



Modélisation des systèmes informatiques

**Responsable : Dr. DIAW
DSTI 2**



Plan

- Organisation et Système d'information
- Langages de modélisation
 - MERISE
 - UML
- Méthode d'analyse et de conception
- Initiation à l'utilisation d'un AGL



Organisation et Système d'information



Notion de Système

- Selon le Robert, un système est un dispositif formé par la réunion d'éléments analogues. En d'autres termes un système est une combinaison de parties qui se coordonnent pour concourir à un résultat, de manière à former un ensemble.
- Tout système fonctionne en transformant des flux d'entrée en flux de sortie selon des processus plus ou moins complexes.



Fonctionnement d'une entreprise

- Une entreprise peut être vue comme une entité fournissant des produits ou services à des clients, en s'appuyant sur les produits ou services de partenaires
- Fonctions de l'entreprise
 - Fonction de réalisation (production, gestion stock et approvisionnement)
 - Fonctions de management (stratégie de gestion de l'entreprise)
 - Fonctions de support (appui aux fonctions de réalisation pour le bon fonctionnement de l'entreprise)

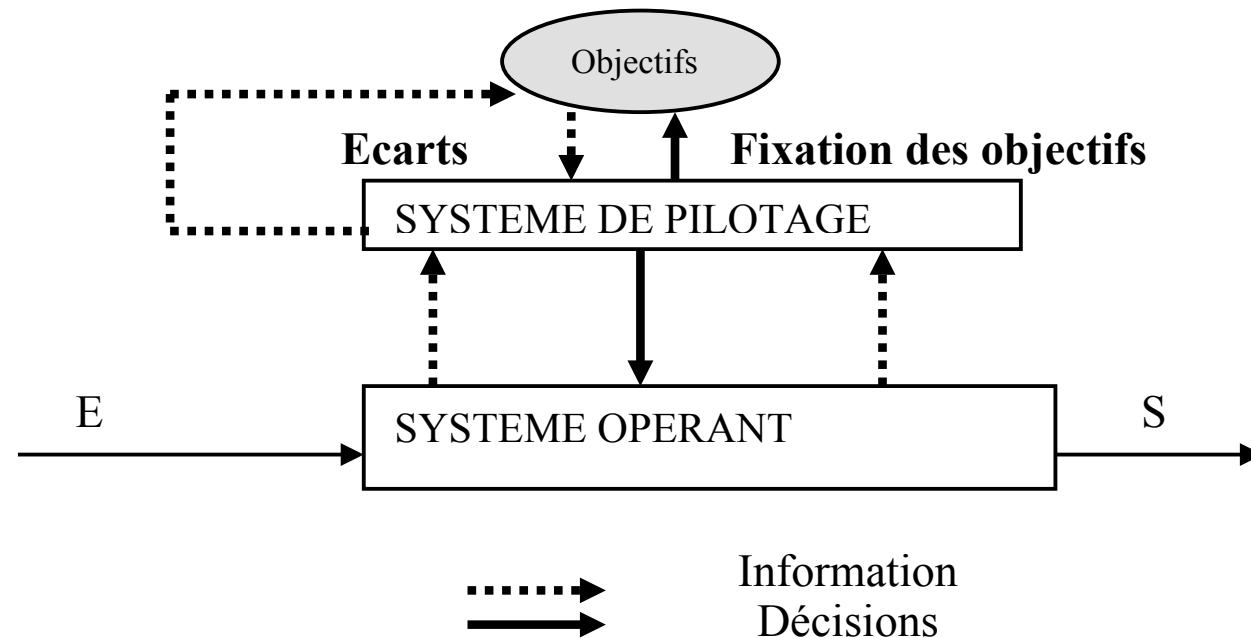


Système opérant et système de pilotage

- Un système physique ou système opérant est l'ensemble des moyens humains, matériels, organisationnels qui exécutent les ordres du système de pilotage.
- Par ailleurs un système de pilotage procède à la régulation et au contrôle du système opérant. Il définit les missions et les objectifs, organise l'emploi des moyens et coordonne l'exécution des travaux.
 - Ce système se compose par exemple de la direction financière, de la direction commerciale, de la direction des ressources humaines, etc.

Systeme operant et systeme de pilotage

- Le système de pilotage reçoit du système opérant des informations sur l'état du système qui lui permettent de mesurer l'écart avec les objectifs
- Il réagit par des décisions sur le processus du système opérant par régulation des flux (fixation des cadences de production, décision de lancer un nouveau produit ou de modifier le prix de vente d'un article).





Système d'Information d'une entreprise

- Complexité grandissante des entreprises à cause des nombreux flux d'informations qui y transitent
- Nécessité de mettre en place un dispositif maîtrisé de ces flux pour faire face à la concurrence
- Nécessité de mettre en place un système pour gérer ces flux avec un temps de réponse bref: d'où la notion de SI
- **Un système d'information** (noté SI) représente l'ensemble des éléments participant à la gestion, à la saisie, au stockage, au traitement, au transport et à la diffusion de l'information au sein d'une entreprise



Fonctions du SI

- Saisie des informations
- Stockage des informations
- Traitement des informations
- Transmission et diffusion des informations
- Restitution des informations

SI: Interface entre les Systèmes opérant et de pilotage

- Le SI peut recevoir du Système de pilotage des décisions destinées à son propre pilotage.
- Le SI peut émettre vers le système opérant des informations-interactions, c'est-à-dire qu'il peut réagir sur le système opérant .

