

CENTRALE  
L Y O N

# Méta-Modélisation

BTD/MAI/MM 1

CENTRALE  
L Y O N

## Méta-modélisation

- But de la méta-modélisation
- Architecture MOF
- 4 niveaux de (méta) modélisation
- Architecture 4 niveaux généralisable en dehors du MOF
- Syntaxes abstraite et concrète
- Profils UML
- Spécialisation et définition de méta-modèles
- Cycle en Y
- Méthodes d'ingénierie : Produit et Processus

BTD/MAI/MM 2

CENTRALE L Y O N	<h2>Normes OMG de modélisation</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● MOF : Meta-Object Facilities           <ul style="list-style-type: none"> <li>→ Langage de définition de méta-modèles</li> </ul> </li> <li>● UML : Unified Modelling Language           <ul style="list-style-type: none"> <li>→ Langage de modélisation</li> </ul> </li> <li>● CWM : Common Warehouse Metamodel           <ul style="list-style-type: none"> <li>→ Modélisation ressources, données, gestion d'une entreprise</li> </ul> </li> <li>● OCL : Object Constraint Language           <ul style="list-style-type: none"> <li>→ Langage de contraintes sur des modèles</li> </ul> </li> <li>● XMI : XML Metadata Interchange           <ul style="list-style-type: none"> <li>→ Standard pour échanges de modèles et méta-modèles</li> </ul> </li> </ul>
BTD/MAI/MM	3

CENTRALE L Y O N	<h2>Normes OMG de modélisation</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Plusieurs de ces normes concernent la définition et l'utilisation de méta-modèles           <ul style="list-style-type: none"> <li>→ MOF : but de la norme</li> <li>→ UML et CWM : peuvent être utilisés pour en définir</li> <li>→ XMI : pour échange de (méta-)modèles entre outils MOF</li> </ul> </li> <li>● MOF est un méta-méta-modèle           <ul style="list-style-type: none"> <li>→ Utilisé pour modéliser des méta-modèles</li> <li>→ Définit les concepts de base (22)</li> <li>→ Entité/classe, relation/association, type de données, référence, package ...</li> <li>→ Le MOF peut définir le MOF</li> </ul> </li> </ul>
BTD/MAI/MM	4

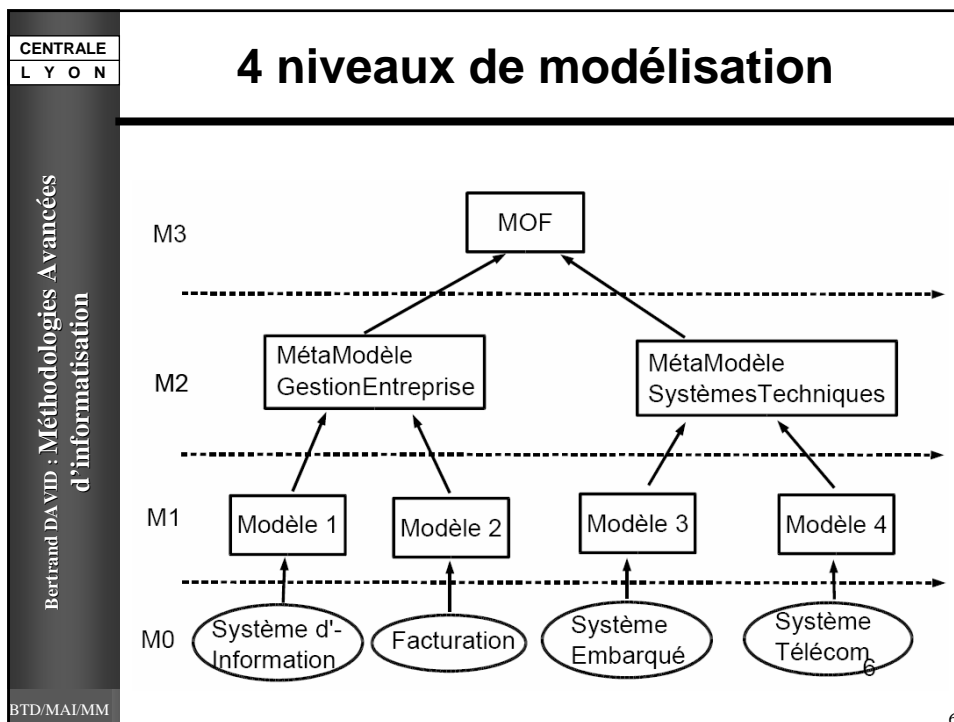
CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées  
d'Informatisation

