

# Informatique - Bases de données

## Génie Biologique et Alimentaire 4<sup>ème</sup> année

Olivier Caron<sup>1</sup>

<http://ocaron.polytech-lille.net>

<sup>1</sup>École d'ingénieurs Polytech Lille  
Université de Lille

13 septembre 2024





# Le programme (1/2)

- 10 heures de Cours

# Le programme (1/2)

- 10 heures de Cours
- 12 heures de TP (un contrôle individuel inclus)

# Le programme (1/2)

- 10 heures de Cours
- 12 heures de TP (un contrôle individuel inclus)
- Contrôle des connaissances :

# Le programme (1/2)

- 10 heures de Cours
- 12 heures de TP (un contrôle individuel inclus)
- Contrôle des connaissances :
  - ▶ 1 DS (1 heure), un contrôle TP individuel

# Le programme (1/2)

- 10 heures de Cours
- 12 heures de TP (un contrôle individuel inclus)
- Contrôle des connaissances :
  - ▶ 1 DS (1 heure), un contrôle TP individuel
  - ▶ Tous documents **papiers** autorisés

# Le programme (2/2)

- Plates-formes :

# Le programme (2/2)

- Plates-formes :
  - ▶ Atelier de modélisation UML :

# Le programme (2/2)

- Plates-formes :
  - ▶ Atelier de modélisation UML :
    - ★ Modelio (<https://www.modelio.org/>)

# Le programme (2/2)

- Plates-formes :
  - ▶ Atelier de modélisation UML :
    - ★ Modelio (<https://www.modelio.org/>)
  - ▶ Système de Gestion de bases de Données Relationnelles (SGBDR) :  
Postgres sous Unix (<http://www.postgresql.org>)

# Le programme (2/2)

- Plates-formes :

- ▶ Atelier de modélisation UML :
  - ★ Modelio (<https://www.modelio.org/>)
- ▶ Système de Gestion de bases de Données Relationnelles (SGBDR) :  
Postgres sous Unix (<http://www.postgresql.org>)
- ▶ *Suite bureautique libreOffice*

# La modélisation, pourquoi faire ?

- Cahier des charges **informel** défini par le "client"

# La modélisation, pourquoi faire ?

- Cahier des charges **informel** défini par le "client"
- Phase d'analyse : échange de questions/réponses, affiner le cahier des charges, **évaluer la faisabilité**, lever toutes les ambiguïtés, etc

# La modélisation, pourquoi faire ?

- Cahier des charges **informel** défini par le "client"
- Phase d'analyse : échange de questions/réponses, affiner le cahier des charges, **évaluer la faisabilité**, lever toutes les ambiguïtés, etc
- Modélisation du cahier des charges : le langage commun entre analystes et gestionnaires de bases de données

# La modélisation, pourquoi faire ?

- Cahier des charges **informel** défini par le "client"
- Phase d'analyse : échange de questions/réponses, affiner le cahier des charges, **évaluer la faisabilité**, lever toutes les ambiguïtés, etc
- Modélisation du cahier des charges : le langage commun entre analystes et gestionnaires de bases de données
- Fournit un guide aux gestionnaires de BD, absence d'imprécisions.

# UML en quelques points



- U.M.L pour Unified Modeling Language <http://www.uml.org>

# UML en quelques points



- U.M.L pour Unified Modeling Language <http://www.uml.org>
- Norme délivrée par le consortium O.M.G. <http://www.omg.org>

# UML en quelques points



- U.M.L pour Unified Modeling Language <http://www.uml.org>
- Norme délivrée par le consortium O.M.G. <http://www.omg.org>
- Une **notation graphique** (ou cadre de modélisation), pas une méthodologie

# UML en quelques points



- U.M.L pour Unified Modeling Language <http://www.uml.org>
- Norme délivrée par le consortium O.M.G. <http://www.omg.org>
- Une **notation graphique** (ou cadre de modélisation), pas une méthodologie
- Indépendante de toute méthodologie, utilisable pour des bases de données, des logiciels,...