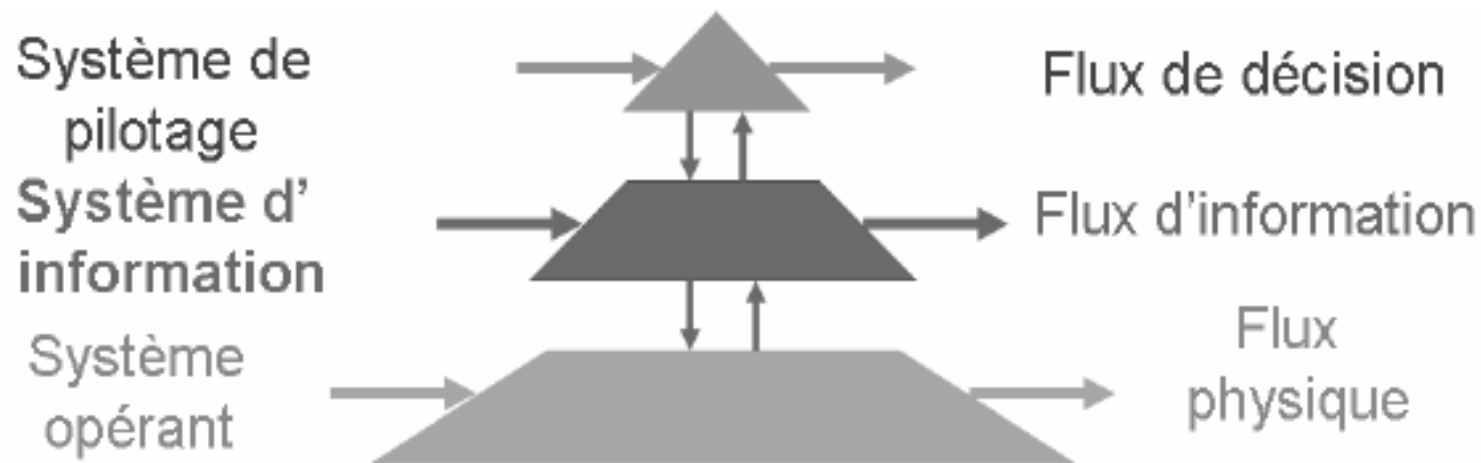


illustration d'un SI d'une entreprise de vente de produits

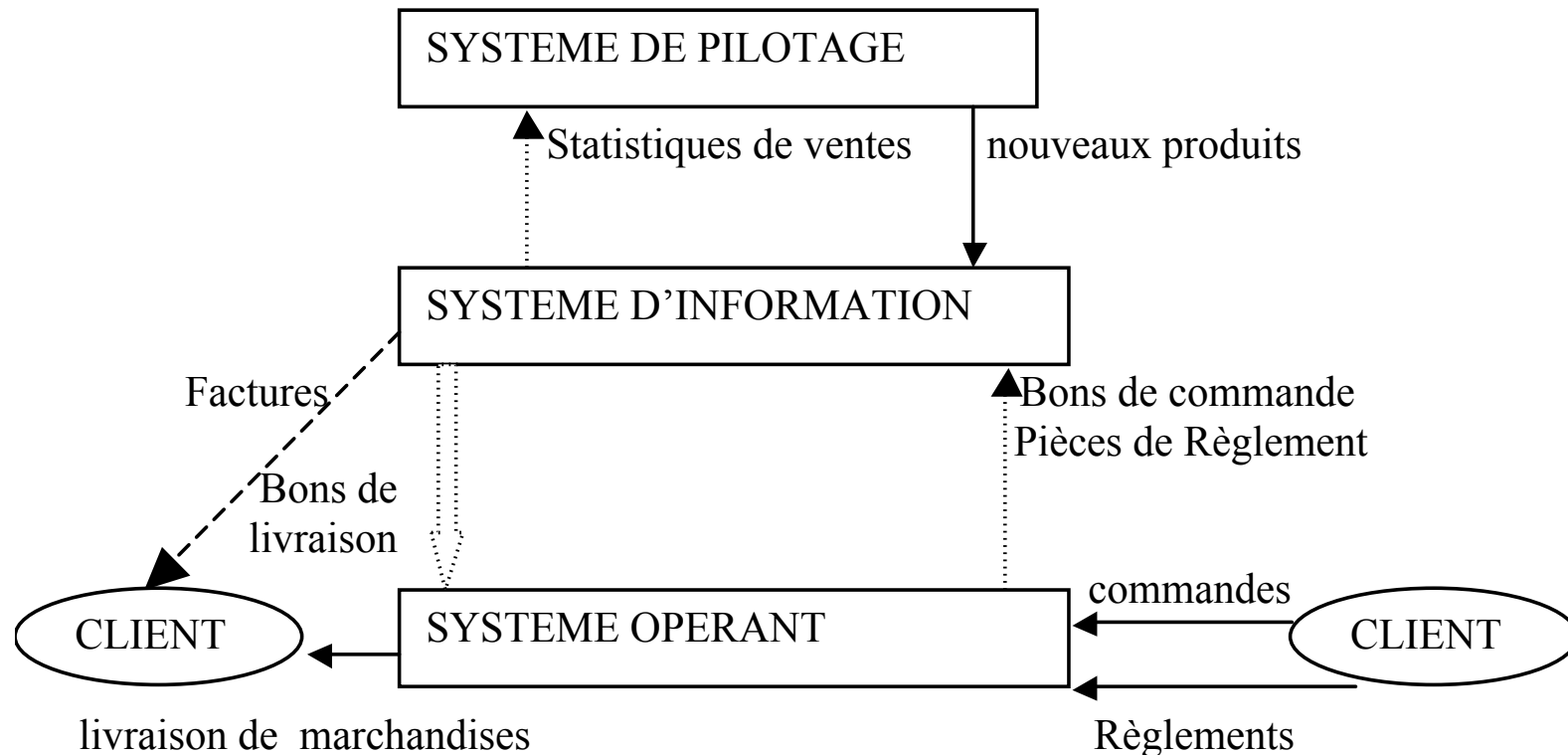




illustration d'un SI

- **Flux physiques:** (bons de commande, bon de livraison, factures, pièces de règlement.)
- **Flux d'information:** Le numéro d'une commande, sa date, la référence d'un produit, sa désignation et son prix, un numéro de client, son nom , etc. Information plus synthétique; nombre de commandes par jour, top-10 des produits les plus vendus
- **Flux de décision:** Le prix d'un produit, le seuil d'approvisionnement du stock
- Le SI de cette entreprise est en liaison d'une part avec un environnement interne (système opérant et système de pilotage) et d'autre part avec un environnement externe (clients, fournisseurs, etc.)



