



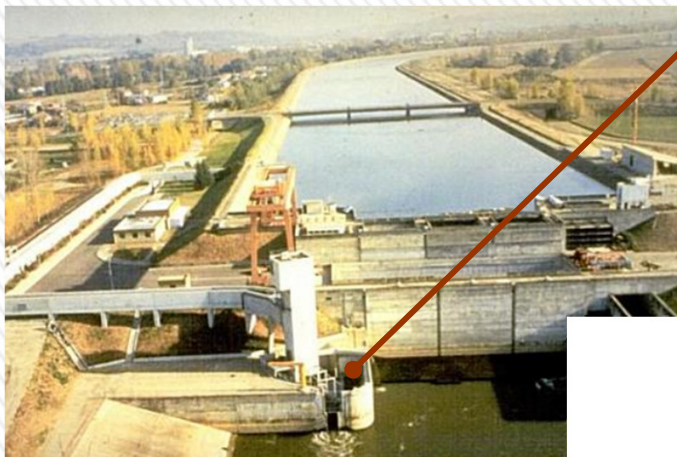
NFE 113

Conception et  
administration de  
bases de données

**André Miralles**

# Parcours professionnel

5 ans



Département  
Eaux

Passes à poissons

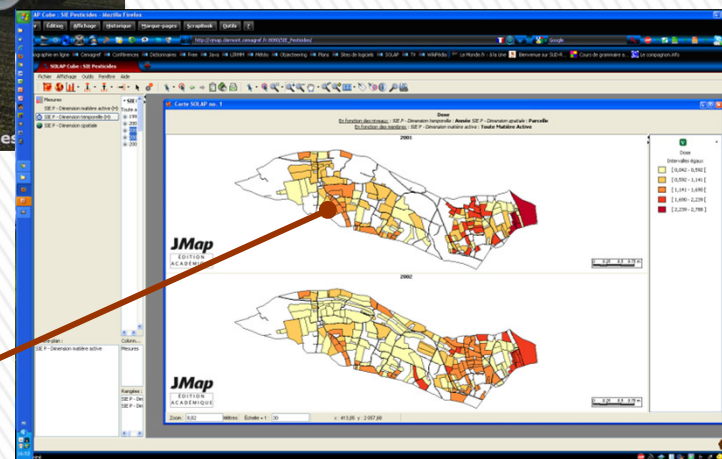
17 ans



Département  
Ecotechnologies

Techniques d'application  
des pesticides

18 ans



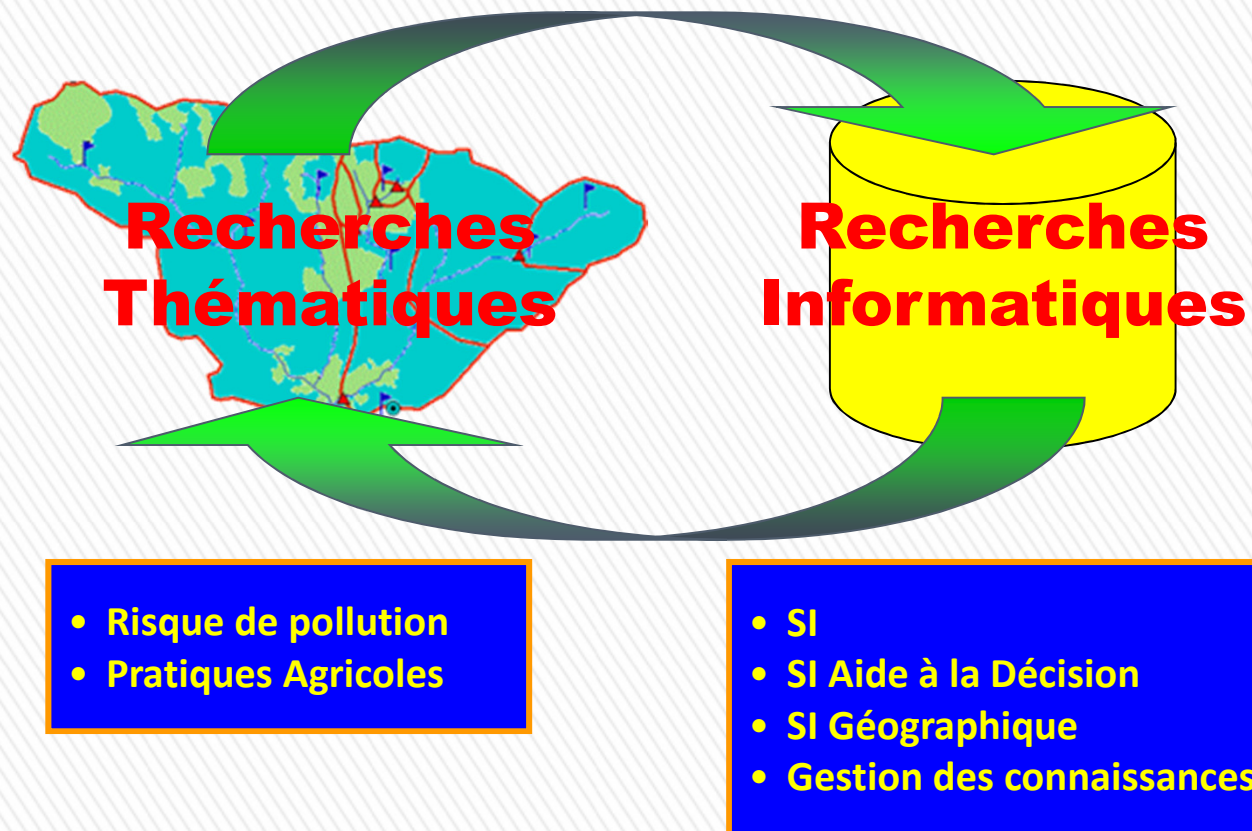
Informatique - Géomatique

Département  
Territoires



# Principe de recherche

- » Recherches Informatiques en **synergie forte** avec des Recherches Thématiques



# Élément de contexte

## » Enquête du Cabinet Vanson Bourne auprès des entreprises

### > En France

- + **22 %** sont des données **stratégiques**
