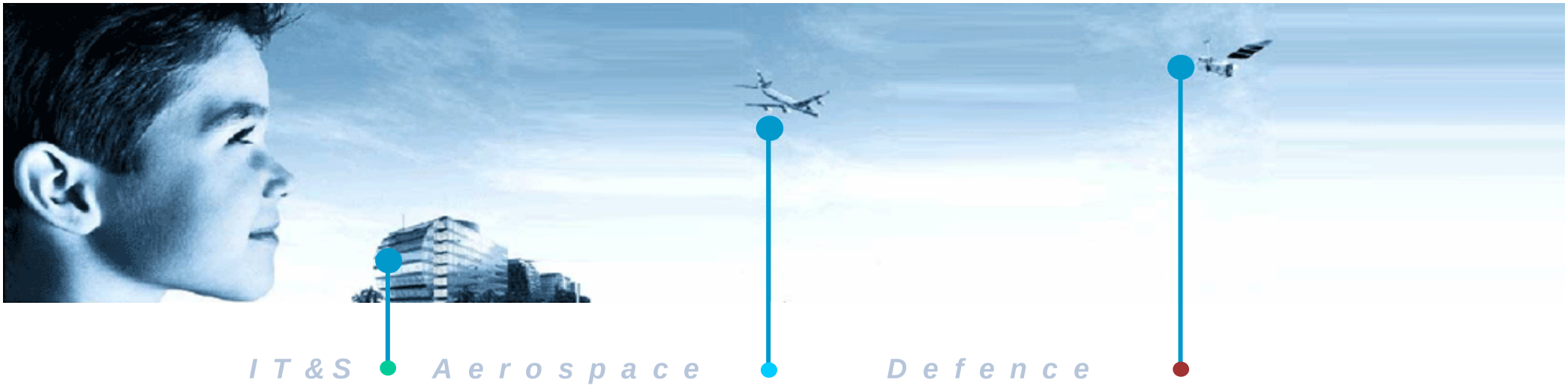
Model Synchronization: a need for MDA



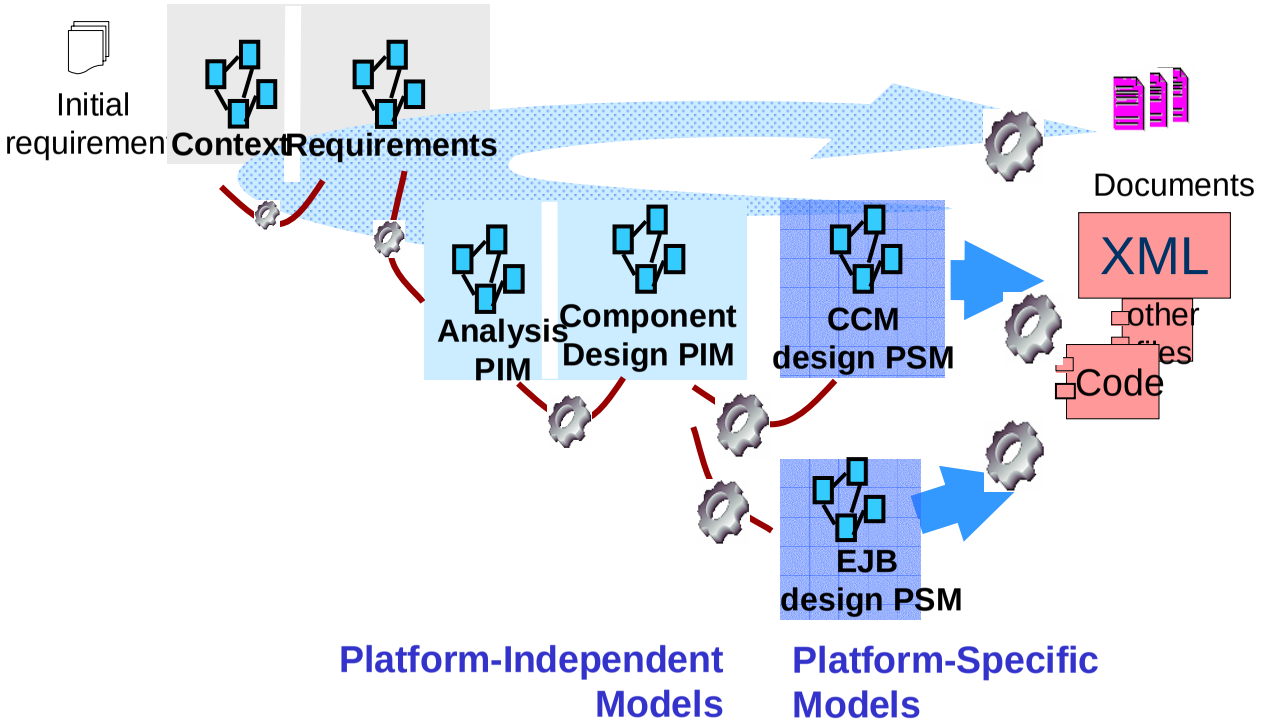
Madeleine Faugère

THALES

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspot
- Synchronization
  - Principles
  - Model
- Demonstration

What is Model Transformation ?