Langages de modélisation



Définition

- Un modèle est une représentation simplifiée d'une réalité (système réel). Le modèle doit répondre aux questions que les utilisateurs se posent sur le système qu'il représente.
- Exemple de modèles: Une carte routière , un schéma, un plan etc.
- Un modèle sert à :
 - Communiquer : vérifier que l'analyse est bien comprise par les utilisateurs (phase analyse)
 - Préparer la réalisation du futur système grâce à un modèle de la solution (phase conception)
- Un langage de modélisation est un langage qui permet de décrire des modèles Exemple: UML; MERISE, etc.



Aperçu sur la méthode MERISE



Présentation générale de MERISE

- La méthode MERISE
 - MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques.
 - La séparation des données et des traitements assure une longévité au modèle.
 - Elle date de 1978-1979, et fait suite à une consultation nationale lancée en 1977 par le ministère de l'Industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information.
 - Version horticole: MERISE provient d'un arbre fruitier le merisier qui sert de porte greffes au cerisier: (La méthode MERISE permet de greffer l'informatique dans une organisation)
 - Version humoristique : Méthode Eprouvée pour Retarder Indéfiniment les Sortie des Etudes



Présentation générale de MERISE

•Analyse d' un SI

- On s' intéresse généralement à un domaine de l' entreprise (ventes, production, logistique, finances, etc.)
- On prend en compte les besoins des utilisateurs
- On définit le problème à résoudre (fonctionnalités et qualités attendues)

•Conception d' un SI (on propose une solution informatique)

- Structuration des données, organisation des traitements
- Définition des postes de travail
- Choix techniques (matériel, langage de programmation , logiciel de gestion des données (SGBD))



Niveaux de MERISE

NIVEAUX	DONNEES	TRAITEMENTS
CONCEPTUEL	MCD : sémantique des données (modèle entité/association)	MCT quoi ? (fonctions du SI)
ORGANISATIONNEL (ou LOGIQUE)	MLD : organisation des données	MOT qui fait quoi, ou, quand ?
PHYSIQUE	MPD implantation des données (SGF, SGBD)	MPT comment on fait ?

MCD : Modèle conceptuel des données
MLD : Modèle logique (organisationnel) des données
MPD : Modèle physique des données
MCT : Modèle conceptuel des traitements
MOT : Modèle organisationnel des traitements
MPT : Modèle physique des traitements




Les questions abordées à chaque niveau

NIVEAU	CHOIX	CONTENU
CONCEPTUEL	GESTION METIER	données traitées, règles de gestion, enchaînements des traitements, ...
ORGANISATIONNEL LOGIQUE	ORGANISATION	partage homme/machine, interactif/différé, organisation des données et traitements, distribution, ...
PHYSIQUE	TECHNIQUE	programmes, écrans, états, organisation physique des données, matériel, réseau, ...



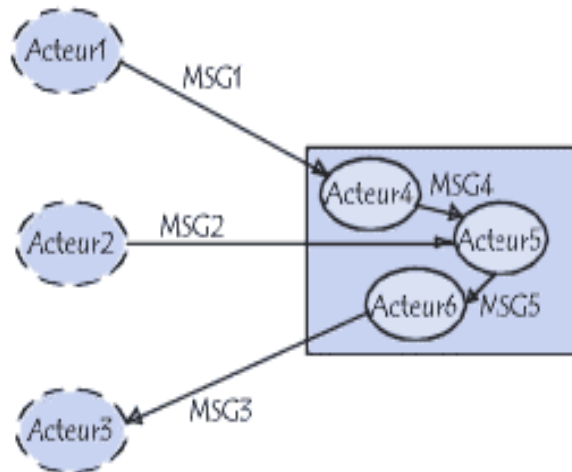
Modèle Conceptuel de la communication



Modèle Conceptuel de la Communication (MCC)

- Ce modèle appelé aussi Diagramme des flux donne une vue d'ensemble de la circulation des informations (les flux) entre des acteurs internes ou externes qui participent à un domaine étudié
- Un acteur est un système actif intervenant dans le domaine d'étude au moyen des flux
- Un acteur peut représenter:
 - Un intervenant extérieur à l'entreprise (clients, fournisseurs, etc.)
 - Un domaine de l'entreprise (service du personnel, comptabilité, etc.)
- Un flux symbolise un échange entre deux acteurs du SI étudié. Il est représenté par une flèche, porte un nom, et peut pour des soucis de lisibilité chronologique être numéroté

Digramme conceptuel de flux ou MCC



Ce diagramme (appelé aussi **modèle conceptuel de la communication**) permet de compléter le diagramme de contexte en décomposant l'organisation en une série d'acteurs internes. Dans ce diagramme la représentation standard est la suivante:

- Les acteurs internes sont représentés par des ellipses,
- Les messages internes sont représentés par des flèches



Exemple de Base

Le champ d'étude concerne la gestion des préinscriptions à l'ESP

Lorsqu'un bachelier souhaite être orienté dans une filière de l'ESP, il dépose un dossier de préinscription. Le service de gestion de la scolarité enregistre le dossier du bachelier.

Deux cas peuvent se produire :

- soit le dossier est complet et dans ce cas la préinscription est acceptée
- soit le dossier est incomplet et il est mis en attente en attendant de recevoir les pièces manquantes.

Dans ce dernier cas, le bachelier dispose d'un délai de 8 jours pour compléter son dossier. Passé ce délai, le dossier sera supprimé.

Dans le cas où le complément du dossier arrive après le délai des 8 jours, une notification de refus est envoyée au bachelier pour cause de réception tardive des pièces manquantes.



Modèle Conceptuel des Traitements



Définition

- Le (MCT) décrit les traitements et plus précisément toutes les activités découlant des échanges entre le domaine étudié et le monde extérieur.
- Il exprime donc ce que fait le domaine sans se poser le problème de savoir qui le fait, quand et comment.



