



# Introduction aux Aspects d'Assemblages (AA)

Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey

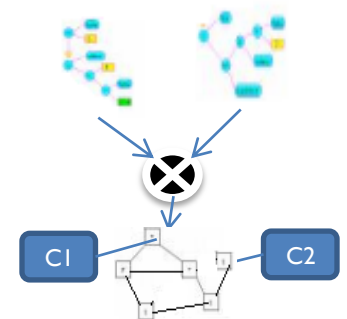
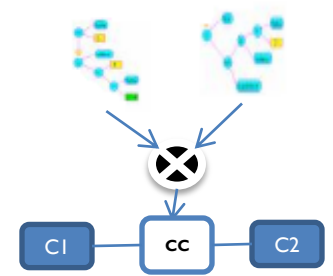
Equipe Rainbow, Laboratoire I3S, UMR CNRS 6070, Université de  
Nice Sophia Antipolis,

Email : <name>@unice.fr

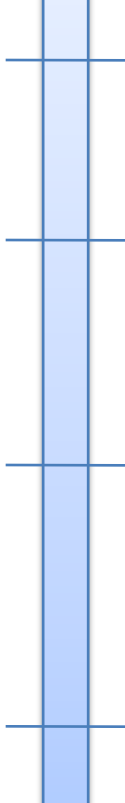
Web : [http://rainbow.i3s.unice.fr/~tigli/doku/doku.php?id=seminaire\\_wcomp\\_2009](http://rainbow.i3s.unice.fr/~tigli/doku/doku.php?id=seminaire_wcomp_2009)

# Aspects d'Assemblage (AA) : une évolution de la composition transverse

- Post ... Script
- Schémas d'Interaction (ex. ISL [Berger01])
- Aspects d'Assemblage [Cheung09]  
= Point de coupe + Schémas d'Assemblage (ex. ISL4WComp)



# Aspects d'Assemblage (AA) : Une évolution des aspects pour l'adaptation

- 
- AOP [Kiczales 97]
  - AspectJ [Kiczales 01]
  - AO4BPEL [Charfi 04]
  - EAOP [Douence 06]
  - AA [Tigli 06, Cheung 07,]
- Minimiser la dispersion de code
  - Préoccupations transverses, non fonctionnelles
  - Modularité Transverse
  - Déclenchement sur événements
  - Auto adaptation transverse (tissage « opportuniste » d'aspects sur SLCA)

# Rappel : Principe AOP

```
public class HelloWorld {  
    public static void main  
    (String[] args) {  
        new HelloWorld().sayHello();  
    }  
    public void sayHello ()  
    {system.out.println("Hello World!");}  
}
```

```
pointcut salutation():  
    execution(*HelloWorld.sayHello(..)  
);
```

```
after():salutation(){  
    System.out.println(« Au revoir ...»);  
}
```

Weaver

Pointcut Matching  
Greffons Application

1

2

```
public class HelloWorld {  
    public static void main  
    (String[] args) {  
        System.out.println(« Hello Two...»);  
        new HelloWorld().sayHello();  
        System.out.println(« Hello One...»);  
    }  
    public void sayHello ()  
    {system.out.println("Hello  
World!");}  
}
```

# Rappel : Principe AOP

```
public class HelloWorld {  
    public static void main  
    (String[] args) {  
        new HelloWorld().sayHello();  
    }  
    public void sayHello ()  
    {system.out.println("Hello World!");}  
}
```

