Analyse Conception UML

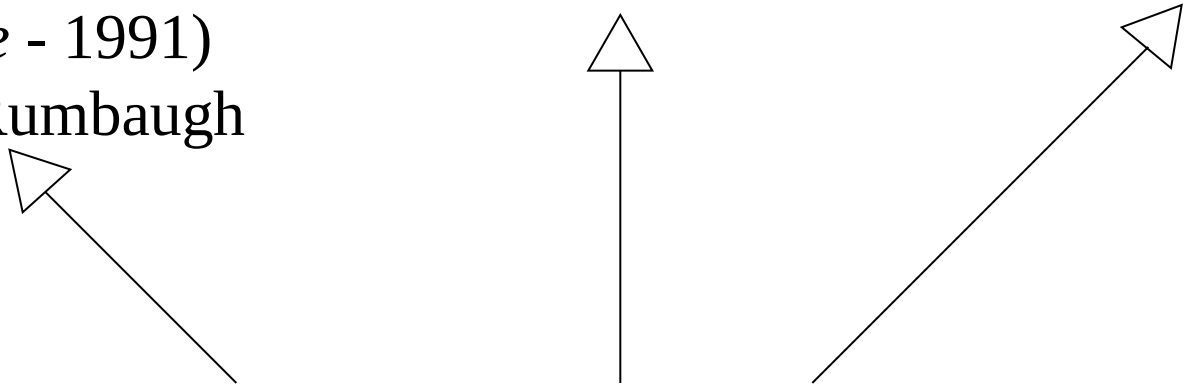
- Vocabulaire, méthodologie et thèmes de l'orienté objet
- Modèle de classes
- Modèle d'états
- Modèle d'interactions

Historique

OMT
(*Object Modeling Technique* - 1991)
de James Rumbaugh

OOSE
de Ivar Jacobson

Notation
de Grady Booch



UML - *Unified Modeling Language*

Standard de modélisation objet
adopté en 1997 par l'*Object Management Group* (OMG)

- Révision des spécifications initiales en 2001 – UML 1
- Approbation de la version **UML 2.0** en 2004
- Depuis 2017 : **UML 2.5.1**

  (cf. `https://www.omg.org/spec/UML/About-UML/`)

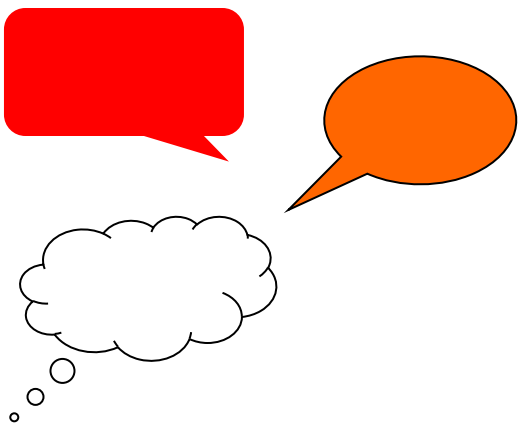
Vocabulaire orienté objet (2/3)

- **Classification** : regroupement des objets ayant même structure de données (attributs) et même comportement (opérations)

  Ex. *TransparentDeCours; Enseignant*

- **Classe** : abstraction décrivant un ensemble d'objets potentiellement infini [BR05]

Objets Bulles et Légendes



Abstraction →

Classe *BulleEtLégende*

Attributs

- forme
- couleurDeTrait
- couleurDeRemplissage

Opérations

- dessiner
- effacer
- déplacer
- modifierTaille

Vocabulaire orienté objet (3/3)

- **Instance d'une classe** : objet appartenant à la classe
- **Héritage** : partage de **propriétés** entre classes sur la base d'une relation hiérarchique
  - **Super-classe** (classe mère)
  - **Sous-classe** (classe fille) : spécialisation de la super-classe
    Héritage des propriétés de la super-classe
- **Polymorphisme** : possibilité de comportements différents d'une même opération dans différentes classes
  - **Opération** : action exécutée par un objet ou transformation subie par un objet
  - **Méthode** : implémentation d'une opération par une classe

    Plusieurs méthodes pour une même opération
    Une méthode par classe pour une opération donnée

Méthodologie orientée objet

- **Spécification initiale** : collaboration entre les analystes métier et les utilisateurs pour la genèse de l'application
- **Analyse**
  - Étude et re-formulation des besoins : collaboration avec le client pour comprendre le problème
  - **Modèle d'analyse** : abstraction concise et précise de l'objectif du système à développer (pas de la façon de le construire)
    - **Modèle de domaine** : description des objets du monde réel manipulés par le système
    - **Modèle de l'application** : parties du système visibles par l'utilisateur
- **Conception** : stratégie de haut niveau (architecture du système) pour résoudre le problème posé
  - Établissement des stratégies de conception générales
  - Allocations prévisionnelles des ressources
- **Conception des classes** : concentration sur les structures de données et algorithmes de chaque classe
- **Implémentation**
- **Test**

Thèmes de l'orienté objet

- **Abstraction**
  Concentration sur ce que représente un objet et sur son comportement avant de décider de la façon de l'implémenter
- **Encapsulation**
  Masquage de l'information : séparation des aspects externes d'un objet accessibles aux autres objets, des détails de l'implémentation cachés aux autres objets
- **Regroupement des données et du comportement**
  Polymorphisme ⇒ transfert de la décision de quelle méthode utiliser à la hiérarchie de classes
- **Partage**
  Héritage ⇒ possibilité de partager des portions de code communes et clarté conceptuelle (mise en évidence de traitement commun)
- **Mise en évidence de la nature intrinsèque des objets** : sur *ce qu'est* un objet et non sur la façon dont il est *utilisé*

Trois modèles (1/2)

- **Modèle de classes**
  - Description de la structure statique des objets du système et de leurs relations
  - **Diagramme de classes** : graphe avec pour sommets les classes et pour arcs les relations entre les classes
- **Modèle d'états**
  - Description des états successifs d'un objet au cours du temps
  - **Diagramme d'états** : graphe avec pour sommets les états et pour arcs les transitions entre états déclenchées par des événements
- **Modèles d'interactions**
  - Description de la manière de coopérer des objets pour obtenir un résultat
  - **Cas d'utilisation** axé sur une fonctionnalité
  - **Diagramme de séquence** : représentation des interactions entre objets et ordonnancement des interactions
  - **Diagramme d'activités** : détails des étapes importantes du traitement

Trois modèles (2/2)

- Parties distinctes de la description du système mais interdépendantes
- Le plus fondamental [BR05] : **le modèle de classes**

> « Il est nécessaire de décrire *ce qui* change ou se transforme avant de décrire *quand* et *comment* les changements ont lieu »

Modélisation orientée objet

- **Modèle** : abstraction pour comprendre un problème avant de mettre en œuvre une solution [BR05]
  - Tester une entité physique avant de la construire
  - Communiquer avec les clients
  - Visualiser
  - Réduire la complexité

  Deux dimensions associées à un modèle :
  1. Une vue d'un système (modèle de classes, d'états ou d'interactions)
  2. Un stade de développement (analyse, conception ou implémentation)
- **Trois modèles en UML**
  - **Modèle de classes** : aspects orientés "données" du système
  - **Modèle d'états** : aspects temporels, comportementaux et de "contrôle " du système
  - **Modèle d'interactions** : collaboration entre objets

  Un seul aspect du système traité par chaque modèle, mais relations entre les trois modèles

  Variation du poids des modèles en fonction du problème posé

Modèle de classes (1/20)

- Description de la structure statique d'un système
- Représentation graphique intuitive d'un système [BR05]
- Vocabulaire :
  - **Objets**
  - **Classes**
  - **Associations**
  - **Liens**
  - **Généralisation**
  - **Héritage**

Modèle de classes (2/20) - Objet

- Concept, abstraction ou entité ayant une signification pour une application [BR05]

- Avec une contrepartie dans le monde réel, ou correspondant à une entité conceptuelle ou introduit pour les besoins de l'implémentation

  Ex. *Maude Manouvrier; la formule pour calculer la moyenne d'un module; le pointeur désigné par la variable p*

- Identifié et distinguable des autres objets

Modèle de classes (3/20) - Classe

- Description d'un groupe d'objets possédant les mêmes propriétés (**attributs**) le même comportement (**opérations**), les mêmes relations et la même sémantique

Classe Cours
Attributs
intitulé
nombreHeures
Opérations
planifier

Classe Enseignant
Attributs
nom
prénom
dateDeNaissance
Opérations
affecterEnseignement

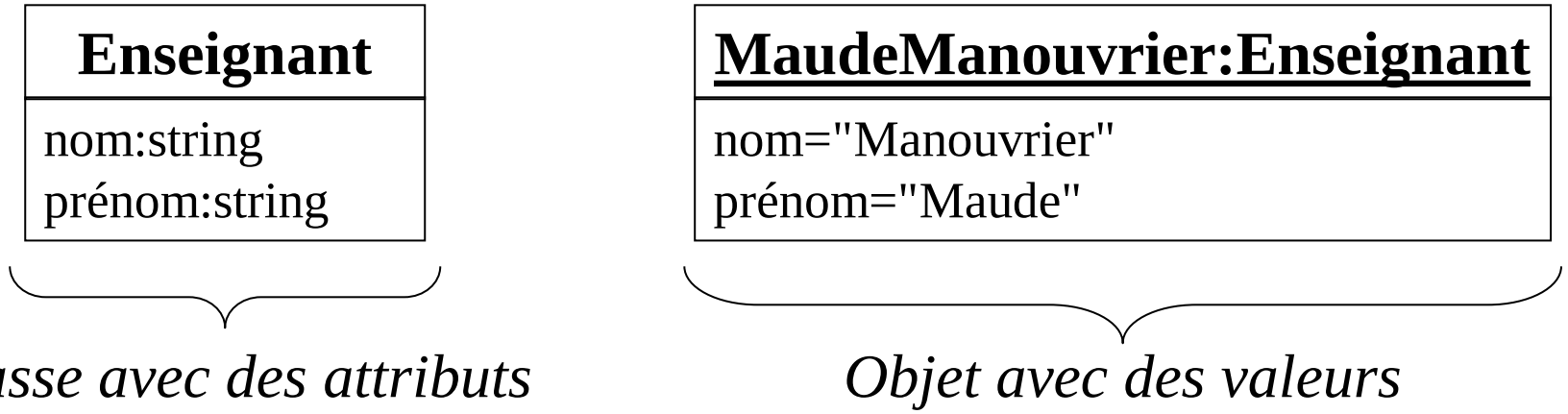
- Objet : **instance** de classe

Modèle de classes (4/20)

- **Diagrammes de classes** : Notation graphique permettant la modélisation des classes et de leurs relations
- **Diagrammes d'objets** : Représentation des objets individuels et de leurs relations

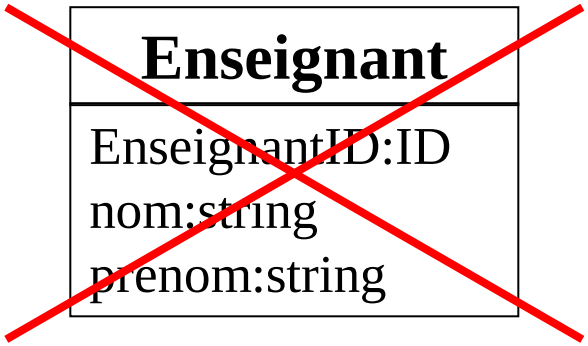
Modèle de classes (5/20)

- **Valeur** : donnée sans identité
- **Attribut** : propriété nommée d'une classe décrivant le type d'une valeur contenue dans chaque objet de la classe
- « *Un objet est à une classe ce qu'une valeur est à un attribut* » [BR05]



Modèle de classes (6/20)

- **Identifiant** : implicite
- Ne pas confondre identifiant interne et attribut d'identification ayant une contrepartie dans le monde réel



Enseignant
nom:string prénom:string

Enseignant
NUMEN:integer nom:string prénom:string

Modèle de classes (7/20)

- **Opération** : fonction ou procédure pouvant être appliquée aux objets ou par les objets d'une classe
- **Méthode :** implémentation d'une opération pour une classe donnée

Enseignant
nom:string prénom:string
affecterEnseignement (e:Enseignement) nombreHeuresEnseignement: integer

Fichier
nom:string localisation:string
imprimer

FichierPowerPoint
nombreTransparents:integer
imprimer

Modèle de classes (8/20)

- **Propriété** : terme générique pour attribut et opération
- **Notation des classes**

NomDeClasse
nomAttribut1 : typeDeDonnées1 = Valeur parDéfaut1 nomAttribut2 : typeDeDonnées2 = Valeur parDéfaut2 …
nomOpération1 (listeArguments1) : TypeDuRésultat1 nomOpération2 (listeArguments2) : TypeDuRésultat2 …

- **Sens de flux** (*direction*) : indication si un argument est en entrée non modifiable (*in*), une sortie (*out*) ou une entrée modifiable (*inout*)

  [*SensDeFlux*] ***nomArgument*** [: *type* = *valeurParDéfaut*]

Modèle de classes (9/20)

Liens et associations

- **Lien** : connexion physique ou conceptuelle entre objets
  Ex. *MaudeManouvrier Enseigne la MiseANiveauUML*
- **Association** : description d'un groupe de liens qui partagent une structure et une sémantique commune
  Ex. *un Enseignant Enseigne un Cours*

Diagramme de classes :

Diagramme d'objets :

Modèle de classes (10/20) - Multiplicité

Nombre d'instances d'une classe pouvant être liées à une instance d'une autre classe / contrainte sur le nombre d'objets liés

- « un-à-un »

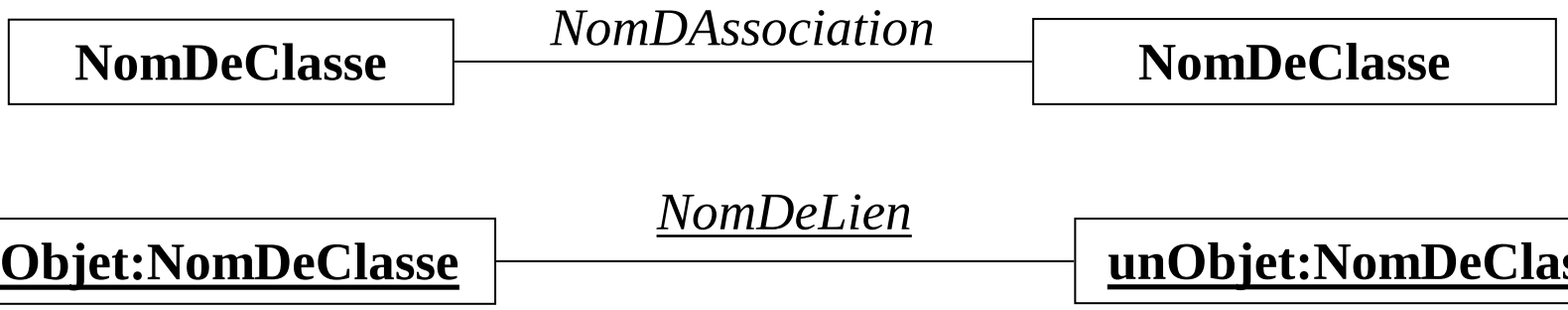
- « zéro-à-un »

- « plusieurs-à-plusieurs »

Modèle de classes (11/20)

Liens et associations

- **Notation**



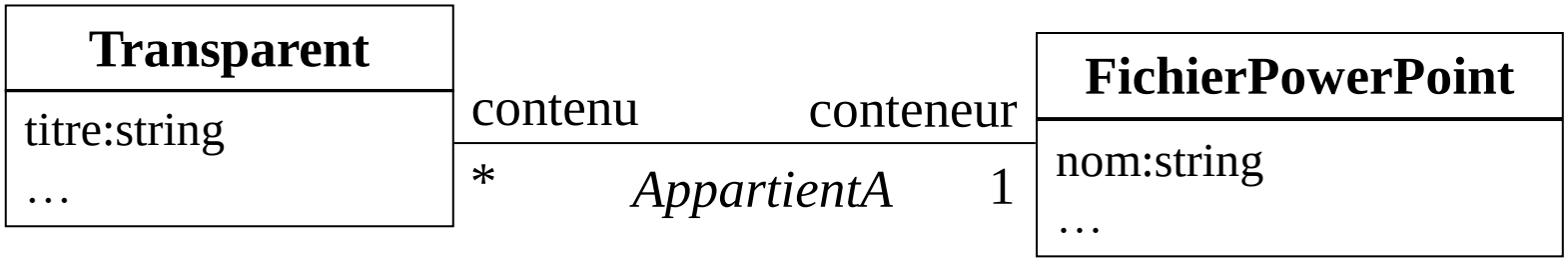
- **Implémentation des associations par référence**

  Ex. Implémentation de l'association Professe par un attribut Enseignements dans la classe Enseignant et/ou un attribut Enseignants dans la classe Enseignement

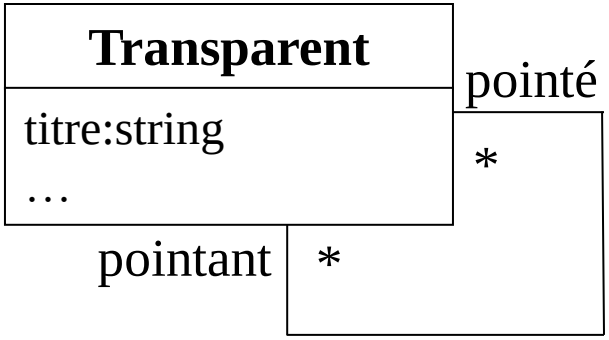
Ne pas confondre implémentation et modélisation

Modèle de classes (12/20) – Noms d'extrémité

- **Possibilité de nommer les extrémités d'association**



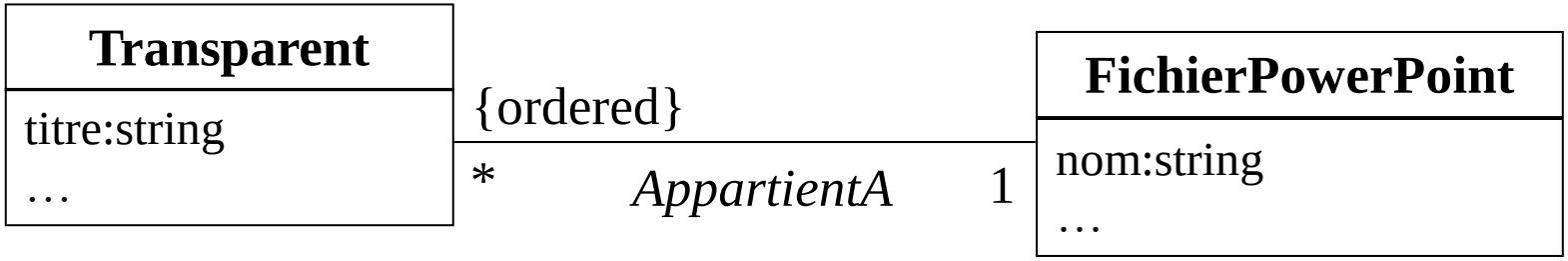
- **Indispensable pour les associations entre objets de même classe**



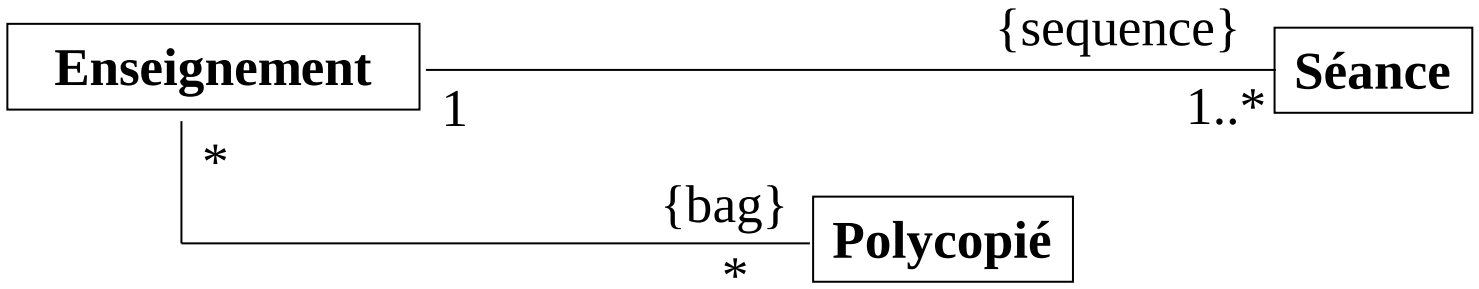
Modèle de classes (13/20)

Ordonnancement, *bags* et séquences

- **Ordonnancement des objets situés à l'extrémité d'une association « plusieurs »**

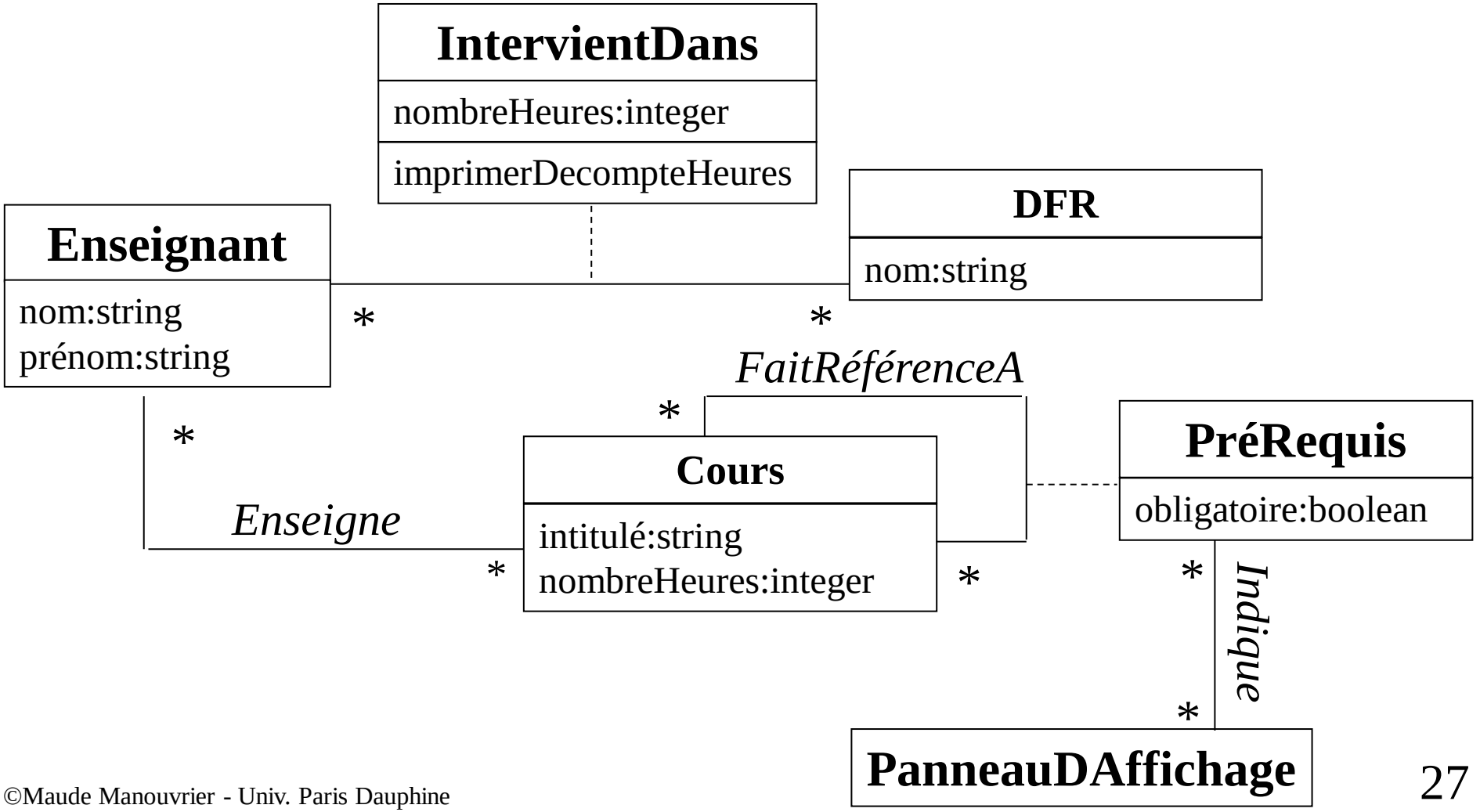


- ***Bag*** **(sac) : collection non ordonnée avec autorisation de doublons**
- **Séquence : collection ordonnée avec autorisation de doublons**



