - Après la méthode
  - Les exceptions
  - Les OnFinally

Caractéristiques

- Pas de dépendances à l'exécution
- Certaines infos non dispo à l'exécution
- Définition des greffons et du tissage simple

NetAspect

- Problématique :

```
public void CreatePerson(string name)
{
    if (name == null)
        throw new Exception ("name must not be null");
    Logger.Log("Create a person : " + name);
    if (!currentUser.IsAdmin())
        throw new Exception ("must be admin");
    var transaction = session.CreateTransaction();
    try
    {
        personDao.Create(new Person(name));
        transation.Commit();
    }
    catch
    {
        transaction.RollBack();
        throw;
    }
}
```

NetAspect

- Avec NetAspet :

```
[Loggable]
[Transactional]
[MustBeAdmin]
public void CreatePerson([NotNull] string name)
{
    personDao.Create(new Person(name));
}
```

NetAspect

- L’aspect Loggable:

```
public class LoggableAttribute : Attribute
{
	public bool NetAspectAttribute = true;

	public void BeforeMethod(MethodBase method, object[] parameters)
	{
		Logger.Log("{0} : {1}", method.Name, string.Join(",", parameters.Select(p.ToString())));
	}
}
```

Possibilités

- Mettre un aspect sur :
  - Les paramètres des méthodes
  - Les appels à des méthodes/fields/constructeurs
  - Les méthodes/constructeurs
- Récupérer des informations comme
  - La méthode
  - Les paramètres
  - Infos du fichier
  - Etc…

NetAspect

- L’état actuel
  - Fonctionnellement complet à 95%
  - Peu de doc
- La suite :
  - Refactoring des tests
  - Terminer le fonctionnel
  - Add-in Visual Studio
  - Documentation

Questions ?