- + **21 %** sont des données **redondantes, obsolètes** ou **triviales** (ROT Data)
- + **57 %** sont des données **obscur**es (Dark Data)
  - Données **capitales** ou **inutiles** pour l'entreprise

**Stockage des données inutiles => 600 000 €/an/entreprise**

**Europe + Moyen-Orient + Afrique => 800 Mds € sur 5 ans**



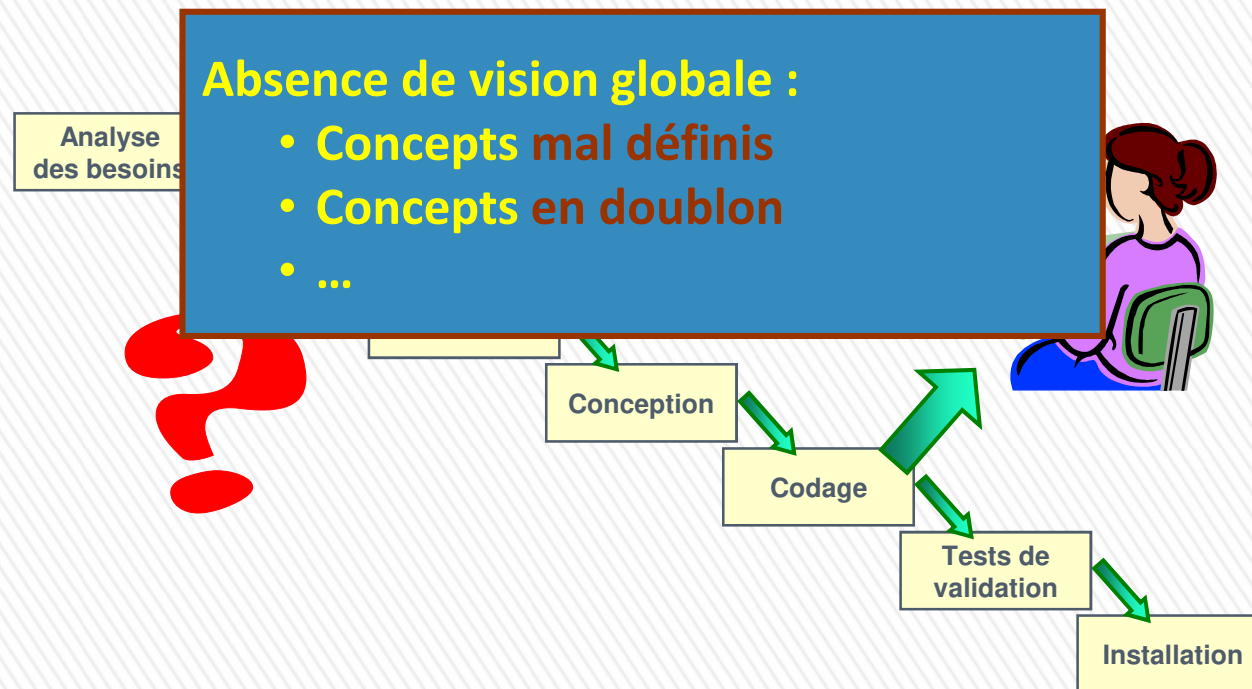
# Travaux de recherche

## » Constat

### > Logiciels souvent développés (surtout en recherche)

+ **Rapidement** pour répondre à un traitement spécifique des données

- Sans **modélisation**
- Sans **méthode de développement** explicite
- ...

