Module de spécialité IS

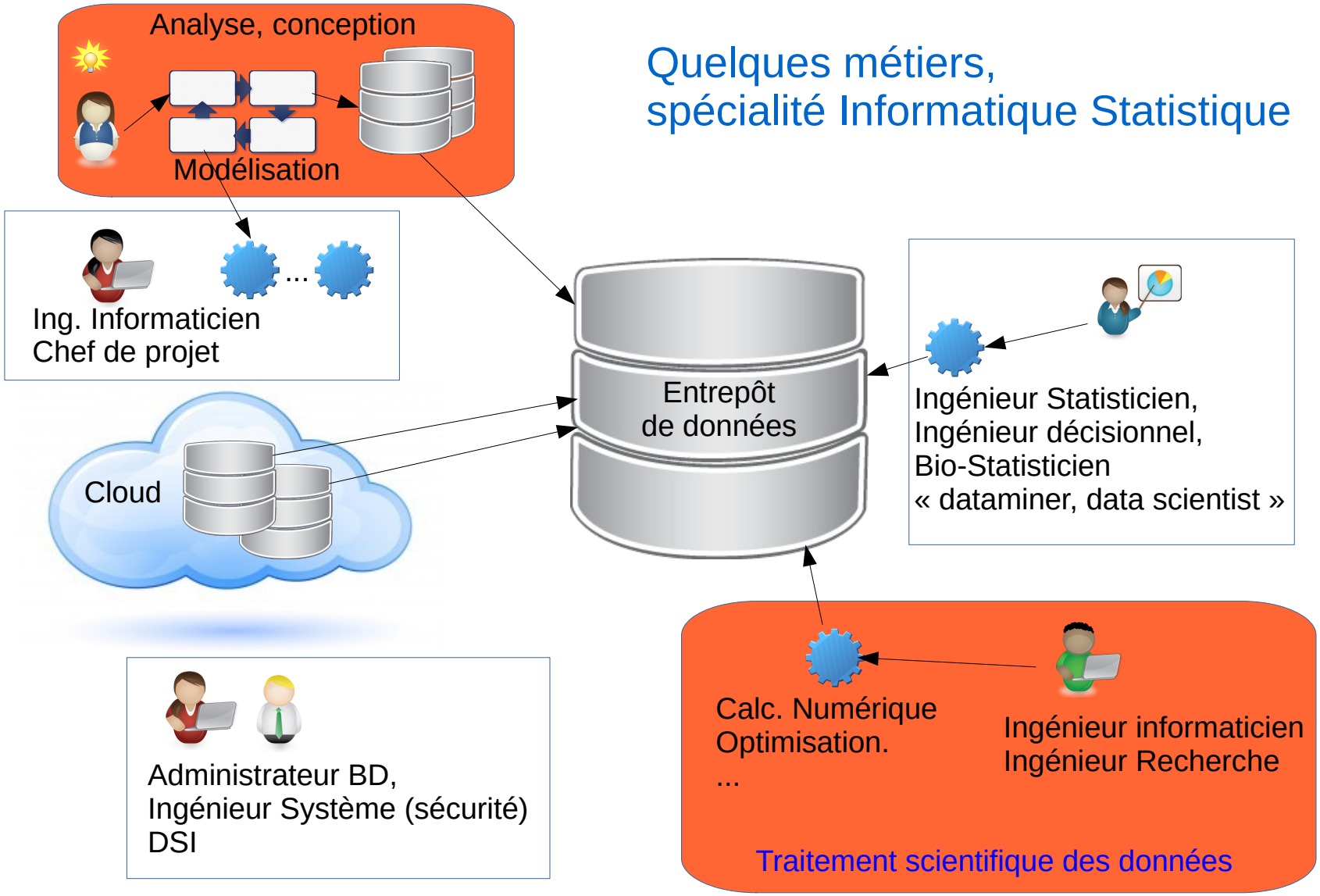
PEIP 2ème année

Olivier Caron, Clarisse Dhaenens, Serguei Dachian
http://ocaron.polytech-lille.net

École d'ingénieurs Polytech Lille
Université de Lille

3 septembre 2024

Le module de spécialité et les métiers IS



Toujours plus de données

- De plus en plus de données à disposition :

Toujours plus de données

- De plus en plus de données à disposition :
  - Au sein de l'entreprise :

Toujours plus de données

- De plus en plus de données à disposition :
  - Au sein de l'entreprise :
    - ★ Issue des bases de données des différents services

Toujours plus de données

- De plus en plus de données à disposition :
  - Au sein de l'entreprise :
    - ★ Issue des bases de données des différents services
    - ★ Activités des clients : opérations, logs webs, etc

Toujours plus de données

- De plus en plus de données à disposition :
  - Au sein de l'entreprise :
    - ★ Issue des bases de données des différents services
    - ★ Activités des clients : opérations, logs webs, etc
  - En dehors de l'entreprise :

Toujours plus de données

- De plus en plus de données à disposition :
  - Au sein de l'entreprise :
    - ★ Issue des bases de données des différents services
    - ★ Activités des clients : opérations, logs webs, etc
  - En dehors de l'entreprise :
    - ★ Des bases de données accessibles (Open Data)
      Exemples : OpenStreetmap, OpenWeathermap, Wikipedia, Twitter, Facebook...

Plusieurs types de données

- Des données structurées :

Plusieurs types de données

- Des données structurées :
  - Issues de bases de données (relationnel, XML, ...)

Plusieurs types de données

- Des données structurées :
  - Issues de bases de données (relationnel, XML, ...)
  - Issues de fichiers structurés (logs webs, ...)

Plusieurs types de données

- Des données structurées :
  - Issues de bases de données (relationnel, XML, ...)
  - Issues de fichiers structurés (logs webs, ...)
- Des données non structurées :

Plusieurs types de données

- Des données structurées :
  - Issues de bases de données (relationnel, XML, ...)
  - Issues de fichiers structurés (logs webs, ...)
- Des données non structurées :
  - Des pages webs
    solution $\longrightarrow$ web sémantique,
    Illustration : rechercher dans google "épouse du président américain"

Plusieurs types de données

- Des données structurées :
  - Issues de bases de données (relationnel, XML, ...)
  - Issues de fichiers structurés (logs webs, ...)
- Des données non structurées :
  - Des pages webs
    solution $\longrightarrow$ web sémantique,
    Illustration : rechercher dans google "épouse du président américain"
  - Du texte (ex : analyse des tweets, analyse des avis clients, ...)
    solution : $\longrightarrow$ text mining

Vers de nouvelles applications

- La convergence numérique est en route :

Vers de nouvelles applications

- La convergence numérique est en route :
  - Utilisation de plusieurs canaux : voix, internet, téléphonie

Vers de nouvelles applications

- La convergence numérique est en route :
  - Utilisation de plusieurs canaux : voix, internet, téléphonie
  - Utilisation de plusieurs équipements : smartphones, PC, objets connectés, capteurs