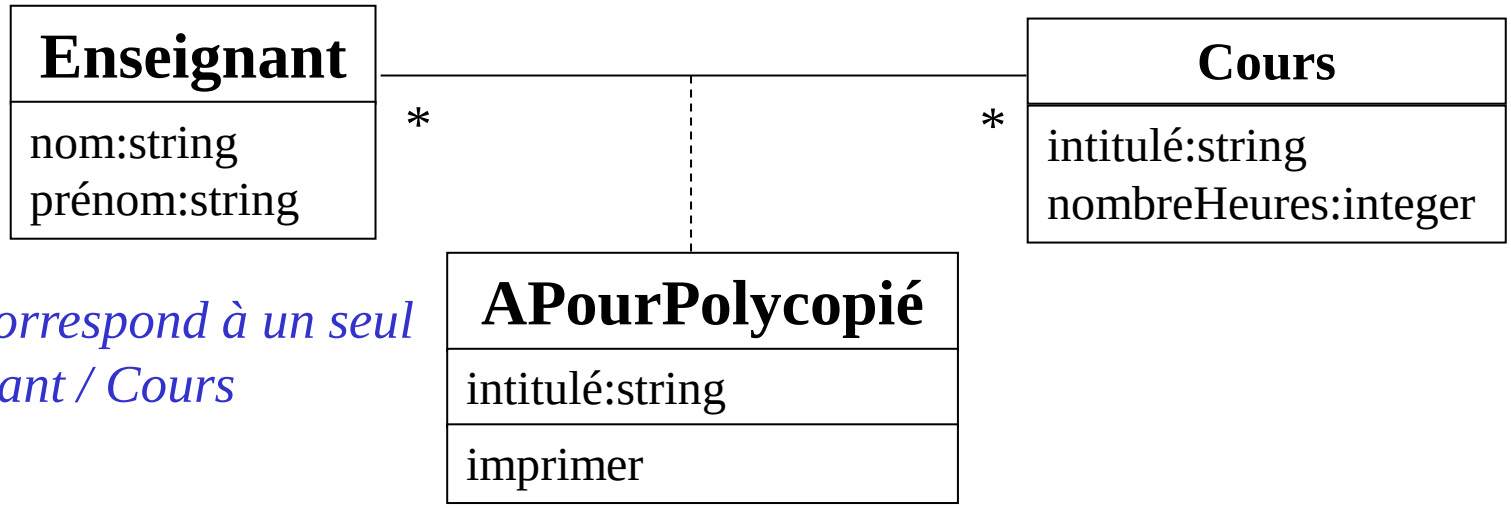
Modèle de classes (14/20) - Classe-association

- Association qui est également une classe
- Caractérisée par des attributs et des opérations

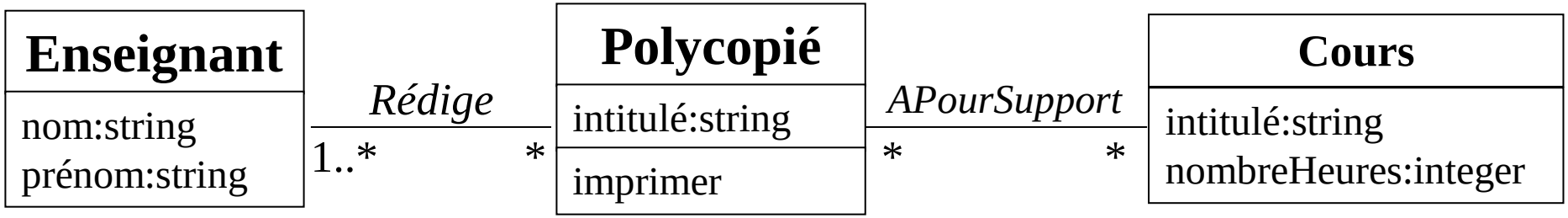


Modèle de classes (15/20) - Classe-association

Ne pas confondre classe-association et association promue au rang de classe



Un polycopié correspond à un seul couple Enseignant / Cours



Un nombre quelconque d'occurrences de Polycopié pour chaque Enseignant et chaque Cours

Modèle de classes (16/20)
Association qualifiée

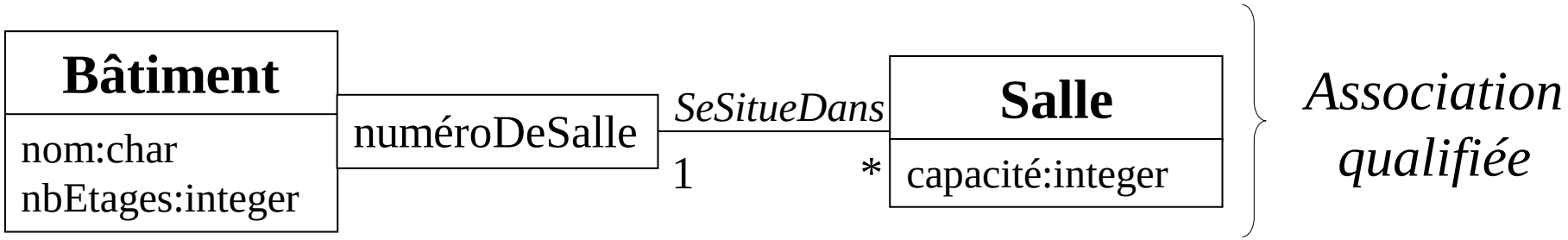
- **Qualificateur** :
  - Attribut permettant de distinguer les objets situés à l'extrémité de multiplicité « plusieurs » d'une association
  - Attribut réduisant la multiplicité « plusieurs » à « un »
- **Association qualifiée** : association contenant un ou plusieurs attributs qualificateurs

Modèle de classes (16bis/20)

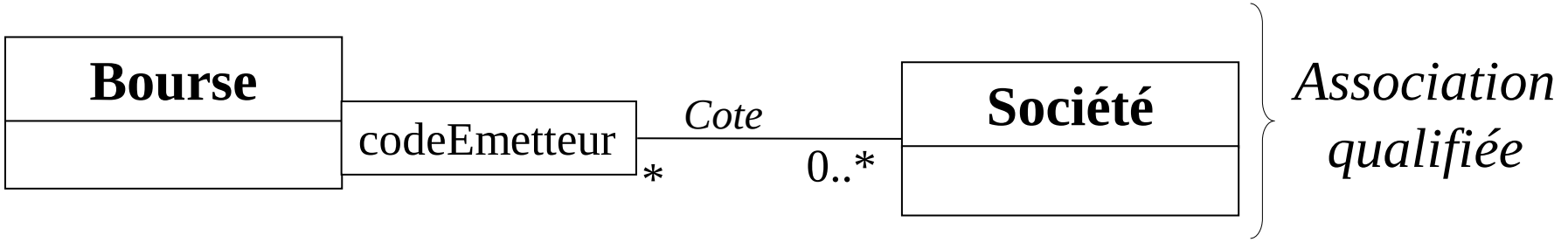
Association qualifiée

Un *numéro* de *Salle* permet d'identifier une salle unique dans un *Bâtiment* donné

Un *numéro* de *Salle* est relatif à un *Bâtiment*



Une société est cotée en bourse et a un code émetteur par Bourse

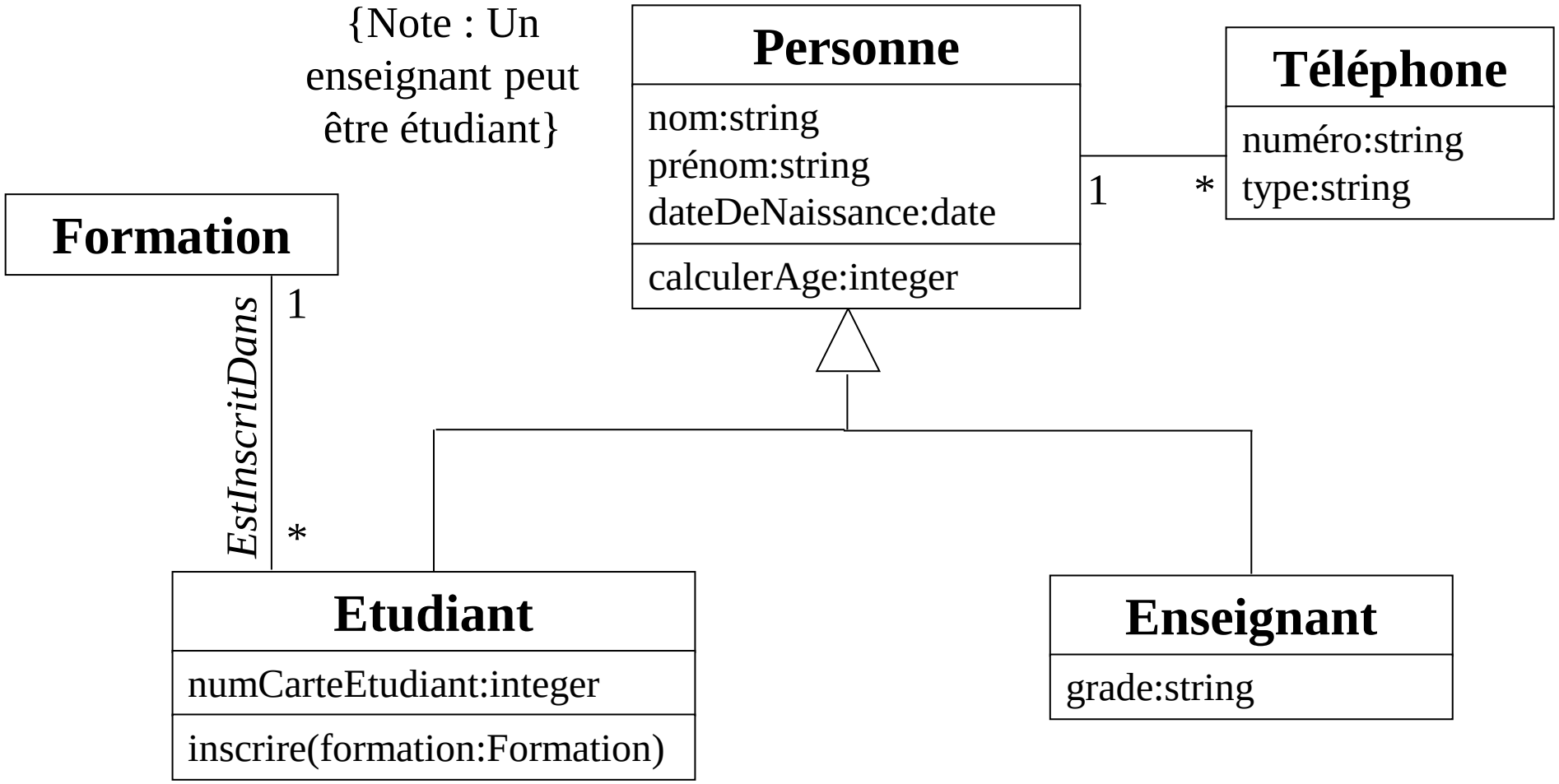


Modèle de classes (17/20)
Généralisation et héritage

- **Généralisation** : relation hiérarchique entre une classe (la super-classe) et une ou plusieurs variantes de cette classe (les sous-classes) [BR05]
- **Super-classe** (ou classe-mère) : attributs, opérations et associations communs
- **Sous-classe** (ou classe fille) : attributs, opérations et associations spécifiques
- **Héritage** des propriétés de la super-classe par ses sous-classes

Modèle de classes (18/20)

Généralisation et héritage



Modèle de classes (19/20)
Généralisation et héritage

Objectifs de la généralisation [BR05] :

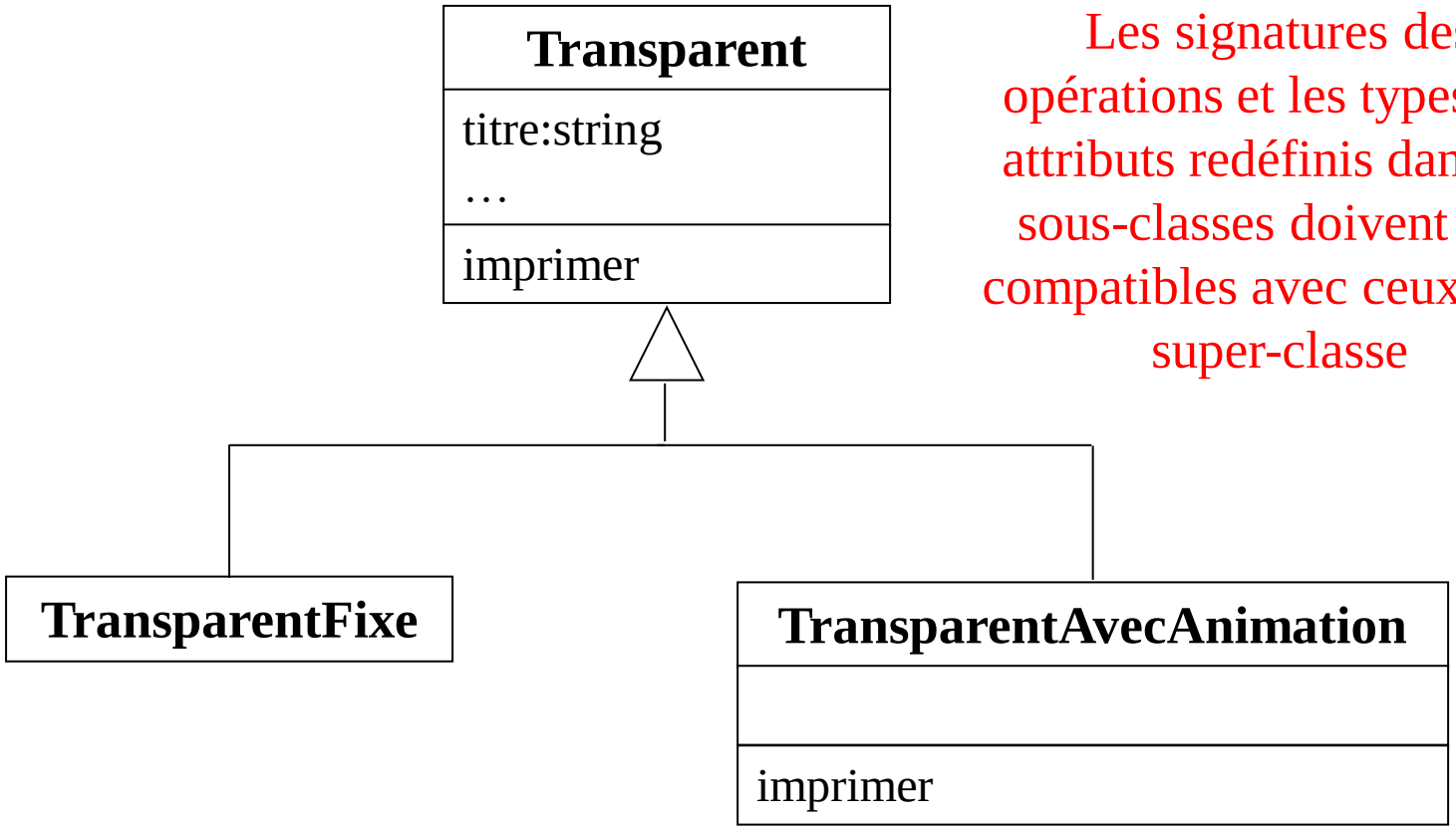
- Prise en charge du polymorphisme
- Structuration de la description des objets
- Possibilité de réutiliser du code

Modèle de classes (20/20)

Généralisation et héritage

Redéfinition des propriétés :

Les signatures des opérations et les types des attributs redéfinis dans les sous-classes doivent être compatibles avec ceux de la super-classe

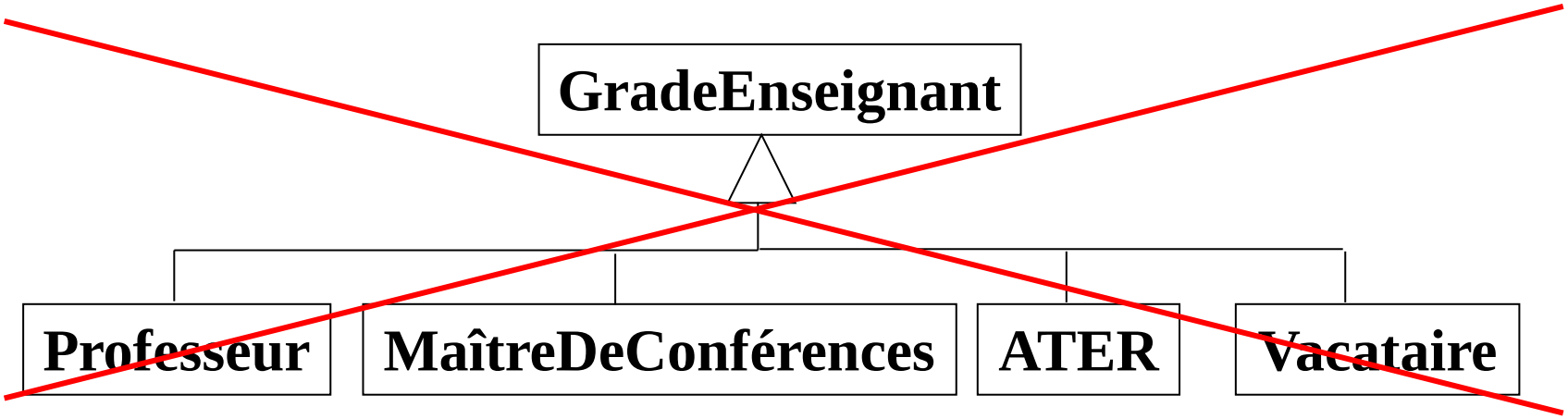


Modèle de classes - Concepts avancés

- Concepts de classe et d'objet avancés
- Extrémité d'association
- Associations n-aires
- Agrégation
- Classes abstraites
- Héritage multiple
- Contraintes
- Données dérivées
- Packages

Modèle de classes - Concepts avancés (1/12)

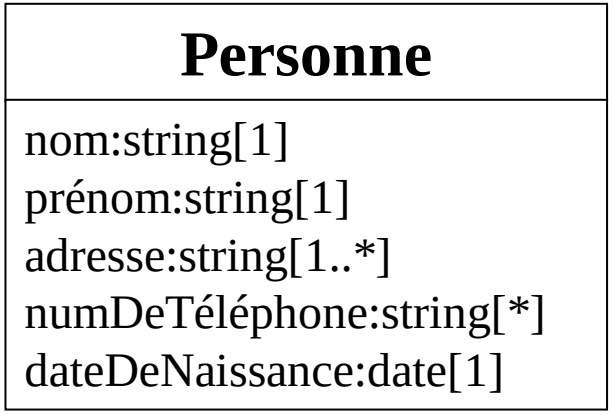
Énumération : type de données ayant un ensemble fini de valeurs



Modèle de classes - Concepts avancés (2/12)

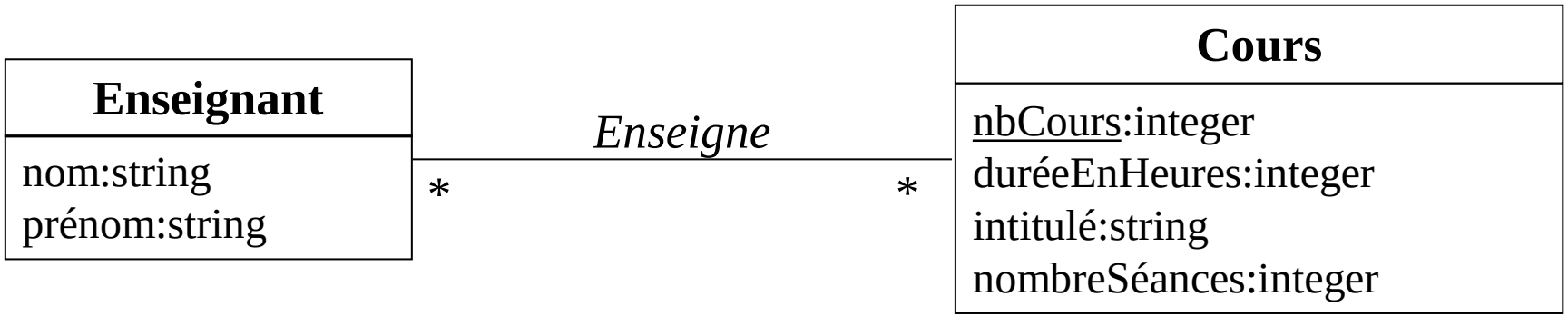
Multiplicité des attributs

- [1] : une valeur obligatoire
- [0..1] : une seule valeur optionnelle
- [*] : zéro ou plusieurs valeurs
- Par défaut : attribut mono-valué ([1])



Modèle de classes - Concepts avancés (3/12)

- **Portée** : indication de l'application d'une propriété à un objet (par défaut) ou à une classe
- **Attribut statique** : attribut dont la portée est la classe



Modèle de classes - Concepts avancés (4/12)

Visibilité : aptitude d'une méthode à référencer une propriété depuis une autre classe

- *public* (+) : propriété accessible par n'importe quelle méthode
- *protected* (#) : propriété protégée d'une classe uniquement visible par les méthodes de la classe et de ses sous-classes
- *private* (-) : propriété uniquement visible par les méthodes de la classe où elle a été définie
- *package* (~) : propriété accessible par les méthodes définies dans le même *package*

Signification pouvant varier en fonction du langage de programmation

Possibilité d'appliquer la visibilité aux extrémités d'association

Modèle de classes - Concepts avancés (5/12)