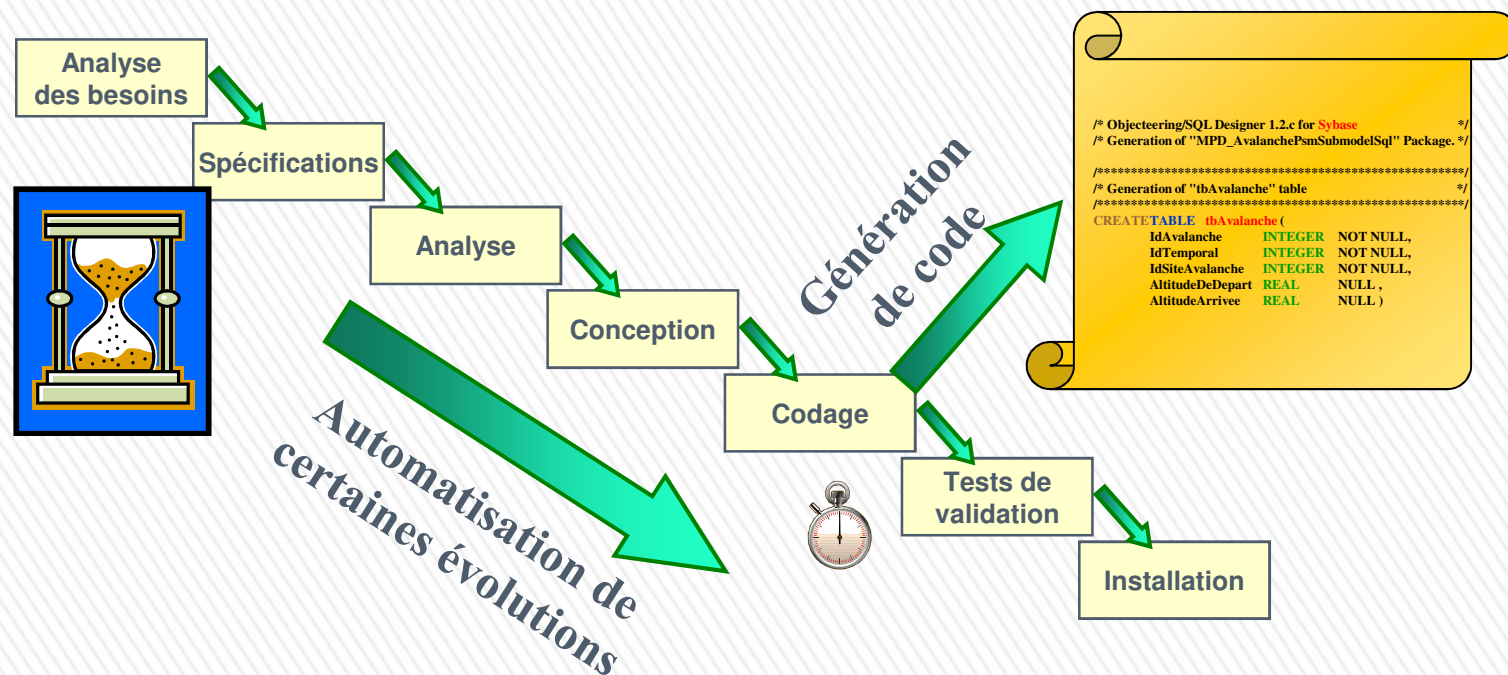


# Travaux de recherche

## » Solution

### > Consacrer plus de temps aux phases amonts

- + Pour **préciser les besoins** des acteurs
- + Pour **appréhender le système** à réaliser
  - Recueillir une **description la plus précise possible**