Vers de nouvelles applications

- La convergence numérique est en route :
  - Utilisation de plusieurs canaux : voix, internet, téléphonie
  - Utilisation de plusieurs équipements : smartphones, PC, objets connectés, capteurs
  - Utilisation de plusieurs sources de données

Exemple : analyse de données externes

- Analyse de la météo et impact sur les ventes d'une entreprise de bricolage

Exemple : analyse de données externes

- Analyse de la météo et impact sur les ventes d'une entreprise de bricolage
- Analyse des tweets sur la popularité d'un produit

Exemple : analyse de données externes

- Analyse de la météo et impact sur les ventes d'une entreprise de bricolage
- Analyse des tweets sur la popularité d'un produit
- Analyse des ventes d'un produit chez ses différents revendeurs

Exemple : analyse de données externes

- Analyse de la météo et impact sur les ventes d'une entreprise de bricolage
- Analyse des tweets sur la popularité d'un produit
- Analyse des ventes d'un produit chez ses différents revendeurs
- Analyser les pages webs :

Exemple : analyse de données externes

- Analyse de la météo et impact sur les ventes d'une entreprise de bricolage
- Analyse des tweets sur la popularité d'un produit
- Analyse des ventes d'un produit chez ses différents revendeurs
- Analyser les pages webs :
  - Algorithme "page rank" de Google (exploration aléatoire)

    ▸ Présentation vidéo de Cédric Villani en 2016 (08:41 → 09:53)

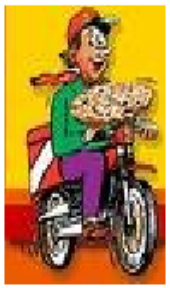
    `https://video-subtitle.tedcdn.com/talk/podcast/2016/None/CedricVillani_2016-480p-fr.mp4`

Exemple :Application "livraison optimisée de pizzas

« Speed Pizza »

Ou suis-je ?

Calcul du plus
court chemin



2 pizzas « royale »
12, rue de Maubeuge
1 pizza « Calzone »
rue des templiers

Au final, des besoins plus sensibles

- Masse d'informations à gérer :

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau
  - Sauvegarde (ex : les bunker)

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau
  - Sauvegarde (ex : les bunker)
  - Sécurité des données (ex : base Orange, réseaux sociaux)

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau
  - Sauvegarde (ex : les bunker)
  - Sécurité des données (ex : base Orange, réseaux sociaux)
  - Temps de réponse

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau
  - Sauvegarde (ex : les bunker)
  - Sécurité des données (ex : base Orange, réseaux sociaux)
  - Temps de réponse
- Nouveau type de données (ex : NoSQL)

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau
  - Sauvegarde (ex : les bunker)
  - Sécurité des données (ex : base Orange, réseaux sociaux)
  - Temps de réponse
- Nouveau type de données (ex : NoSQL)
- L'éthique des données et programmes (CNIL, etc)

Au final, des besoins plus sensibles

- Masse d'informations à gérer :
  - Stockage (datacenter, cloud)
  - Accès réseau
  - Sauvegarde (ex : les bunker)
  - Sécurité des données (ex : base Orange, réseaux sociaux)
  - Temps de réponse
- Nouveau type de données (ex : NoSQL)
- L'éthique des données et programmes (CNIL, etc)
  - Testez chez vous : `http://www.amiunique.org`

Revenons au point de départ

- **Point de départ** : concevoir puis développer une application logicielle ou une base de données

Revenons au point de départ

- **Point de départ** : concevoir puis développer une application logicielle ou une base de données
  - Applications de plus en plus complexe, nécessité de bien **décrire** la future application, les données, les interactions.

Revenons au point de départ

- **Point de départ** : concevoir puis développer une application logicielle ou une base de données
  - Applications de plus en plus complexe, nécessité de bien **décrire** la future application, les données, les interactions.
  - Permet de découvrir (parfois) si l'objectif est programmable.

Revenons au point de départ

- **Point de départ** : concevoir puis développer une application logicielle ou une base de données
  - Applications de plus en plus complexe, nécessité de bien **décrire** la future application, les données, les interactions.
  - Permet de découvrir (parfois) si l'objectif est programmable.
  - Permet de lever toutes les ambiguïtés

Revenons au point de départ

- **Point de départ** : concevoir puis développer une application logicielle ou une base de données
  - Applications de plus en plus complexe, nécessité de bien **décrire** la future application, les données, les interactions.
  - Permet de découvrir (parfois) si l'objectif est programmable.
  - Permet de lever toutes les ambiguïtés
  - Diminue les risques d'erreur

Revenons au point de départ

- **Point de départ** : concevoir puis développer une application logicielle ou une base de données
  - Applications de plus en plus complexe, nécessité de bien **décrire** la future application, les données, les interactions.
  - Permet de découvrir (parfois) si l'objectif est programmable.
  - Permet de lever toutes les ambiguïtés
  - Diminue les risques d'erreur
- La solution : **La modélisation du logiciel**

Besoin de spécification des projets informatiques

- Pour 10 projets informatiques,__ sont des succès

Besoin de spécification des projets informatiques

- Pour 10 projets informatiques,__ sont des succès
- Fiabilité des projets, spécification des règles de gestion, tests

Besoin de spécification des projets informatiques

- Pour 10 projets informatiques,__ sont des succès
- Fiabilité des projets, spécification des règles de gestion, tests

- Vol 501 d'ariane 5, bug informatique : débordement de capacité.

Besoin de spécification des projets informatiques

- Pour 10 projets informatiques,__ sont des succès
- Fiabilité des projets, spécification des règles de gestion, tests

- Vol 501 d'ariane 5, bug informatique : débordement de capacité.
- Réalisées après la catastrophe, les simulations en laboratoire ont permis de vérifier que l'explosion était inéluctable.