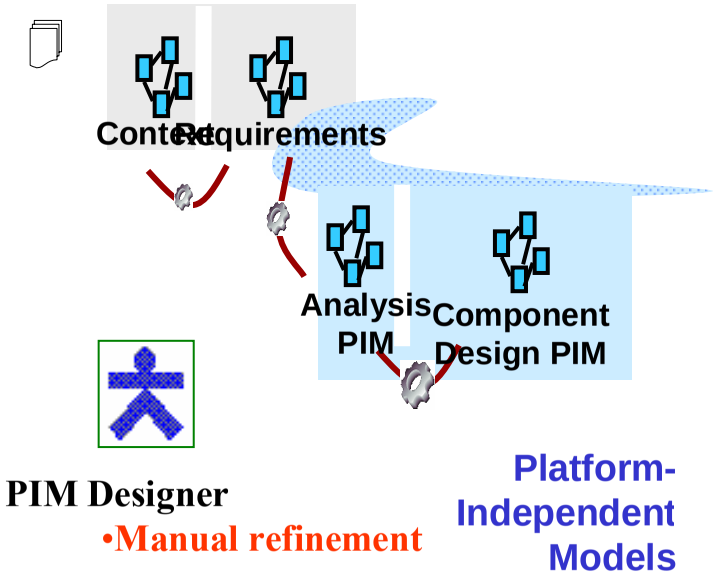
- Separation of concern
  - is the key concept of the MDE approach
  - uses a specific semantical language for each abstraction level
  - Implies a technology to jump from one semantical language to the other with information reuse: this is model transformation technology

Model Transformation is the keystone of a MDE environment

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspot
- Synchronization
  - Principles
  - Model
- Demonstration

What is Model Synchronization (1/4) ?



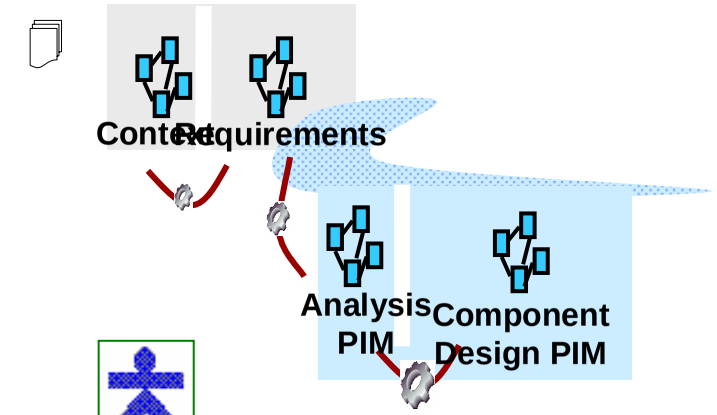
I

II

III

IV

What is Model Synchronization (2/4) ?