# Travaux de recherche

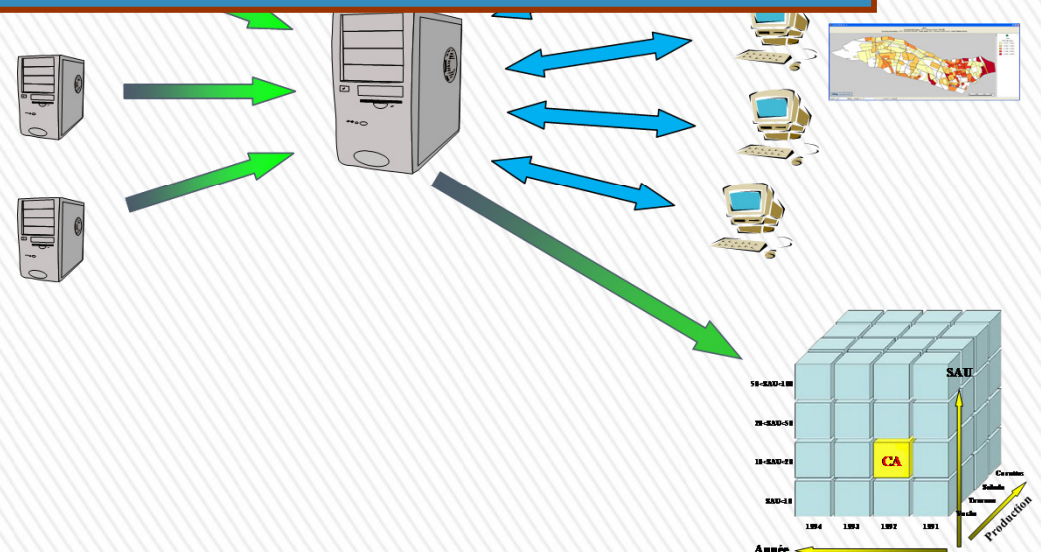
## » Méthode

- > Développer des **méthodes** et des **outils**
  - + Automatiser le **processus de développement** depuis l'analyse jusqu'au code
    - Code SQL en particulier
  - + Créer des Systèmes d'information et des entrepôts de données
    - Composantes **spatiales**
    - Composantes **temporelles**

## Full MDA (Model Driven Architecture)

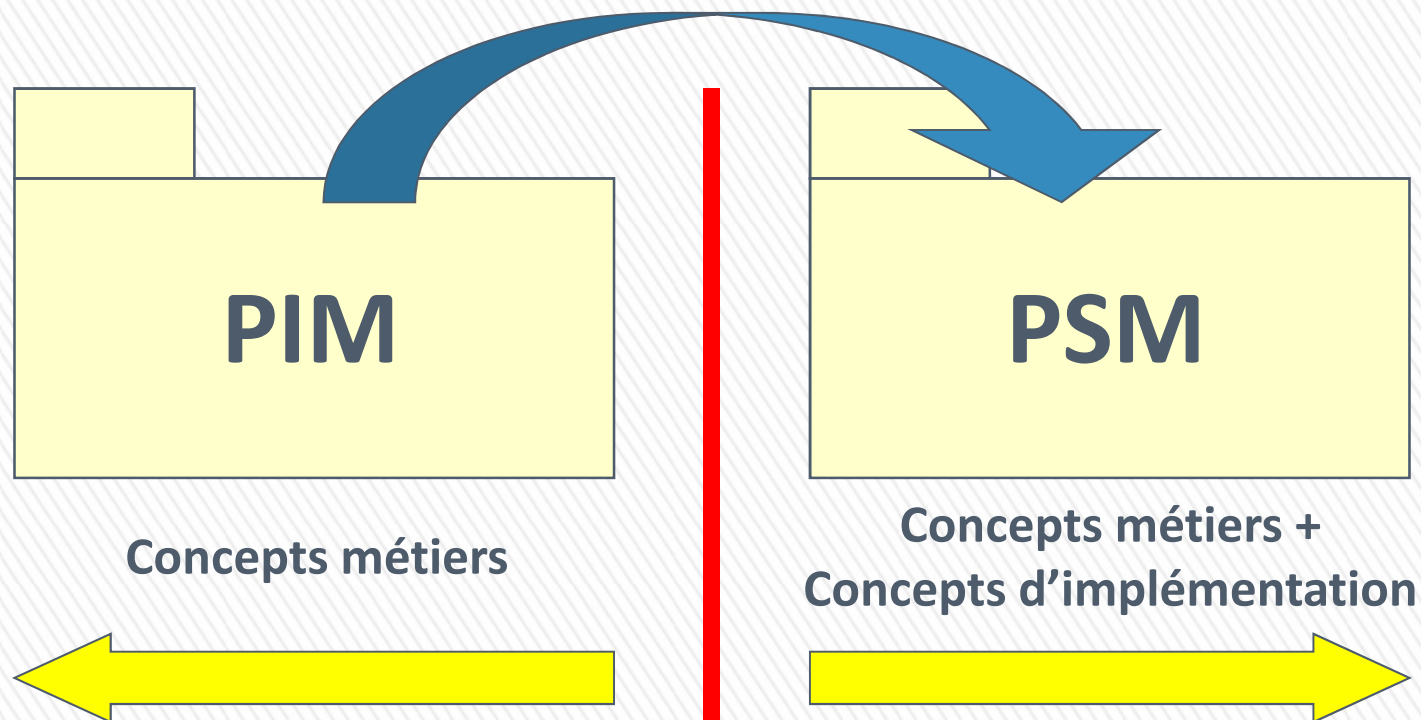
## » Domaine d'Application

- > Environnement
  - + Suivi des **pesticides**



# Principe de l'Ingénierie dirigée par les modèles

## Transformations de modèles



**PIM** : Platform Independent Model  
**PSM** : Platform Specific Model

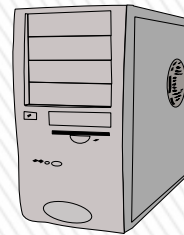


# Processus Full MDA

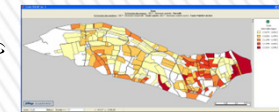
**Systèmes  
d'information  
transactionnels**