Association n-aire

A éviter ! Peut être décomposée en associations binaires

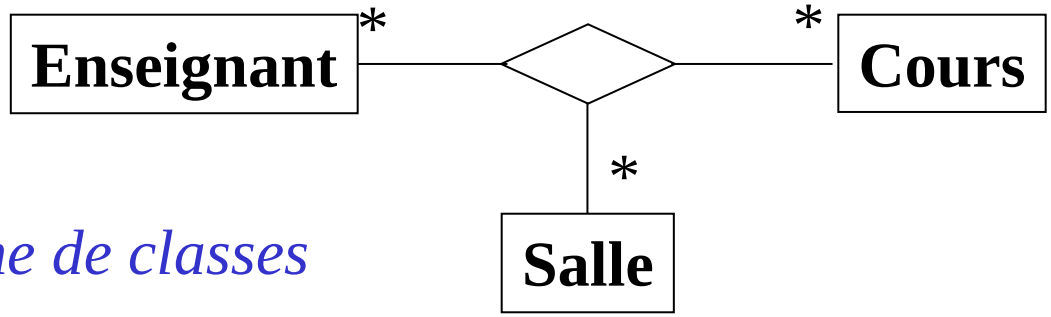


Diagramme de classes

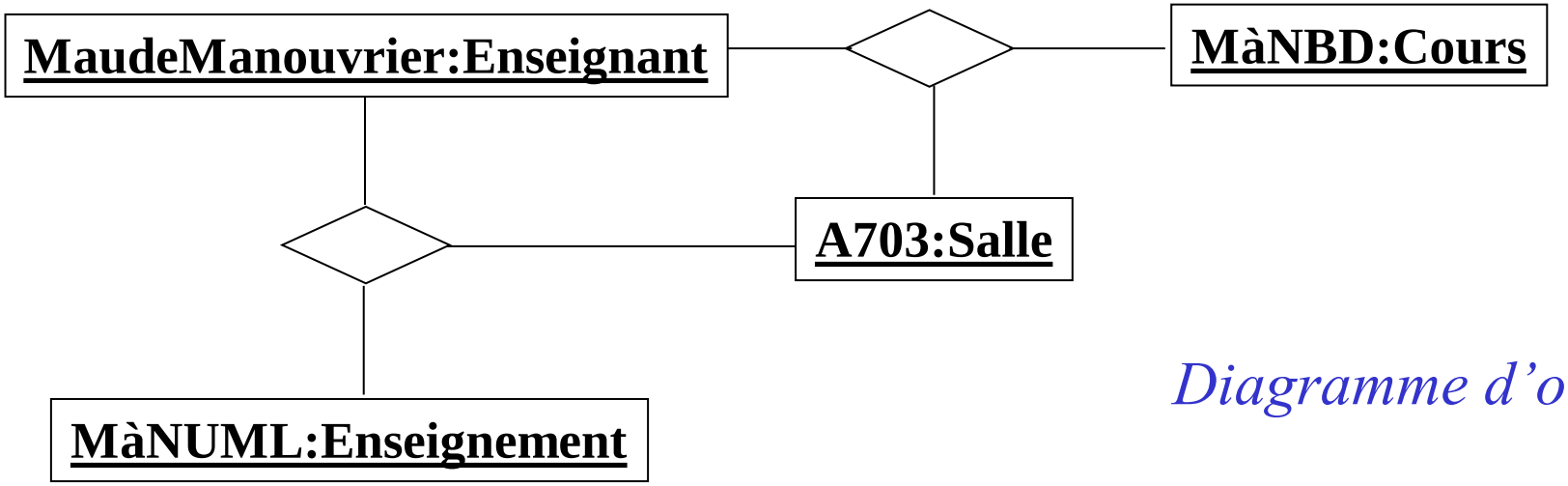
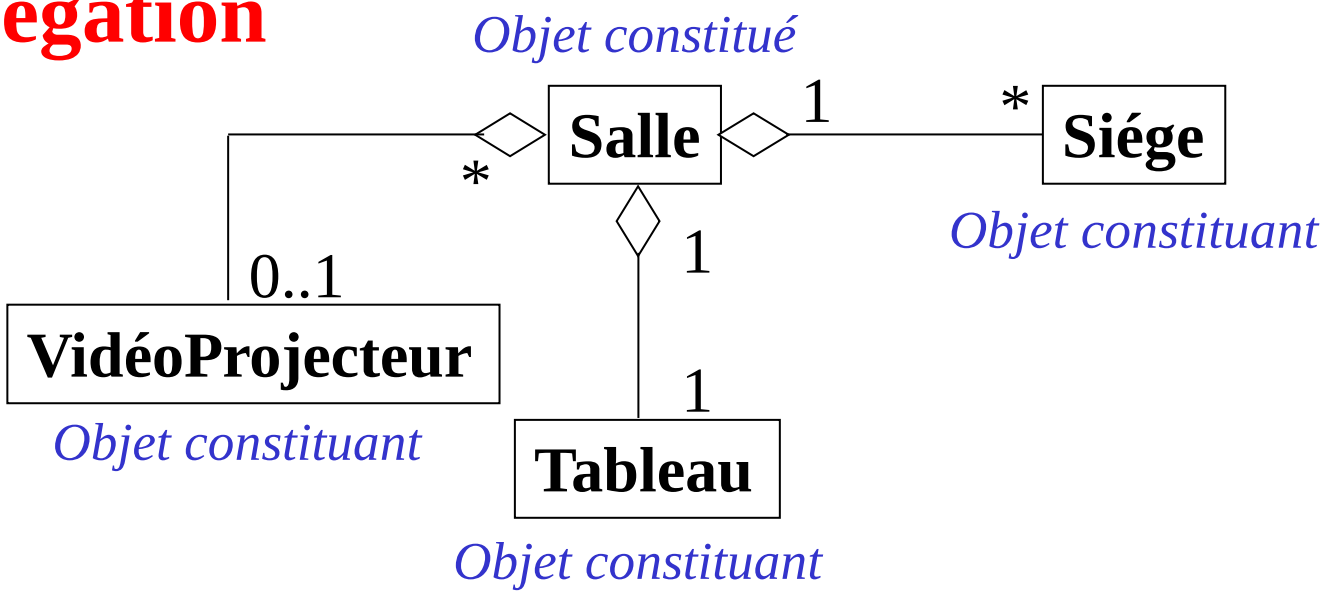


Diagramme d'objets

Modèle de classes - Concepts avancés (6/12)

Agrégation

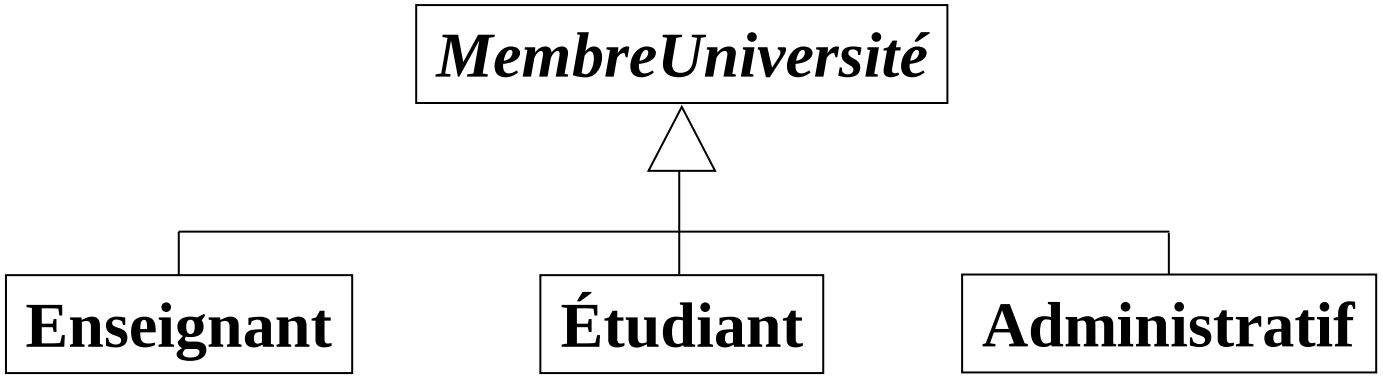


Composition

Appartenance des parties constituantes à un assemblage et coïncidence de leur durée de vie

Modèle de classes - Concepts avancés (7/12)

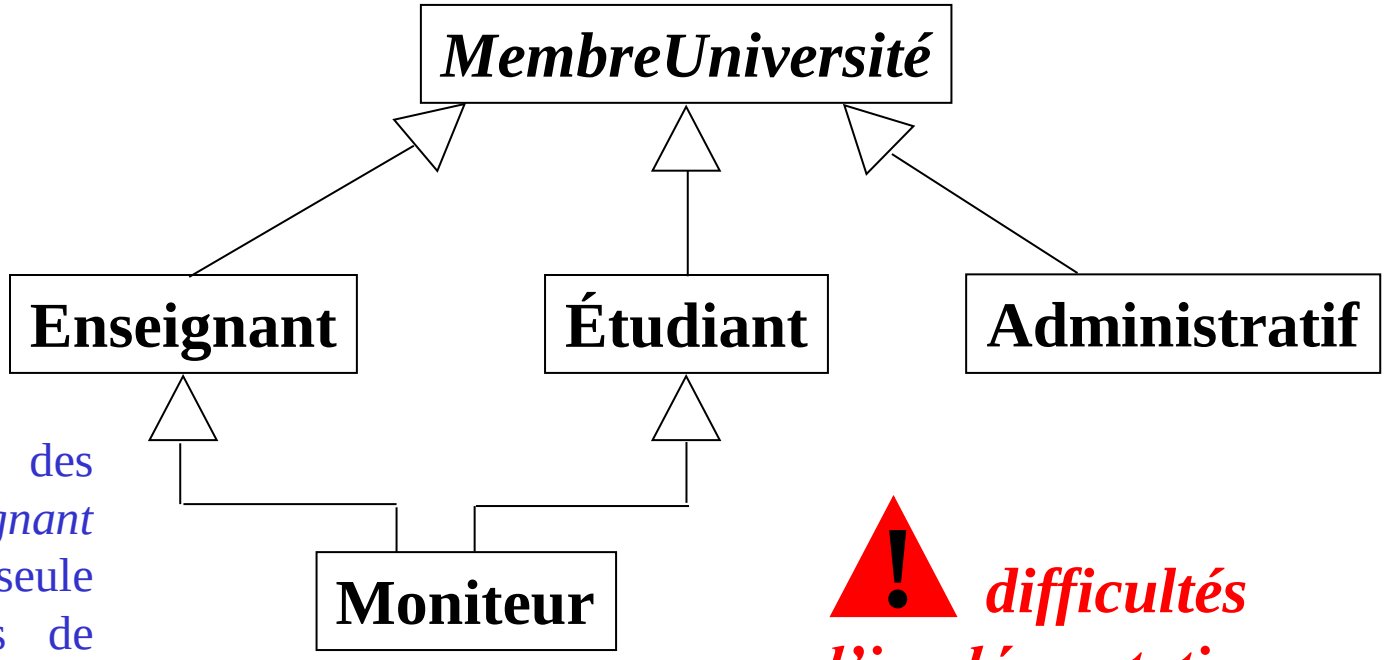
Classe abstraite : classe ne pouvant pas être instanciée en tant que telle



- Sous-classes d'une classe abstraite : obligatoirement concrètes
- Possibilité d'utiliser {abstract}

Modèle de classes - Concepts avancés (8/12)

Héritage multiple : héritage permettant à une classe d'avoir plusieurs super-classes et d'hériter des propriétés de tous ses parents



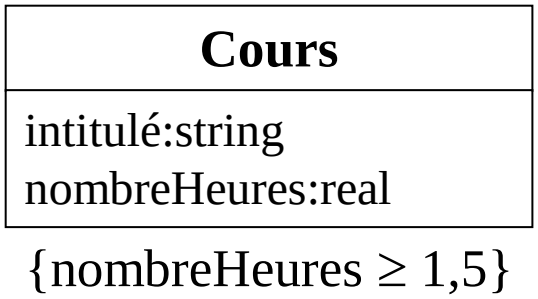
Moniteur hérite des propriétés de *Enseignant* et d'*Etudiant* et une seule fois des propriétés de *MembreUniversité*.

difficultés d'implémentation

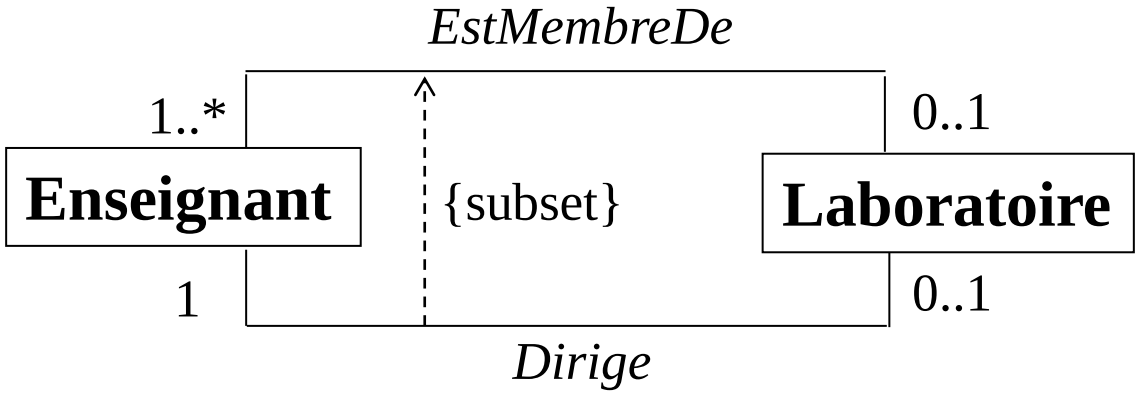
Modèle de classes - Concepts avancés (9/12)

Contrainte : condition booléenne s'appliquant aux éléments d'un modèle UML

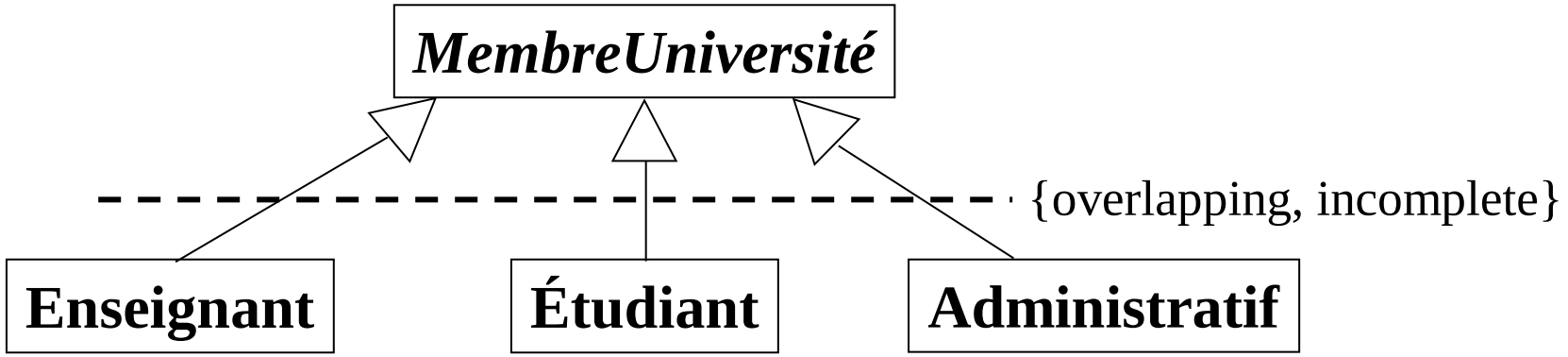
Contrainte sur les objets :



Contrainte sur les associations :

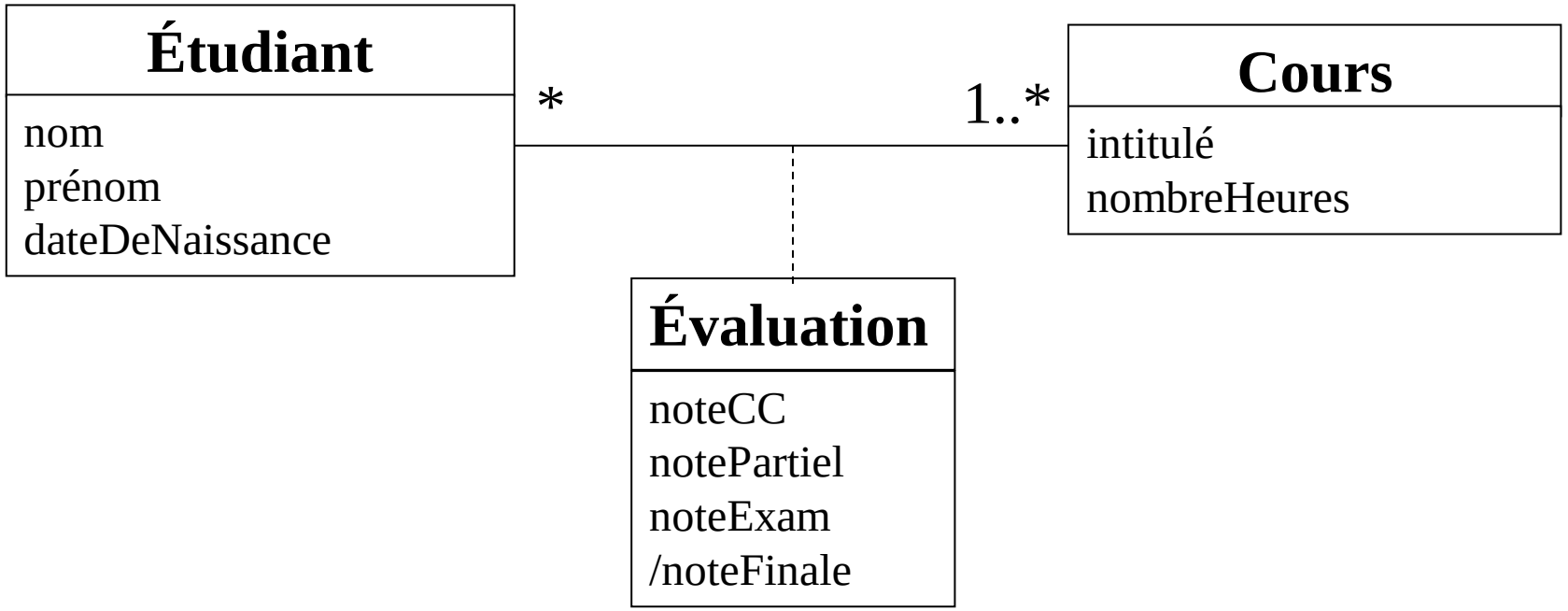


Contrainte sur les ensembles de généralisation :



Modèle de classes - Concepts avancés (10/12)

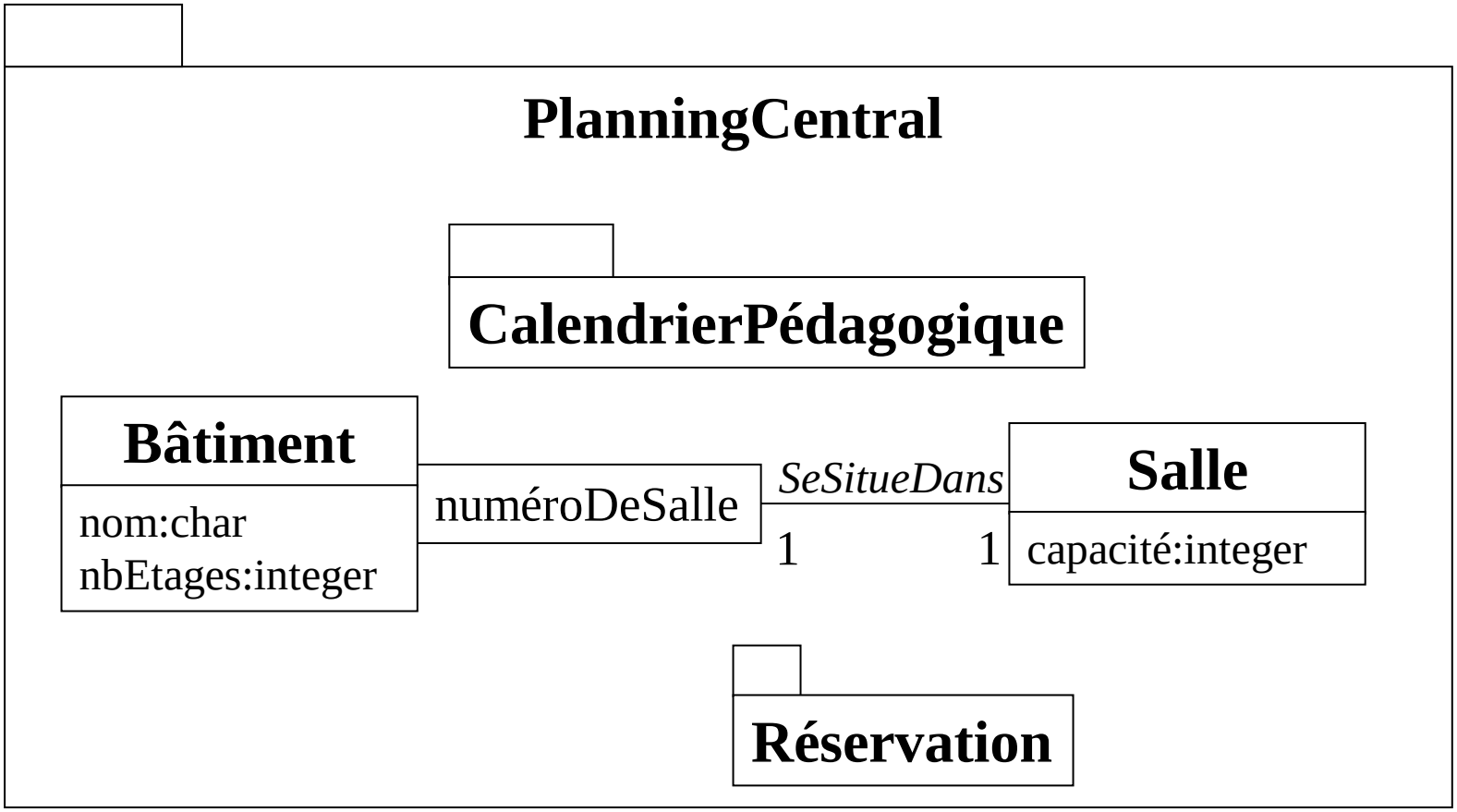
Élément dérivé : donnée définie en terme d'autres éléments



Pour des raisons de complexité d'implémentation, à utiliser avec parcimonie !

Modèle de classes - Concepts avancés (11/12)

Package : groupe d'éléments partageant un thème commun



Modèle de classes - Concepts avancés (12/12)

Package [BR05] :

- Utile pour organiser les grands modèles
- Ne définir une classe (i.e. représenter ses propriétés) que dans un seul *package*
- Référencer une classe d'un autre package en n'utilisant que le nom de la classe

Modèle d'états (1/15)

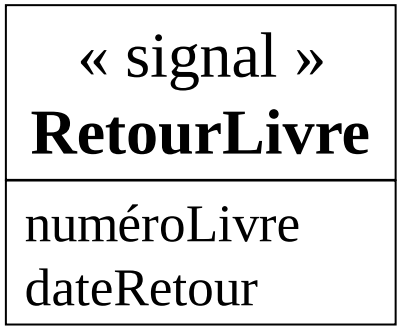
- Description des aspects d'un système relatifs à la durée et au séquençage des opérations
- Composé d'autant **diagrammes d'états** que de classes dotées d'un comportement temporel significatif pour l'application
- Vocabulaire :
  - **Événements**
  - **États**
  - **Transitions et conditions de franchissement**

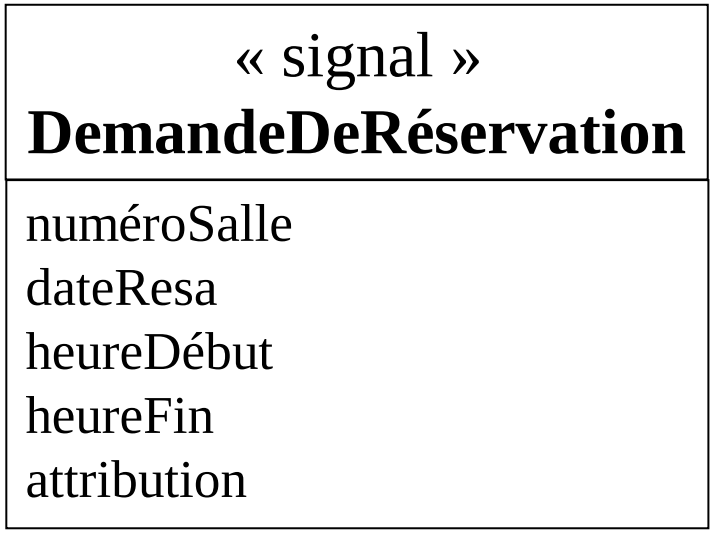
Modèle d'états (2/15) - Évènement

- Occurrence ou fait ayant lieu à un moment donné
- Modification intervenue dans l'environnement
  Ex. Réservation annulée
- Vérification de conditions d'erreur
  Ex. nombre d'emprunts > 6
- Vocabulaire :
  - **Évènement concurrents** : événements sans relation de causalité, sans effet l'un sur l'autre, dont l'exécution peut se chevaucher dans le temps
  - **Évènement de signal**
  - **Évènement de changement**
  - **Évènement temporel**

Modèle d'états (3/15) – Évènement de signal

- **Signal** : transmission d'information explicite et unidirectionnelle d'un objet à un autre
- Regroupement des signaux dans des **classes de signaux**
- **Évènement de signal** : événement d'envoi ou de réception d'un signal





Le livre 055.7 RAM *est retourné le 15/09/2014*

Instance de la classe de signaux *RetourLivre*

Modèle d'états (4/15) – Évènement

- **Évènement de changement** : Évènement engendré par la satisfaction d'une expression booléenne

  Passage de l'expression de faux à vrai ⇒ déclenchement de l'événement de changement

```
when (nombre d'étudiants > capacité de la salle)
when (nombre d'étudiants < 10)
when (nombre d'absences en TD > 3)
```

- **Évènement temporel** : Évènement engendré par l'occurrence d'un temps absolu ou l'écoulement d'une durée

```
when (date < 18/12/2005)
after (15 minutes)
```