La modélisation UML en quelques points

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

La modélisation 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`

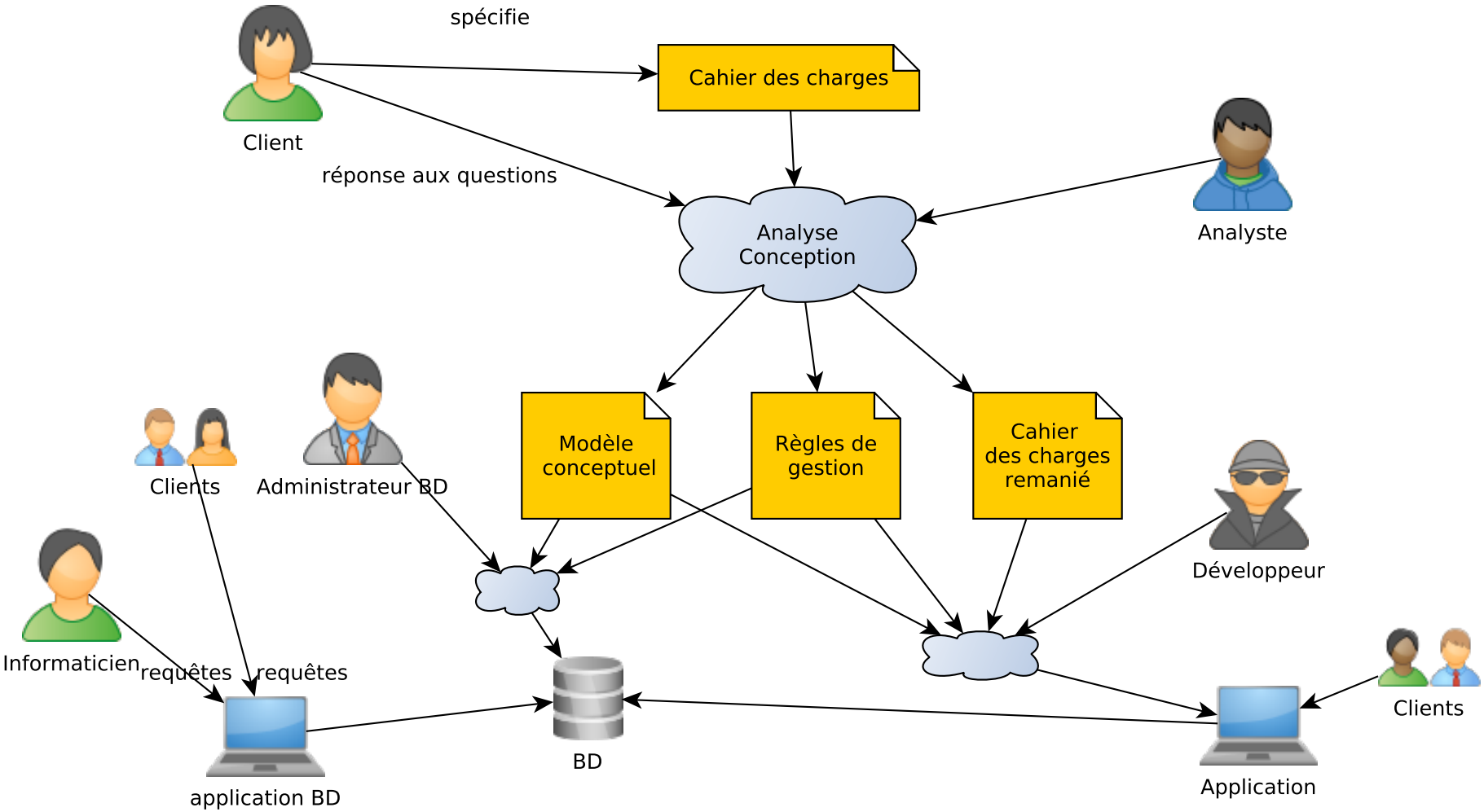
La modélisation UML en quelques points

- 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 objet** (ou cadre de modélisation), pas une méthodologie

La modélisation 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 objet** (ou cadre de modélisation), pas une méthodologie
- Indépendante de toute méthodologie, utilisables pour des bases de données, des logiciels

De la modélisation à un logiciel (programme IS)



F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*
  - ne permet pas de prouver si l'objectif est réalisable.

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*
  - ne permet pas de prouver si l'objectif est réalisable.
    - ★ Exemple : *concevoir un programme qui à partir du code d'un programme quelconque indique si ce dernier va s'arrêter ou pas.*

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*
  - ne permet pas de prouver si l'objectif est réalisable.
    - ★ Exemple : *concevoir un programme qui à partir du code d'un programme quelconque indique si ce dernier va s'arrêter ou pas.*
    - ★ **problème impossible** (Alan Turing, 1936)

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*
  - ne permet pas de prouver si l'objectif est réalisable.
    - ★ Exemple : *concevoir un programme qui à partir du code d'un programme quelconque indique si ce dernier va s'arrêter ou pas.*
    - ★ **problème impossible** (Alan Turing, 1936)
  - ne permet pas d'estimer les temps de réponse (ex :calcul météo)

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*
  - ne permet pas de prouver si l'objectif est réalisable.
    - ★ Exemple : *concevoir un programme qui à partir du code d'un programme quelconque indique si ce dernier va s'arrêter ou pas.*
    - ★ **problème impossible** (Alan Turing, 1936)
  - ne permet pas d'estimer les temps de réponse (ex :calcul météo)
- A quoi sert donc UML ?

  $\Rightarrow$ à décrire, spécifier un système complexe, sous plusieurs angles afin de mieux le comprendre

  $\Rightarrow$ à avoir un formalisme moins ambigu que le langage naturel

  $\Rightarrow$ disposer d'un format d'échange, de travail collaboratif

F.A.Q. UML

- UML sert-il à concevoir un système sans erreurs ?
  - $\Rightarrow$ ben, non... *(mais peut-être avec moins d'erreurs)*
  - ne permet pas de prouver si l'objectif est réalisable.
    - ★ Exemple : *concevoir un programme qui à partir du code d'un programme quelconque indique si ce dernier va s'arrêter ou pas.*
    - ★ **problème impossible** (Alan Turing, 1936)
  - ne permet pas d'estimer les temps de réponse (ex :calcul météo)
- A quoi sert donc UML ?

  $\Rightarrow$ à décrire, spécifier un système complexe, sous plusieurs angles afin de mieux le comprendre

  $\Rightarrow$ à avoir un formalisme moins ambigu que le langage naturel

  $\Rightarrow$ disposer d'un format d'échange, de travail collaboratif
- Quelles sont les tendances d'UML ?

  $\Rightarrow$ Opérationnaliser UML : outil de construction de modèles, outil de contrôle de modèles, outil de génération de code, ...

Modéliser des données : le diagramme de classes

- Issu des travaux de Chen (US) et travaux européens

Modéliser des données : le diagramme de classes

- Issu des travaux de Chen (US) et travaux européens
- Objectifs :

Modéliser des données : le diagramme de classes

- Issu des travaux de Chen (US) et travaux européens
- Objectifs :
  - Puissance de représentation

Modéliser des données : le diagramme de classes

- Issu des travaux de Chen (US) et travaux européens
- Objectifs :
  - Puissance de représentation
  - Stabilité et flexibilité : un **ajout** de donnée ne doit pas remettre en cause le schéma.