# F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?  
⇒ ben, non...

# F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - ⇒ ben, non...
- A quoi sert UML ?
  - ⇒ à décrire, spécifier un système complexe, sous plusieurs angles afin de mieux le comprendre
  - ⇒ à avoir un formalisme moins ambiguë que le langage naturel
  - ⇒ disposer d'un format d'échange, de travail collaboratif

# F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?  
⇒ ben, non...
- A quoi sert UML ?  
⇒ à décrire, spécifier un système complexe, sous plusieurs angles afin de mieux le comprendre  
⇒ à avoir un formalisme moins ambiguë que le langage naturel  
⇒ disposer d'un format d'échange, de travail collaboratif
- Quelles sont les tendances d'UML ?  
⇒ "opérationnaliser" UML : outil de construction de modèles, outil de contrôle de modèles, outil de génération de code, ...

# Le diagramme de classes

- Un parmi les nombreux types de diagrammes UML

# Le diagramme de classes

- Un parmi les nombreux types de diagrammes UML
- Objectifs :

# Le diagramme de classes

- Un parmi les nombreux types de diagrammes UML
- Objectifs :
  - ▶ Puissance de représentation

# Le diagramme de classes

- Un parmi les nombreux types de diagrammes UML
- Objectifs :
  - ▶ Puissance de représentation
  - ▶ Stabilité et flexibilité : un **ajout** de donnée ne doit pas remettre en cause le schéma.

# Le diagramme de classes

- Un parmi les nombreux types de diagrammes UML
- Objectifs :
  - ▶ Puissance de représentation
  - ▶ Stabilité et flexibilité : un **ajout** de donnée ne doit pas remettre en cause le schéma.
  - ▶ Simplicité : facilité de compréhension et d'utilisation

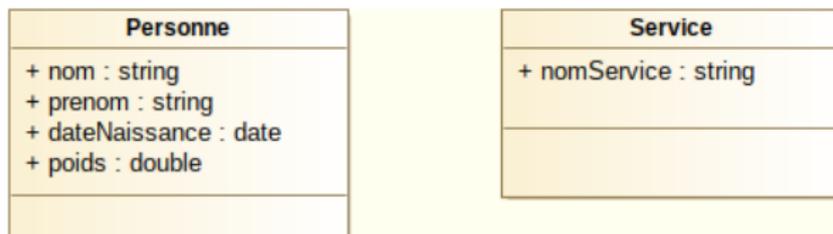
# Le diagramme de classes

- Un parmi les nombreux types de diagrammes UML
- Objectifs :
  - ▶ Puissance de représentation
  - ▶ Stabilité et flexibilité : un **ajout** de donnée ne doit pas remettre en cause le schéma.
  - ▶ Simplicité : facilité de compréhension et d'utilisation
  - ▶ Indépendance par rapport à l'implémentation cible (SGBDR, fichiers, programmes. . .)

# L'entité ou classe

## Définition

*Classe* : c'est un objet discernable d'autres objets comme, par exemple, une personne, une voiture mais qui peut être aussi un concept ou une grandeur abstraite. L'entité est définie par une liste d'attributs qui la caractérisent. Celles-ci constituent le plus petit élément d'information ayant un sens par lui-même.



**Règle** : Une classe décrit **un** concept, donc **jamais** de pluriel au nom de la classe.

# Les attributs ou propriétés d'une classe

- Liste d'éléments qui caractérisent une classe

# Les attributs ou propriétés d'une classe

- Liste d'éléments qui caractérisent une classe
- Les propriétés/attributs peuvent être typé(e)s.  
fixé dès le départ ou à tout moment : UML sert à la conception (processus itératif).

# Les attributs ou propriétés d'une classe

- Liste d'éléments qui caractérisent une classe
- Les propriétés/attributs peuvent être typé(e)s.  
fixé dès le départ ou à tout moment : UML sert à la conception (processus itératif).
- UML dispose de types primitifs (string, double, boolean, integer, char).

# Les attributs ou propriétés d'une classe

- Liste d'éléments qui caractérisent une classe
- Les propriétés/attributs peuvent être typé(e)s.  
fixé dès le départ ou à tout moment : UML sert à la conception (processus itératif).
- UML dispose de types primitifs (string, double, boolean, integer, char).
- Certains ateliers proposent d'autres types (ex : date sur modelio)

# Les attributs ou propriétés d'une classe

- Liste d'éléments qui caractérisent une classe
- Les propriétés/attributs peuvent être typé(s).  
fixé dès le départ ou à tout moment : UML sert à la conception (processus itératif).
- UML dispose de types primitifs (string, double, boolean, integer, char).
- Certains ateliers proposent d'autres types (ex : date sur modelio)
- Possibilité d'ajout de nouveaux types via les DataType.