**Entrepôt  
de  
données**



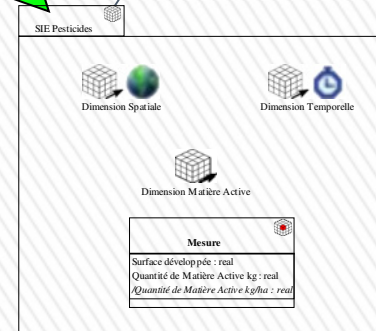
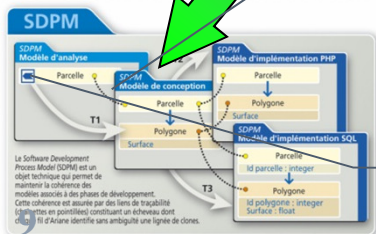
**Outils  
d'aide à la  
décision**



**Profil SDPM**

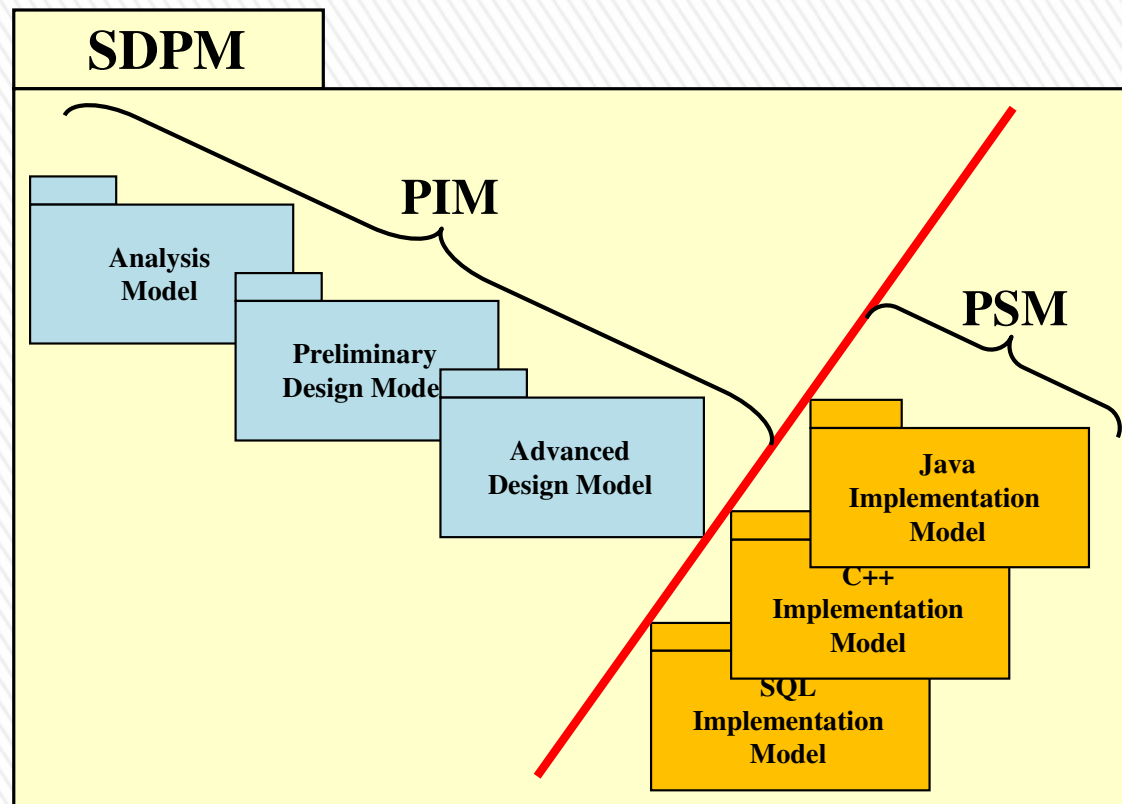
**Profil SOLAP**

**Profil UML- SIG**



# Profil SDPM

- » Le **SDPM** est un **artefact de modélisation** qui permet de **créer** et de **gérer en cohérence** plusieurs modèles reproduisant les phases de développement