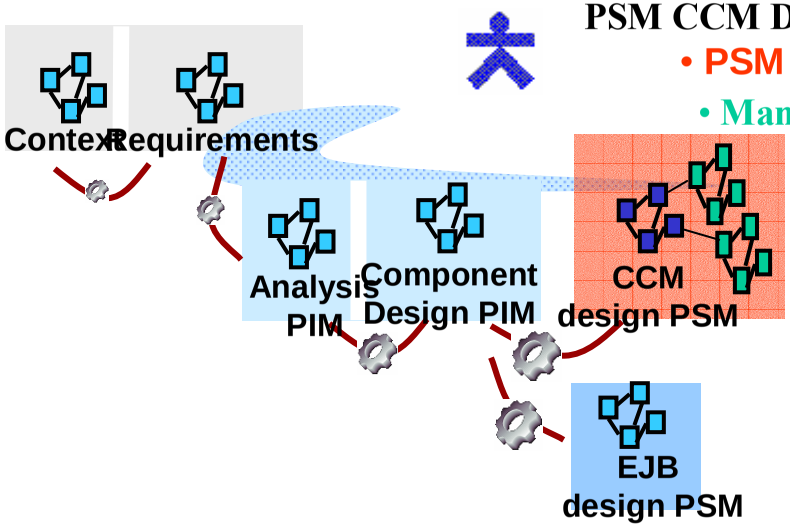


PIM Designer

•Manual refinement

Platform-Independent Models

I



PSM CCM Designer

• PSM GENERATION

• Manual refinement

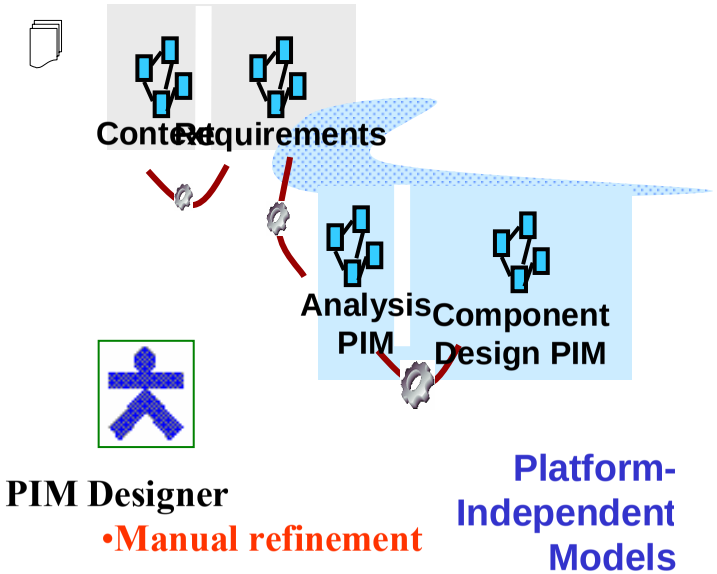
Platform-Specific Models

II

III

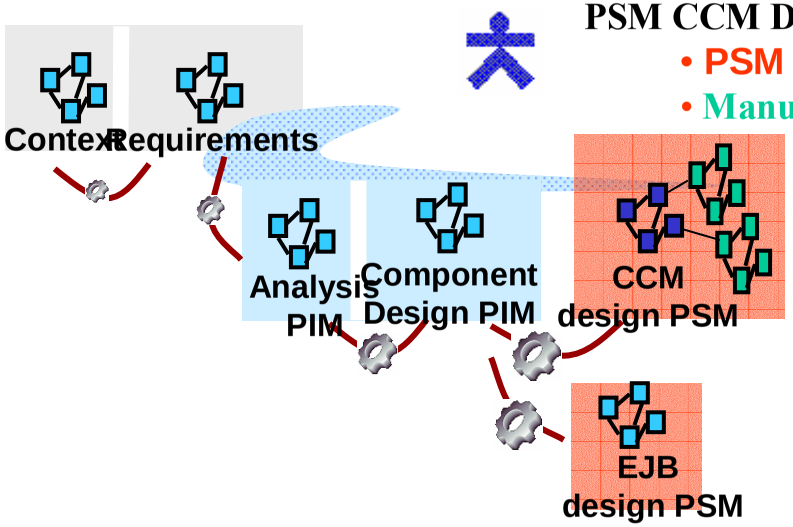
IV

What is Model Synchronization (3/4) ?