# Classe d'entité

## Définition

*Classe d'entité* : c'est l'ensemble des entités de même type qu'il est possible de définir au cours du temps.

Exemple : le `personnel`, classe d'entité de l'entité employé  
On parle également d'*instances* ou d'*objets*

# Association

## Définition

C'est un lien logique entre l'ensemble des entités appartenant à plusieurs classes différentes.

# Association

## Définition

C'est un lien logique entre l'ensemble des entités appartenant à plusieurs classes différentes.

- Souvent perçue comme une action menée vis à vis des entités (et se traduit alors par un verbe).

## Définition

C'est un lien logique entre l'ensemble des entités appartenant à plusieurs classes différentes.

- Souvent perçue comme une action menée vis à vis des entités (et se traduit alors par un verbe).
- La mise en relation d'entités peut faire apparaître des propriétés qui n'appartiennent en propre à aucune des entités. On distingue :

## Définition

C'est un lien logique entre l'ensemble des entités appartenant à plusieurs classes différentes.

- Souvent perçue comme une action menée vis à vis des entités (et se traduit alors par un verbe).
- La mise en relation d'entités peut faire apparaître des propriétés qui n'appartiennent en propre à aucune des entités. On distingue :
  - ▶ Les associations binaires relient les différentes instances de deux classes d'entité (imposé par OMT et ODMG)

## Définition

C'est un lien logique entre l'ensemble des entités appartenant à plusieurs classes différentes.

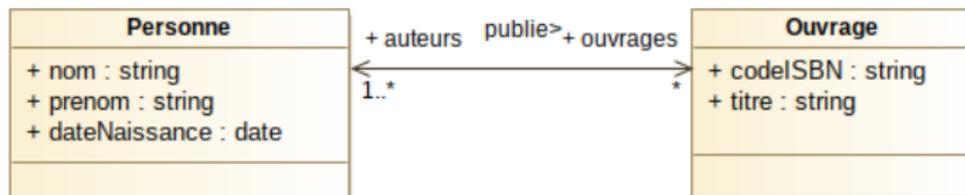
- Souvent perçue comme une action menée vis à vis des entités (et se traduit alors par un verbe).
- La mise en relation d'entités peut faire apparaître des propriétés qui n'appartiennent en propre à aucune des entités. On distingue :
  - ▶ Les associations binaires relient les différentes instances de deux classes d'entité (imposé par OMT et ODMG)
  - ▶ Les associations n-aires relient les instances de n-classes d'entité.

## Définition

C'est un lien logique entre l'ensemble des entités appartenant à plusieurs classes différentes.

- Souvent perçue comme une action menée vis à vis des entités (et se traduit alors par un verbe).
- La mise en relation d'entités peut faire apparaître des propriétés qui n'appartiennent en propre à aucune des entités. On distingue :
  - ▶ Les associations binaires relient les différentes instances de deux classes d'entité (imposé par OMT et ODMG)
  - ▶ Les associations n-aires relient les instances de n-classes d'entité.
  - ▶ Les associations réflexives relient une instance d'une classe d'entité à d'autres instances d'une même classe.

# Association binaire



- Utilisation de '`<`' ou '`>`' pour précisez le sens de lecture du nom de l'association (facultatif)

# Les constituants d'une association

- Facultatifs :

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association
  - ▶ La navigation

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association
  - ▶ La navigation
  - ▶ Les propriétés (ou attributs) d'une association

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association
  - ▶ La navigation
  - ▶ Les propriétés (ou attributs) d'une association
- Obligatoires (sinon sémantique trop faible) :

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association
  - ▶ La navigation
  - ▶ Les propriétés (ou attributs) d'une association
- Obligatoires (sinon sémantique trop faible) :
  - ▶ Les cardinalités de l'association

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association
  - ▶ La navigation
  - ▶ Les propriétés (ou attributs) d'une association
- Obligatoires (sinon sémantique trop faible) :
  - ▶ Les cardinalités de l'association
- Préférable d'en dire plus que pas assez.