- » **Software Development Process Model**



# Profil SDPM

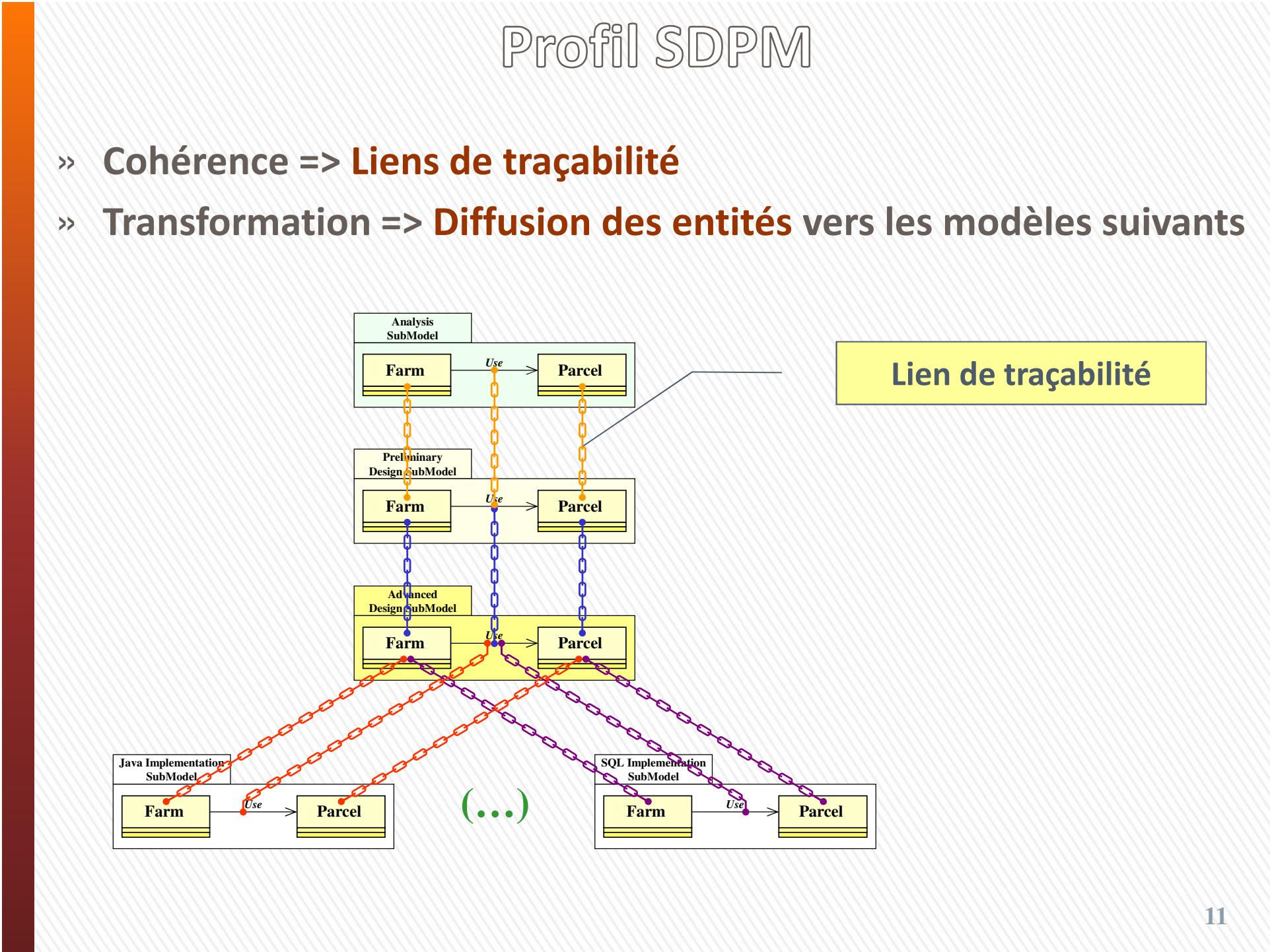
- » Cohérence => Liens de traçabilité
- » Transformation => Diffusion des entités vers les modèles suivants

The diagram illustrates the SDPM profile structure, showing a hierarchy of submodels and their relationships. The submodels are arranged in a vertical stack, with the top submodel being the 'Analysis SubModel' and the bottom submodels being 'Java Implementation SubModel' and 'SQL Implementation SubModel'. Each submodel contains two entities: 'Farm' and 'Parcel'. The 'Farm' entity is connected to the 'Parcel' entity via a 'Use' relationship. The 'Analysis SubModel' is connected to the 'Preliminary Design SubModel' via a 'Use' relationship. The 'Preliminary Design SubModel' is connected to the 'Advanced Design SubModel' via a 'Use' relationship. The 'Advanced Design SubModel' is connected to the 'Java Implementation SubModel' and the 'SQL Implementation SubModel' via 'Use' relationships. The 'Java Implementation SubModel' and the 'SQL Implementation SubModel' are connected to the 'Advanced Design SubModel' via 'Use' relationships. The 'Farm' entity in the 'Java Implementation SubModel' is connected to the 'Farm' entity in the 'Advanced Design SubModel' via a 'Use' relationship. The 'Parcel' entity in the 'Java Implementation SubModel' is connected to the 'Parcel' entity in the 'Advanced Design SubModel' via a 'Use' relationship. The 'Farm' entity in the 'SQL Implementation SubModel' is connected to the 'Farm' entity in the 'Advanced Design SubModel' via a 'Use' relationship. The 'Parcel' entity in the 'SQL Implementation SubModel' is connected to the 'Parcel' entity in the 'Advanced Design SubModel' via a 'Use' relationship. A yellow box labeled 'Lien de traçabilité' (Traceability link) points to the 'Use' relationship between the 'Farm' and 'Parcel' entities in the 'Analysis SubModel'.