Weaver

Pointcut Matching  
Greffons Application

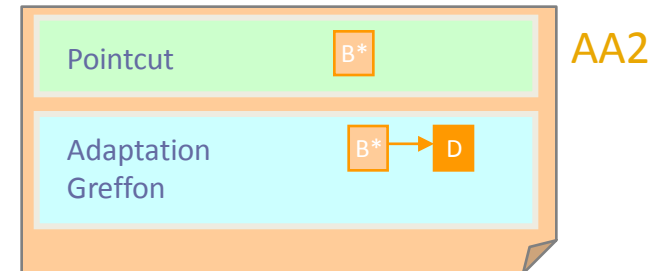
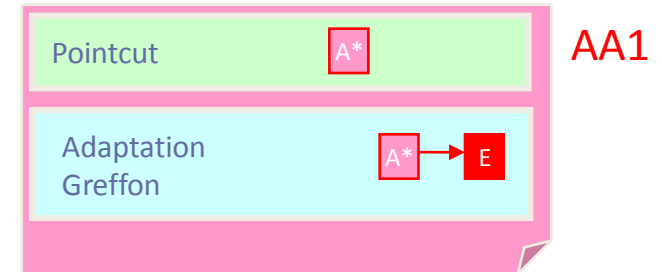
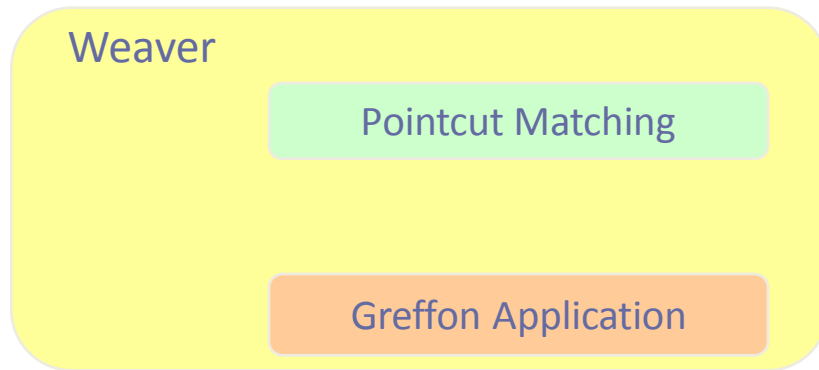
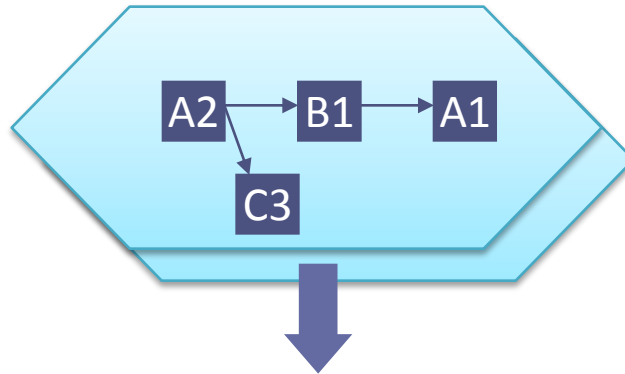
1  
2

```
public class HelloWorld {  
    public static void main  
    (String[] args) {  
        System.out.println(« Hello Two...»);  
        new HelloWorld().sayHello();  
        System.out.println(« Hello One...»);  
    }  
    public void sayHello ()  
    {system.out.println("Hello  
World!");}  
}
```