Modéliser des données : le diagramme de classes

- Issu des travaux de Chen (US) et travaux européens
- 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

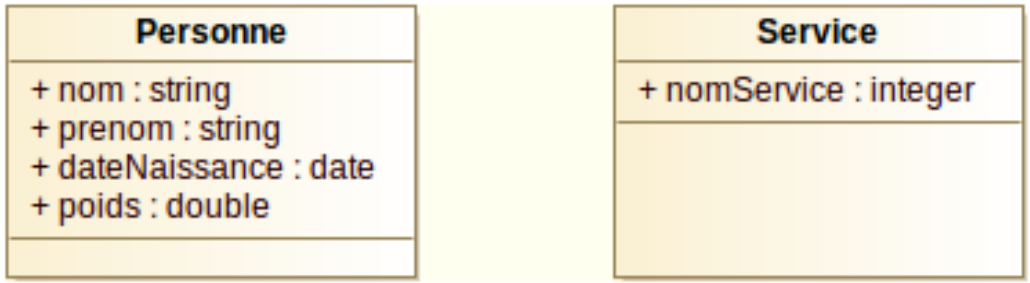
Modéliser des données : le diagramme de classes

- Issu des travaux de Chen (US) et travaux européens
- 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.



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.
  non obligatoire : UML sert à la conception.

Les attributs ou propriétés d'une classe

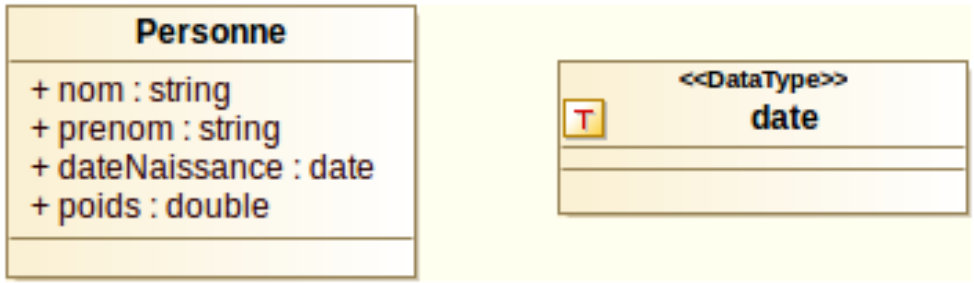
- Liste d'éléments qui caractérisent une classe
- Les propriétés/attributs peuvent être typé(e)s.
  non obligatoire : UML sert à la conception.
- UML dispose de types primitifs (string, real, boolean, int).

Les attributs ou propriétés d'une classe

- Liste d'éléments qui caractérisent une classe
- Les propriétés/attributs peuvent être typé(e)s.
  non obligatoire : UML sert à la conception.
- UML dispose de types primitifs (string, real, boolean, int).
- 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(e)s.
  non obligatoire : UML sert à la conception.
- UML dispose de types primitifs (string, real, boolean, int).
- 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).

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).
- La mise en relation d'entités peut faire apparaître des propriétés qui n'appartiennent en propre à aucune des entités. On distingue :

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).
- 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)

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).
- 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-aire relient les instances de n-classes d'entité.

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).
- 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-aire 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

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.
- définies 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.
- définies au niveau de chaque extrémité de l'association
- Conventions d'affichage :

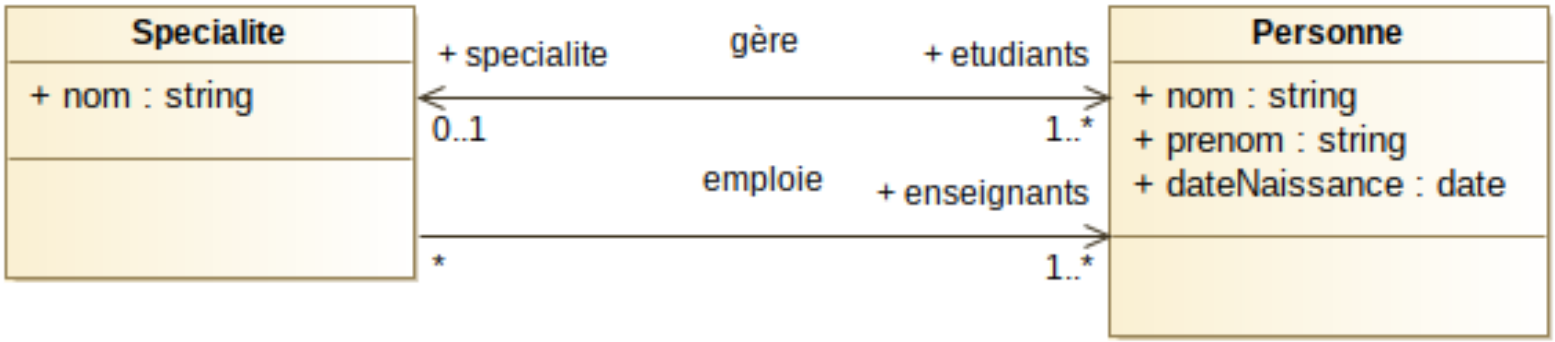
représentation	**signification**
1	un et un seul
0..1	zéro et un
M..N	de M à N
*	de zéro à plusieurs
0..*	de zéro à plusieurs
1..*	de un à plusieurs

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

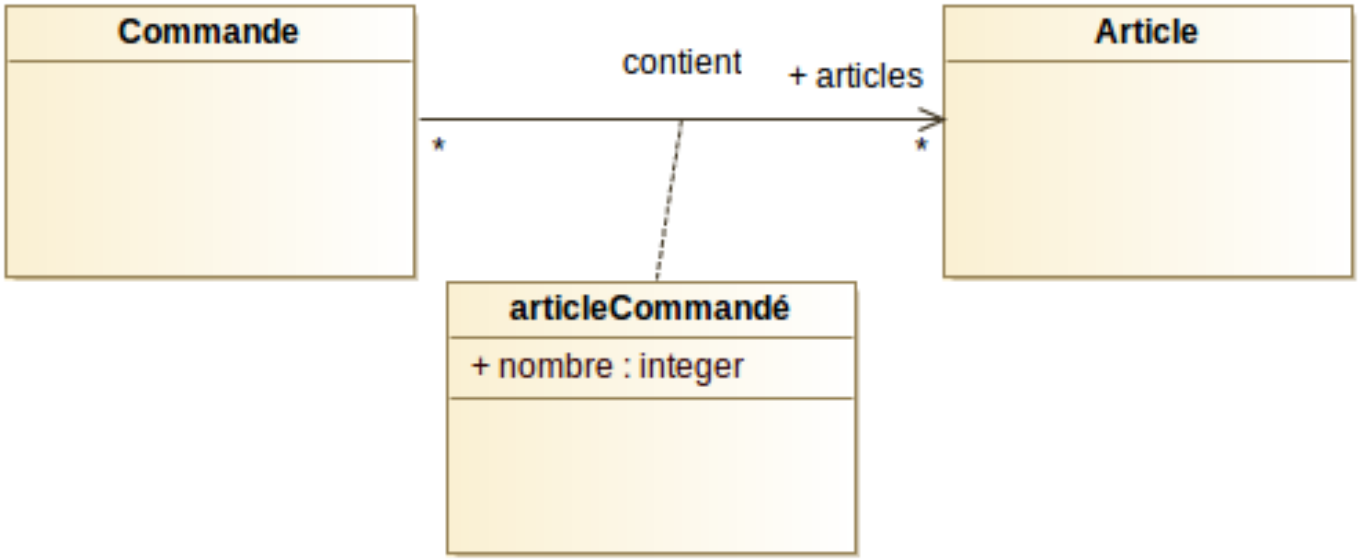
- L'attribut n'a de sens que pour l'association, ne dépend donc pas d'une seule classe.