Modèle d'états (5/15) - État

- Abstraction de valeurs et de liens d'un objet
- Spécification de la réponse d'un objet à des événements entrants
- **État vs Événement :**
  - Évènement : représentation d'un moment précis dans le temps
  - État : intervalle séparant la réception de deux événements par un objet

Modèle d'états (6/15) - État

Caractérisation d'un état

État : *LivreEmprunté*

Description : Le livre est emprunté par l'emprunteur d'identifiant *numéroEmprunteur*, à la date *dateEmprunt*

Séquence d'événements qui produit l'état :

Emprunt(numéroEmprunteur,dateEmprunt)

Condition qui caractérise l'état :

Emprunteur.nbreEmprunt < *NbreMaxEmprunt* et Emprunteur non pénalisé

Évènements acceptés dans l'état :

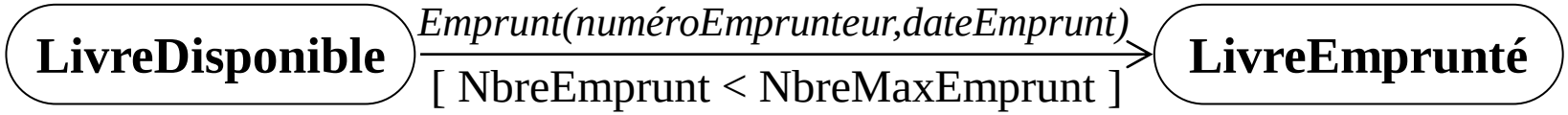
événement	**action**	**état suivant**
retourLivre	*terminerEmprunt*	*LivreDisponible*

Exemple inspiré de http://www-inf.int-evry.fr/COURS/IO21/COURS/CoursENSTBr.pdf

Modèle d'états (7/15)

Transition et conditions de franchissement

- **Transition** : passage instantané d'un état à un autre
- **Condition de franchissement (*guard condition*)** : expression booléenne devant être vraie pour le franchissement de la transition

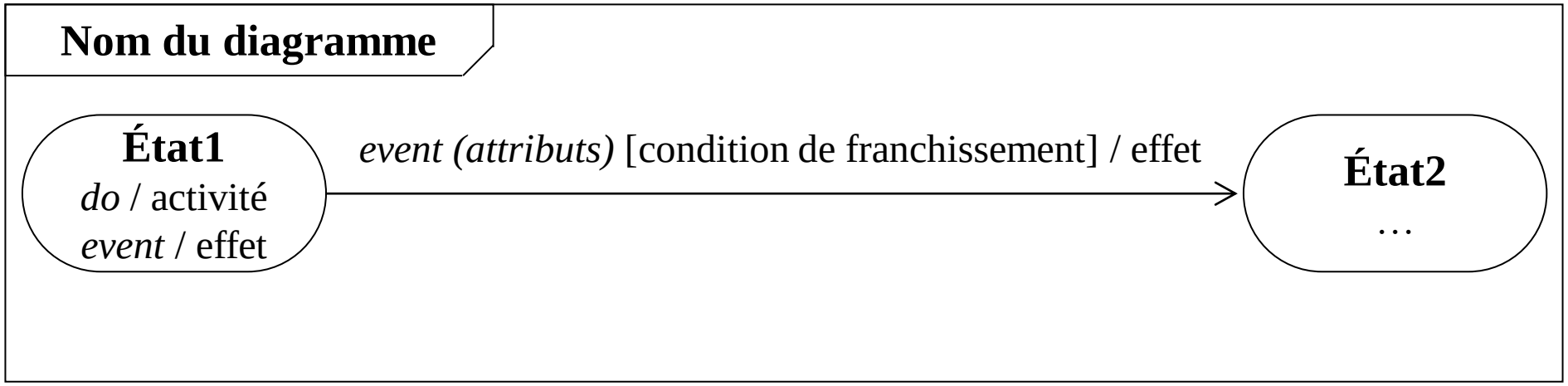


<u>Attention</u> : Condition de franchissement ≠ Événement

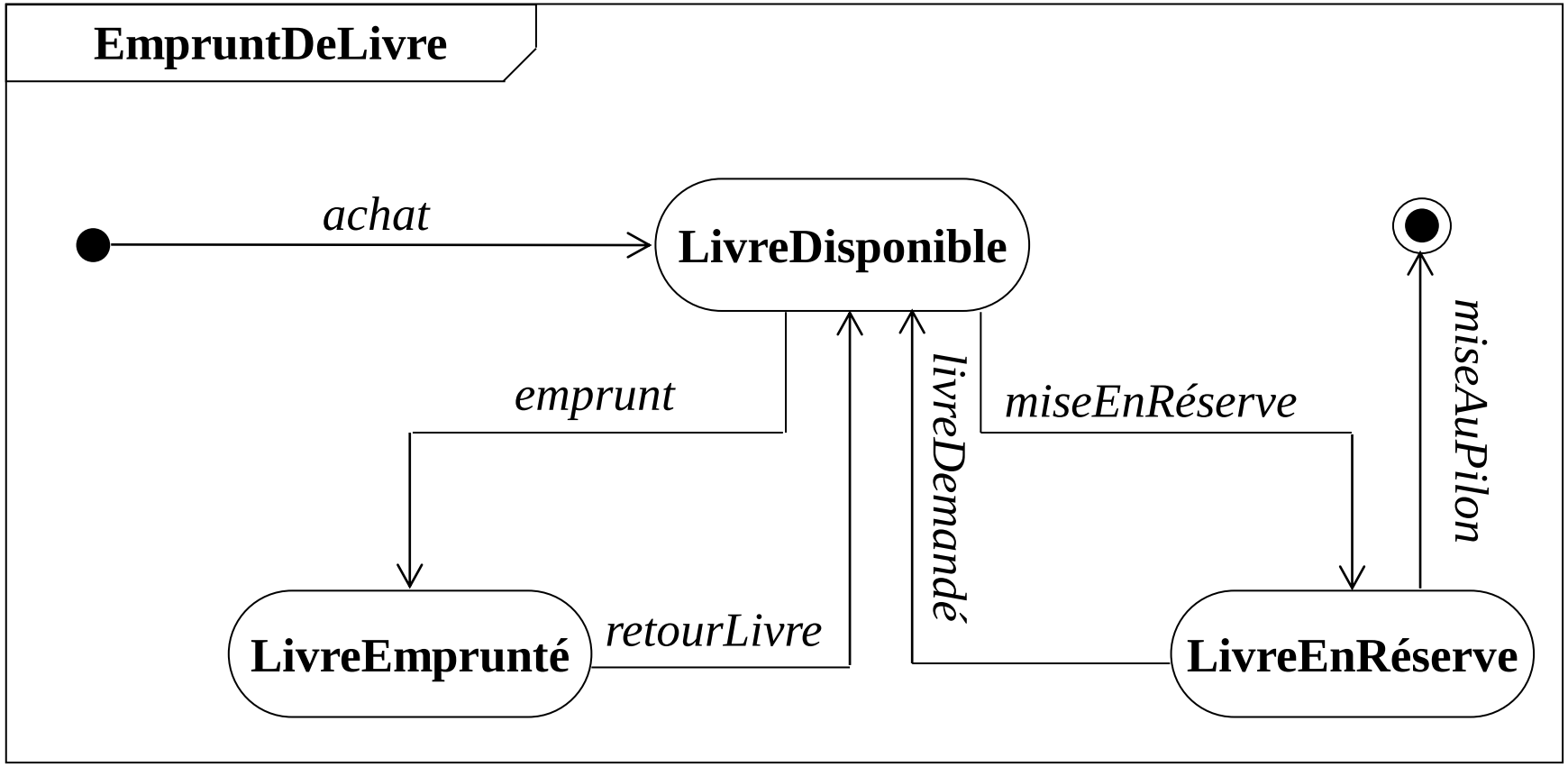
- ***Condition de franchissement évaluée une seule fois***
- ***Événement de changement évalué en permanence***

Modèle d'états (8/15) - Diagramme d'états

- Graphe orienté dont les sommets sont les états et les arcs les transitions entre les états
- Spécification des successions d'états provoqués par des successions d'évènements
- Associé à une classe
- En boucle infinie ou irréversible (*one-shot*)

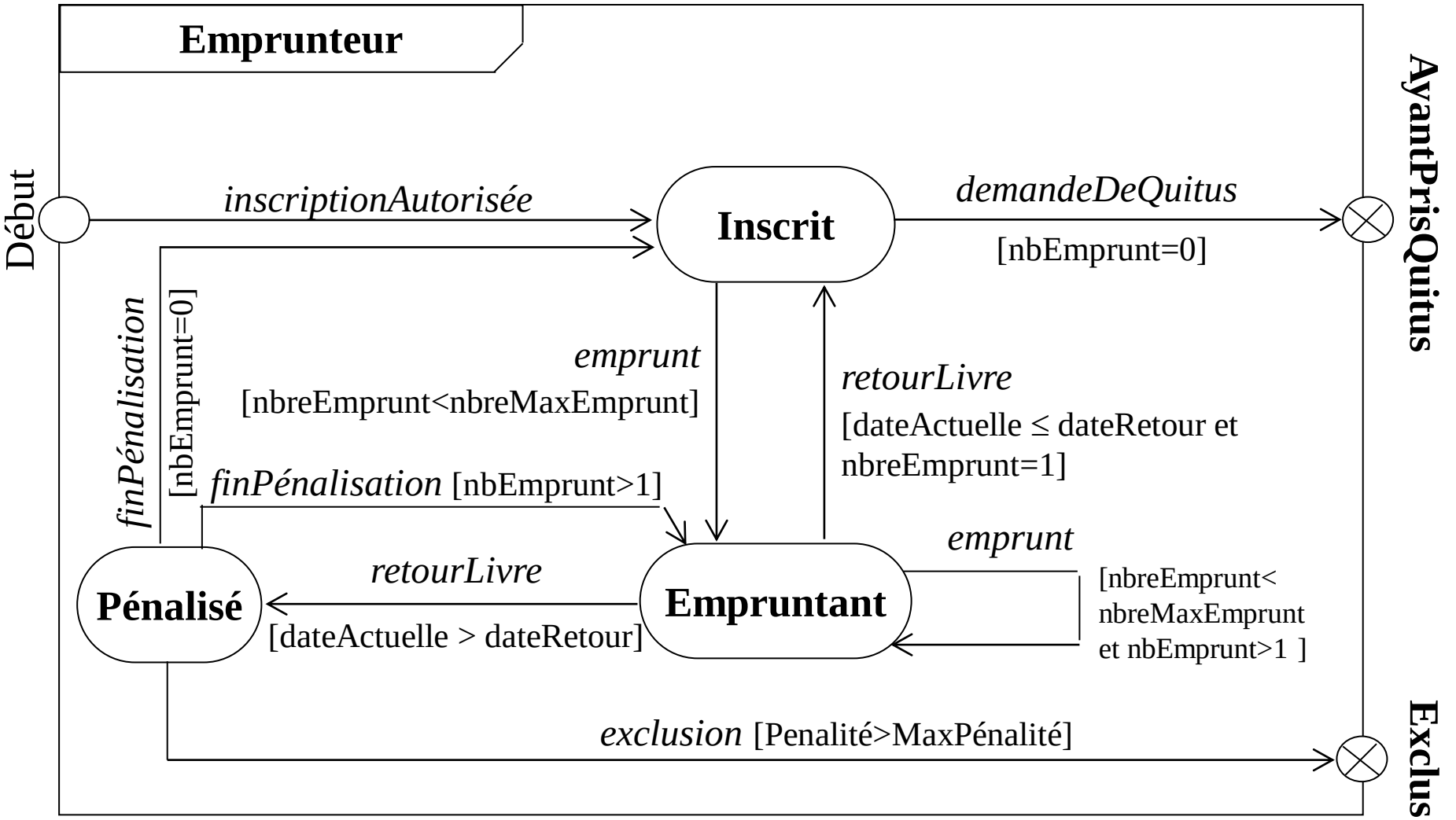


Modèle d'états (9/15) - Diagramme d'états



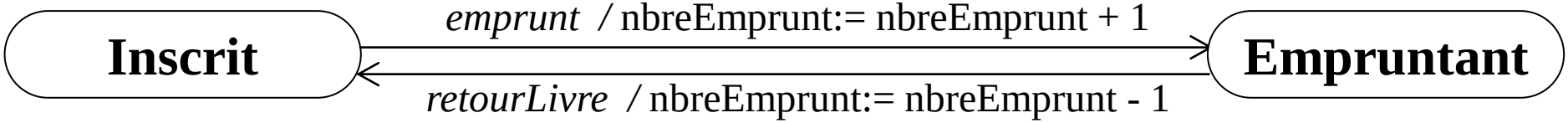
- Entrée dans l'*état initial* à la création de l'objet
- Destruction de l'objet à l'*état final*

Modèle d'états (10/15) - Diagramme d'états



Modèle d'états (11/15) – Effets et activités

- **Effet** : Référence à un comportement exécuté en réponse à un événement – noté par / suivi du nom de l'activité
- **Activité** : Comportement réel invoqué par un nombre quelconque d'effets
  - Effectuée suite à une transition, à l'entrée ou à la sortie d'un état, ou suite à un autre événement au sein d'un état
  - Pouvant représenter des opérations de contrôle internes (Ex. affectation de valeur à un attribut ou génération d'un autre événement)
  - Sans contrepartie dans la monde réel mais pour structurer le flux de contrôle dans une implémentation



Modèle d'états (12/15)

Activités *do* - Activités d'entrée et de sortie

- **Activité associée au mot-clé** *do* : Activité continue ou séquentielle exécutée sur une longue durée
  - Associée uniquement à un état (et non à une transition)
  - Pouvant être exécutée sur tout ou partie de la durée pendant laquelle un objet est dans l'état
  - Pouvant être interrompue par un événement reçu pendant son exécution

LivreEnRéserve
do / clignoter « livre en réserve »

- **Activité d'entrée ou de sortie** : exécutée à l'entrée (*entry /*) ou à la sortie d'un état (*exit /*)

LivreEmprunté
entry / enregistrer date de retour

Modèle d'états (13/15) – Activités

Ordre d'exécution des activités pour un état :

1. Activités de la transition entrante
2. Activités d'entrée
3. Activités *do*
4. Activités de sortie
5. Activités de la transition sortante

Si événement provoquant des transitions hors de l'état

⇒ Interruption des activités *do*

⇒ Mais exécution de l'activité de sortie

Auto-transition

⇒ exécution des activités d'entrée et de sortie

Modèle d'états (14/15)

Transitions d'achèvement et envoi de signaux

- **Transition d'achèvement** : Transition automatique, sans événement associé, exécutée à la fin de l'exécution des activités d'un état

Condition de franchissement testée une seule fois.

Si vérification d'aucune condition de franchissement ⇒ état toujours actif ou objet « bloqué » au sein de l'état

Pour prévoir toutes les conditions possibles : *else*

- **Envoi de signaux** : Interaction du système d'objets par échange de signaux

```
send cible.S(attributs)
```

- **Condition de concurrence critique (*race condition*)** : état final affecté par l'ordre d'exécution des signaux reçus

Modèle d'états (15/15)

Conseils pratiques [BR05] :

- Ne construire de diagrammes d'états que pour les classes ayant un comportement temporel significatif, i.e. les classes répondant différemment à différents événements ou ayant plus d'un état
- Pas de nécessité de construire un diagramme d'états pour toutes les classes
- Bien concevoir les conditions de franchissement pour ne pas bloquer un objet dans un état
- Faire attention aux conditions de concurrence quand un état peut recevoir des signaux de plus d'un objet

Modèle d’états - Concepts avancés

- Diagrammes d’états imbriqués
- États imbriqués
- Concurrence
- Relations entre modèle de classes et modèle d’états

Modèle d'états - Concepts avancés (1/7)

Diagrammes d'états imbriqués

Possibilité de détailler un état par un sous-automate

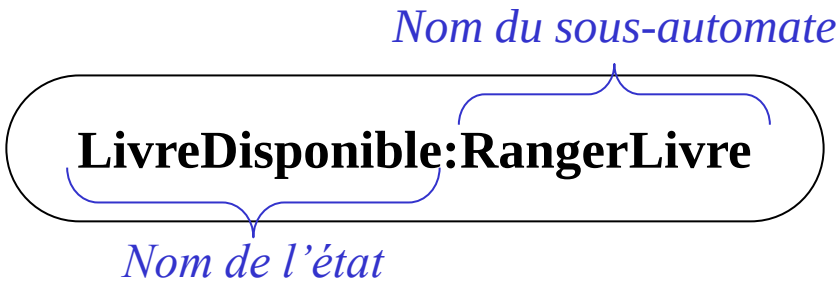
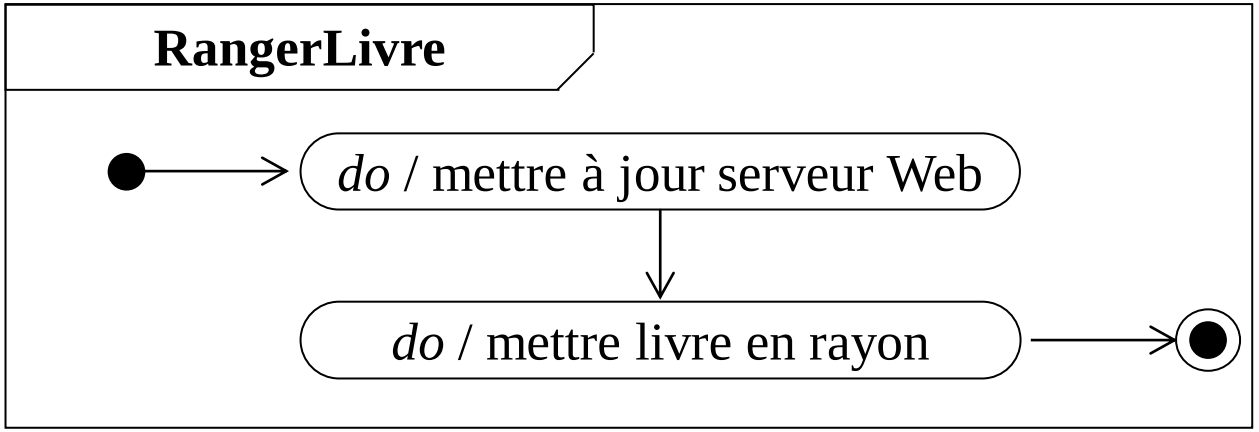


Diagramme d'états de plus bas niveau détaillant l'état LivreDisponible



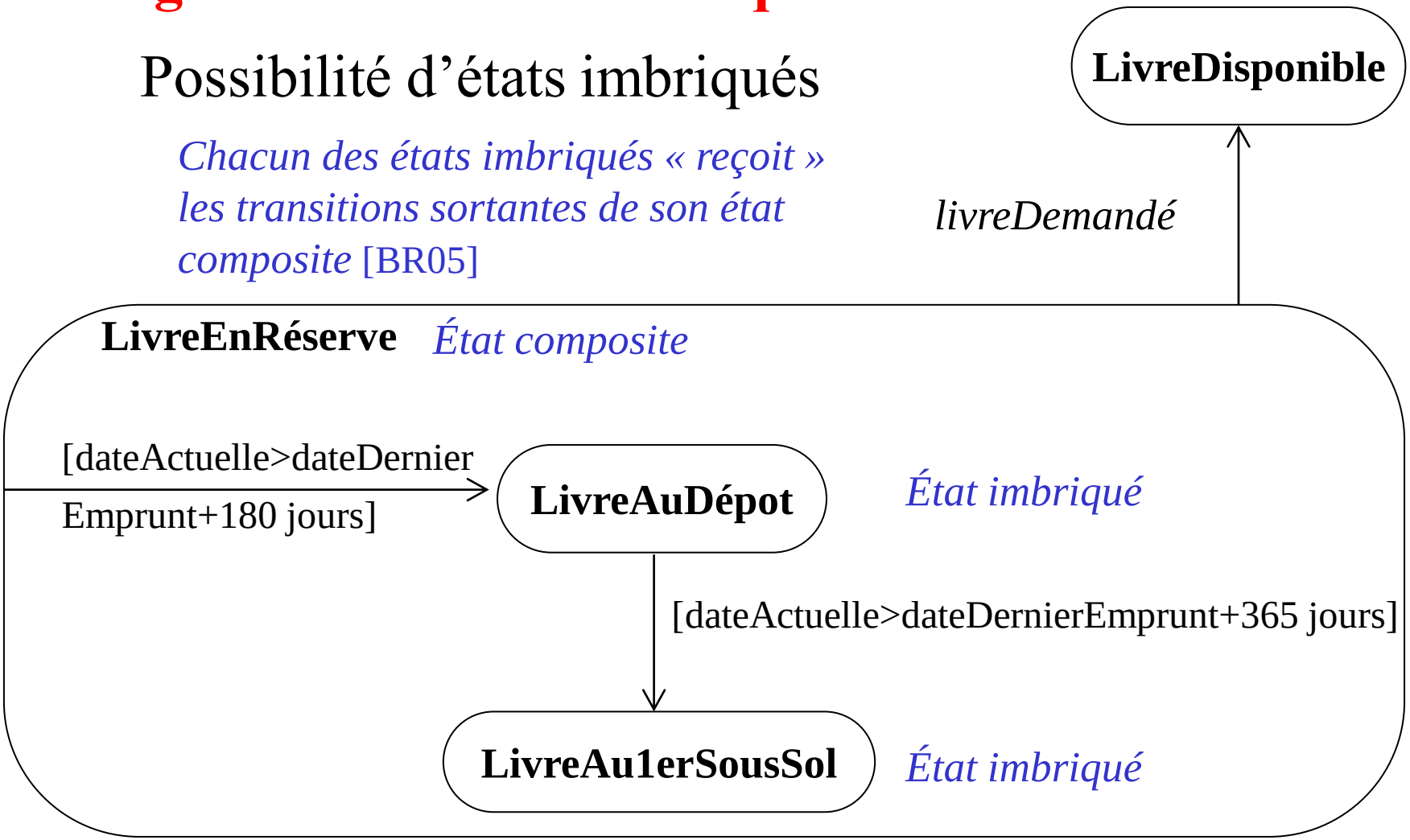
A utiliser de préférence pour des modèles de plus de 10 à 15 états [BR05]

Modèle d'états - Concepts avancés (2/7)

Diagrammes d'états imbriqués

Possibilité d'états imbriqués

Chacun des états imbriqués « reçoit » les transitions sortantes de son état composite [BR05]

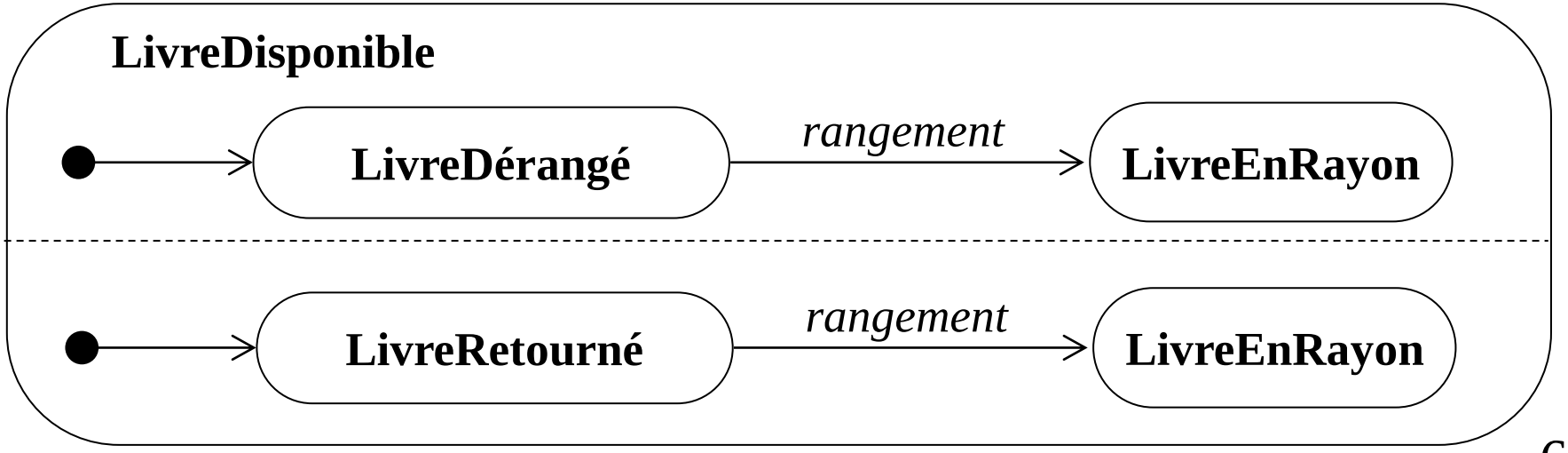


A utiliser quand une même transition s'applique à plusieurs états [BR05]