## 4 Niveaux du MOF

- Le MOF définit 4 niveaux de modélisation
  - M0 : système réel, système modélisé
  - M1 : modèle du système réel défini dans un certain langage
  - M2 : méta-modèle définissant ce langage
  - M3 : méta-méta-modèle définissant le méta-modèle
- Le niveau M3 est le MOF
  - Dernier niveau, il est méta-circulaire : il peut se définir lui même
  - Le MOF est – pour l'OMG – le méta-méta-modèle unique servant de base à la définition de tous les méta-modèles

BTD/MAI/MM 5



CENTRALE L Y O N	<h2>Hiérarchie 4 Niveaux</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● On retrouve cette hiérarchie à 4 niveaux en dehors du MOF et d'UML, dans d'autres espaces technologique que celui de l'OMG</li> <li>● Langage de programmation             <ul style="list-style-type: none"> <li>→ M0 : l'exécution d'un programme</li> <li>→ M1 : le programme</li> <li>→ M2 : la grammaire du langage dans lequel est écrit le programme</li> <li>→ M3 : le concept de grammaire EBNF</li> </ul> </li> <li>● XML             <ul style="list-style-type: none"> <li>→ M0 : données du système</li> <li>→ M1 : données modélisées en XML</li> <li>→ M2 : DTD XML</li> <li>→ M3 : le langage XML</li> </ul> </li> </ul>
BTD/MAI/MM	7

CENTRALE L Y O N	<h2>Méta-modélisation UML</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Dans UML, on retrouve également les 4 niveaux mais avec le niveau M3 définissable en UML directement à la place du MOF</li> <li>● Exemple de système à modéliser (niveau M0)             <ul style="list-style-type: none"> <li>→ Une pièce possède 4 murs, 2 fenêtres et une porte</li> <li>→ Un mur possède une porte ou une fenêtre mais pas les 2 à la fois</li> <li>→ Deux actions sont associées à une porte ou une fenêtre : ouvrir et fermer                 <ul style="list-style-type: none"> <li>➢ Si on ouvre une porte ou une fenêtre fermée, elle devient ouverte</li> <li>➢ Si on ferme une porte ou une fenêtre ouverte, elle devient fermée</li> </ul> </li> </ul> </li> </ul>
BTD/MAI/MM	8

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées  
d'informatisation

BTD/MAI/MM

## Méta-modélisation UML

- Pour modéliser ce système, il faut définir 2 diagrammes UML : niveau M1
  - Un diagramme de classe pour représenter les relations entre les éléments (portes, murs, pièce)
  - Un diagramme d'état pour spécifier le comportement d'une porte ou d'une fenêtre (ouverte, fermée)
- On peut abstraire le comportement des portes et des fenêtres en spécifiant les opérations d'ouverture fermeture dans une interface.
- Le diagramme d'état est associé à cette interface.
- Il faut également ajouter des contraintes OCL pour préciser les contraintes entre les éléments d'une pièce.