Lien de traçabilité

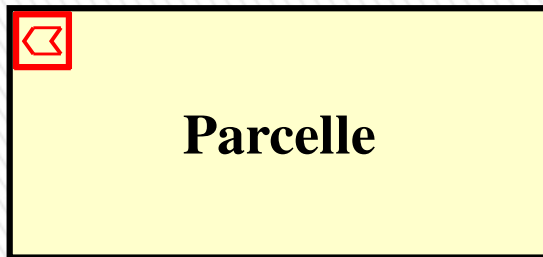
(...)

- # Profil SDPM
- » Cohérence => Liens de traçabilité
  - » Transformation => Diffusion des entités vers les modèles suivants
- 
- The diagram illustrates the SDPM profile structure, showing a hierarchy of submodels and their relationships. The submodels are arranged in a vertical stack, with the top submodel being the 'Analysis SubModel' and the bottom submodels being 'Java Implementation SubModel' and 'SQL Implementation SubModel'. Each submodel contains two entities: 'Farm' and 'Parcel'. The 'Farm' entity is connected to the 'Parcel' entity via a 'Use' relationship. The 'Analysis SubModel' is connected to the 'Preliminary Design SubModel' via a 'Use' relationship. The 'Preliminary Design SubModel' is connected to the 'Advanced Design SubModel' via a 'Use' relationship. The 'Advanced Design SubModel' is connected to the 'Java Implementation SubModel' and the 'SQL Implementation SubModel' via 'Use' relationships. The 'Java Implementation SubModel' is connected to the 'SQL Implementation SubModel' via a 'Use' relationship. The diagram also shows a 'Lien de traçabilité' (Traceability Link) connecting the 'Farm' entity in the 'Analysis SubModel' to the 'Farm' entity in the 'Java Implementation SubModel'. The 'Lien de traçabilité' is represented by a yellow box with the text 'Lien de traçabilité' and a line pointing to the traceability link between the 'Farm' entities.