Modèle d'états - Concepts avancés (3/7)

Concurrence :

- **Concurrence d'agrégation**
  Diagramme d'état d'un assemblage = collection des diagrammes d'états de ses sous-parties
- **Concurrence à l'intérieur d'un objet**
  Possibilité de partitionner un objet en sous-ensembles d'attributs et de liens, chacun des sous-ensembles ayant un diagramme d'états

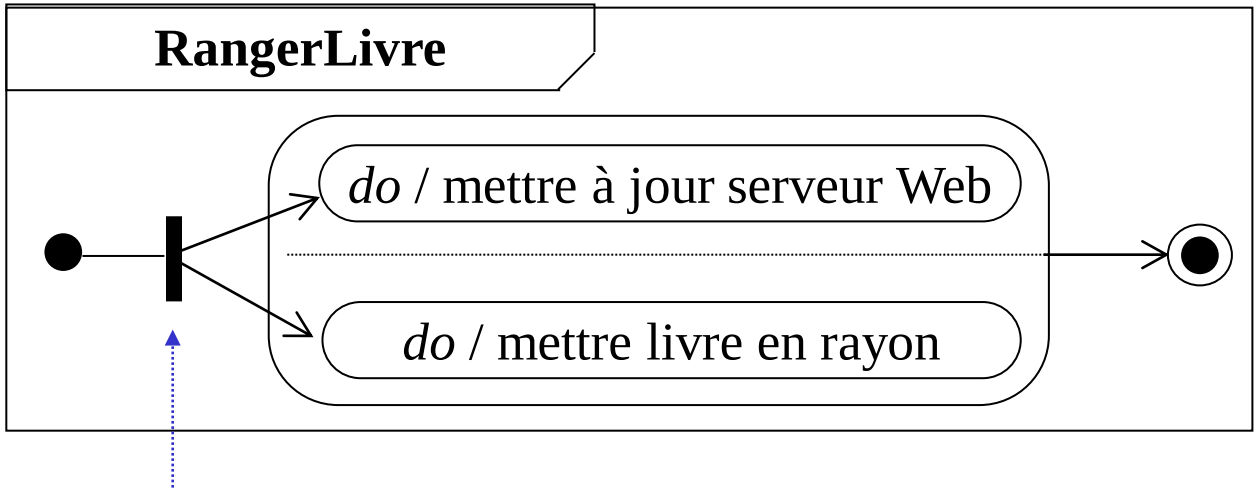


Modèle d'états - Concepts avancés (4/7)

Synchronisation du contrôle des activités concurrentes

Possibilité pour un même objet d'exécuter des activités concurrentes

⇒ Pas de synchronisation des activités
⇒ Mais division du contrôle des activités
⇒ Et synchronisation du contrôle



Division du contrôle en deux parties concurrentes (fork)

Modèle d'états - Concepts avancés (5/7)

Relations modèle de classes – modèle d'états

- **Modèle de classes** :
  - Description des objets, valeurs et liens pouvant exister dans un système
  - Modélisation des différences intrinsèques entre objets
- **Modèle d'états** :
  - Spécification des séquences possibles de modification des objets du modèle de classes
  - Modélisation des différences temporaires entre objets
- **Diagramme d'états** = description de tout ou partie du comportement des objets d'une classe donnée
- **État** = valeurs et liens détenus par un objet

Modèle d'états - Concepts avancés (6/7)

Relations modèle de classes – modèle d'états (suite)

- **Agrégation d'objets** $\Rightarrow$
  - Des états indépendants propres à chaque partie d'une agrégation
  - L'état de l'assemblage = combinaison des états de toutes ses parties
- **Hiérarchie de classes d'objets** $\Rightarrow$
  - Héritage par les sous-classes des modèle d'états de leur classe ancêtre
  - Possibilité pour les sous-classes d'avoir leur propre diagramme d'états, traitant de préférence uniquement des attributs propres aux sous-classes

Modèle d'états - Concepts avancés (7/7)

Relations modèle de classes – modèle d'états (suite)

- Possibilité de définir les signaux à travers différentes classes - parallèles aux classes d'objets
- Possibilité d'implémenter les transitions comme des opérations sur des objets (avec comme nom d'opération : le nom du signal correspondant)

Pouvoir d'expression plus puissant des signaux car dépendance entre la réponse à un événement et l'état de l'objet recevant l'événement

Modèle d'interactions (1/17)

- **Modèle de classes** = représentation des objets et de leurs relations
- **Modèle d'états** = description du cycle de vie des objets
- **Modèle d'interactions** = expression de la façon dont les objets interagissent pour produire des résultats utiles à l'application [BR05]
- Plusieurs niveaux d'abstraction du modèle d'interactions :
  - **Cas d'utilisation** : description de l'interaction du système avec les acteurs extérieurs
  - **Diagrammes de séquence** : représentation des messages échangés entre ensemble d'objets au fil du temps
  - **Diagramme d'activités** : représentation du flux de contrôle entre les étapes de traitement

Modèle d'interactions (2/17)
Cas d'utilisation

- **Acteur**
  - = Utilisateur externe direct du système
  - = Objet ou ensemble d'objets communiquant directement avec le système sans en faire partie
  - = Tout ce qui interagit directement avec le système

  Ex. Un employé d'une bibliothèque

- **Cas d'utilisation**
  - Identification des fonctionnalités pouvant être fournies par un système en interagissant avec les acteurs

    Ex. L'employé enregistre un emprunt
  - Organisation des fonctionnalités selon le point de vue utilisateur

Modèle d'interactions (3/17) – Cas d'utilisation

Caractérisation d'un cas d'utilisation

Cas d'utilisation :Enregistrer un emprunt de livre

Résumé : Un emprunt d'un livre pour un membre de la bibliothèque est enregistré

Acteur : Un employé de la bibliothèque

Pré-conditions : L'emprunteur doit être inscrit à la bibliothèque et ne pas avoir atteint le quota d'emprunts ou être exclu ou pénalisé et le livre doit pouvoir être emprunté

Description : Le système de gestion de la bibliothèque est dans l'état « Enregistrement d'un emprunt ». L'employé lit la carte de membre de l'emprunteur. Après saisie, le système de prêts indique s'il reconnaît l'emprunteur et si l'emprunteur est autorisé à emprunter. L'employé lit le code barre du livre à emprunter. Si le livre peut être emprunté, (1) l'emprunt est enregistré pour l'emprunteur et le livre, (2) la date de retour du livre est enregistrée et affichée, (3) le nombre de livres pouvant être encore empruntés par l'emprunteur est mis à jour et affiché.

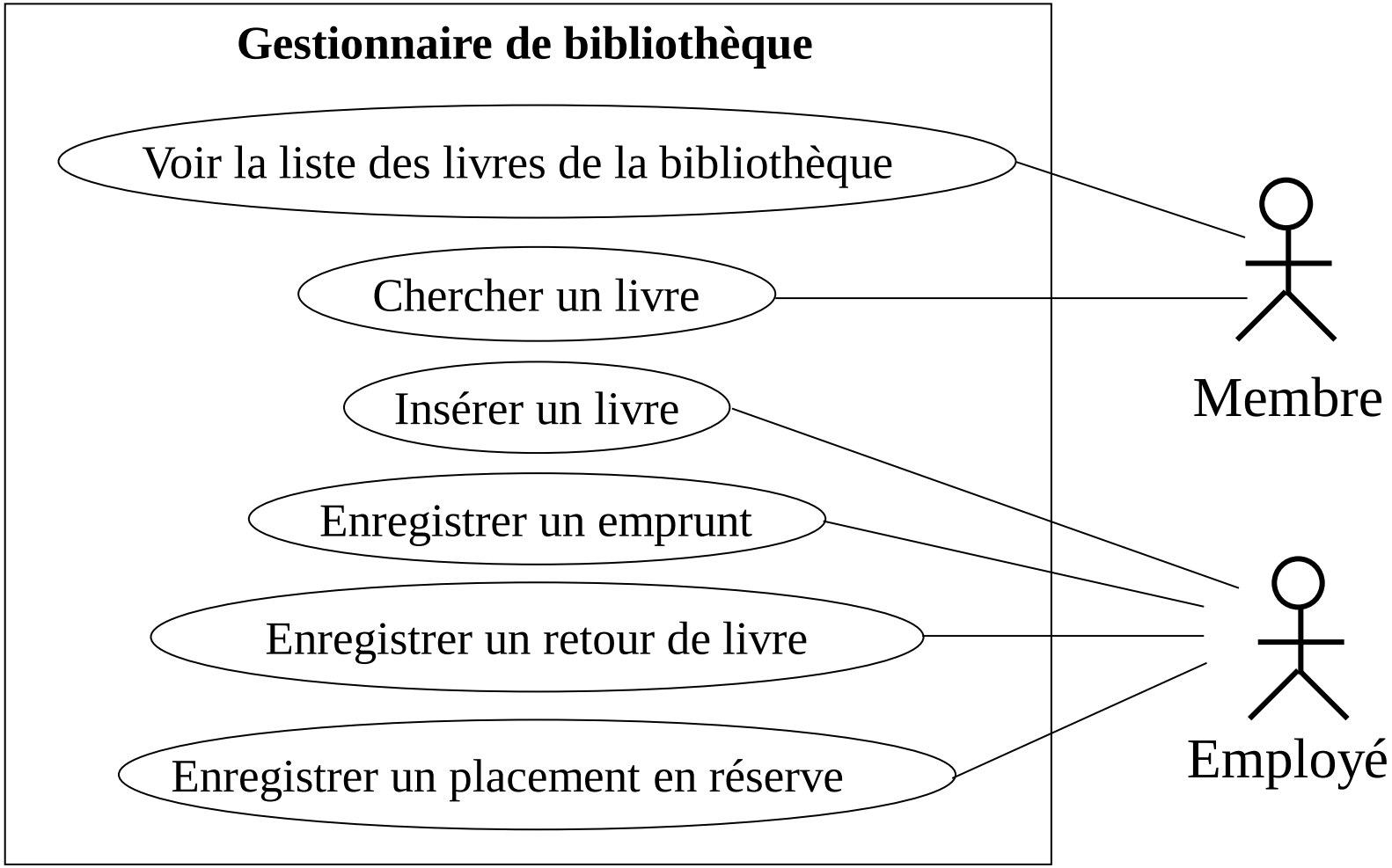
Exceptions :

Annulation : Si l'emprunteur ne peut pas emprunter ou si le livre ne peut pas être emprunté, le système de prêt revient à l'écran « Enregistrement d'un emprunt ».

Post-conditions : La date de retour du livre emprunté et le nombre de livres pouvant être encore empruntés par l'emprunteur sont affichés.

Modèle d'interactions (4/17) – Cas d'utilisation

Diagramme d'un cas d'utilisation d'un système de gestion d'une bibliothèque



Modèle d'interactions (5/17)

Conseils pratiques de [BR05] :

- Fixer précisément les limites du systèmes
- Limiter un acteur à un objectif unique et cohérent quitte à capturer les objectifs d'un même objet réel à travers plusieurs acteurs
- Ne pas définir trop étroitement les cas d'utilisation
- Lier les acteurs et les cas d'utilisation
- Ne pas chercher à trop formaliser
- Structurer les cas d'utilisation des grands systèmes

Modèle d'interactions (6/17)

Modèles de séquences

- Précision des thèmes fonctionnels introduits par les cas d'utilisation
- Ajout de détails et précision de la description informelle des cas d'utilisation
- Deux modèles :
  - **Scénarios** : séquence d'événements ayant lieu lors du fonctionnement du système (ex. exécution d'un cas d'utilisation) - décrite sous forme textuelle
  - **Diagrammes de séquence** : Représentation des participants à une interaction et de leurs messages échangés

Modèle d'interactions (7/17) – Scénario

- Séquence d'événements se produisant lors d'une exécution particulière d'un système
- Représentation de l'historique de l'exécution d'un système réel existant ou d'un prototype d'exécution d'un système envisagé
- De portée variée :
  - Comprenant tous les événements du système
  - Ou n'incluant que les événements affectant certains objets ou générés par certains objets
- Étape d'un scénario

  = Commandes logiques

  ≠ Simples clics de souris

Modèle d'interactions (8/17) – Scénario

Scénario d'une session d'un système de gestion de bibliothèque

Maude Manouvrier se connecte au module de « Recherche d'un livre »

Le système affiche le formulaire de saisie de recherche d'un livre

Maude Manouvrier saisit le terme « UML » dans le champ « Mot-clé »

Le système recherche, parmi les livres, ceux dont la liste de mots-clés correspondante contient le mot « UML »

Le système retourne les titres des livres répondant à la requête

Maude Manouvrier clique sur l'ouvrage intitulé « Modélisation et conception orientées objet avec UML 2 »

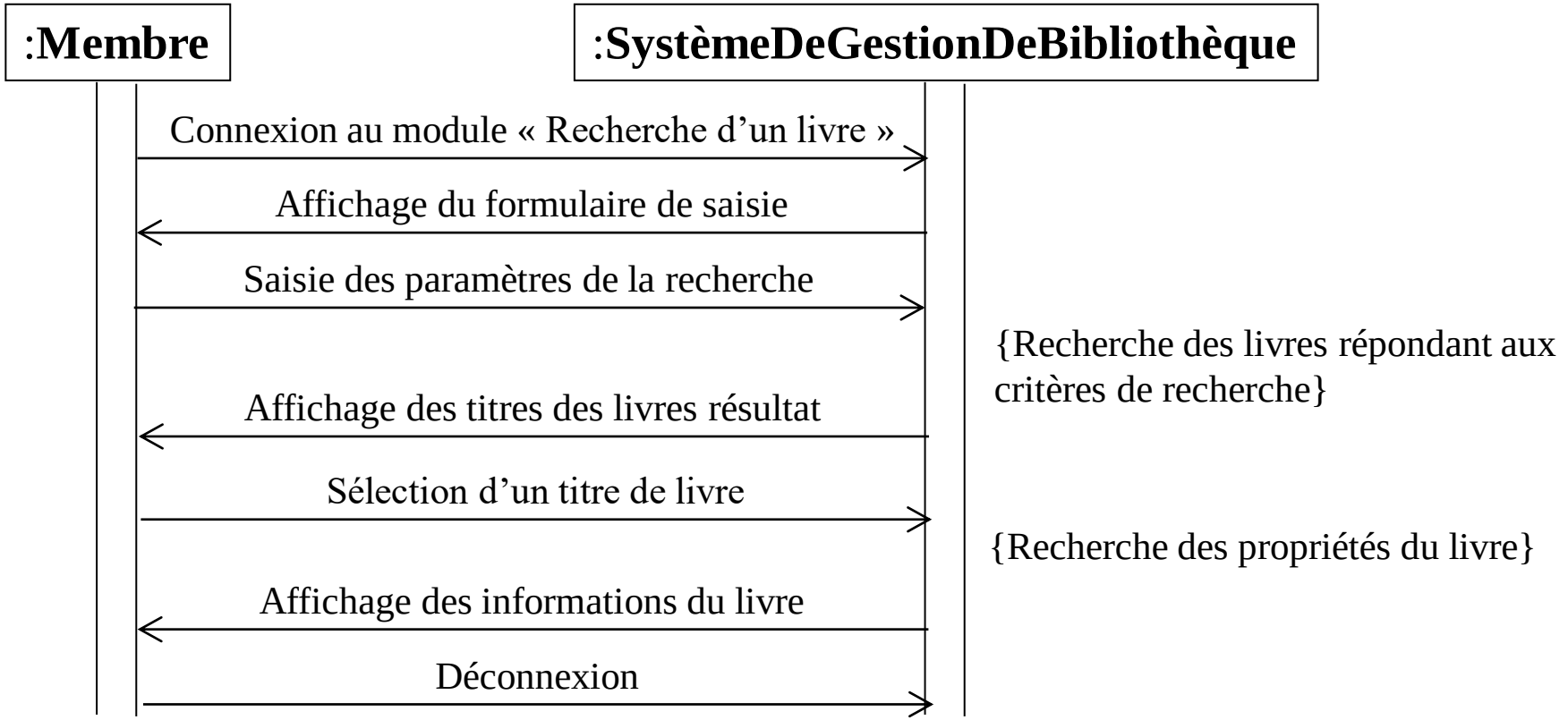
Le système affiche les informations correspondant à ce livre

Maude Manouvrier se déconnecte

Modèle d'interactions (9/17)

Diagrammes de séquence

Représentation des participants à une interactions et de leurs messages échangés



Modèle d'interactions (10/17)
Diagrammes de séquence

- Nécessité d'avoir plusieurs diagrammes de séquence pour décrire le comportement de chaque cas d'utilisation
- Représentation d'une séquence de comportement par diagramme de séquence
- Nécessité de tracer un diagramme de séquence pour chaque condition d'exception contenue dans un cas d'utilisation
- Impossibilité de représenter tous les scénarios
- Mais nécessité de détailler tous les cas d'utilisation et tous les types de comportement possibles avec des diagrammes de séquence

Modèle d'interactions (11/17)

Conseils pratiques de [BR05] :

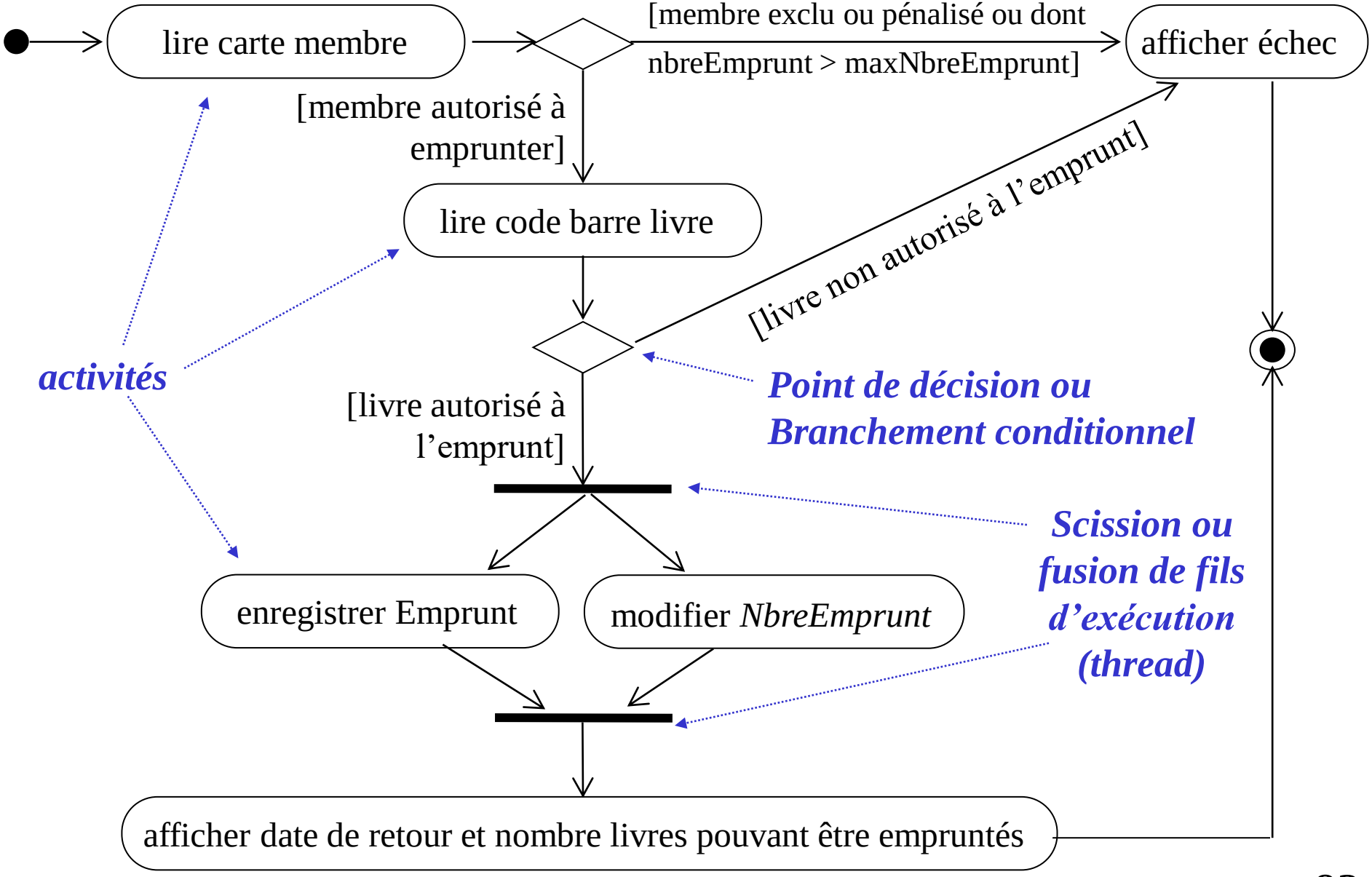
- Réaliser au moins un scénario par cas d'utilisation
- Synthétiser les scénarios par des diagrammes de séquence
- Subdiviser les interactions complexes
- Réaliser un diagramme de séquence par condition d'erreur

Modèle d'interactions (12/17)
Modèle d'activités

Diagramme d'activités :

- Représentation des étapes d'un processus complexe (ex. algorithme ou *workflow*) et des contraintes de séquencement
- Expression du flux de contrôle, comme un diagramme de séquences, mais avec une attention particulière sur les opérations plutôt que les objets
- Suite d'étapes correspondant aux activités décrites dans le modèle d'états

Modèle d'interactions (13/17) - Modèle d'activités



Modèle d'interactions (14/17)

Modèle d'activités

Diagramme d'activités :

- Possibilité de décomposer une activité en activités plus fines

- Nécessité d'avoir un même niveau de détail pour toutes les activités d'un même diagramme
- Possibilité d'utiliser la condition [*else*]

- Aucune garantie possible sur le choix de l'activité exécutée en cas de satisfaction de plusieurs conditions
- Possibilité d'avoir des activités concurrentes

Modèle d'interactions (15/17)
Modèle d'activités

Diagramme d'activités exécutables

- Possibilité de placer un *jeton d'activité* sur une activité pour indiquer son exécution
- Possibilité d'avoir plusieurs jetons en cas de concurrence
- Scission de contrôle ⇒ augmentation du nombre de jetons

Modèle d'interactions (16/17)

Conseils pratiques de [BR05] :

- Ne pas se servir des diagramme d'activités comme d'organigrammes pour développer des logiciels
- Équilibrer les diagrammes d'activités
- Concevoir avec soin les branchements conditionnels et les conditions
- Utiliser avec prudence les activités concurrentes
- Exécuter les diagrammes d'activités pour comprendre le déroulement d'un processus

Modèle d'interactions (17/17)

- **Cas d'utilisation** : partition d'un système en fonctionnalités discrètes et significatives pour les acteurs (extérieurs au système)
- Possibilité de détailler les cas d'utilisation par des **scénarios** et des **diagrammes de séquence**
- **Diagramme de séquence** : représentation claire des objets participant à une interaction et des messages émis ou reçus par ces objets
- **Diagramme d'activités** : description des détails d'un traitement

Modèle d'interactions
Concepts avancés

- Relations entre cas d'utilisation
- Modèles de séquence procédurale
- Notation spéciales des modèles d'activités