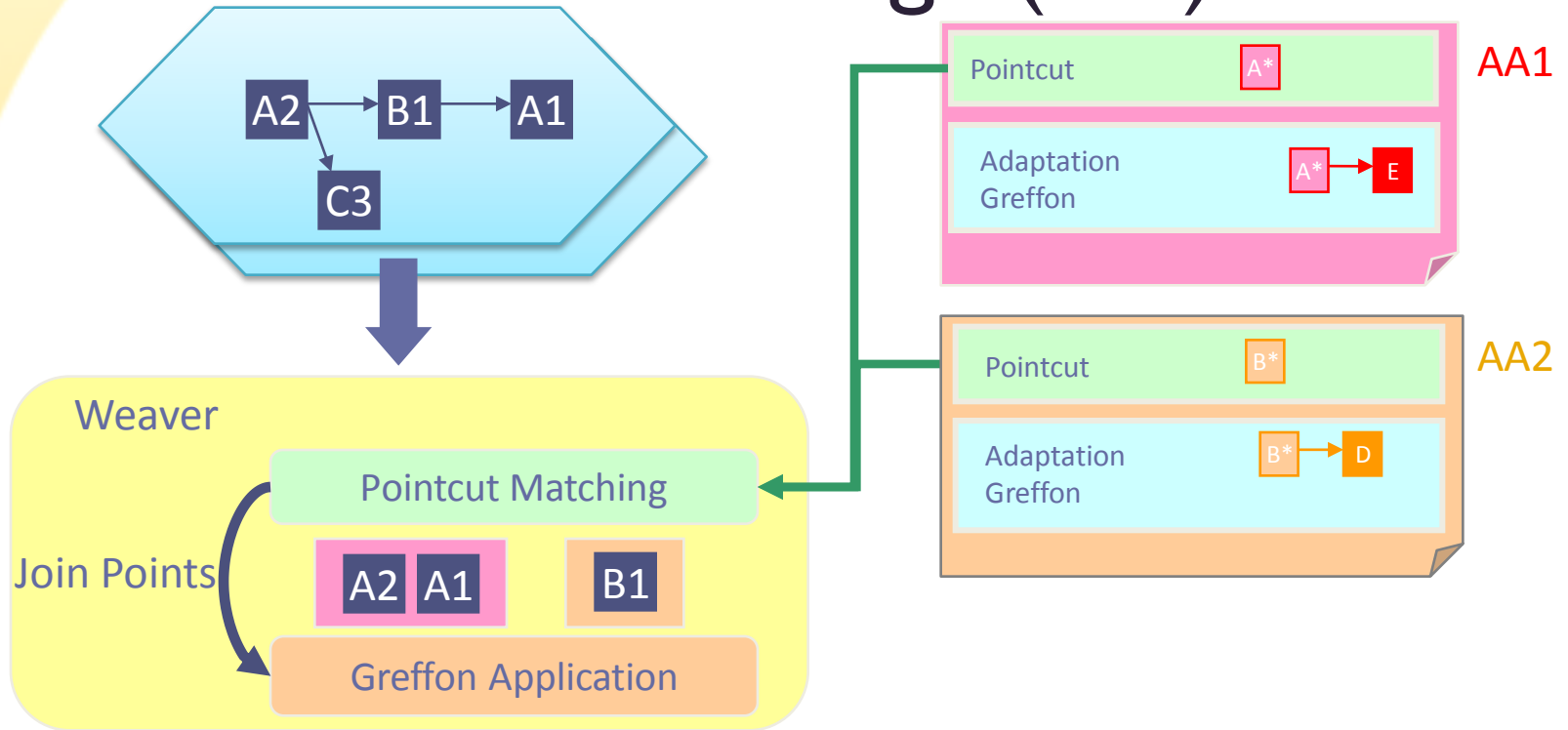
```
pointcut Two():  
    execution(*HelloWorld.sayHello(..));
```

```
before():Two(){  
    System.out.println(« Hello Two ...»);  
}
```