1. Les concepts de base

- L'acteur

- Un acteur est une personne morale ou physique capable d'émettre ou de recevoir des informations. Par exemple, le bachelier qui souhaite faire une préinscription

- On distingue deux types d'acteurs :

- les acteurs internes qui appartiennent au système d'information étudié.
- les acteurs externes qui n'appartiennent pas au système d'information mais qui sont l'origine ou la destination de flux d'informations reçus ou émanant du système



Les concepts de base

•L' événement

- L' événement matérialise par un fait, qui en se produisant, doit déclencher une réaction du SI. Plus précisément cette notion recouvre deux aspects :
 - le fait qui survient et sa perception. La décision d'un bachelier de s' inscrire à l'ESP illustre cet aspect.
 - le compte rendu de cette perception faite auprès du système d' information.
- Ainsi, dans le cas de la préinscription, c' est l'enregistrement du dossier qui constitue le compte rendu du souhait du bachelier..
- l' événement (au sens du MCT) est porteur d' informations qui doivent être obligatoirement digérées par le système d' information sans quoi il ne répondrait pas à ses objectifs



Les concepts de base

- **L' événement**
- Parmi les événements, on distingue les événements externes et les événements internes :
 - les événements déclencheurs externes sont des événements émis par un acteur externe. Par exemple le dépôt d' un dossier de préinscription est un événement, externe déclenché par un bachelier.
 - les événements internes sont des événements qui surviennent lorsqu'une opération se termine. Cela peut être par exemple l' acceptation de la préinscription ou la mise en attente du dossier



Les concepts de base

•L' événement

- Certains événements externes sont liés au temps. Par exemple, pour déclencher un traitement en début d' année civil, on introduira l' événement «Début d' année ».
- L' événement « Date actuelle est JJ/MM/AAAA » peut être considéré comme un événement qui permettra d' exécuter un traitement à une date donnée.
- Dans le MCT, chaque événement est identifié au moyen d' un libellé générique tel que « Dépôt d' un dossier de préinscription »
- Cet intitulé est très insuffisant pour décrire l' événement car il ne fait pas apparaître les données associées à l' événement. Par exemple, le dépôt d' un dossier de préinscription apporte de nombreuses informations telles que l' état civil du bachelier, des données sur sa scolarité actuelle, la formation qu' il souhaite intégrer, etc.



Les concepts de base

- **L' événement.**
 - Par exemple, l' événement « Dépôt de dossier de préinscription » sera détaillé ainsi : « Dépôt du dossier de préinscription du bachelier B à la formation F ».
 - Ce texte, qui constitue le message associé à l' événement, fait intervenir deux « entités » : Bachelier; et Formation.
 - Notons enfin que l' intitulé générique étant parfois long, on lui associe, sur le MCT un alias « ext » (externe) ou « int » (interne) afin de le référencer plus facilement.



Les concepts de base

- **L'opération**
- La réponse à l'arrivée d'un événement est le déclenchement d'un ensemble de traitements appelé opération.
- Le traitement d'enregistrement d'une préinscription est une opération déclenchée à la suite d'un événement « dépôt de dossier de préinscription »
- L'exécution d'une opération se ramène à l'exécution d'actions élémentaires à partir des données portées par le ou les événement(s) externes.
- Ces actions élémentaires peuvent appartenir à l'un des quatre types suivants :
 - insertion
 - la modification
 - l'effacement
 - recherche



Les concepts de base

•L'opération

- Sur le MCT, une opération est identifiée par un libellé et peut être décrite, de manière détaillée, en annexe du MCT, en présentant la logique algorithmique du déclenchement des actions élémentaires.
- Par exemple l'opération d'enregistrement du dossier de préinscription pourrait être détaillée ainsi :
 - création d'une occurrence de l'entité « Bachelier » et d'une occurrence de l'entité « Formation »
 - création d'une occurrence de l'association « Demande » (reliant l'entité « Bachelier » à l'entité « Formation »)
- Il est possible d'associer à une opération une durée qui représente le temps maximal qui lui est alloué pour qu'elle s'exécute.



Les concepts de base

- La règle d'émission

- La production effective d'une ou de plusieurs occurrences d'un événement interne est soumise à une règle d'émission, c'est-à-dire à une proposition logique qui s'applique au contenu de la base d'information après exécution de l'opération.
- L'événement est produit si la proposition logique est vraie. A l'issue de l'enregistrement d'un dossier de préinscription deux cas peuvent se présenter :
 - soit le dossier est complet et une occurrence de l'événement « Préinscription du Bachelier B à la formation F réalisée le JJ/MM/AAAA » est émise
 - soit le dossier est incomplet (certaines propriétés du MCD n'ont pas été renseignées) et une occurrence de l'événement « Dossier D Mis en attente le JJ/MM/AAAA » est produite.