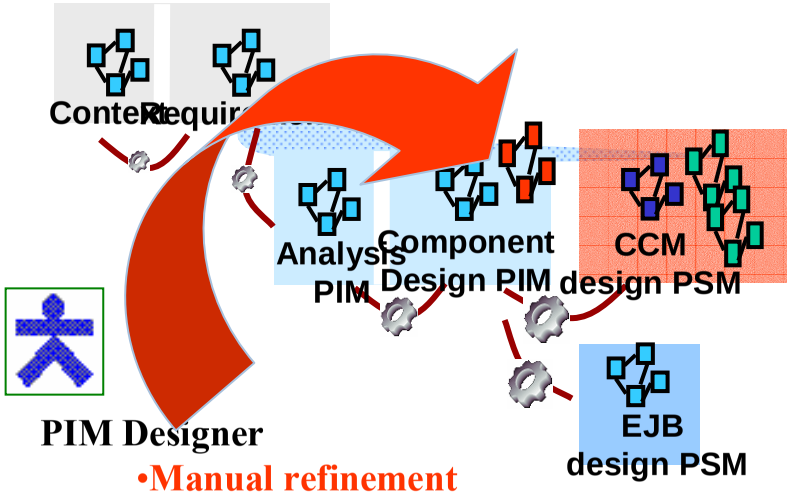
Platform-Independent Models

I



Platform-Specific Models

II

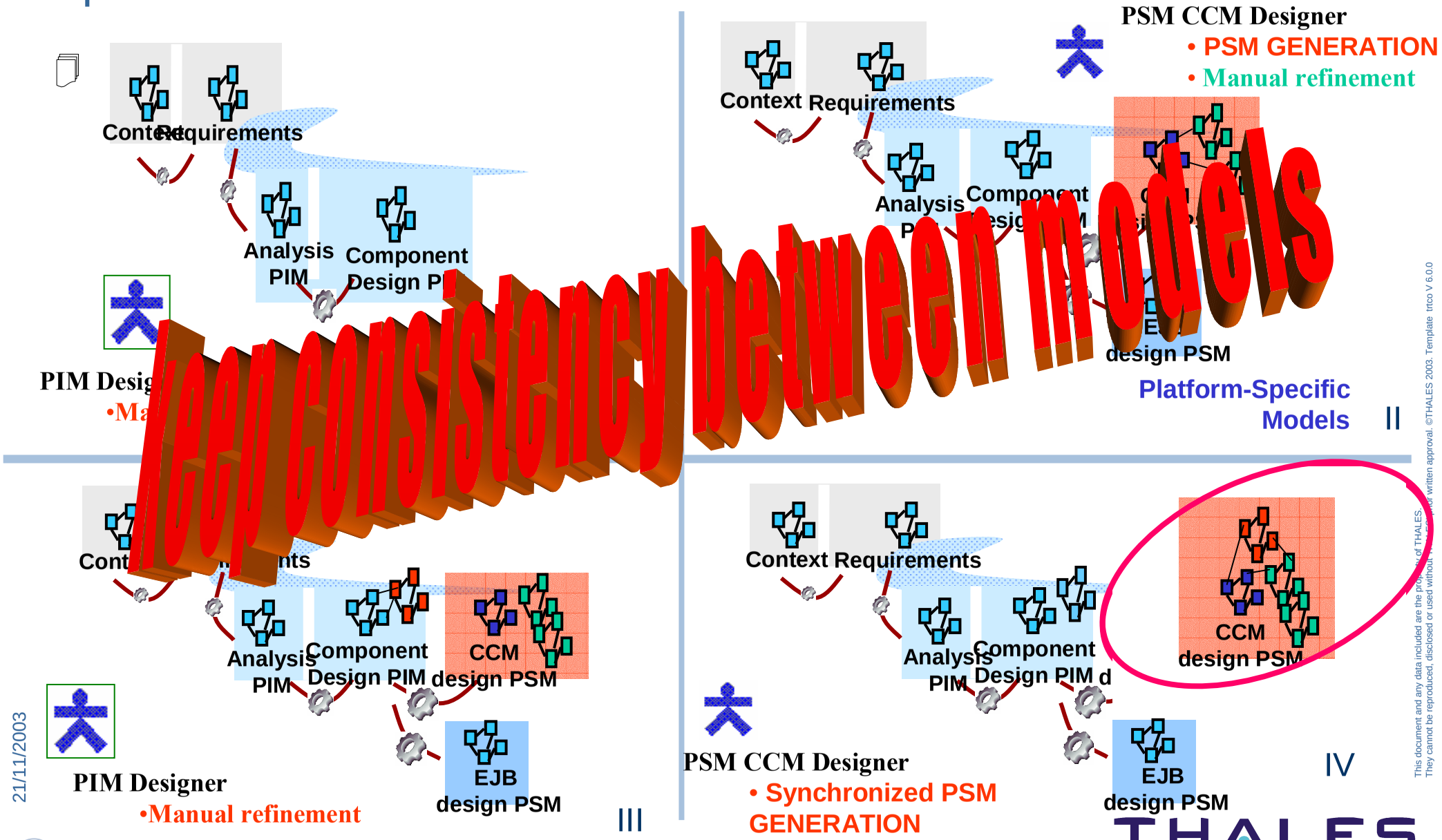


III

IV

THALES

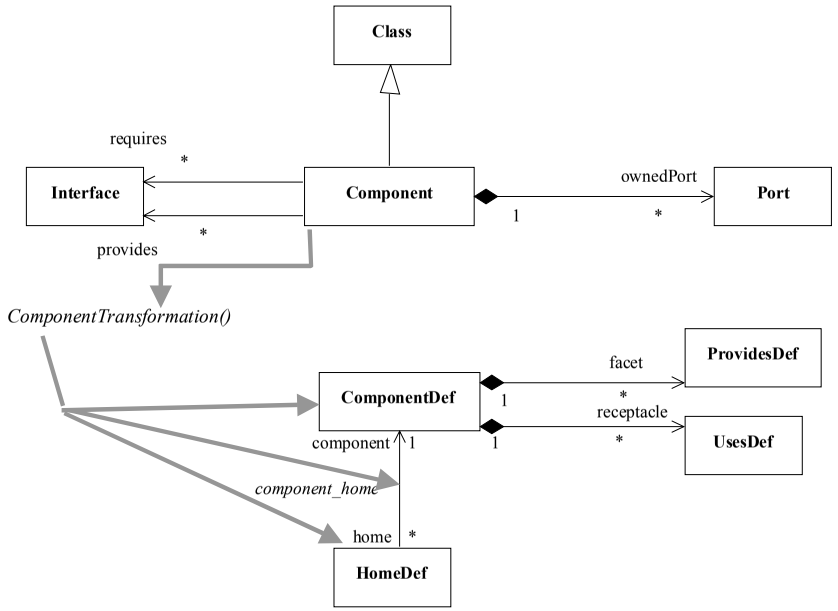
What is Model Synchronization (4/4) ?



Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspots
- Synchronization
  - Principles
  - Model
- Demonstration

Rule declaration principles



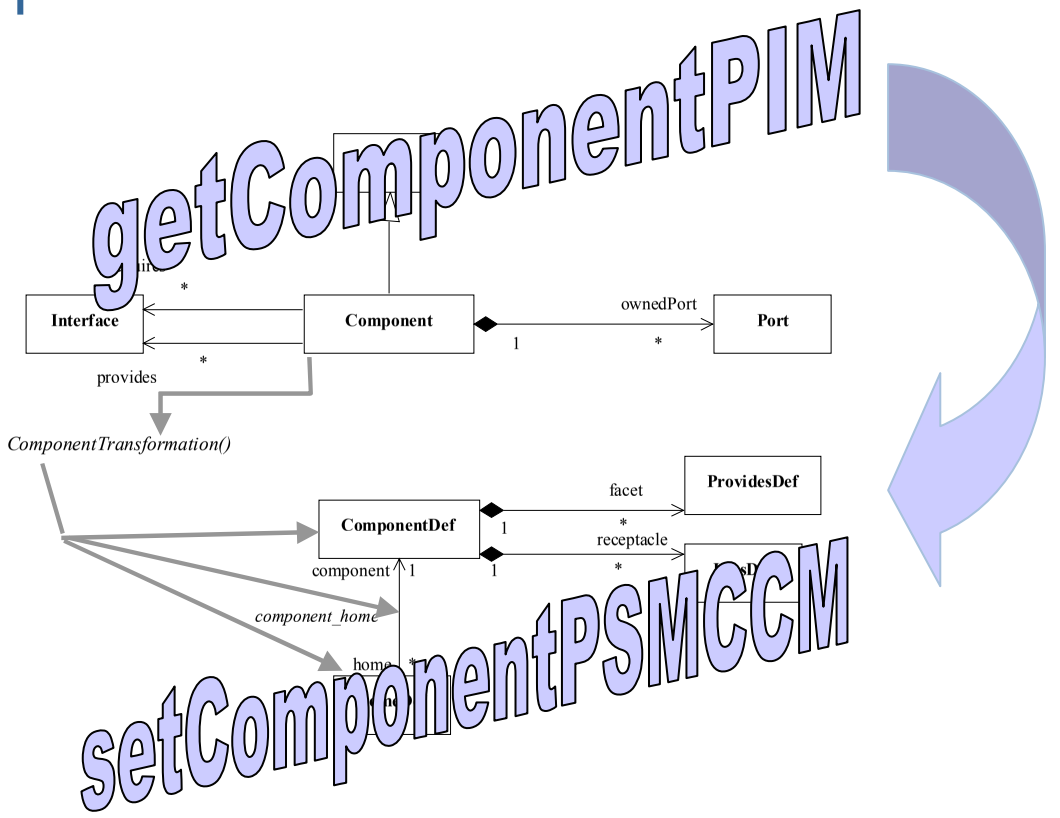

```
// OCL Constraints
ComponentTransformation()
{
context Component inv:
                        let component_name = self.name in
            ComponentDef.allInstances->exists(comp_def: ComponentDef |
                        comp_def.name = component_name and
                        comp_def.home->exists (home_def:HomeDef |
home_def.name = component.name));
}
```

- We choose to define transformation rule between concepts
  - Concepts are meaningful
  - Metamodel level is independent of the projection towards the UML world
- We explicit the transformation with a textual language
  - Richer semantics than graphical language
- We choose
  - OCL 2.0 for its nice model features manipulation and navigation capabilities

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspot
- Synchronization
  - Principles
  - Model
- Demonstration