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

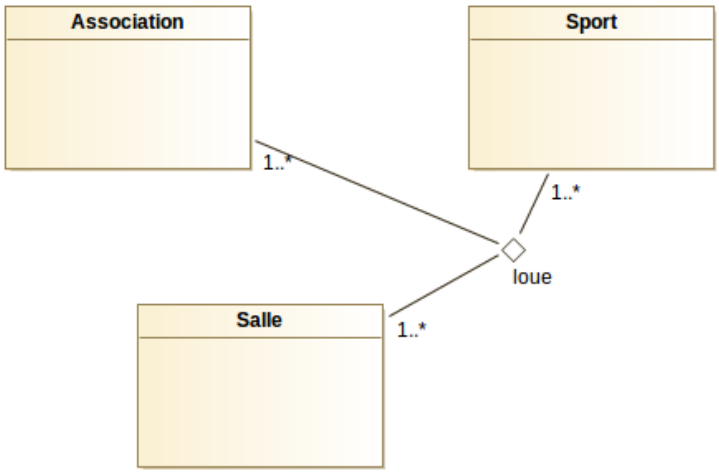
Les propriétés (attributs) d'une association

- 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 :



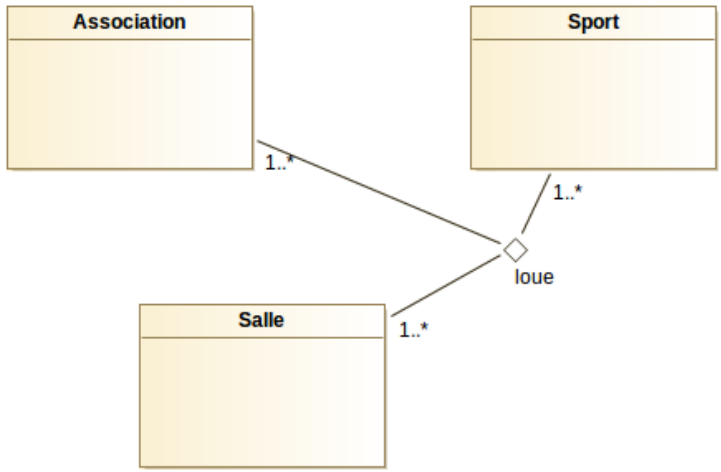
Association n-aire

- Représentation graphique UML :



Association n-aire

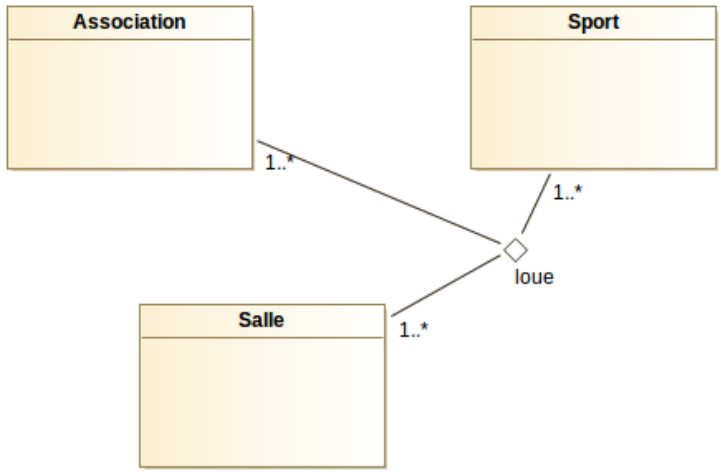
- Représentation graphique UML :



- 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

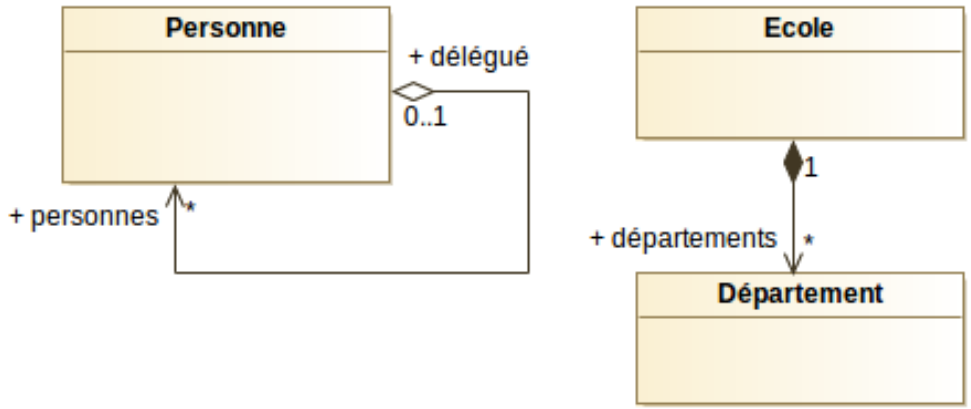
- Composition (losange noir) et Agrégation (losange blanc) : formes particulière d'une association
- Une des classes joue un rôle plus important : la classe Maître
- Composition (losange noir) :
  - La durée de vie des éléments sont liés à la durée de vie de la classe maître