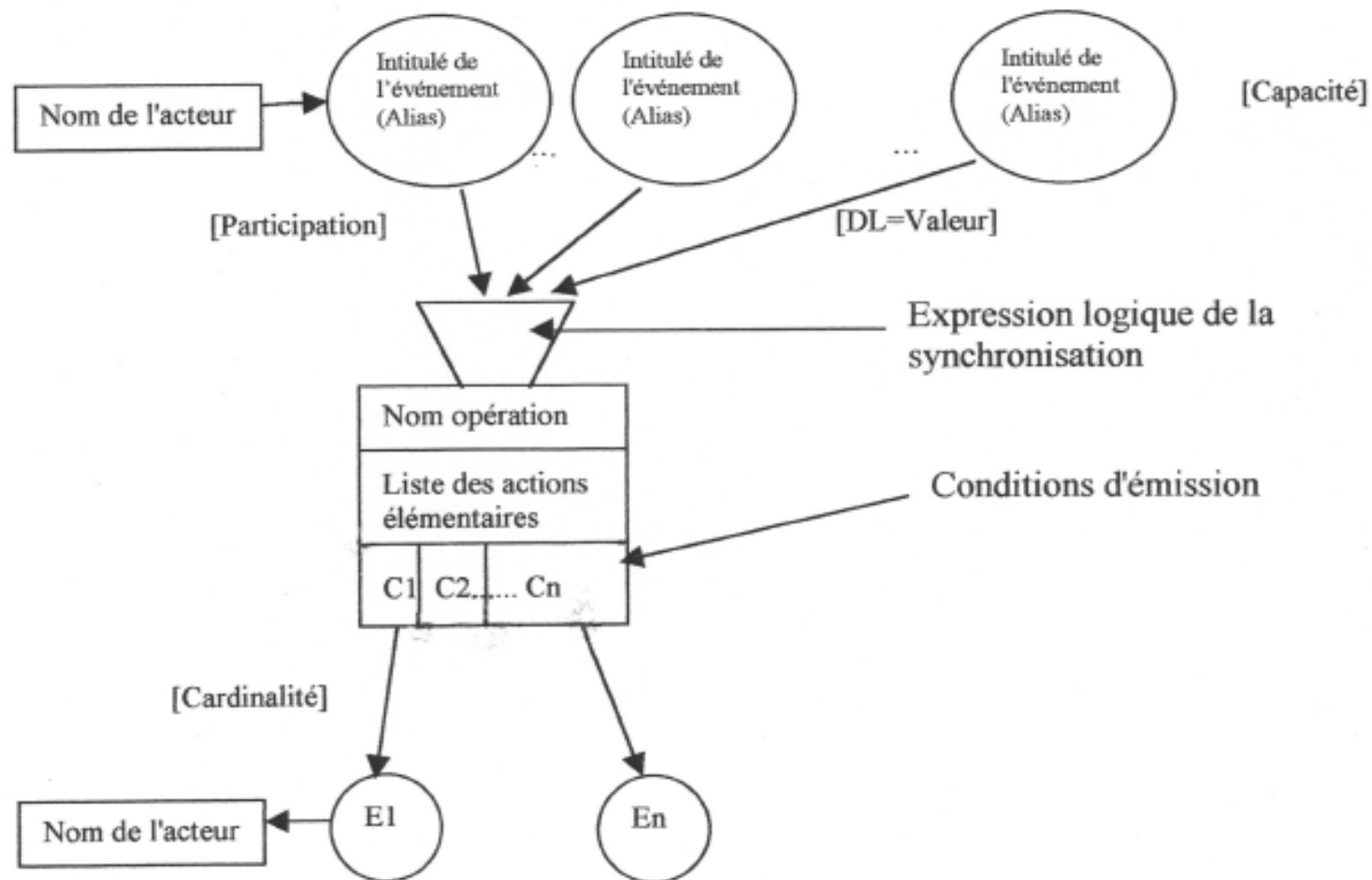


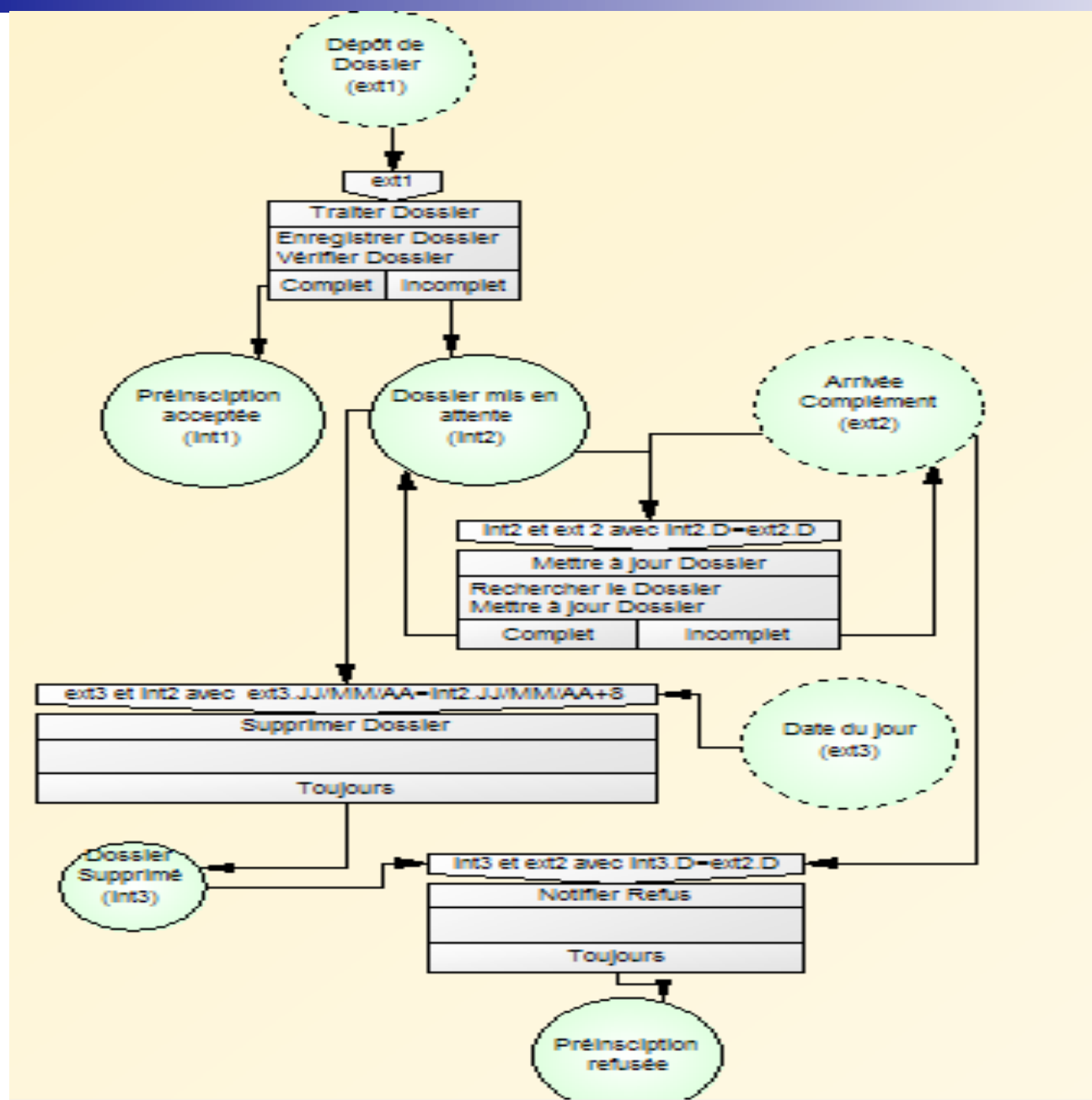
Les concepts de base

- **La règle d'émission**

- Si la plupart des règles d'émission sont basées sur une structure alternative et donne donc lieu à une seule occurrence d'événement interne, certaines peuvent intégrer une structure itérative de type « Pour – Tout » afin de produire n occurrences d'un événement interne.
- Par exemple pour envoyer en début d'année des lettres de renouvellement d'adhésion, on introduira la règle d'émission suivante :
 - Pour tout adhérent A enregistré dans la base d'informations
 - créer une occurrence de int1 (Renouvellement d'adhésion envoyé à A le JJ/MM/AAAA)
 - Fin pour

Représentation graphique et exemple de base







Modèle Organisationnel des Traitements



MOT

- Le MOT complète le MCT en y intégrant la