# Les constituants d'une association

- Facultatifs :
  - ▶ Le nom de l'association
  - ▶ Le nom de rôle de l'association aux extrémités de l'association
  - ▶ La navigation
  - ▶ Les propriétés (ou attributs) d'une association
- Obligatoires (sinon sémantique trop faible) :
  - ▶ Les cardinalités de l'association
- Préférable d'en dire plus que pas assez.
- Même si plusieurs constituants, pris séparément, sont facultatifs, certains constituants deviennent obligatoires en l'absence d'autres.  
Exemple : au moins un nom de rôle et/ou association si aucun autre nom fixé (rôle/association).

# La navigation

- Permet de se déplacer dans le modèle

# La navigation

- Permet de se déplacer dans le modèle
- Correspond aux différents scénarios (diagrammes dynamiques)

# La navigation

- Permet de se déplacer dans le modèle
- Correspond aux différents scénarios (diagrammes dynamiques)
- Par défaut, les associations sont navigables dans les deux directions (un simple trait suffit)

# La navigation

- Permet de se déplacer dans le modèle
- Correspond aux différents scénarios (diagrammes dynamiques)
- Par défaut, les associations sont navigables dans les deux directions (un simple trait suffit)
- Dépend de la cible du modèle : programmes, bases de données, etc

# La navigation

- Permet de se déplacer dans le modèle
- Correspond aux différents scénarios (diagrammes dynamiques)
- Par défaut, les associations sont navigables dans les deux directions (un simple trait suffit)
- Dépend de la cible du modèle : programmes, bases de données, etc
- **Information peu sensible pour bases de données relationnelles car tout est navigable.**

# Cardinalités d'une association

- Elles précisent les nombres minimum et maximum d'occurrences d'une entité pouvant être impliquées dans les occurrences d'association.

# Cardinalités d'une association

- Elles précisent les nombres minimum et maximum d'occurrences d'une entité pouvant être impliquées dans les occurrences d'association.
- Définie au niveau de chaque extrémité de l'association

# Cardinalités d'une association

- Elles précisent les nombres minimum et maximum d'occurrences d'une entité pouvant être impliquées dans les occurrences d'association.
- Définie au niveau de chaque extrémité de l'association

<b>représentation</b>	<b>signification</b>
1	un et un seul
0..1	zéro et un
M..N	de M à N
*	de zéro à plusieurs
0..*	de zéro à plusieurs
1..*	de un à plusieurs

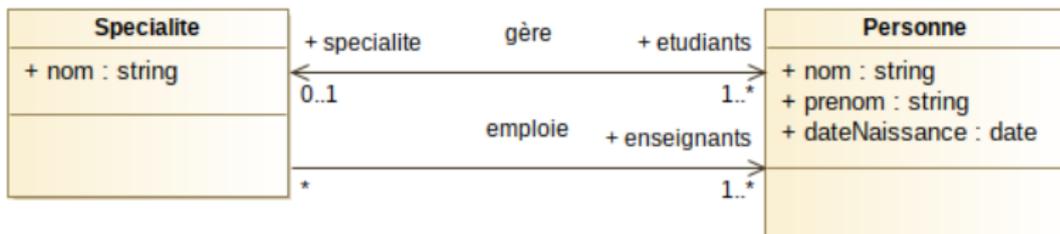
- Conventions d'affichage :

# Les rôles d'une association

- Chaque classe joue des rôles différents selon les associations où elle intervient.

# Les rôles d'une association

- Chaque classe joue des rôles différents selon les associations où elle intervient.
- Représentation graphique UML :