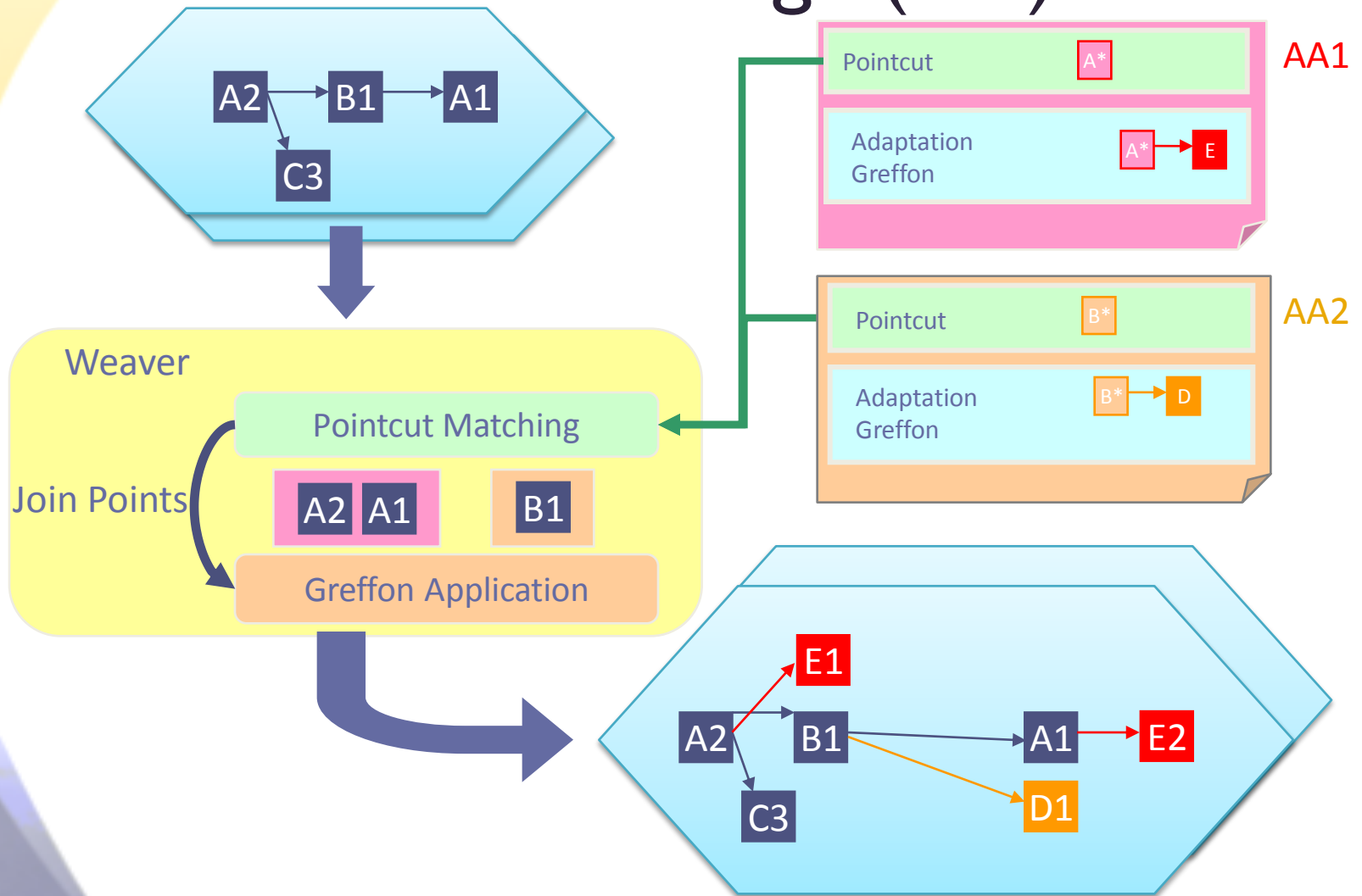
# Le principe des Aspects d'Assemblage (AA)



# Le principe des Aspects d'Assemblage (AA)



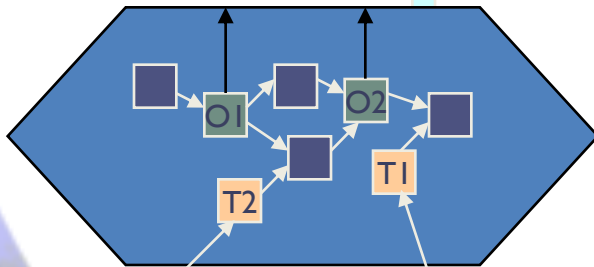
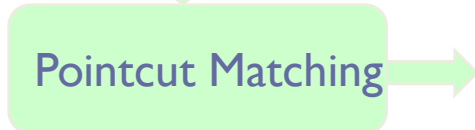
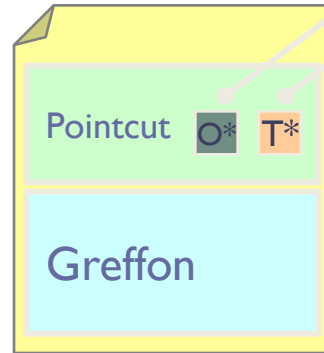
# Le principe des Aspects d'Assemblage (AA)





# AA: Point de Coupe

```
observed := /t*/ ;  
timeout :=  
  /ct*/ { a[substr($1,3)]=$1 }  
END    { for(i=1;i<=NR;i++){print a[i]} } ;
```



Enfin, les Greffons se basent sur des méthodes (entrée) ou des événements (en sortie)

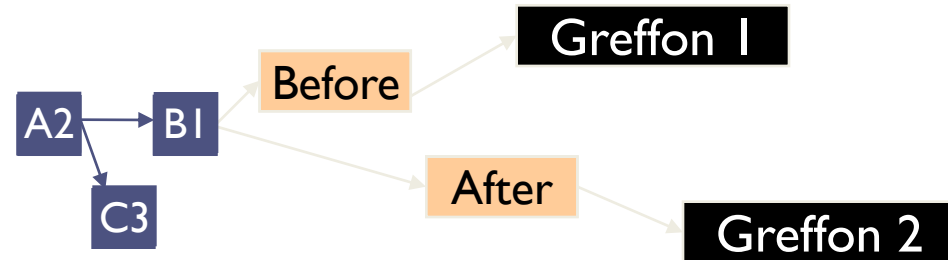
```
observed.^Out1 ...  
Timeout.check ...
```