 - Notion de temps (Période)
 - Notion d'acteurs
 - Type d'opération (interactif, automatique, manuel)



Langage UML



Historique

Des Méthodes de modélisation

- L' apparition du paradigme objet à permis la naissance de plusieurs méthodes de modélisation
 - OMT, OOSE, Booch,
- Chacune de ces méthodes fournit une notation graphique et des règles pour élaborer les modèles
- Certaines méthodes sont outillées



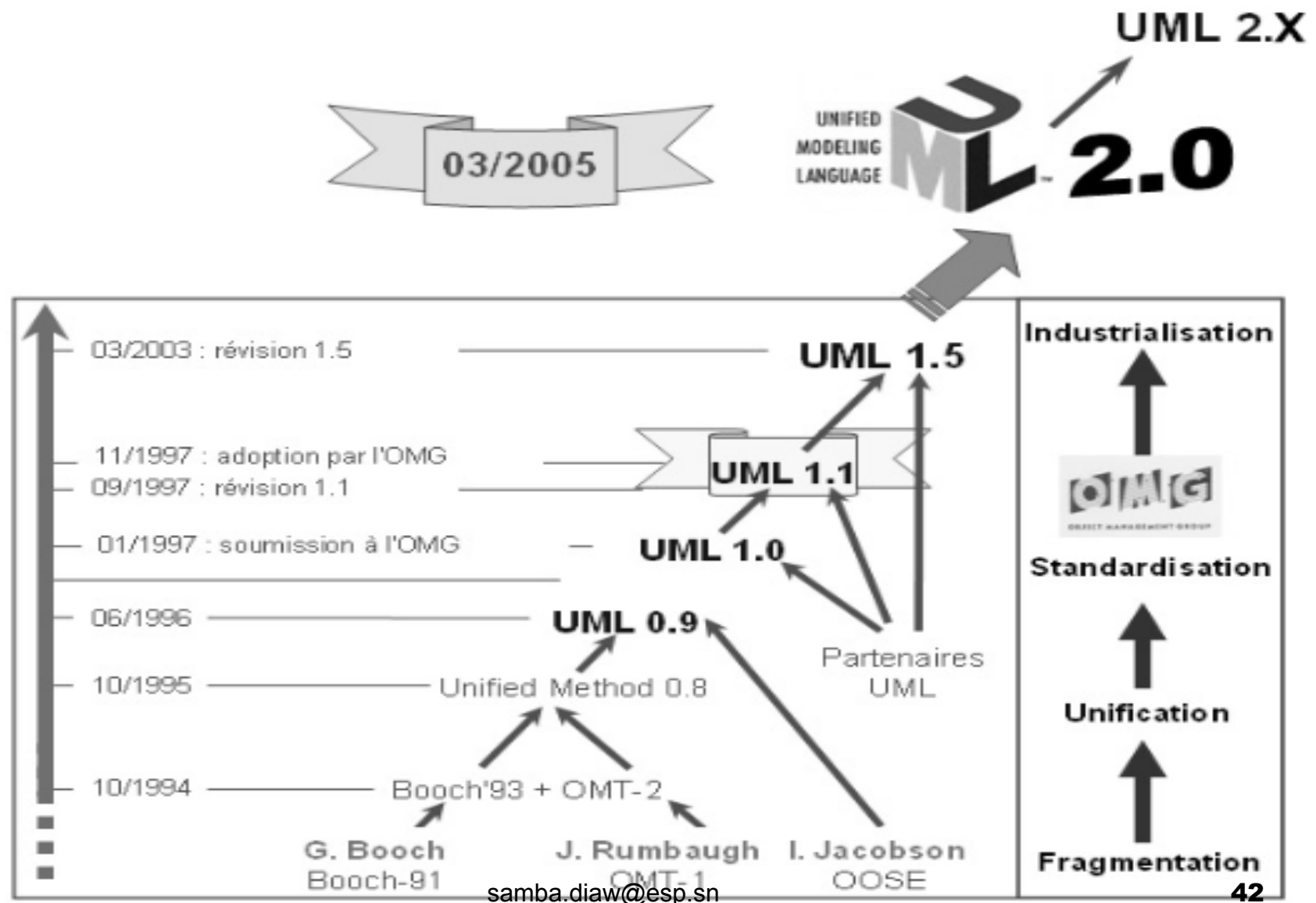
Historique

Trop de Méthodes

- Entre 89 et 94 : le nombre de méthodes orientées objet est passé de 10 à plus de 50
- Toutes les méthodes avaient pourtant d'énormes points communs (objets, méthode, paramètres, ...)
- Au milieu des années 90, G. Booch, I. Jacobson et J. Rumbaugh ont chacun commencé à adopter les idées des autres. Les 3 auteurs ont souhaité créer un langage de modélisation unifié