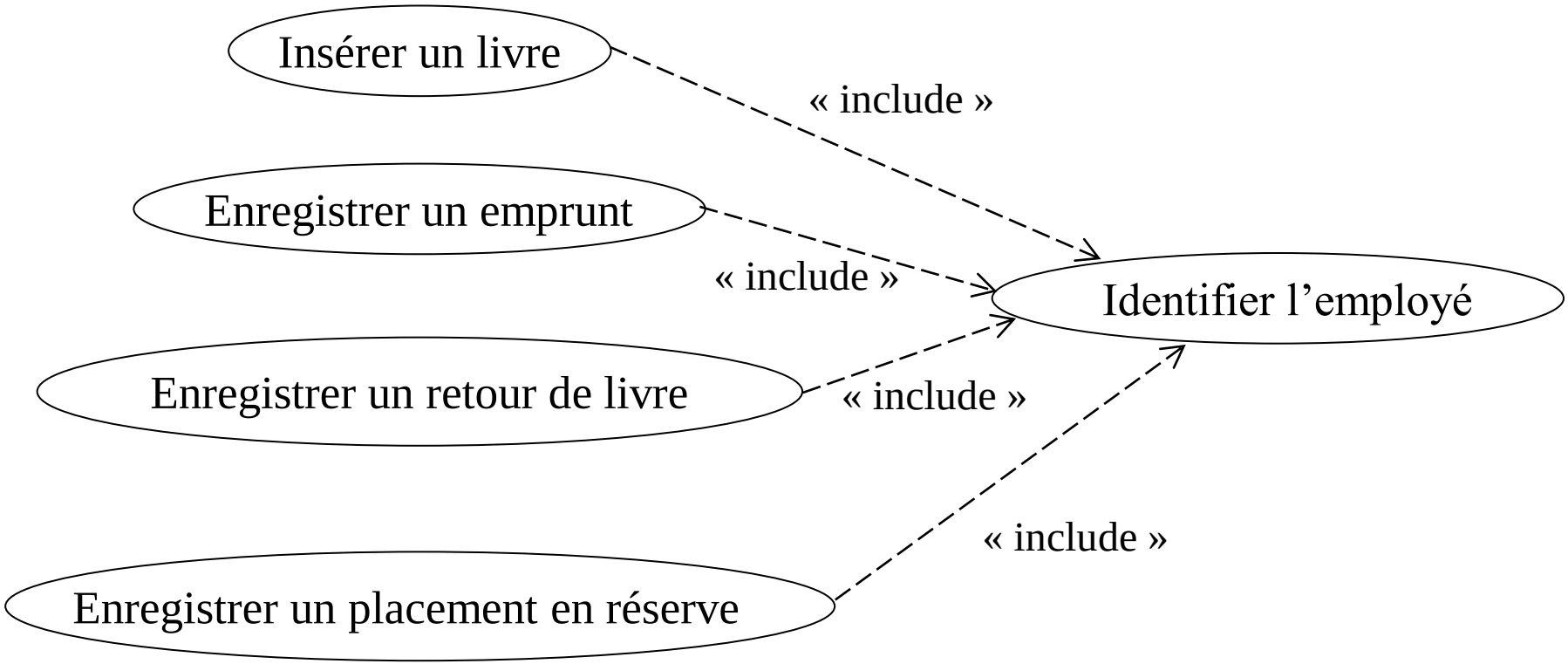
Modèle d'interaction
Concepts avancés (1/10)

Relation *include*

- Insertion d'un cas d'utilisation dans la séquence de comportements d'un autre cas d'utilisation
- Mise en commun de comportements communs à plusieurs cas d'utilisation
- Cas d'utilisation inclus :
  - Sous-routine
  - Unité de comportement significative pour les acteurs
  - Possibilité d'utiliser les cas d'utilisation inclus isolément

Modèle d'interaction
Concepts avancés (2/10)

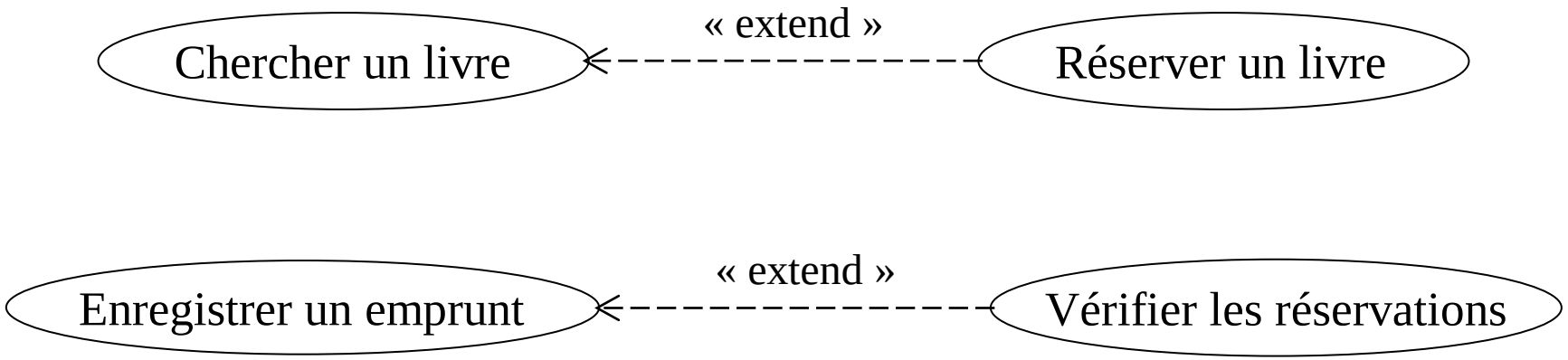
Relation *include*



Modèle d'interaction
Concepts avancés (3/10)

Relation *extend*

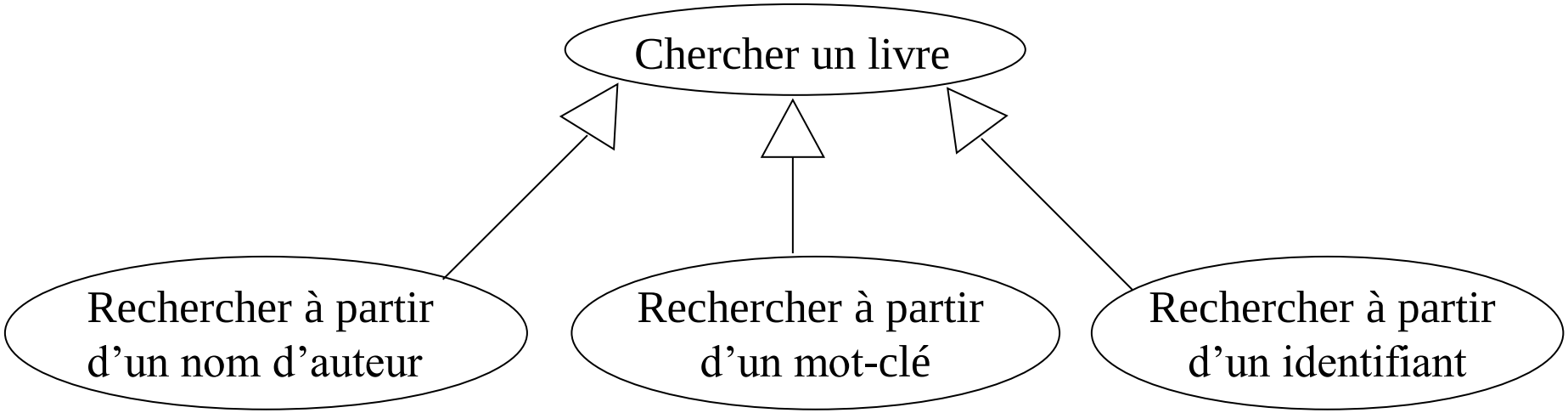
- Ajout d'un comportement incrémental à un cas d'utilisation
- Extension possible d'un cas d'utilisation de base
- Association d'une condition à la relation « *extend* »



Modèle d'interaction
Concepts avancés (4/10)

Généralisation des cas d'utilisation

- Représentation des variantes d'un cas d'utilisation
- Cas d'utilisation parent = représentation d'une séquence de comportements générale
- Cas d'utilisation enfant = insertion d'étapes supplémentaires ou affinage de certaines étapes du cas d'utilisation parent



Les cas d'utilisation enfant ajoutent des étapes de comportement devant apparaître à la position appropriée dans la séquence de comportement du parent

Modèle d'interaction
Concepts avancés (5/10)

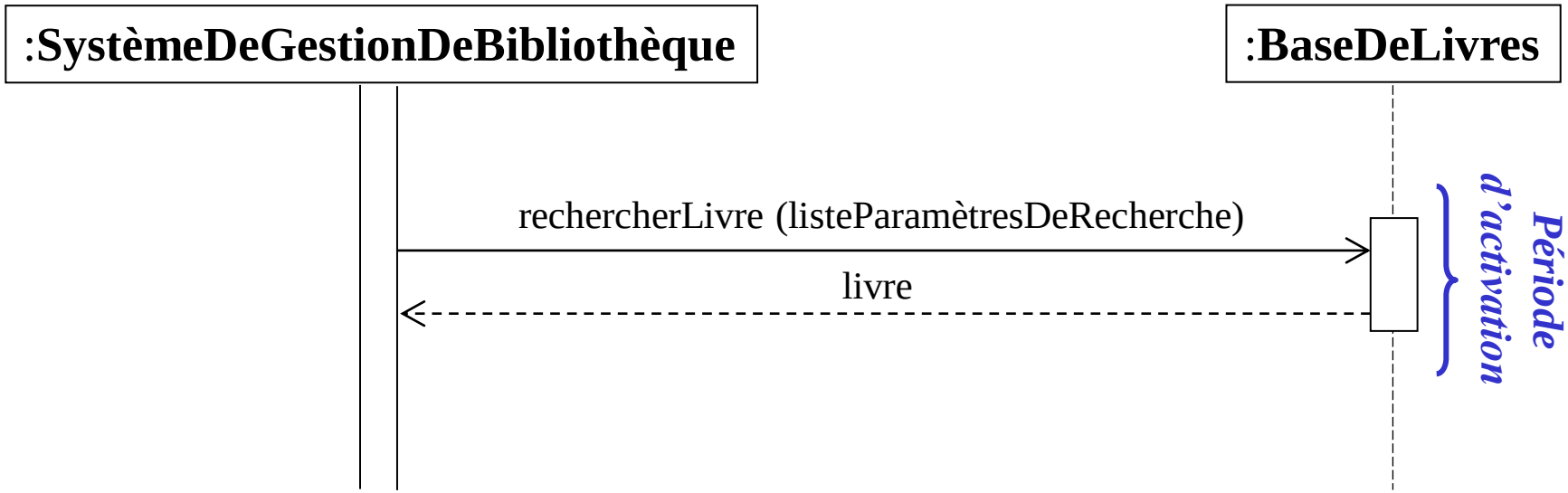
Conseils pratiques de [BR05] :

- Utiliser la généralisation de cas d'utilisation pour représenter un cas d'utilisation à plusieurs variantes mais pas pour partager un fragment de comportement
- Utiliser la relation « *include* » pour partager un fragment de comportement correspondant à une activité significative
- Utiliser la relation « *extend* » pour définir un cas d'utilisation ayant des caractéristiques optionnelles

Modèle d'interaction
Concepts avancés (6/10)

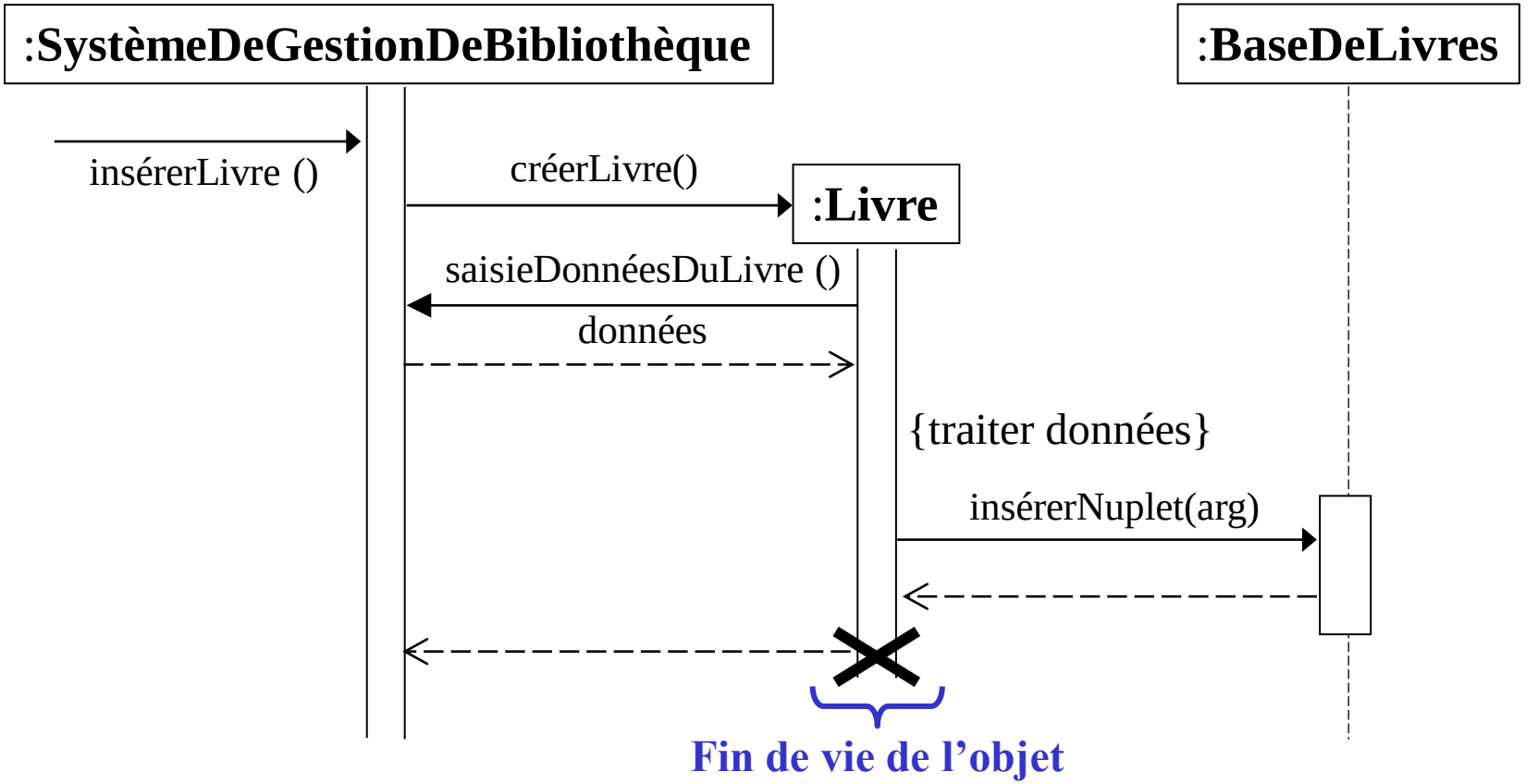
Diagramme de séquence avec objets passifs

- Notation dédiée permettant d'illustrer les appels de procédures
- Déclenchement d'un comportement d'un objet passif ⇒ activation de l'objet passif



Modèle d'interaction
Concepts avancés (7/10)

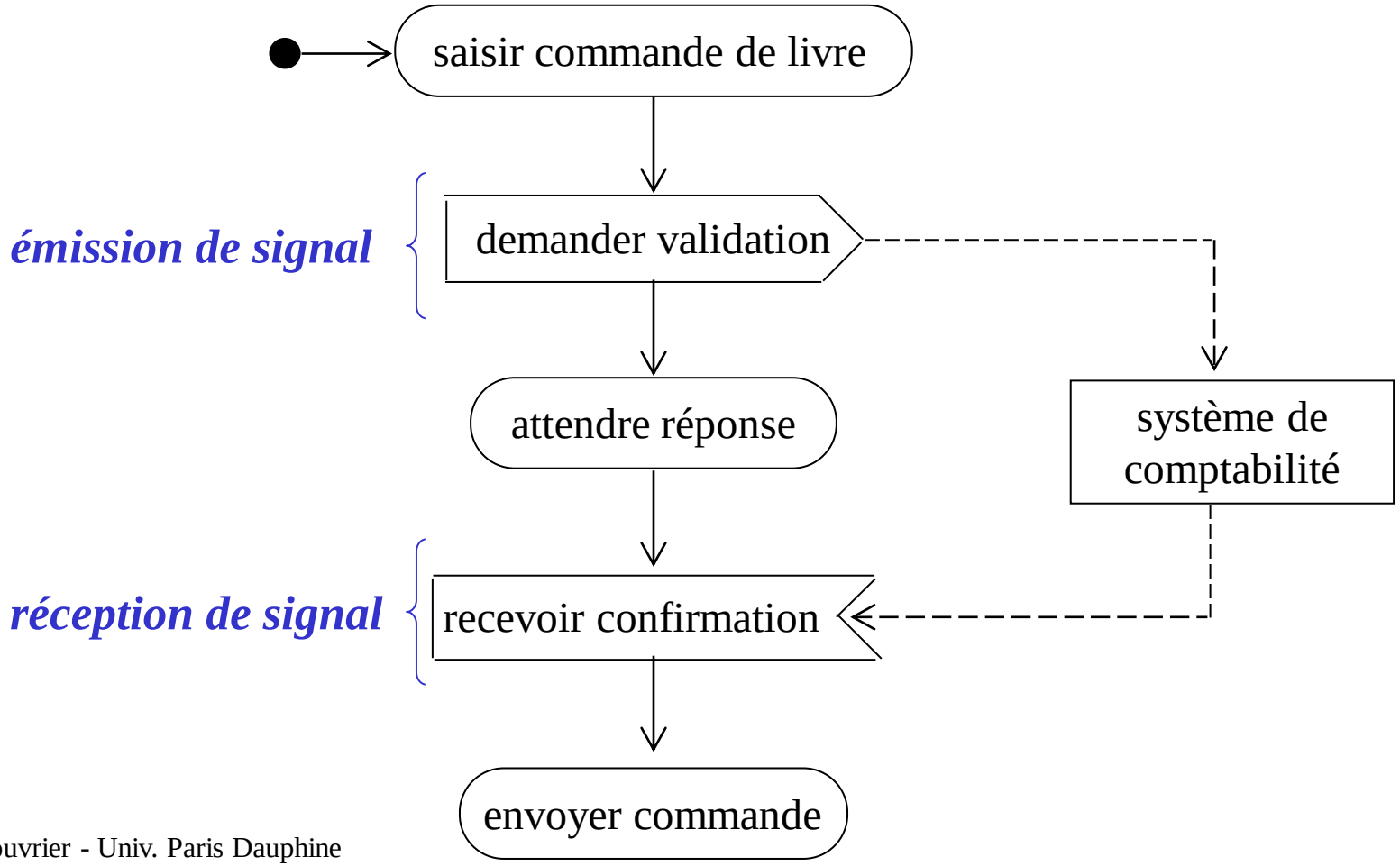
Diagramme de séquence avec objets temporaires



Ne représenter les détails d'implémentations que pour les diagrammes difficiles ou particulièrement importants [BR05]

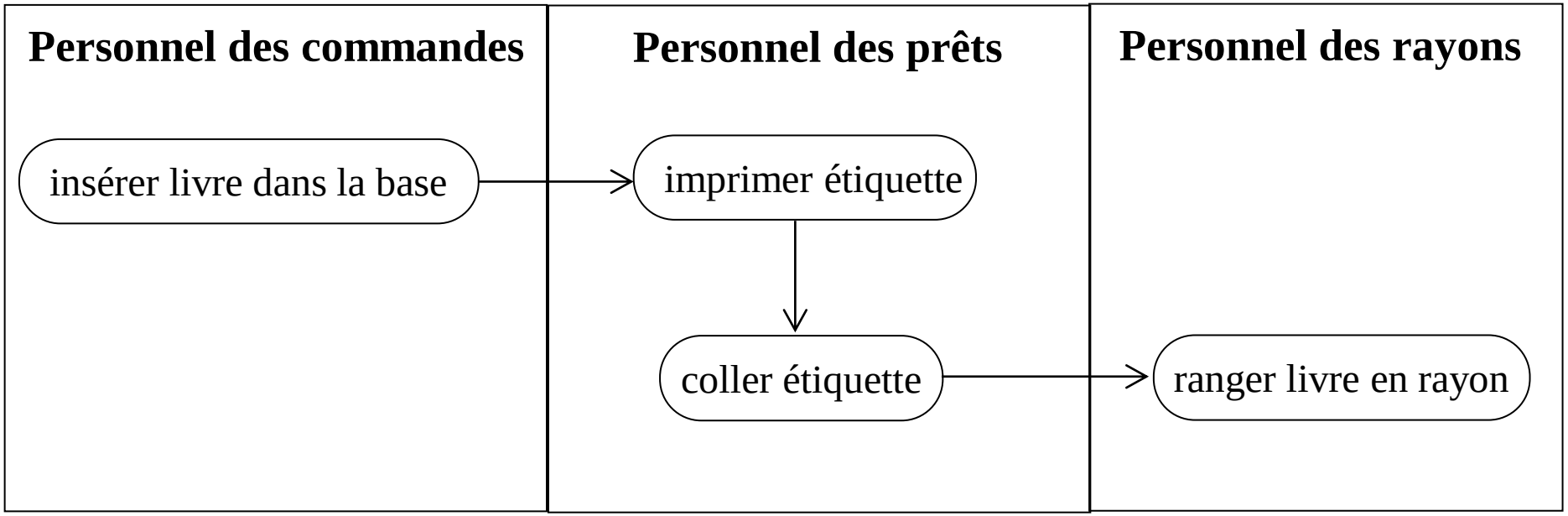
Modèle d'interaction
Concepts avancés (8/10)

Émission et réception de signaux dans les diagrammes d'activités



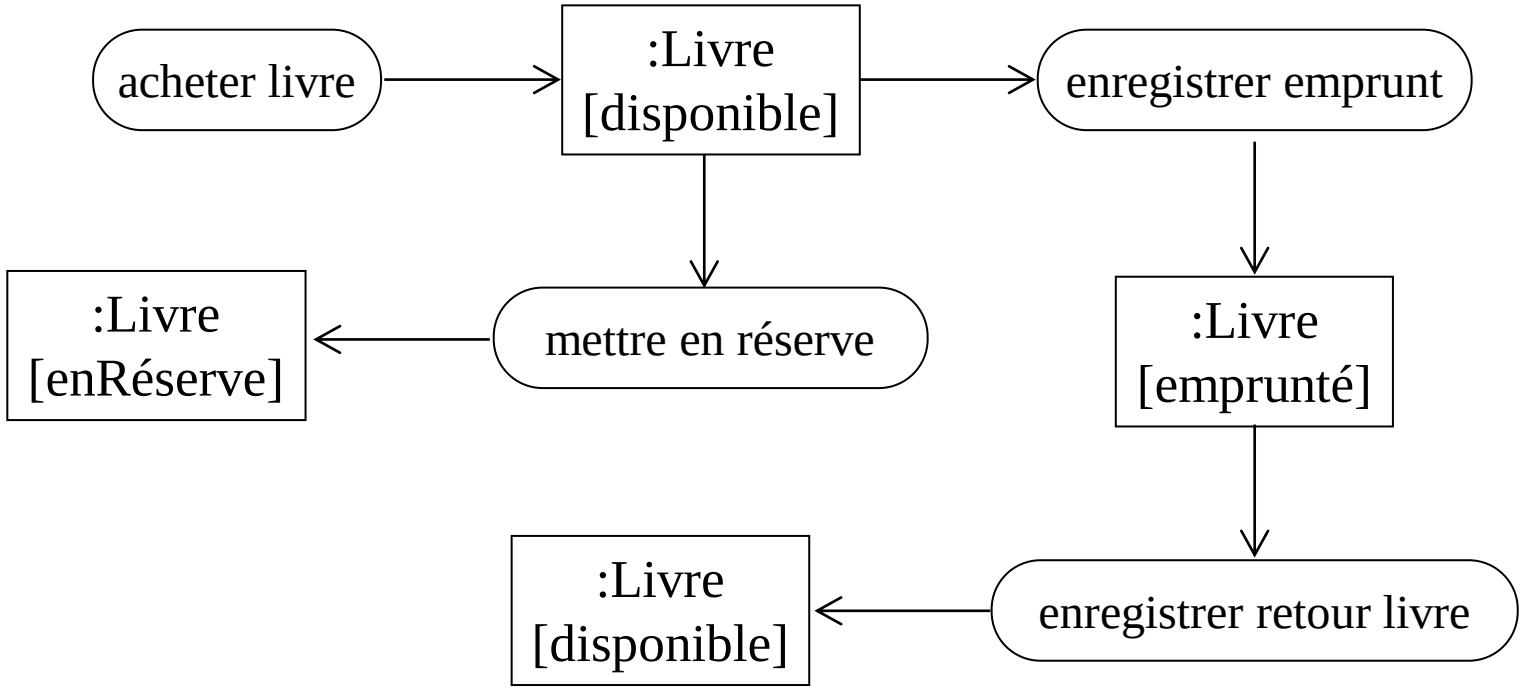
Modèle d'interaction
Concepts avancés (9/10)