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

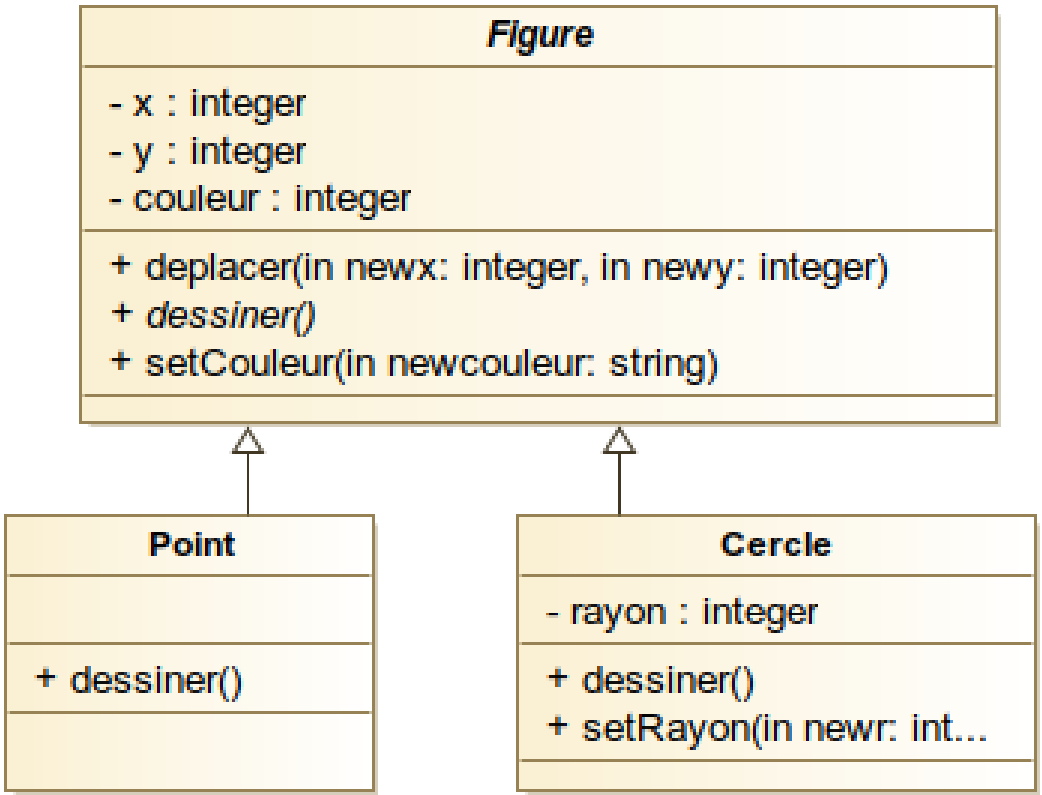
Composition - Agrégation

- 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 :

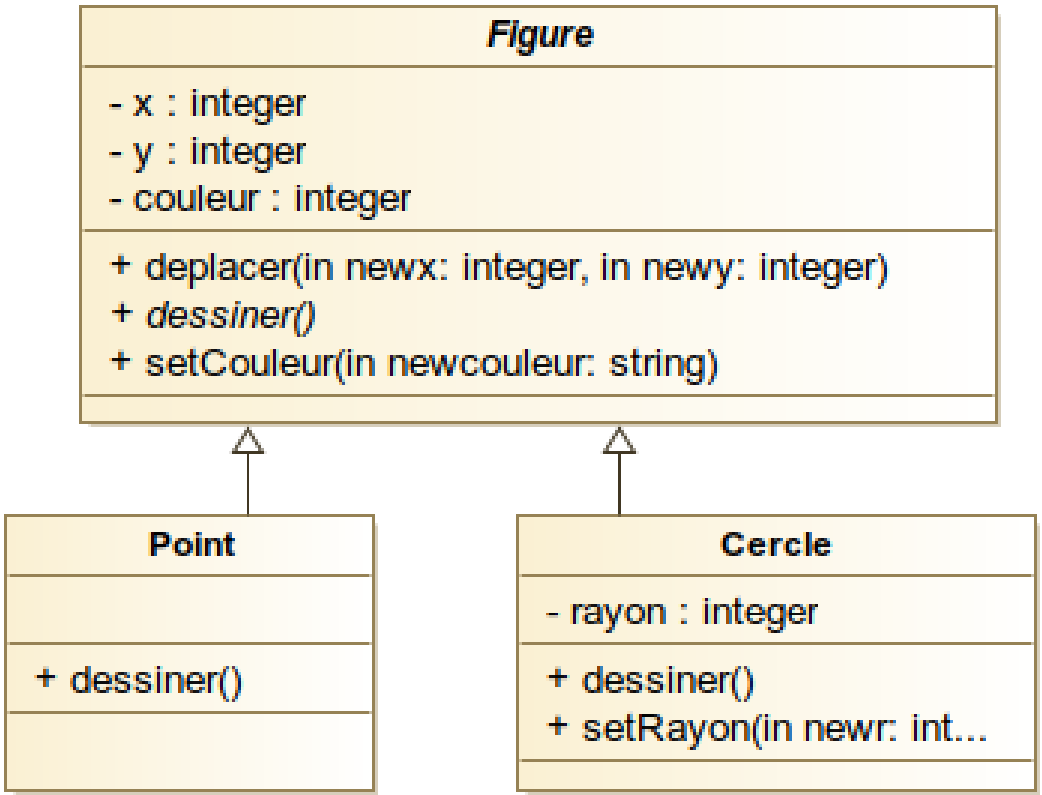


La dimension objet d'UML, cas de l'héritage

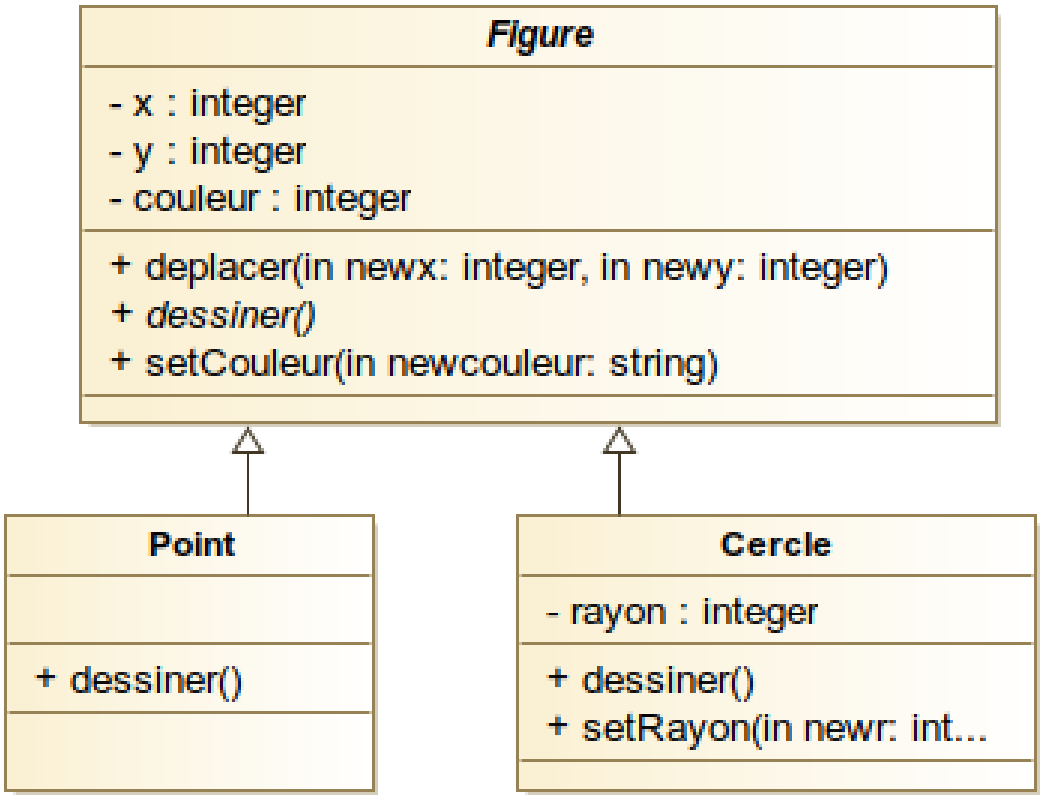
La dimension objet d'UML, cas de l'héritage