## Les propriétés (attributs) d'une association

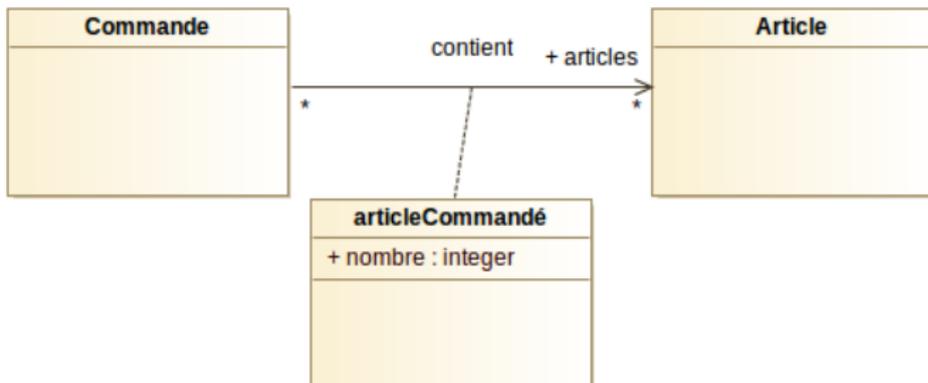
- L'attribut n'a de sens que pour l'association, ne dépend donc pas d'une seule classe.

# Les propriétés (attributs) d'une association

- L'attribut n'a de sens que pour l'association, ne dépend donc pas d'une seule classe.
- On parle de classe-association pour contenir ces attributs

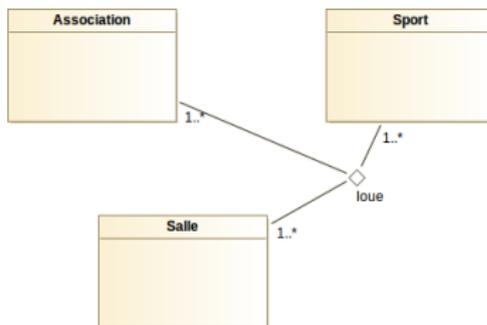
# Les propriétés (attributs) d'une association

- L'attribut n'a de sens que pour l'association, ne dépend donc pas d'une seule classe.
- On parle de classe-association pour contenir ces attributs
- Représentation graphique UML :



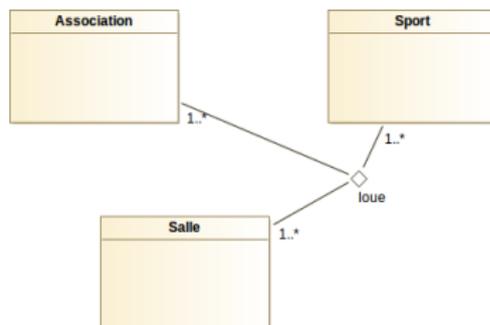
# Association n-aire

- Représentation graphique UML :



# Association n-aire

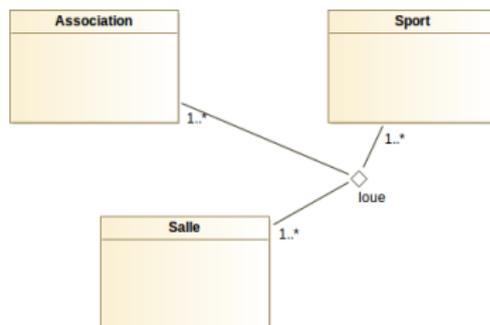
- Représentation graphique UML :



- Définition des cardinalités : le couple ( min , max ) correspond au nombre d'occurrences possibles d'entités associées dans la relation quand les autres valeurs sont fixées

# Association n-aire

- Représentation graphique UML :



- Définition des cardinalités : le couple ( min , max ) correspond au nombre d'occurrences possibles d'entités associées dans la relation quand les autres valeurs sont fixées
- On peut transformer toute association n-aire en associations binaires (sémantique plus simple pour les cardinalités)

# Composition - Agrégation

- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association

# Composition - Agrégation

- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association
- Une des classes joue un rôle plus important : la classe Maître

# Composition - Agrégation

- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association
- Une des classes joue un rôle plus important : la classe Maître
- Composition (losange noir) :

# Composition - Agrégation

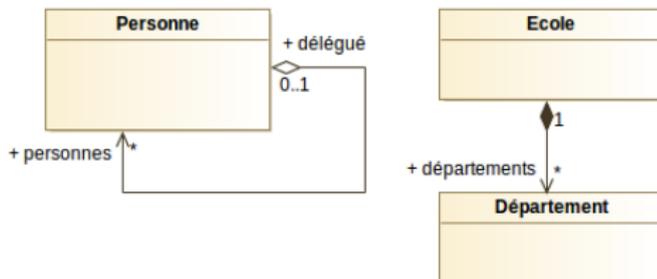
- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association
- Une des classes joue un rôle plus important : la classe Maître
- Composition (losange noir) :
  - ▶ La durée de vie des éléments sont liés à la durée de vie de la classe maître