Couloir d'activités : répartition des activités aux entités organisationnelles



Modèle d'interaction
Concepts avancés (10/10)

Flux d'objets : visualisation des relations entre une opération et les objets apparaissant en arguments ou en résultat de l'opération



Résumé (1/7)

Trois points de vue différents mais apparentés :

- **Modèle de classes** : description des objets d'un système et de leurs relations
- **Modèle d'états** : description du cycle de vie des objets
- **Modèle d'interactions** : description de la façon dont les objets interagissent

Résumé (2/7)

Modèle de classes

- Description de la structure statique des objets : identité, relations avec les autres objets, attributs et opérations
- Cadre d'insertion pour les modèles d'états et d'interactions
- Concepts importants :
  - **Classe** : ensemble d'objets similaires
  - **Association** : ensemble de liens similaires entre objets
  - **Généralisation** : structuration de la description des objets en les organisant en fonction de leurs différences et de leurs similarités

Résumé (3/7)

Modèle d'états

- Description des aspects temporels d'un objet
- **Événement** :
  - Marque d'un changement
  - Stimuli externe
- **État** :
  - Définition du contexte d'un événement
  - Valeurs d'un objet
- **Diagramme d'états** :
  - Description du comportement générique des objets d'une classe
  - Représentation des séquences d'états et d'événements pour une classe d'objets donnée

Résumé (4/7)

Modèle d'interactions

- Description de la façon dont les objets collaborent pour obtenir des résultats
- Complément du modèle d'états
- Différents niveaux d'abstraction pour modéliser les interactions :

Niveau de détails : - (haut) → + (bas)

- **Cas d'utilisation** : représentation des interactions du système avec les acteurs extérieurs
- **Diagrammes de séquence** : représentation des interactions entre objets et de leur succession dans le temps
- **Diagramme d'activités** : représentation des interactions avec mise en évidence du flux de contrôle entre les différentes étapes de traitement

Résumé (5/7)

Relations entre les 3 modèles

- Mêmes concepts (données, séquencement et opérations) mais avec accentuation différente
- Modèle de classes :
  - Description de la structure des données sur lesquelles les modèles d'états et d'interactions opèrent
  - Correspondance entre les opérations du modèle de classe et les événements, les conditions et les activités
- Modèle d'états :
  - Description de la structure du contrôle des objets
  - Représentation des décisions dépendant des valeurs des objets, entraînant les modifications de ces valeurs et les changements d'états
- Modèle d'interactions :
  - Concentration sur les échanges entre les objets
  - Vue globale du système

Résumé (6/7)

Relations entre les 3 modèles

- Généralisation de classes
  - Héritage des attributs, opérations, associations, diagrammes d'états de la super-classe par ses sous-classes
  - Possibilité d'utiliser ou redéfinir les propriétés de la super-classe dans les sous-classes
  - « *Le diagramme d'états d'une sous-classe doit être une addition orthogonale au diagramme d'états de la super-classe* » [BR05]
- Généralisation de signaux
  - Héritage d'attributs de signaux
  - Signal réel = feuille d'un arbre de généralisation de signaux
  - Signal entrant $\Rightarrow$ franchissement des transitions associées à ses signaux ancêtres
- Généralisation de cas d'utilisation
  - Cas d'utilisation parent = séquence générale de comportements
  - Cas d'utilisation enfant = spécialisation du cas parent par insertion d'étapes supplémentaires ou redéfinition d'étapes existantes

Résumé (7/7)

Relations entre les 3 modèles

- Agrégation d'objets :
  - Décomposition d'un assemblage en éléments orthogonaux ayant une interaction limitée
  - Diagramme d'états d'un agrégat = collection des diagrammes d'états de chacun de ses éléments constituants
- Agrégation d'états :
  - Possibilité de décomposer un état en états plus petits, chacun opérant indépendamment et possédant son propre diagramme
  - État de l'objet constitué d'un état de chaque sous-diagramme

Résumé des notations (1/16)

Modèles de classes

NomDeClasse

NomDObjet:NomDeClasse

NomDeClasse
nomAttribut1 [Mult.] : typeDeDonnées1 = Valeur parDéfaut1 nomAttribut2 [Mult.] : typeDeDonnées2 = Valeur parDéfaut2 …
nomOpération1 (listeArguments1) : TypeDuRésultat1 nomOpération2 (listeArguments2) : TypeDuRésultat2 …

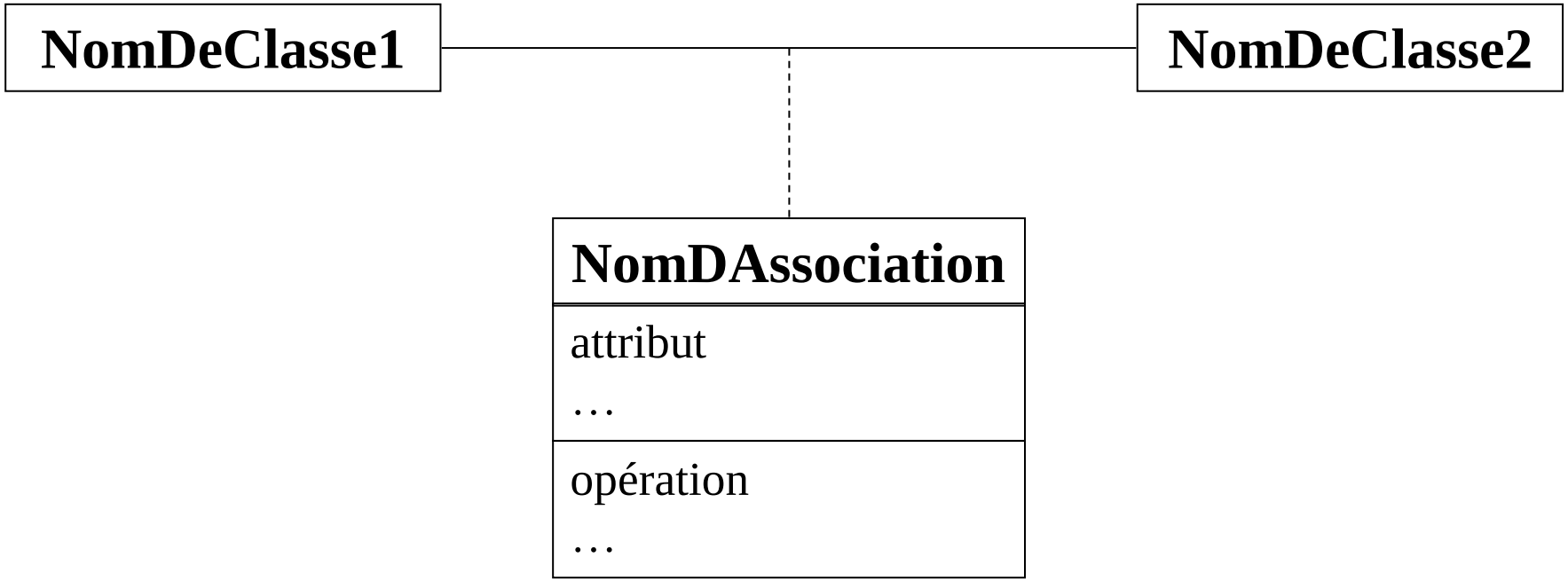
NomDObjet:NomDeClasse
nomAttribut1=valeur1 nomAttribut2=valeur2

Résumé des notations (2/16)

Modèles de classes

Résumé des notations (3/16)

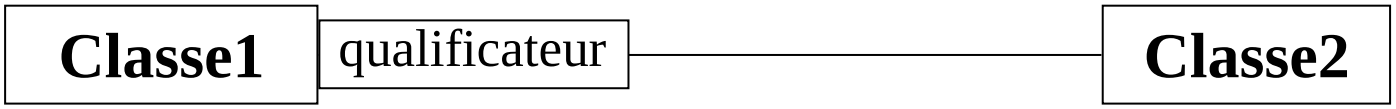
Modèles de classes



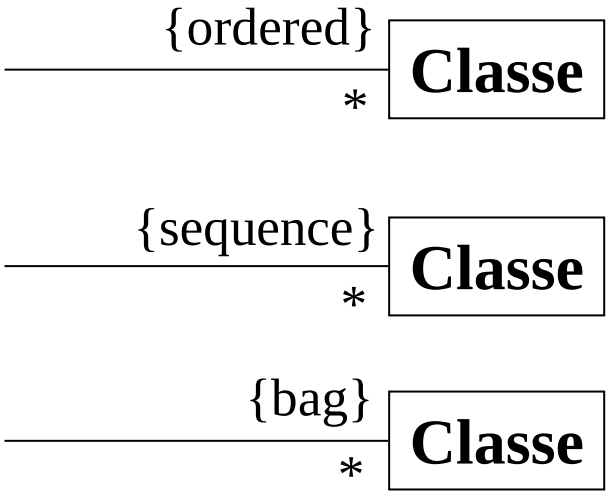
Classe-Association

Résumé des notations (4/16)

Modèles de classes



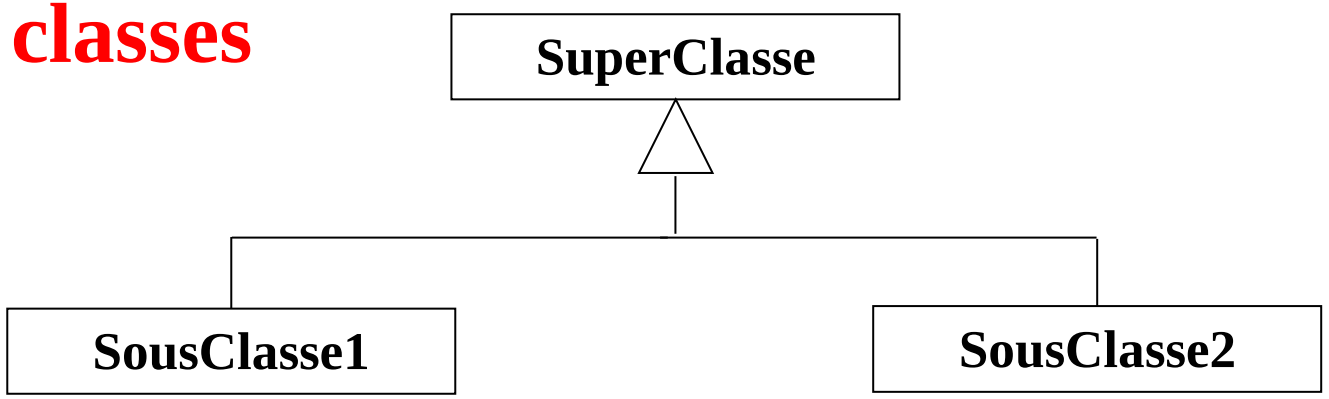
Association qualifiée



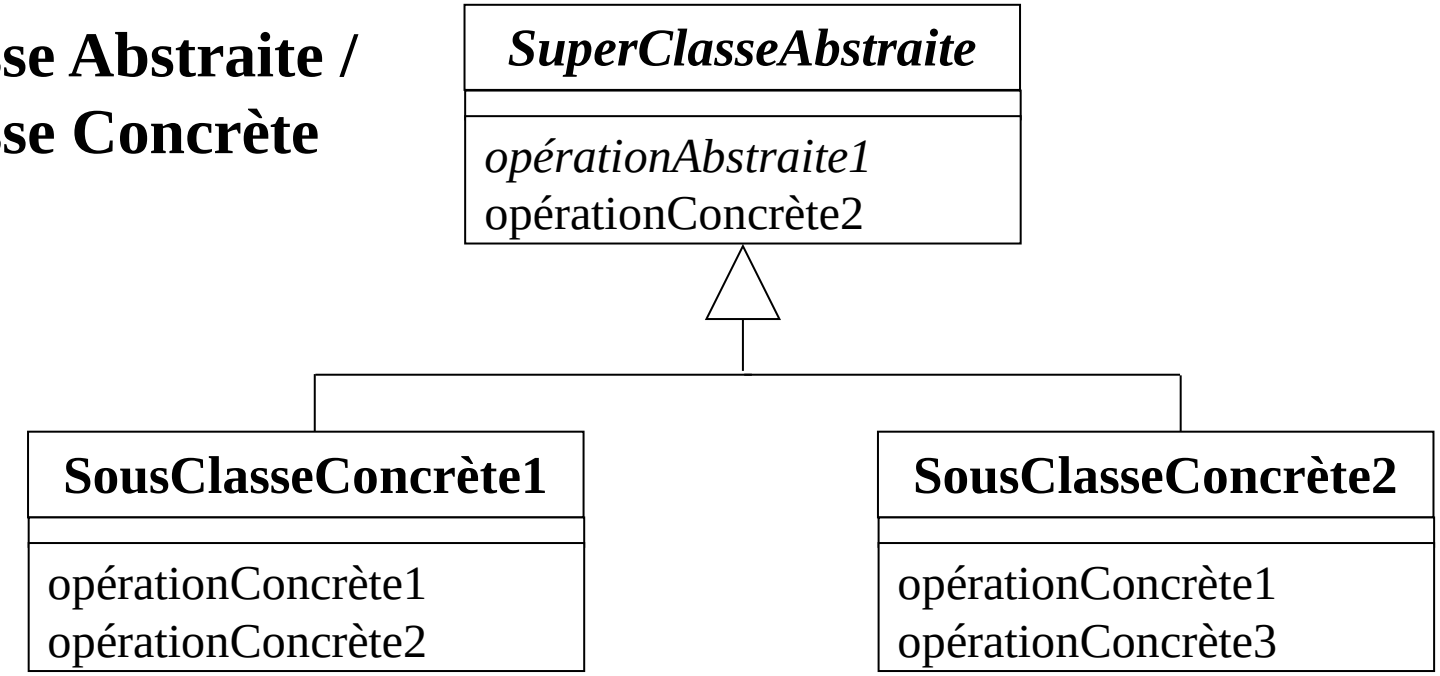
Résumé des notations (5/16)

Modèles de classes

Héritage



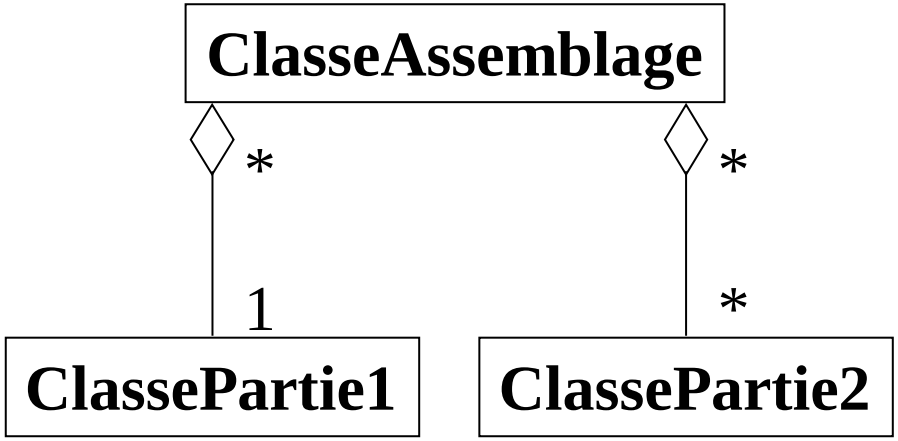
Classe Abstraite / Classe Concrète



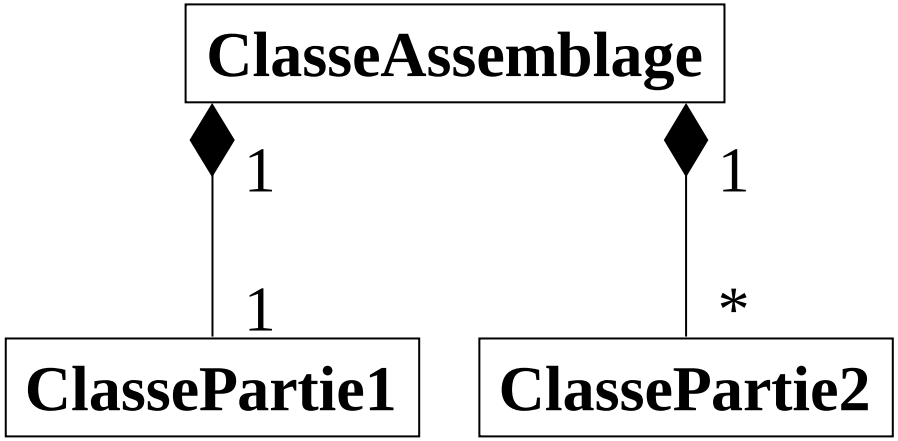
Résumé des notations (6/16)

Modèles de classes

Agrégation



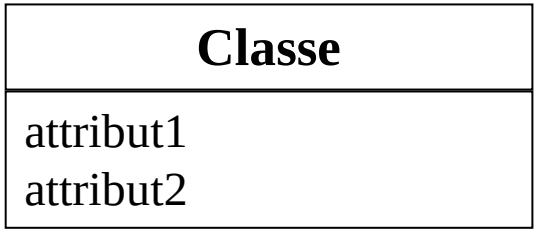
Composition



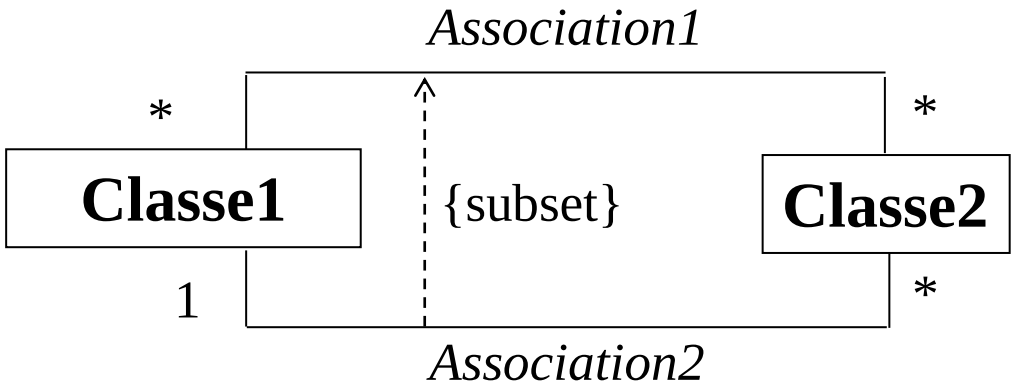
Résumé des notations (7/16)

Modèles de classes

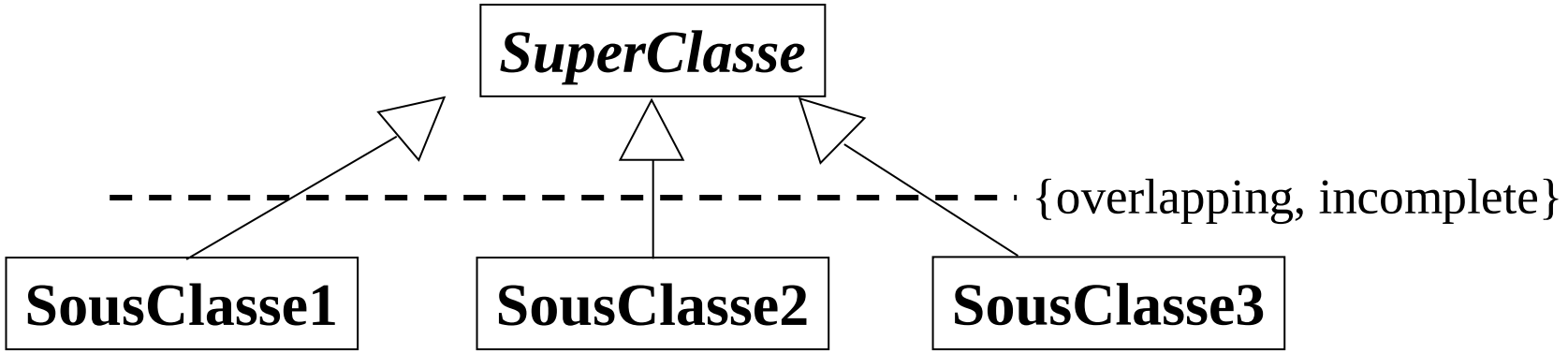
Contrainte sur les objets :



Contrainte sur les associations :

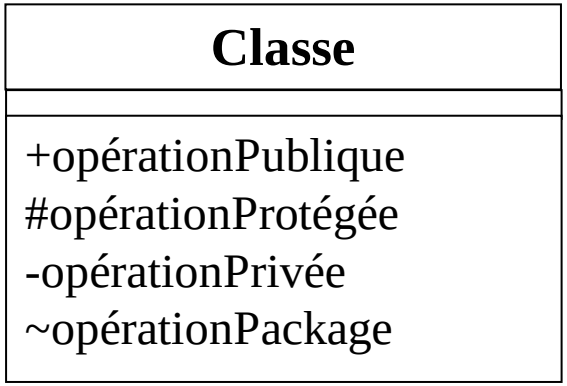


Contrainte sur les ensembles de généralisation :



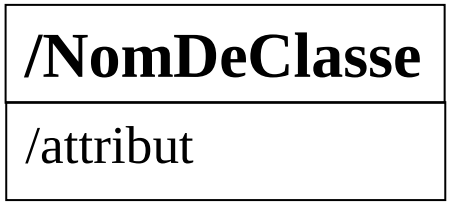
Résumé des notations (8/16)

Modèles de classes



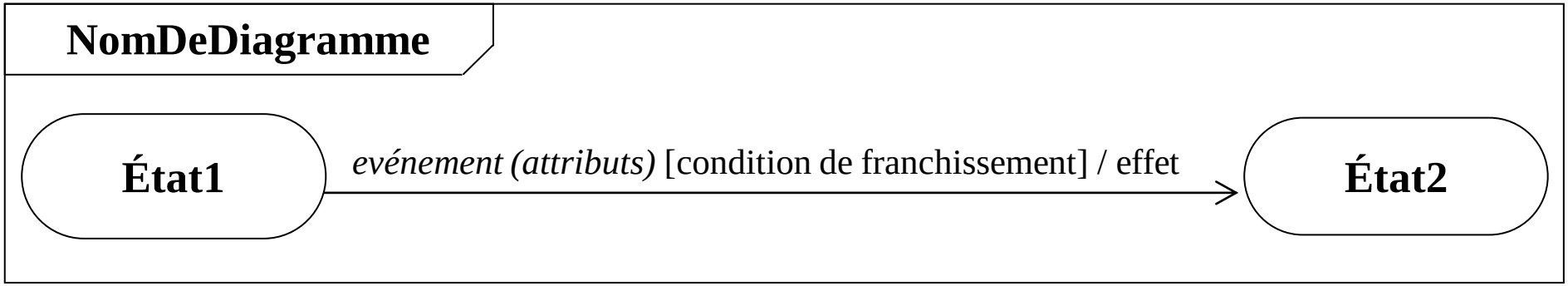
Visibilité

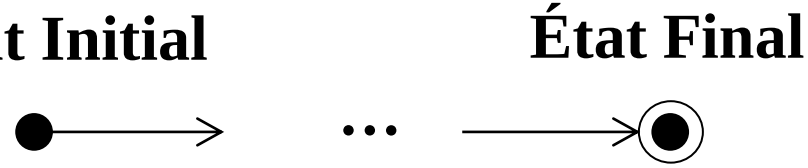
Élément dérivé :