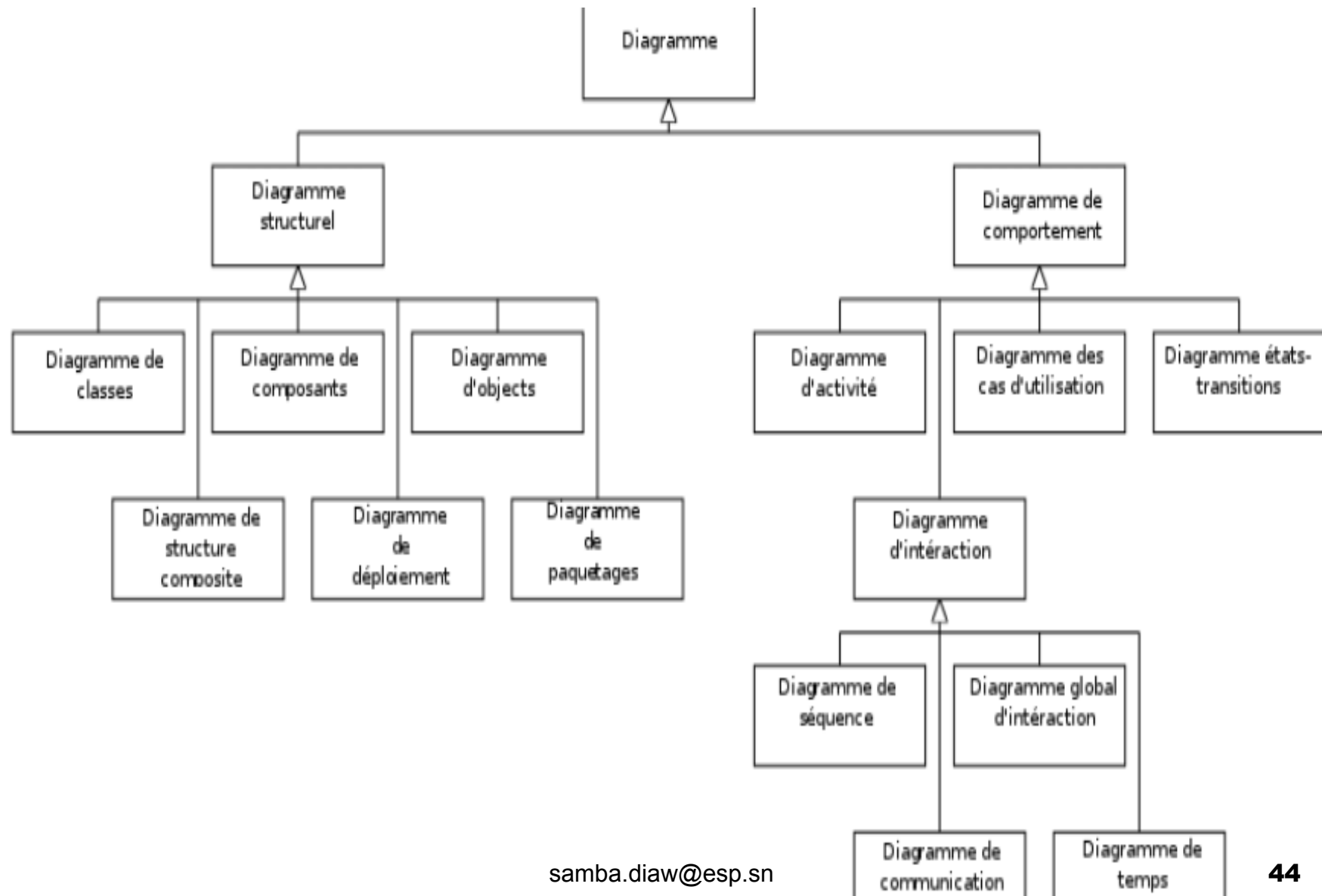
Historique





- UML propose diverses notations pour décrire les différentes facettes d'un système matériel ou logiciel
- UML 2.3 s'articule autour de quatorze types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en trois grands groupes:
 - Sept diagrammes structurels ou statiques
 - Trois diagrammes comportementaux
 - Quatre diagrammes d'interactions ou dynamiques

Vue d'ensemble d'UML





Le Diagramme d'objets

Représentation des objets et leurs relations



■ Définition

- Un diagramme d'objets représente des objets (instances de classes) et leurs liens (instances de relations) pour donner une vue de l'état du système à un instant donné afin :
 - d'illustrer le modèle de classes en montrant un exemple explicatif ;
 - de préciser certains aspects du système en mettant en évidence des détails imperceptibles dans le diagramme de classes ;



■ Objet

□ Définition

■ Objet du monde réel :

- Perception fondée sur le concept de masse

■ Objet informatique :