# Profil UML-SIG

## » Annotation pictogrammique des entités métiers



### » Spatial

> Point, Ligne, Polygone

### » Temporel

> Instant, Période

### » Spatialité & Temporalité alternative

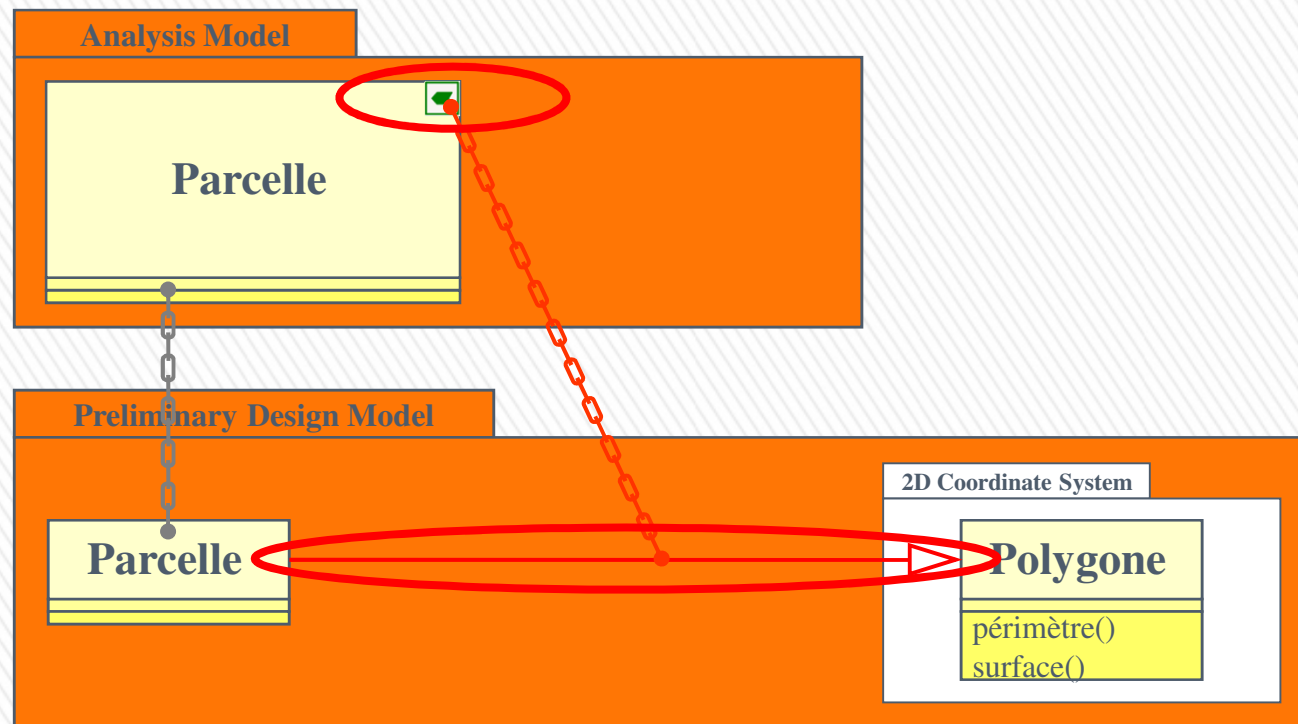
### » Spatialité & Temporalité multiple

### » Spatio-temporel

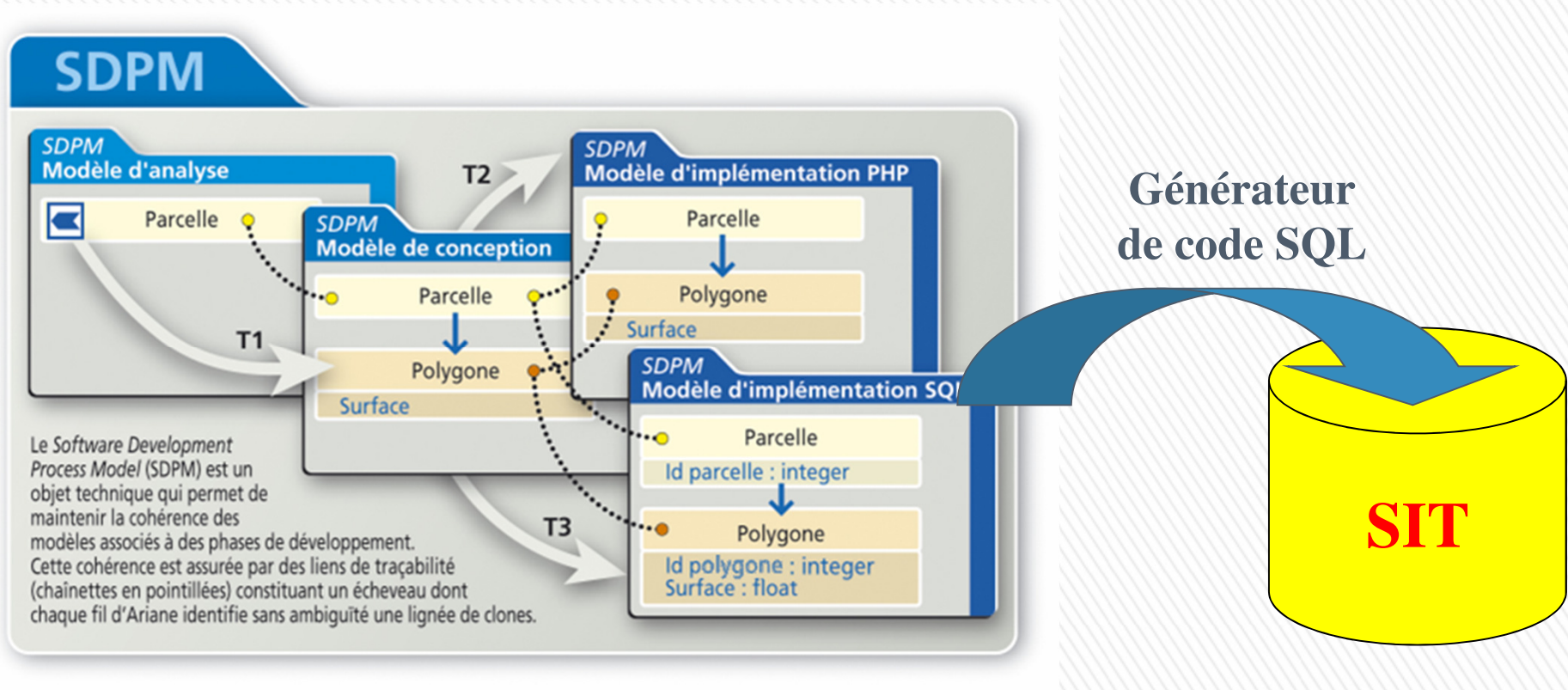


# Profil SDPM + Profil UML-SIG

- » Une transformation **crée** la classe **Polygone**
- » Une transformation ajoute une **relation** (association/généralisation) entre la classe **Parcelle** et la classe **Polygone**



# Atelier de génie logiciel





# Objectif et Programme

## » Objectifs du cours :

- > Comprendre le fonctionnement des systèmes de gestion de bases de données relationnelles, de la conception à l'utilisation et l'administration

## » Programme

- > Modélisation des bases de données relationnelles
- > Requêtes SQL et Programmation PL/pgSQL (dont triggers)
- > Administration et optimisation des bases de données

## » Comment on va travailler ?

- > Utilisation de PostgreSQL voire PostGIS
- > Cours
- > TPs pour bien appréhender les concepts
- > Évoluer une base de données existantes
  - + Évolution améliorer le schéma
  - + Évolution pour traiter l'héritage de tables (Mise en œuvre de triggers)

**Modélisation + Implémentation de la BD + Transferts des données**

# Questions + Tour de table