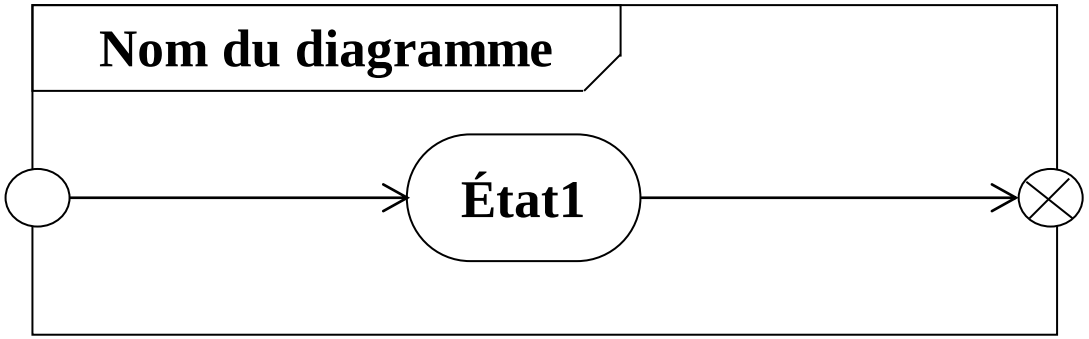
Résumé des notations (9/16)

Modèles d'états



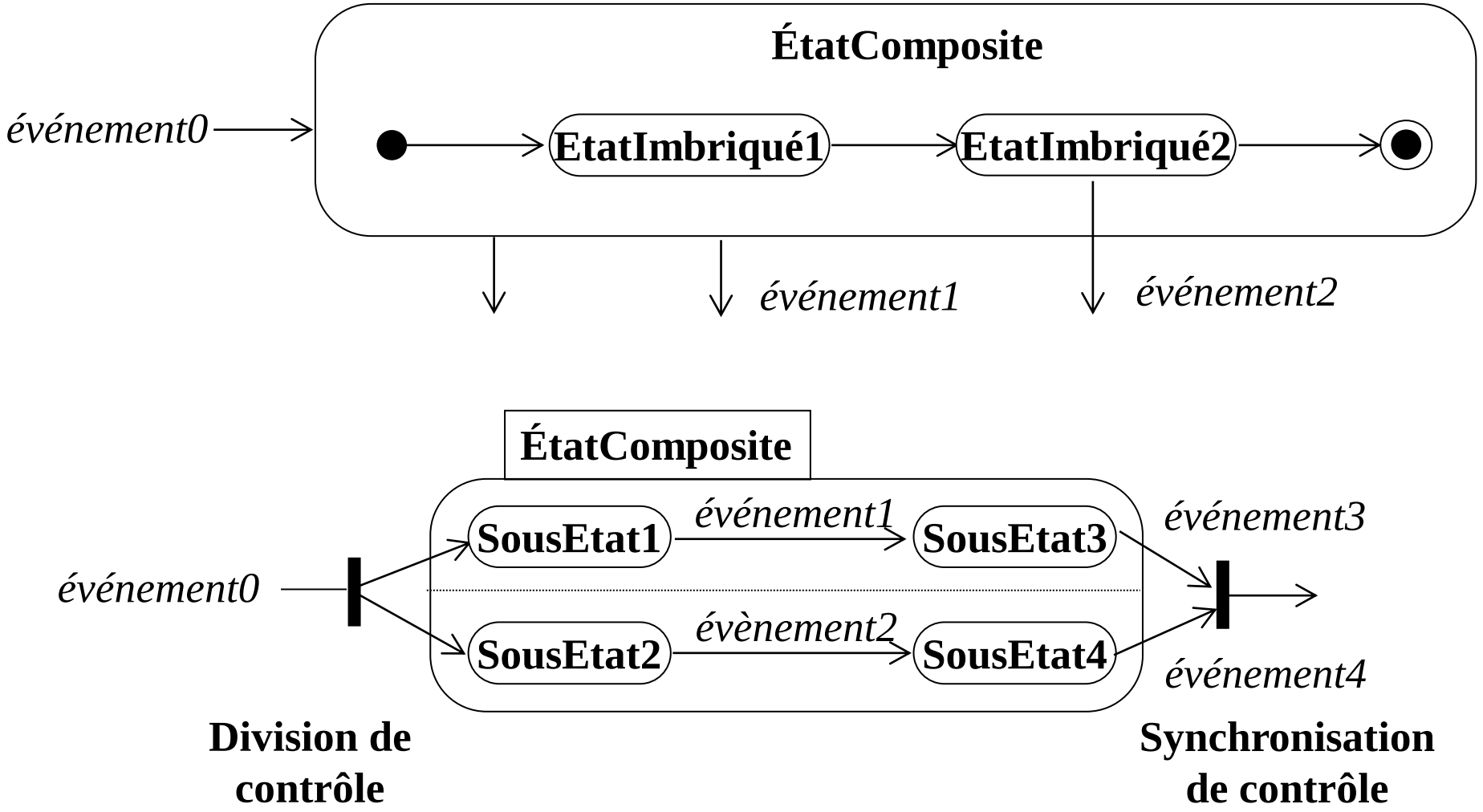


Points d'entrée et de sortie



Résumé des notations (10/16)

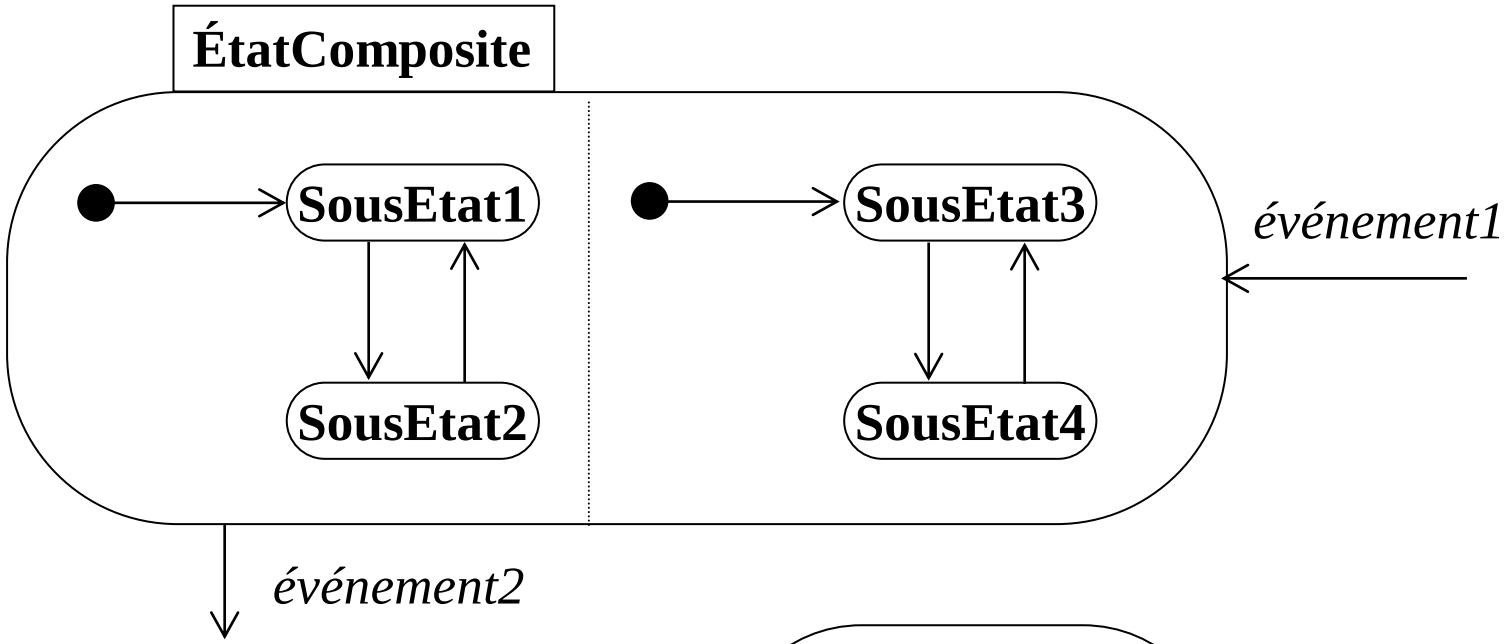
Modèles d'états



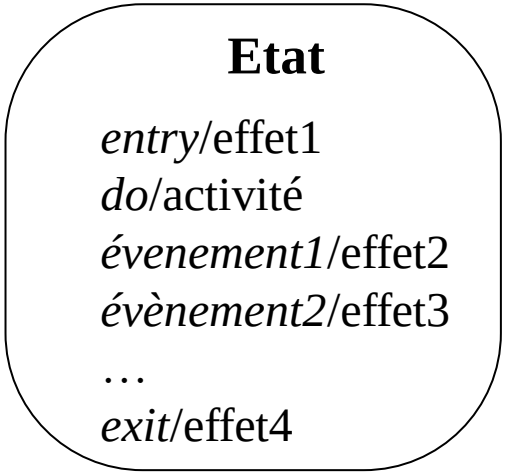
Résumé des notations (11/16)

Modèles d'états

Concurrence à l'intérieur d'un objet



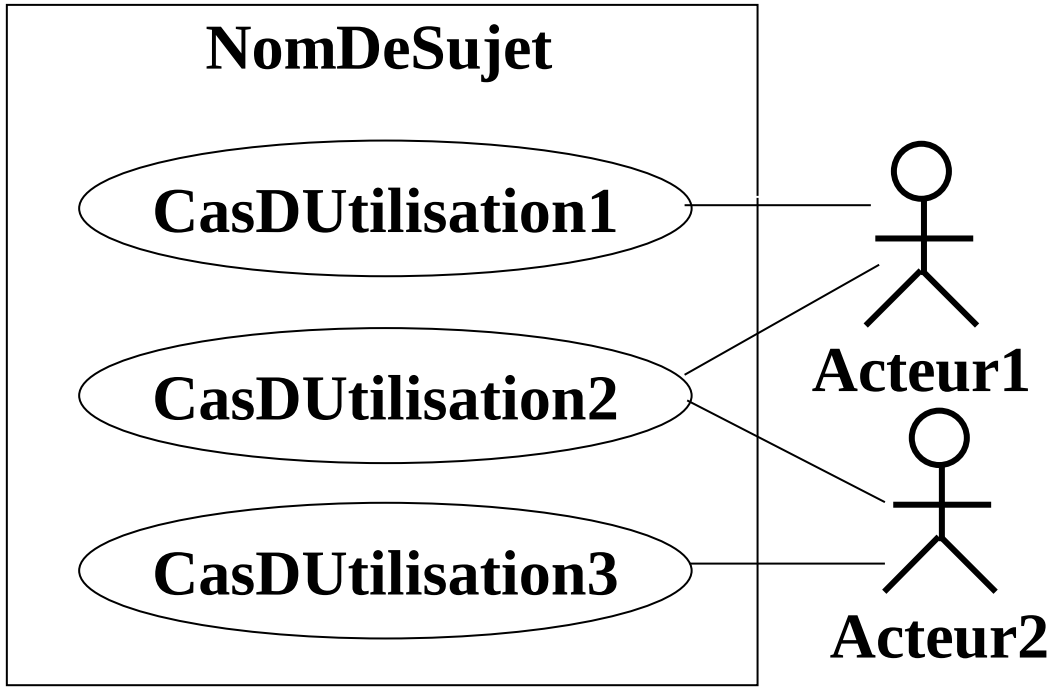
Activités internes à un état



Résumé des notations (12/16)

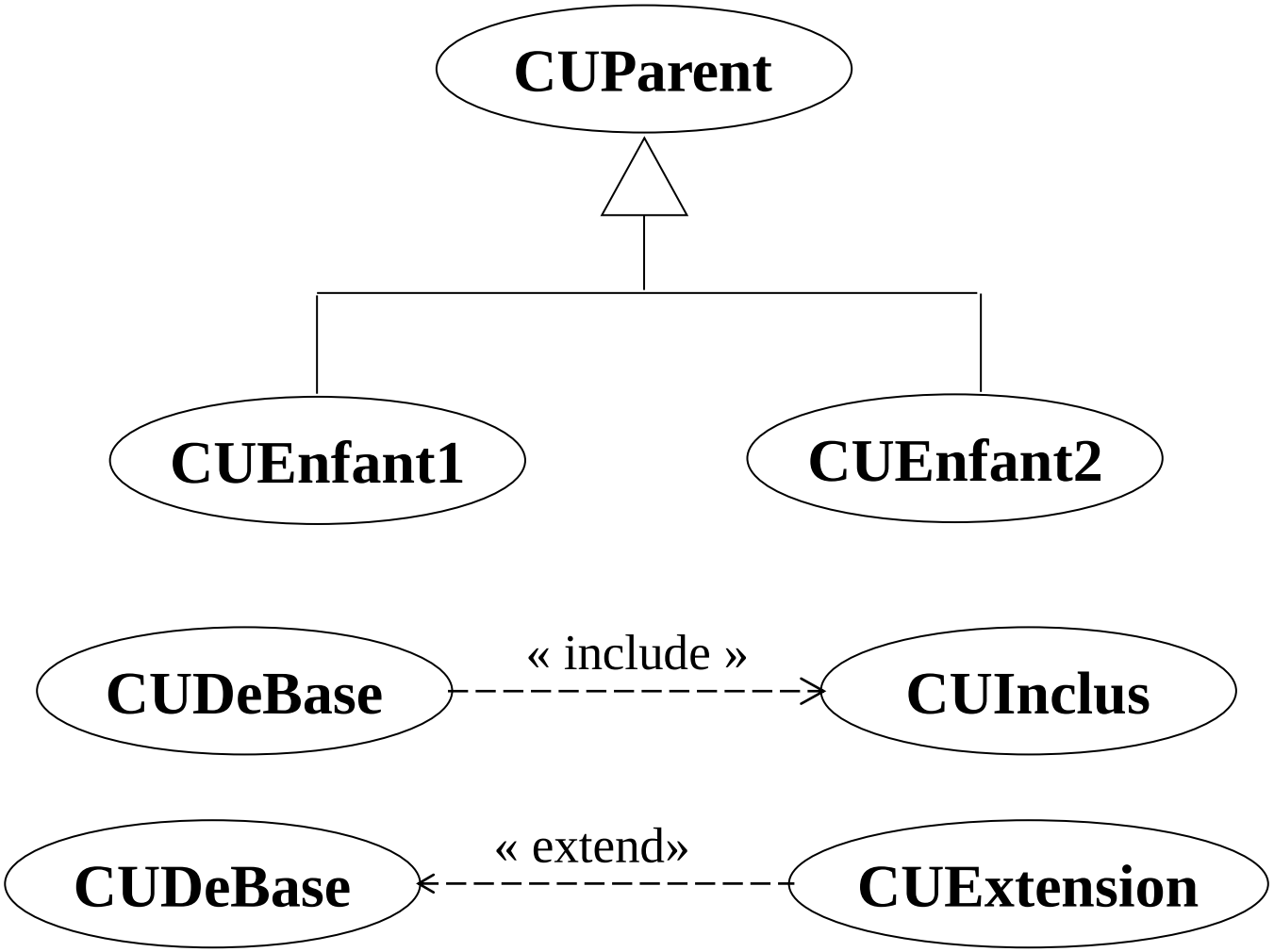
Modèles d'interactions

Diagramme de cas d'utilisation



Résumé des notations (13/16)

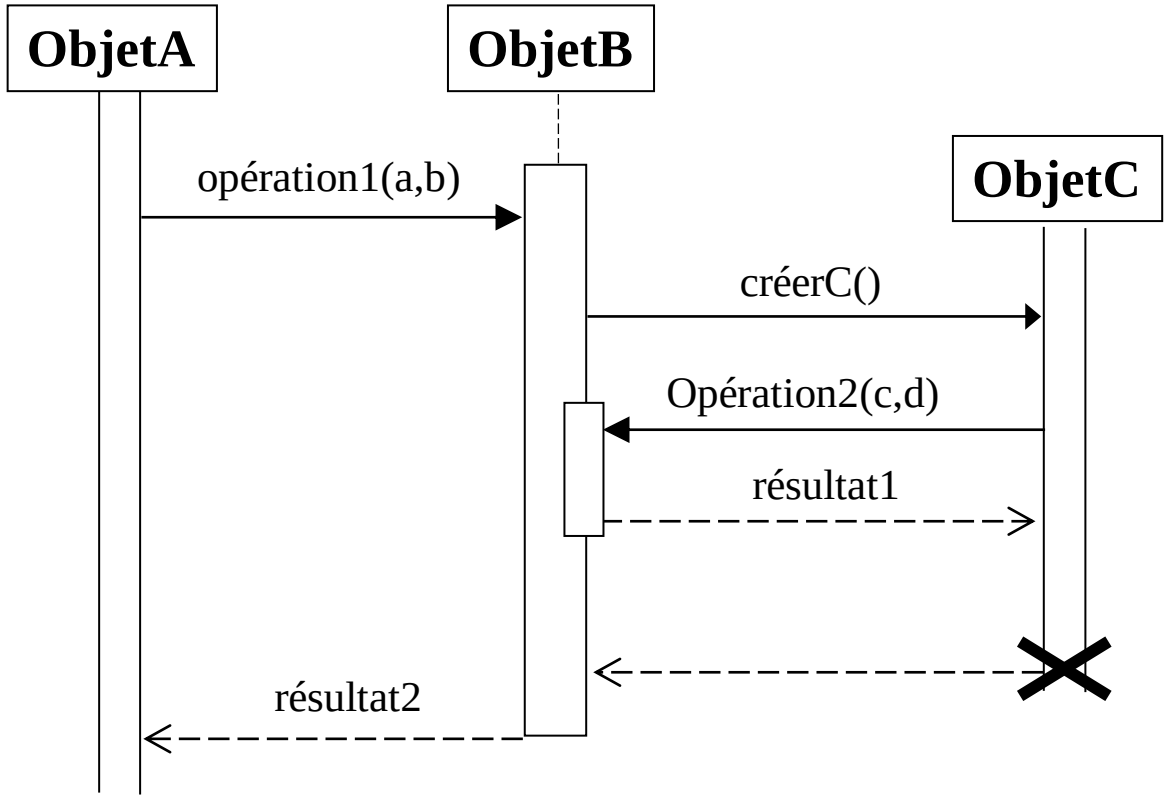
Modèles d'interactions



Résumé des notations (14/16)

Modèles d'interactions

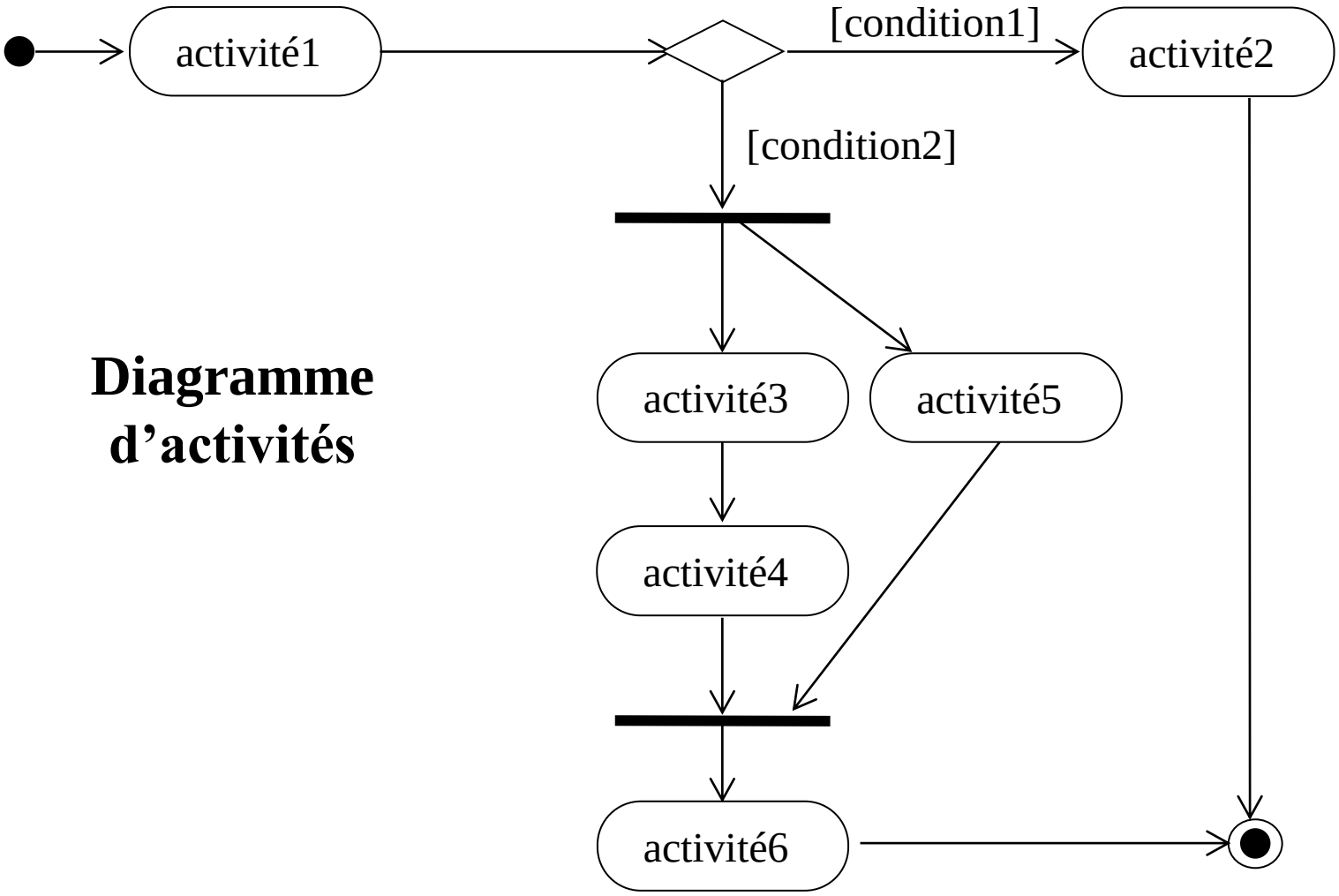
Diagramme de séquence



Résumé des notations (15/16)

Modèles d'interactions

Diagramme d'activités



Résumé des notations (16/16)

Modèles d'interactions

Diagramme d'activités avec couloirs d'activités

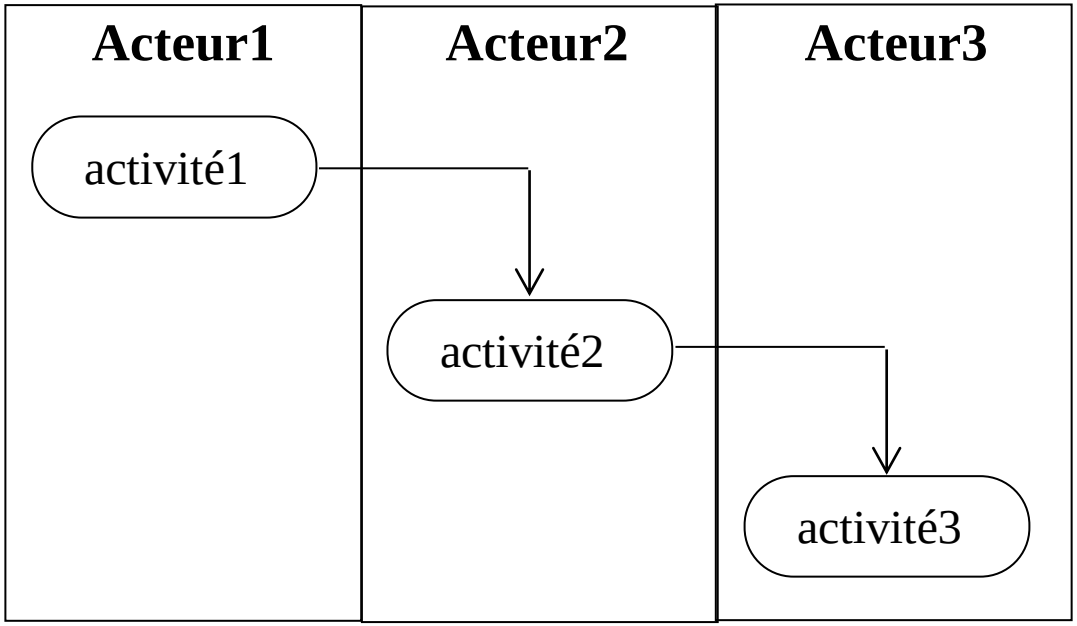


Diagramme d'activités avec flux d'objets