# AA: Deux types de Greffons

- **Blackbox Greffons:**

- Tissage externe entre les Greffons

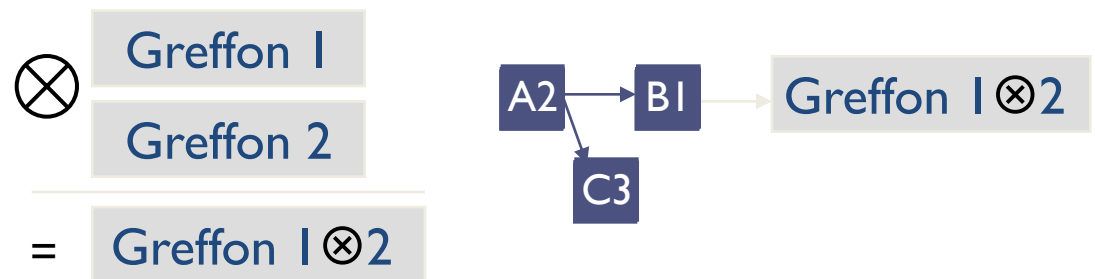
```
SCHEMA Ex (observed):
observed.^Out1 -> Bcomp.do; CALL /* before
observed.^Out2 -> CALL ; Bcomp.do; /* after
```



- **Greybox Greffons:**

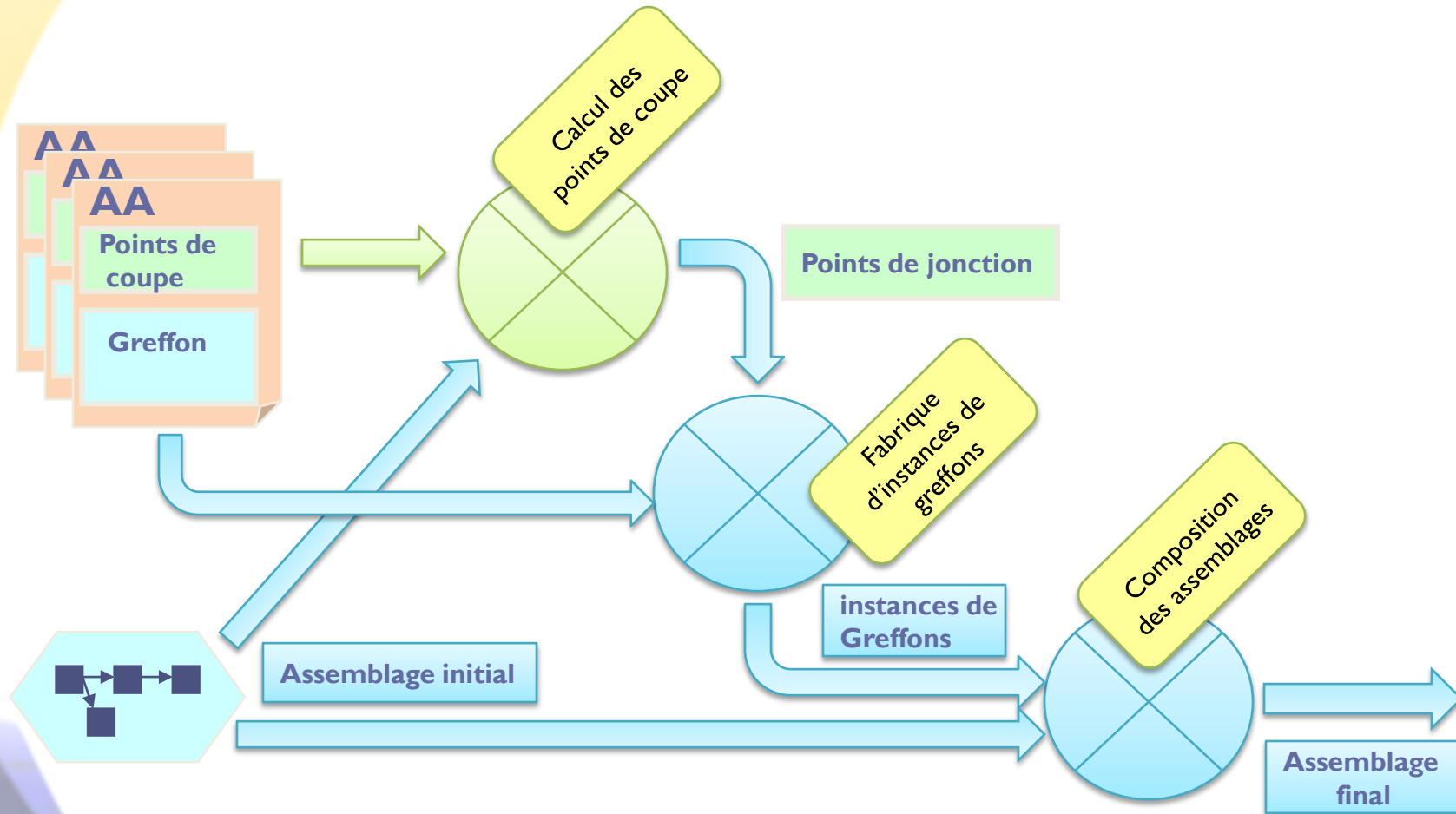
- Nous connaissons en partie la sémantique des greffons
- Possibilité de Fusion en cas de conflit

```
SCHEMA Ex (observed, timeout):
observed.^Out ->
  ( IF ( timeout.Check ) CALL )
timeout.Check ->
  ( timeout.Start ; CALL )
```