BTD/MAI/MM

9

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées  
d'informatisation

BTD/MAI/MM

## M1 : spécification du Système

- **context Mur inv:**
  - fenetre -> union(porte) -> size() <= 1 -- un mur a soit une fenêtre soit une porte (soit rien)
- **context Piece inv:**
  - mur.fenetre -> size() = 2 -- 2 murs de la pièce ont une fenêtre
  - mur.porte -> size() = 1 -- 1 mur de la pièce a une porte

```

classDiagram
    class Piece
    class Mur
    class Porte
    class Fenetre
    Piece "4" *-- "*" Mur
    Piece "1" *-- "*" Porte
    Mur "1" *-- "*" Fenetre
    Mur "1" *-- "*" Porte
    Porte "0..1" *-- "*" Fenetre
    Fenetre "0..1" *-- "*" Porte
    Mur "1" *-- "*" Fenetre
  
```

```

stateDiagram-v2
    [*] --> Ouvert
    Ouvert --> Ferme : fermer
    Ferme --> Ouvert : ouvrir
  
```

*diagramme d'état associé à l'interface Ouverture*

BTD/MAI/MM

10

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées d'informatisation

BTD/MAI/MM

## Méta-modélisation UML

- Les 2 diagrammes de ce modèle de niveau M1 sont des diagrammes UML valides
- Les contraintes sur les éléments des diagrammes UML et leurs relations sont définies dans le méta-modèle UML : niveau M2
  - Un diagramme UML (classe, état ...) doit être conforme au méta-modèle UML
- Méta-modèle UML
  - Diagramme de classe spécifiant la structure de tous types de diagrammes UML
  - Avec contraintes OCL pour spécification précise

BTD/MAI/MM

11

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées d'informatisation

BTD/MAI/MM

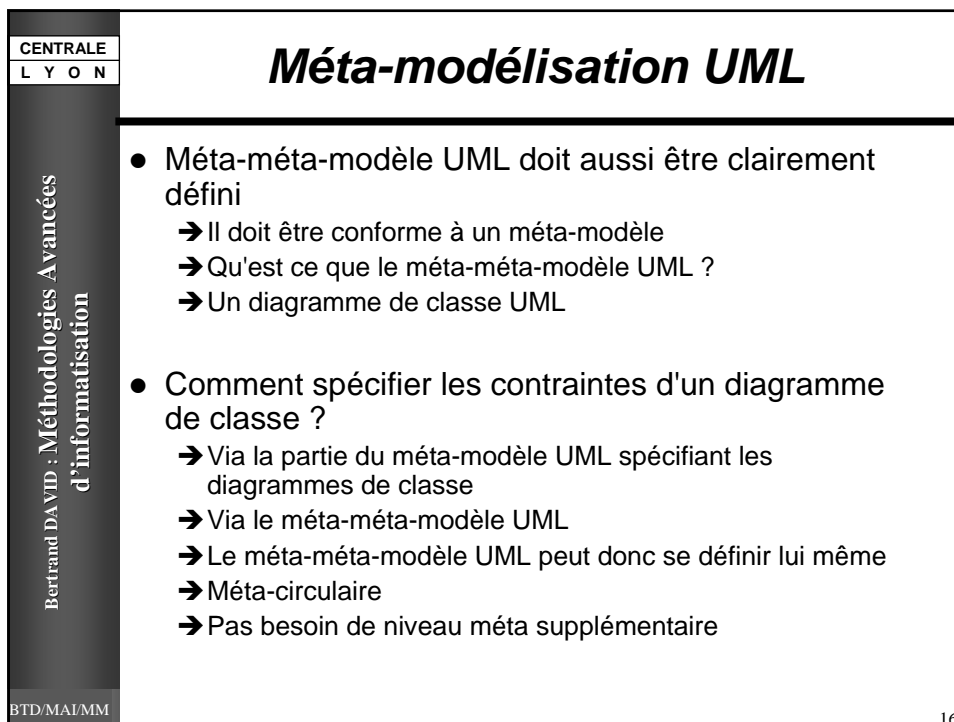
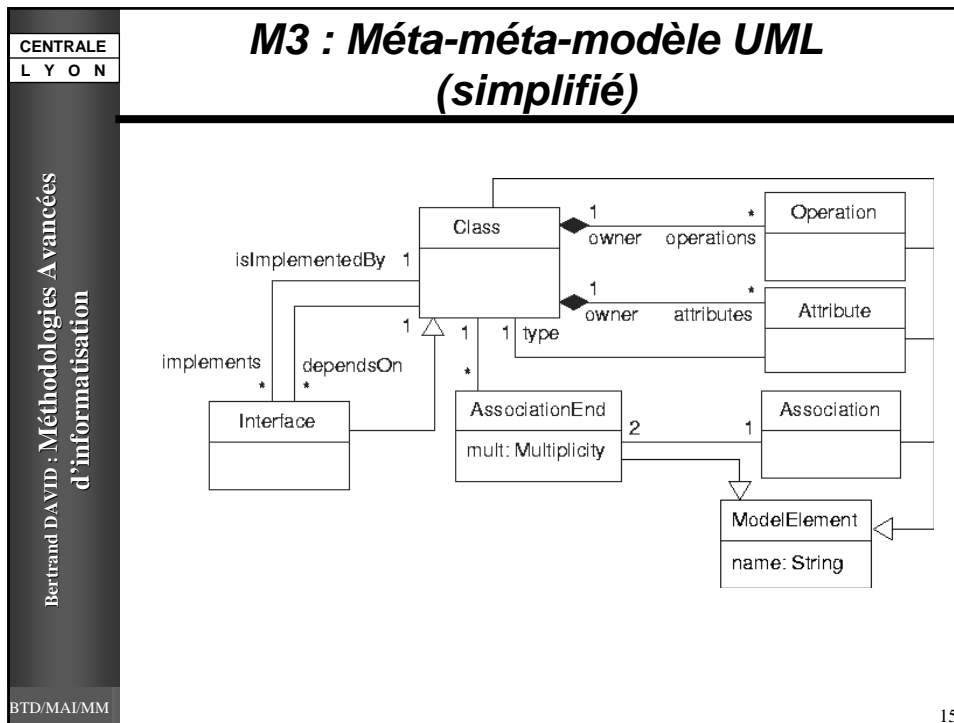
## M2 : Méta-modèle UML (simplifié)