Rule implementation principles

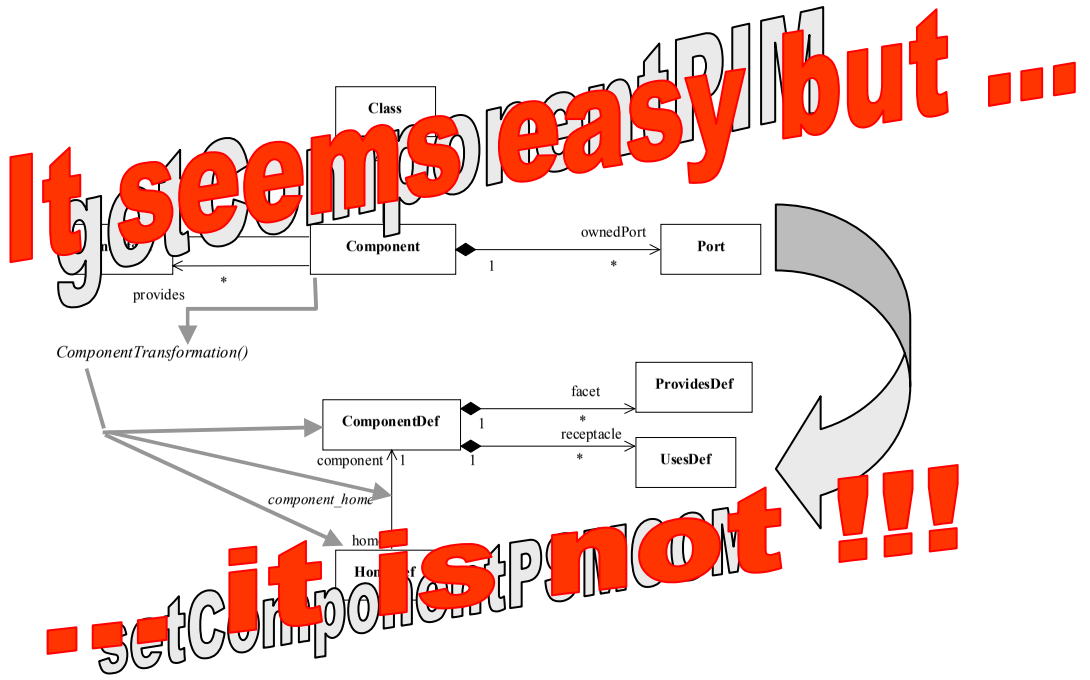


- Use of a MOF API
  - To simulate Meta model (M2) level at Model (M1) level
  - To be profile (projection of Meta model language to UML language) independent
- This API can be generated automatically from meta model and associated profile
  - getComponent, setComponent, append….operations
- Rules result in a combination of get and set operations

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspot
- Synchronization
  - Principles
  - Model
- Demonstration