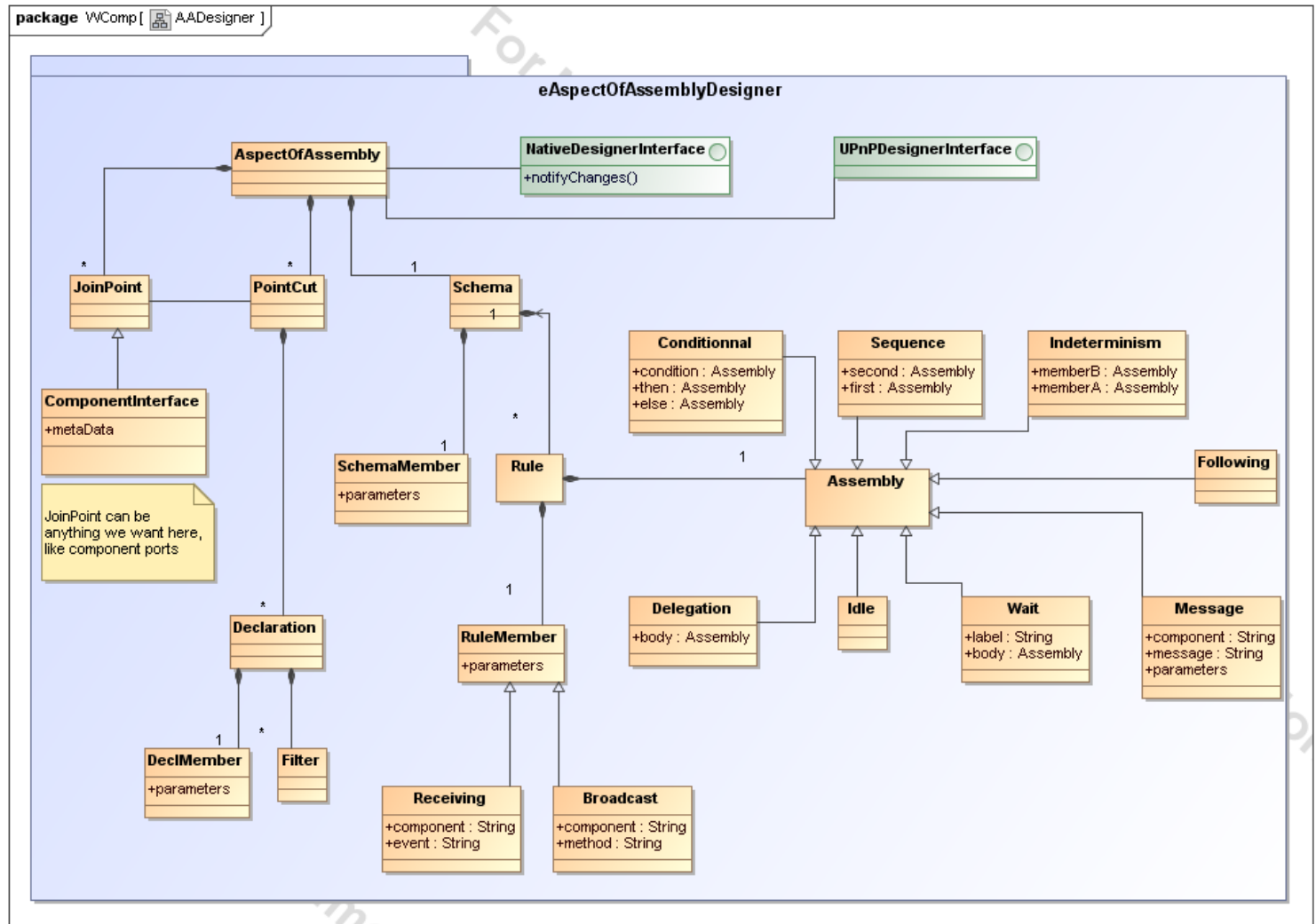




# Architecture fonctionnelle du Tisseur d'AA

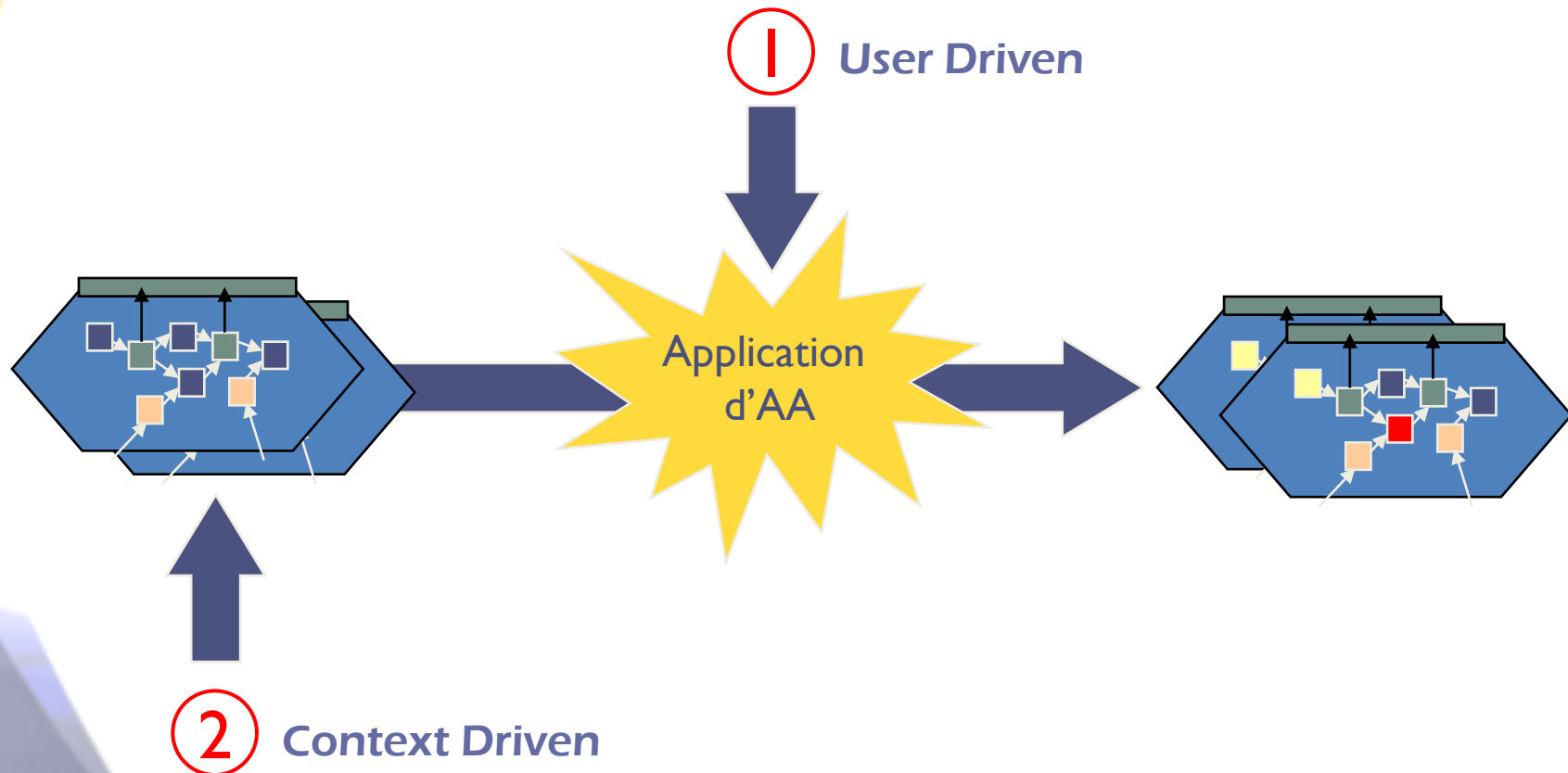


# Modèle de l'AA Designer



# Mise en œuvre : Deux modes d'application des AAs

- « a posteriori » et « a priori »





# Outils

- UPnP Designer
  - Crée les composants proxy des dispositifs découverts
  - Détruit les composants proxy des dispositifs disparus
- AA Designer
  - Sélectionne les AAs
  - Tisse les AAs et modifie l'assemblage du Services Composites cibles
- Et toujours Sharpdevelop ...



# Manipulation des Aspects d'Assemblages sous WComp

<http://rainbow.i3s.unice.fr/~tigli/cours/WComp/PDF/Demo AA WComp.pdf>





# Questions ?





# Pour aller plus loin

- Sites Web

- UPnP

- Universal Plug and Play forum
    - <http://www.upnp.org/>
    - Intel's UPnP SDK for Linux
    - <http://developer.intel.com/ial/upnp/>

- WComp

- Site de Distribution Wcomp :  
<http://rainbow.polytech.unice.fr/wcomp/>