La dimension objet d'UML, cas de l'héritage

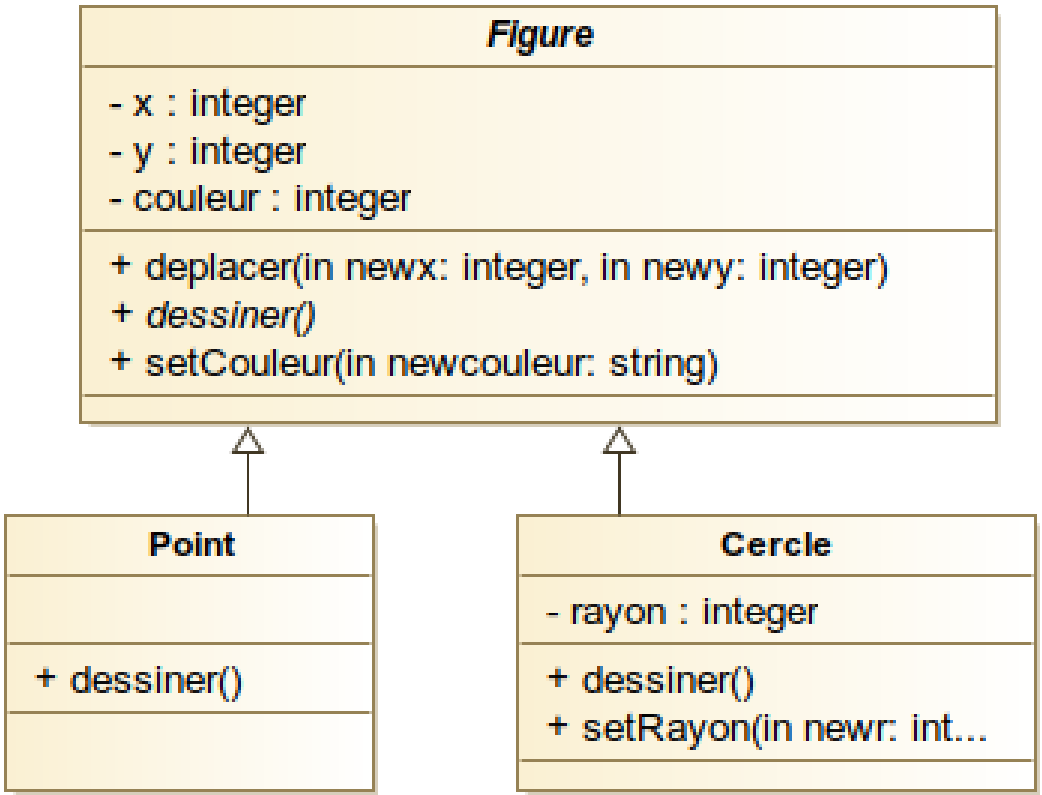


La dimension objet d'UML, cas de l'héritage



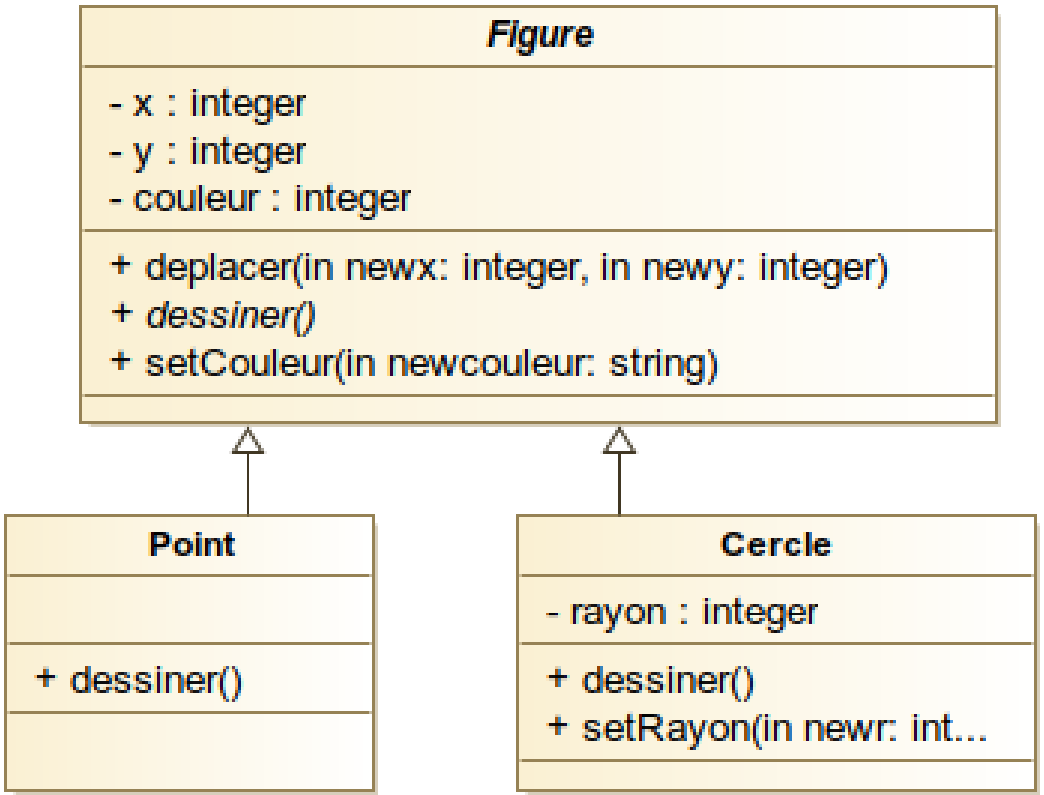
- Méthodes/opérations associées aux classes