Rule implementation hotspots

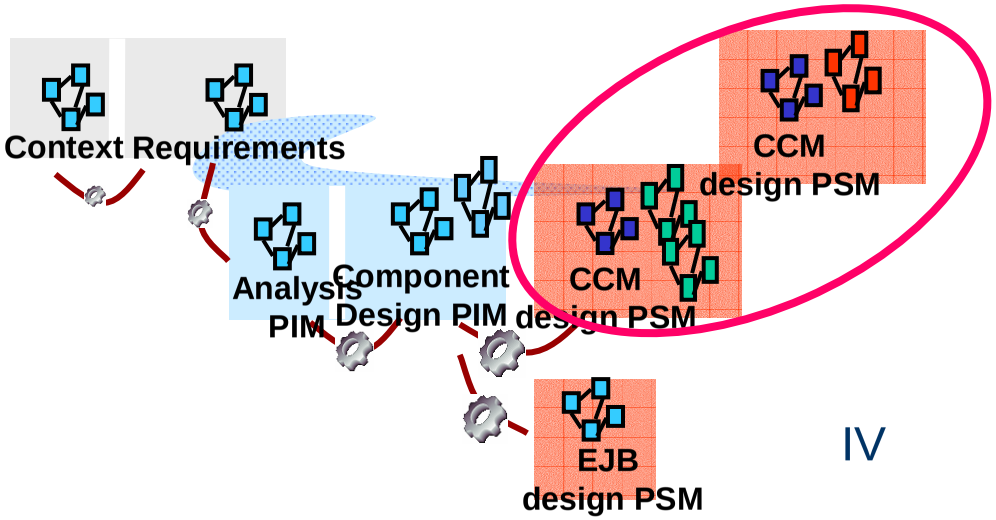


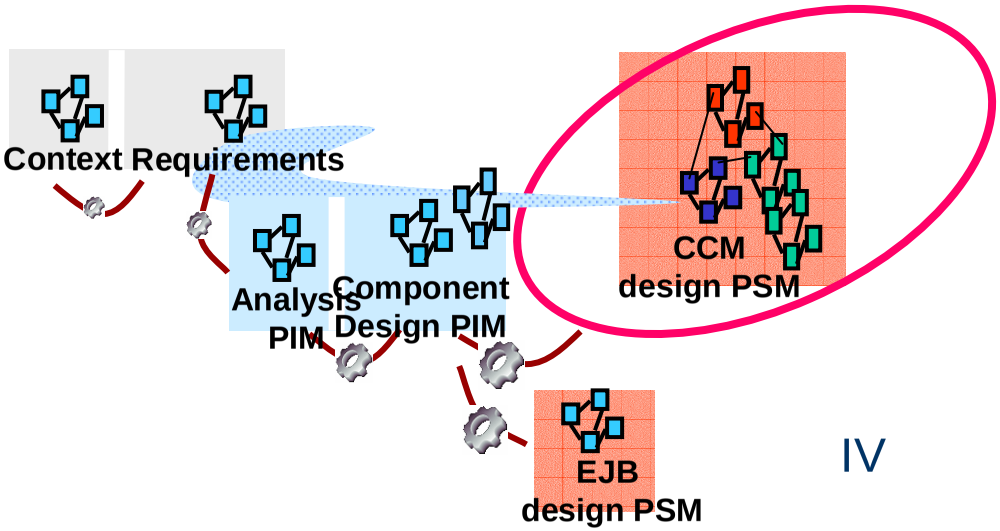
- There is still a gap between rule declaration and rule implementation
  - Visibility: Structure of the target model is not the same structure of the source model
    - Of the target meta model template must be take into account by the transformation rules
  - UML meta model dependency : the AL meta models are Uml meta models extensions
    - Transformation rules must be defined at “Model Element” granularity
  - Rule ordering independency
    - Rule schedule must be rule implementation independent
  - Where storing user specific information ?

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspot
- Synchronization
  - Principles
  - Model
- Demonstration