    - Site de Distribution Ubiquarium :  
<http://rainbow.polytech.unice.fr/wcomp/>



# Dépôts Logiciels de WComp

- [2007] Jean Merlin, Vincent Hourdin, Stéphane Lavirotte, Jean-Yves Tigli et Michel Riveill. « BridgeBox USB/UPnP ». Dans Dépôt APP, 2007. En cours.
- [2007] Daniel Cheung-Foo-Wo, Jean-Yves Tigli, Stéphane Lavirotte, Michel Riveill et Eric Pascual. « AAComp: Aspect of Components Assembly Designer ». Dans Dépôt APP, 2007. En cours.
- [2007] Nicolas Oudit, Stéphane Lavirotte, Nicolas Dalmasso, Jean-Yves Tigli et Michel Riveill. « Ubiquarium: Scènes 3D pour l'Environnement Virtuel ». Dans Dépôt APP, juin 2007.  
IDDN.FR.001.230042.000.S.C.2007.000.10600.
- [2007] Samuel Weibel, Jean-Yves Tigli, Stéphane Lavirotte, Daniel Cheung, Vincent Hourdin et Michel Riveill. « SharpWComp ». Dans Dépôt APP, février 2007.  
IDDN.FR.001.090016.000.S.C.2007.000.10600.
- [2006] Vincent Hourdin, Stéphane Lavirotte, Jean-Yves Tigli et Michel Riveill. « CyrLink UPnP ». Dans Dépôt APP, septembre 2006.  
IDDN.FR.001.360013.000.S.C.2006.000.10600.



# Sélection de Publications sur WComp

- [2009] J.-Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, D. Cheung, E. Callegari, M. Riveill “WComp middleware for ubiquitous computing: Aspects and composite event-based Web services” dans la revue Annals of Telecommunications, éditeur Springer Paris, ISSN 0003-4347 (Print) 1958-9395 (Online), Vol. 64, No 3-4, March-April 2009.
- [2008] Vincent Hourdin, Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey, Michel Riveill, “SLCA, Composite Services for Ubiquitous Computing”, in International Conference on Mobile Technology, Applications and Systems, Sep 2008.
- [2007] Daniel Cheung-Foo-Wo, Jean-Yves Tigli, Stéphane Lavirotte et Michel Riveill. « Self-adaptation of event-driven component-oriented Middleware using Aspects of Assembly ». Dans 5th International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), California, USA, novembre 2007.