BTD/MAI/MM

12

CENTRALE L Y O N	<h2>M2 : Méta-modèle UML (simplifié)</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Contraintes OCL, quelques exemples           <ul style="list-style-type: none"> <li>→ <b>context</b> Interface <b>inv</b>: attributs -&gt; isEmpty()               <ul style="list-style-type: none"> <li>➢ Une interface est une classe sans attribut</li> </ul> </li> <li>→ <b>context</b> Class <b>inv</b>: attributs -&gt; forAll ( a1, a2   a1 &lt;&gt; a2 <b>implies</b> a1.name &lt;&gt; a2.name)               <ul style="list-style-type: none"> <li>➢ 2 attributs d'une même classe n'ont pas le même nom</li> </ul> </li> <li>→ <b>context</b> StateMachine <b>inv</b>: transition -&gt; forAll ( t   self.state -&gt; includesAll(t.state))               <ul style="list-style-type: none"> <li>➢ Une transition d'un diagramme d'état connecte 2 états de ce diagramme d'état</li> </ul> </li> </ul> </li> </ul>
BTD/MAI/MM	13

CENTRALE L Y O N	<h2>Méta-modélisation UML</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Le méta-modèle UML doit aussi être précisément défini           <ul style="list-style-type: none"> <li>→ Il doit être conforme à un méta-modèle</li> <li>→ C'est le méta-méta-modèle UML</li> </ul> </li> <li>● Qu'est ce que le méta-modèle UML ?           <ul style="list-style-type: none"> <li>→ Un diagramme de classe UML (avec contraintes OCL)</li> </ul> </li> <li>● Comment spécifier les contraintes d'un diagramme de classe ?           <ul style="list-style-type: none"> <li>→ Via le méta-modèle UML</li> <li>→ Ou plus précisément : via la partie du méta-modèle UML spécifiant les diagrammes de classes</li> </ul> </li> <li>● Méta-méta-modèle UML = copie partielle du méta-modèle UML : niveau M3</li> </ul>
BTD/MAI/MM	14



CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées  
d'informatisation

BTD/MAI/MM

## Diagrammes d'instances UML

- Un diagramme d'instances est particulier car
  - Il doit être conforme au méta-modèle UML,
  - qui définit la structure générale des diagrammes d'instances,
  - Il doit aussi être conforme à un diagramme de classe.
- Diagramme de classe est un méta-modèle
  - qui doit être conforme également au méta-modèle UML

```

graph TD
    MMU[Méta-modèle UML]
    UD[Un diagramme de classe]
    UDI[Un diagramme d'instances]
    UD -- conforme à --> MMU
    UDI -- conforme à --> MMU
    UDI -- conforme à --> UD
      
```

Le diagramme illustre les relations de conformité entre trois éléments UML : le Méta-modèle UML, un diagramme de classe, et un diagramme d'instances. Des flèches pointent de 'Un diagramme de classe' et 'Un diagramme d'instances' vers 'Méta-modèle UML', toutes deux étiquetées 'conforme à'. Une autre flèche pointe de 'Un diagramme d'instances' vers 'Un diagramme de classe', également étiquetée 'conforme à'.

BTD/MAI/MM

17

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées  
d'informatisation

BTD/MAI/MM

## Syntaxe

- Un langage est défini par un méta-modèle
- Un langage possède une syntaxe respectant le méta-modèle
- Syntaxe textuelle
  - Ensemble de mots-clés et de mots respectant des contraintes définies selon des règles précises
  - Notions de syntaxe et de grammaire dans les langages
- Exemple pour langage Java
  - `public class MaClasse implements MonInterface { ... }`
- Grammaire Java pour déclaration de classe
  - `class_declaration ::= { modifier } "class" identifieur [ "extends« class_name ] [ "implements" interface_name { ",« interface_name } ] "{ { field_declaration } }`

BTD/MAI/MM

18

CENTRALE L Y O N	<h1>Syntaxe</h1>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Syntaxe graphique           <ul style="list-style-type: none"> <li>→ Notation graphique, chaque élément a une forme graphique particulière</li> </ul> </li> <li>● Exemple : associations entre classes/interfaces sur les diagrammes de classe UML           <ul style="list-style-type: none"> <li>→ Trait normal : association</li> <li>→ Flèche, trait pointillé : dépendance</li> <li>→ Flèche en forme de triangle, trait en pointillé : implémentation</li> <li>→ Flèche en forme de triangle, trait plein : spécialisation</li> </ul> </li> </ul>
BTD/MAI/MM	19

CENTRALE L Y O N	<h1>Syntaxe</h1>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Syntaxe abstraite/concrète</li> <li>● Abstraite           <ul style="list-style-type: none"> <li>→ Les éléments et leurs relations sans une notation spécialisée</li> <li>→ Correspond à ce qui est défini au niveau du méta-modèle</li> </ul> </li> <li>● Concrète           <ul style="list-style-type: none"> <li>→ Syntaxe graphique ou textuelle définie pour un type de modèle</li> <li>→ Plusieurs syntaxes concrètes possibles pour une même syntaxe abstraite</li> <li>→ Un modèle peut être défini via n'importe quelle syntaxe               <ul style="list-style-type: none"> <li>➢ L'abstraite</li> <li>➢ Une des concrètes</li> </ul> </li> </ul> </li> <li>● MOF : langage pour définir des méta-modèles           <ul style="list-style-type: none"> <li>→ Pas de syntaxe concrète définie</li> </ul> </li> </ul>
BTD/MAI/MM	20



CENTRALE L Y O N	<h2>Spécification de méta-modèles</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● But : définir un type de modèle avec tous ces types d'éléments et leurs contraintes</li> <li>● Trois approches possibles             <ul style="list-style-type: none"> <li>→ Définir un méta-modèle nouveau à partir de rien</li> <li>→ Modifier un méta-modèle existant : ajout, suppression, modification d'éléments et des contraintes sur leurs relations</li> <li>→ Correspond au MOF, décomposé en 2 parties                 <ul style="list-style-type: none"> <li>➢ E-MOF : essential MOF, les méta-éléments de base réutilisés tels quels dans tous les méta-modèles</li> <li>➢ MOF : un méta-modèle particulier défini via E-MOF</li> </ul> </li> </ul> </li> <li>● Spécialiser un méta-modèle existant en rajoutant des éléments et des contraintes (sans en enlever)             <ul style="list-style-type: none"> <li>→ Correspond aux profils UML</li> </ul> </li> </ul>
BTD/MAI/MM	23