La dimension objet d'UML, cas de l'héritage



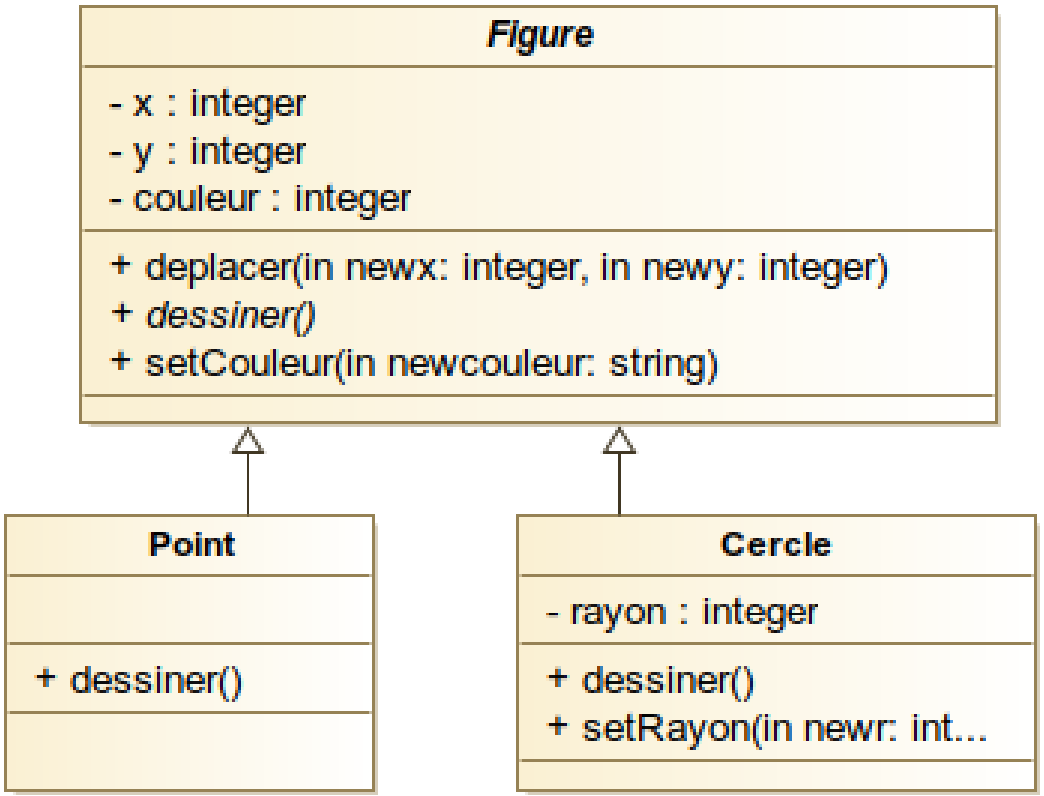
- Méthodes/opérations associées aux classes
- Notion de droits d'accès (public/privé)

La dimension objet d'UML, cas de l'héritage



- Méthodes/opérations associées aux classes
- Notion de droits d'accès (public/privé)
- Héritage de classes

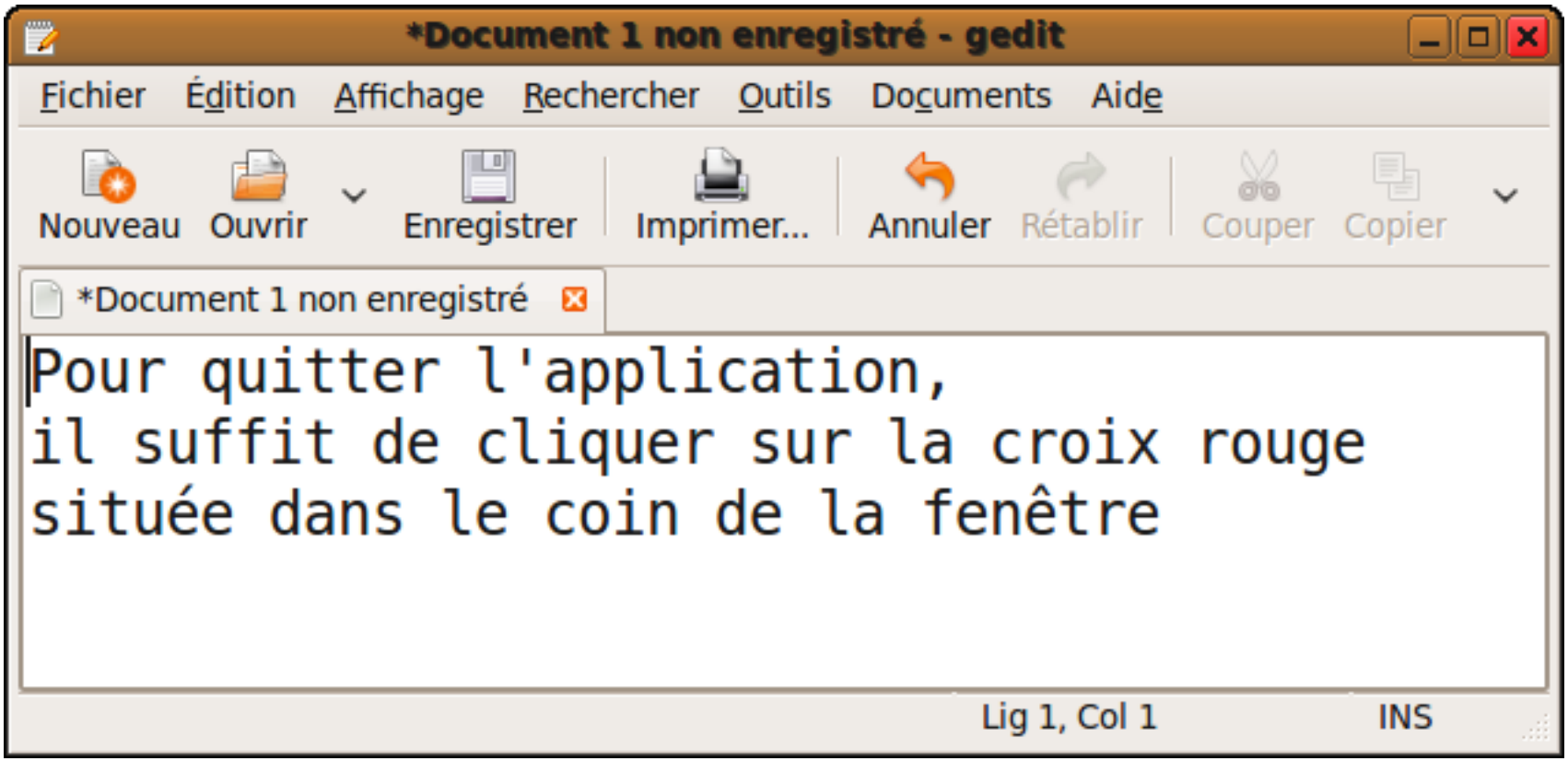
La dimension objet d'UML, cas de l'héritage



- Méthodes/opérations associées aux classes
- Notion de droits d'accès (public/privé)
- Héritage de classes
- Classes et méthodes abstraites ou concrètes (en italique)

Conclusion

- UML, un outil pour se faire comprendre avec le moins d'incohérences.



Problème classique - Gestion d'une bibliothèque

Soit le cahier des charges suivants *volontairement flou* pour gérer une bibliothèque.
Un livre est caractérisé par son titre, son auteur, ses éditeurs. Pour chaque livre édité, on veut connaître la date d'édition.
Un auteur est caractérisé par son nom, prénom.
Un éditeur est caractérisé par son nom et son adresse.
Un utilisateur est caractérisé par son nom, prénom et age.
Le but de cette gestion est de gérer les emprunts de livres (maximum 3 livres par utilisateur) et la possibilité de réserver des livres.

$\Longrightarrow$ Proposez un schéma conceptuel