Objectives remainder

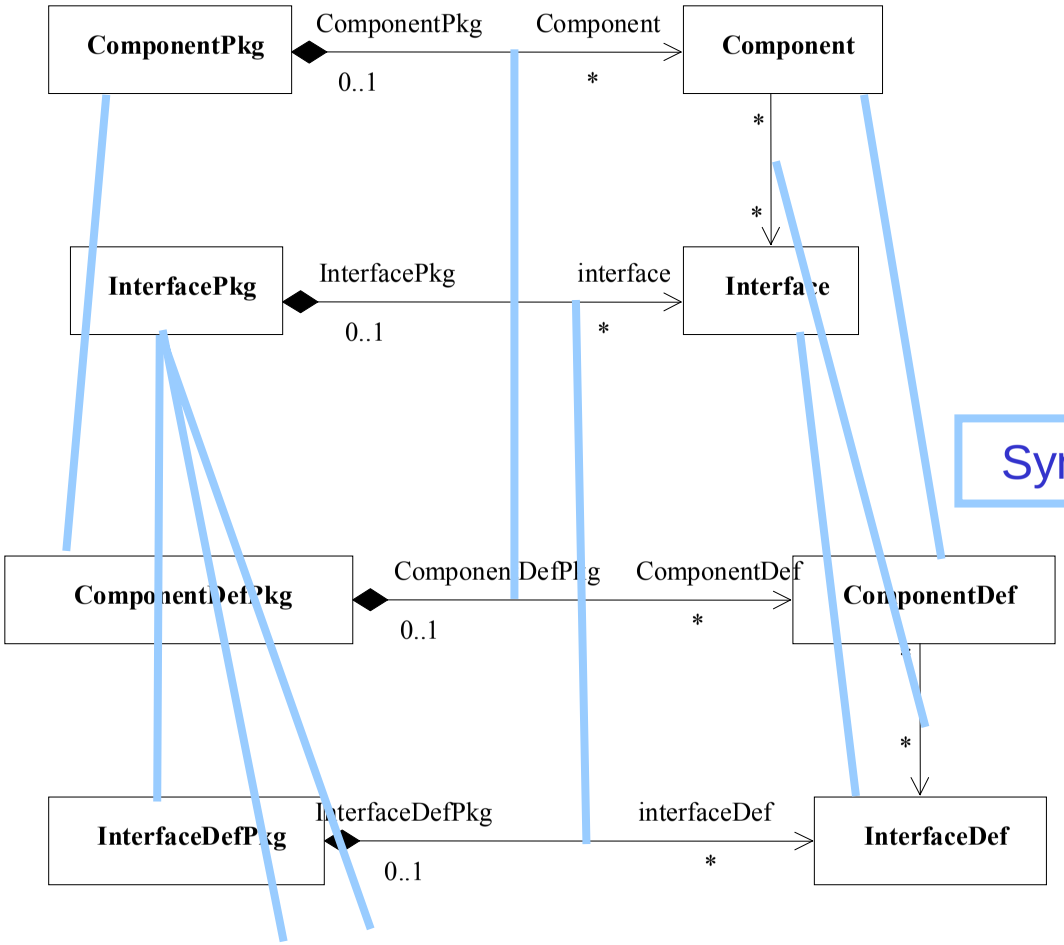




- Why do we need incremental consistency features ?
  - Models evolves currently during lifecycle
  - Modification must be propagate without information loose at the lower levels

Synchronization principles

- How to manage these features?
  - Use synchronization links between model element to update characteristics independently of model element creation
  - Associate synchronization links with transformation rules



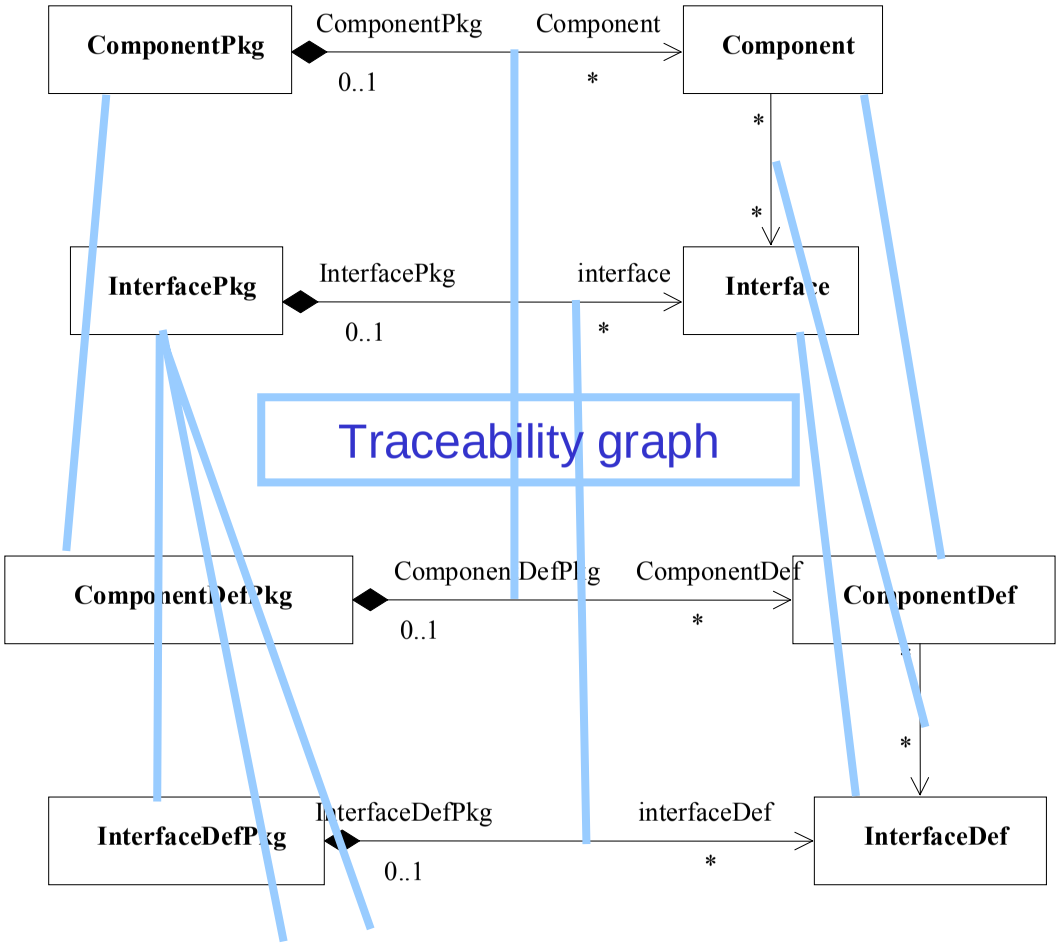
Synchronization links

Meta model level

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
  - Rule implementation hotspot
- Synchronization
  - Principles
  - Model
- Demonstration

Synchronization model



Meta model level

- Traceability information form a graph enhanced with
  - User specific information
    - entity/session bean,…
  - Rule specific information
    - Rule number,…
  - Fine grain traceability link to deal with 1 To M links

Plan

- What is Model Transformation ?
- What is Model Synchronization ?
- Model to model transformation principles
  - Rule declaration
  - Rule implementation
- Synchronization principles
- Demonstration