CENTRALE L Y O N	<h2>Profils UML</h2>
Bertrand DAVID : Méthodologies Avancées d'informatisation	<ul style="list-style-type: none"> <li>● Un profil est une spécialisation du méta-modèle UML             <ul style="list-style-type: none"> <li>→ Ajouts de nouveaux types d'éléments et des contraintes sur leurs relations entre eux et avec les éléments d'UML</li> <li>→ Ajouts de contraintes sur éléments existants d'UML</li> <li>→ Ajouts de contraintes sur relations existantes entre les éléments d'UML</li> <li>→ Aucune suppression de contraintes ou d'éléments</li> </ul> </li> <li>● Profil : mécanisme d'extension d'UML pour l'adapter à un contexte métier ou technique particulier             <ul style="list-style-type: none"> <li>→ Profil pour composants EJB</li> <li>→ Profil pour gestion bancaire</li> <li>→ Profil pour architecture logicielle</li> </ul> </li> </ul>
BTD/MAI/MM	24

CENTRALE  
L Y O N

## Profils UML

Bertrand DAVID : Méthodologies Avancées d'informatisation

- Stéréotype : extension, spécialisation d'un élément du méta-modèle
  - Classe, association, attribut, opération ...
  - Le nom d'un stéréotype est marqué entre << ... >>
  - Il existe déjà des stéréotypes dans UML
  - << interface >> : une interface est une classe particulière (sans attribut)
  - On peut marquer des attributs d'une classe pour préciser une contrainte ou un rôle particulier : tagged value
- Exemple {unique} id: int

```

classDiagram
    class Client {
        <<process>>
    }
    class Server {
        <<process>>
        {unique} IPadd: IP
    }
    class BDD {
        <<data>>
    }
    Client --> Server : <<uses>>
    Server --> BDD : <<query>>
          
```

BTD/MAI/MM

25

CENTRALE  
L Y O N

## Profils UML

Bertrand DAVID : Méthodologies Avancées d'informatisation

- Profil UML est composé de 3 types d'éléments
  - Des stéréotypes
  - Des tagged value
  - Des contraintes (exprimables en OCL)
    - Sur ces stéréotypes, tagged value
    - Sur des éléments du méta-modèle existant
    - Sur les relations entre les éléments
- Un profil UML est défini sous la forme d'un package stéréotypé << profile >>
- Exemple de profil : architecture logicielle
  - Des composants clients et serveur
  - Un client est associé à un serveur via une interface de service par l'intermédiaire d'un proxy

BTD/MAI/MM

26

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées d'informatisation

## Exemple Profil

- Profil nommé ClientProxyServer
- Définit trois stéréotypes : Trois classes jouant un rôle particulier : extensions de la méta-class Class du méta-modèle UML
  - Server, Proxy, Client

BTD/MA/MM

27

CENTRALE  
L Y O N

Bertrand DAVID : Méthodologies Avancées d'informatisation

## Exemple Profil

- Pour compléter le profil, ajout de contraintes OCL
  - Navigation sur le méta-modèle (simplifié) en considérant que la méta-class Class a trois spécialisations (Server, Client, Proxy)
    - **context** Client **inv:**let proxies = self.associationEnd.associationEnd.class -> select ( c | c.oclIsTypeOf(Proxy)) **in** let interfaces = self.dependsOn **in** interfaces -> forAll ( i | proxies.implements -> includes (i) **and** proxies -> forAll ( p | p.implements -> includes (i) **implies** p.hasClassRefWith(self)))
    - **context** Class **def:** hasClassRefWith(cl : Class) : Boolean = self.associationEnd.associationEnd.class -> exists ( c | c = cl )
  - Un proxy associé à un client doit implémenter une des interfaces dont dépend le client et un proxy implémentant une interface d'un client doit avoir une association avec ce client

BTD/MA/MM

28