# Composition - Agrégation

- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association
- Une des classes joue un rôle plus important : la classe Maître
- Composition (losange noir) :
  - ▶ La durée de vie des éléments sont liés à la durée de vie de la classe maître
  - ▶ Cardinalité max côté losange noir toujours égale à 1

# Composition - Agrégation

- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association
- Une des classes joue un rôle plus important : la classe Maître
- Composition (losange noir) :
  - ▶ La durée de vie des éléments sont liés à la durée de vie de la classe maître
  - ▶ Cardinalité max côté losange noir toujours égale à 1
- Représentation graphique UML :



# Les ateliers UML

- Les produits commerciaux :

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)
  - ▶ Visio (éditeur graphique)

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)
  - ▶ Visio (éditeur graphique)
  - ▶ GenMyModel (<http://www.genmymodel.com/>)

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)
  - ▶ Visio (éditeur graphique)
  - ▶ GenMyModel (<http://www.genmymodel.com/>)
- Les produits libres :

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)
  - ▶ Visio (éditeur graphique)
  - ▶ GenMyModel (<http://www.genmymodel.com/>)
- Les produits libres :
  - ▶ Argo UML (Java, Poseidon)

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)
  - ▶ Visio (éditeur graphique)
  - ▶ GenMyModel (<http://www.genmymodel.com/>)
- Les produits libres :
  - ▶ Argo UML (Java, Poseidon)
  - ▶ TCM (éditeur graphique)

# Les ateliers UML

- Les produits commerciaux :
  - ▶ Visual paradigm
  - ▶ Modelio (Softeam)
  - ▶ Visio (éditeur graphique)
  - ▶ GenMyModel (<http://www.genmymodel.com/>)
- Les produits libres :
  - ▶ Argo UML (Java, Poseidon)
  - ▶ TCM (éditeur graphique)
- et bien d'autres...

# Particularités des ateliers UML professionnels

- Partage du modèle par plusieurs utilisateurs

# Particularités des ateliers UML professionnels

- Partage du modèle par plusieurs utilisateurs
- Distinction représentation interne et représentation graphique :

# Particularités des ateliers UML professionnels

- Partage du modèle par plusieurs utilisateurs
- Distinction représentation interne et représentation graphique :
  - ▶ Autorise plusieurs diagrammes pour des mêmes données

# Particularités des ateliers UML professionnels

- Partage du modèle par plusieurs utilisateurs
- Distinction représentation interne et représentation graphique :
  - ▶ Autorise plusieurs diagrammes pour des mêmes données
  - ▶ Possibilité de masquer ou pas des éléments

# Particularités des ateliers UML professionnels

- Partage du modèle par plusieurs utilisateurs
- Distinction représentation interne et représentation graphique :
  - ▶ Autorise plusieurs diagrammes pour des mêmes données
  - ▶ Possibilité de masquer ou pas des éléments
  - ▶ Extensible (plugins, modules, etc)

# Conclusion

- UML : un langage simple et intuitif de description de systèmes.

# Conclusion

- UML : un langage simple et intuitif de description de systèmes.
- UML massivement utilisé.

# Conclusion

- UML : un langage simple et intuitif de description de systèmes.
- UML massivement utilisé.
- Mais...

# Conclusion

- UML : un langage simple et intuitif de description de systèmes.
- UML massivement utilisé.
- Mais...
  - ▶ Besoins d'apports sémantiques pour aller plus loin (génération de code)

# Conclusion

- UML : un langage simple et intuitif de description de systèmes.
- UML massivement utilisé.
- Mais...
  - ▶ Besoins d'apports sémantiques pour aller plus loin (génération de code)
  - ▶ Le diagramme de classes est insuffisant pour décrire complètement un système.

# Conclusion

- UML : un langage simple et intuitif de description de systèmes.
- UML massivement utilisé.
- Mais...
  - ▶ Besoins d'apports sémantiques pour aller plus loin (génération de code)
  - ▶ Le diagramme de classes est insuffisant pour décrire complètement un système.
  - ▶ Il existe une dizaine de types de diagrammes UML (objets, états, collaborations, séquences, ...)