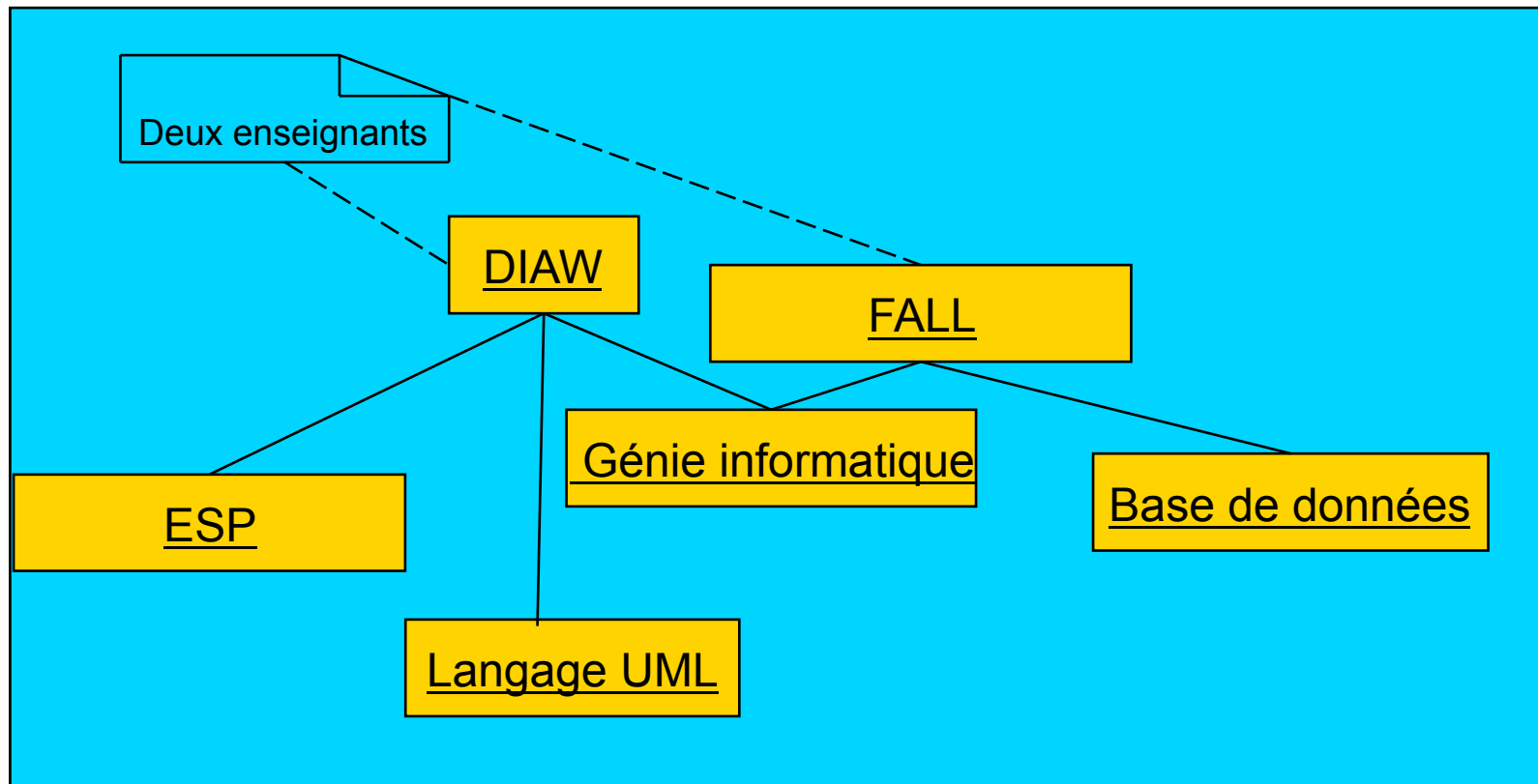
- Est une unité atomique formée de l'union d'un état et d'un comportement
- Définit une représentation abstraite d'une entité du monde réel ou virtuel, dans le but de la piloter ou de la simuler
 - Grain de sable, étoile
 - Compte en banque, police d'assurance, équation mathématiques, etc.

■ Les objets du monde réel et du monde informatique naissent, vivent et meurent

■ Objet

- Représentation d'objets en UML

Un objet



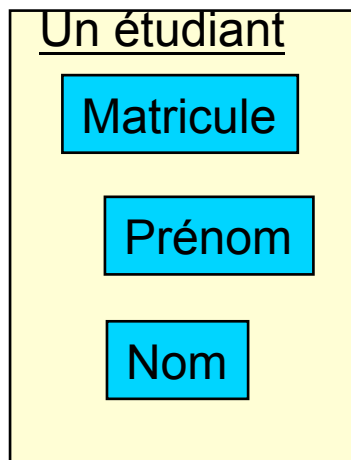
■ Objet

- Caractéristique fondamentales d' un objet (informatique)

Objet = État + Comportement + Identité

■ État

- Regroupe les valeurs instantanées de tous les attributs d' un objet :
 - attribut est une information qui qualifie l' objet qui le contient
 - Chaque attribut peut prendre une valeur dans un domaine de définition donné



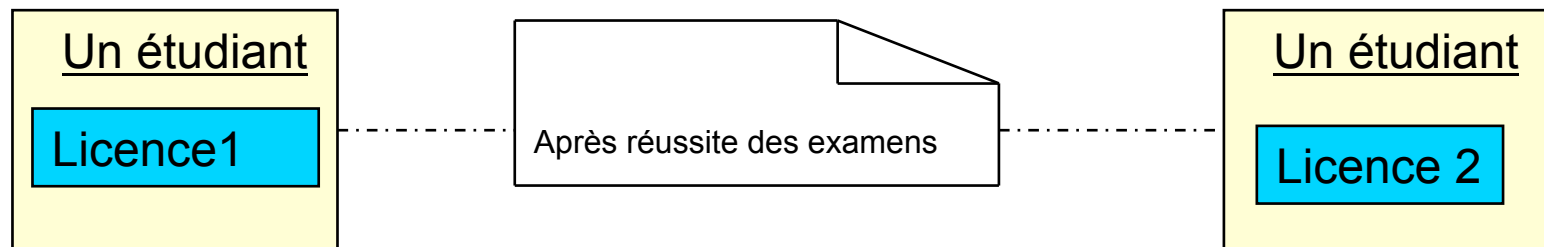
Exemple : Un objet Etudiant regroupe les valeurs des attributs matricule, Prénom et Nom

■ Objet

- Caractéristique fondamentales d'un objet (informatique)

■ État

- L'état d'un objet à un instant donné, correspond à une sélection de valeurs, parmi toutes les valeurs possibles des différent attributs
- L'état évolue au cours du temps, il est la conséquence de ses comportement passés
 - Un étudiant passe en classe supérieure, son niveau change et son cycle éventuellement.



- Certaines composantes de l'état peuvent être constantes
 - Le prénom et le nom d'un étudiant par exemple



■ Objet

- Caractéristique fondamentales d' un objet (informatique)

- **Le comportement**

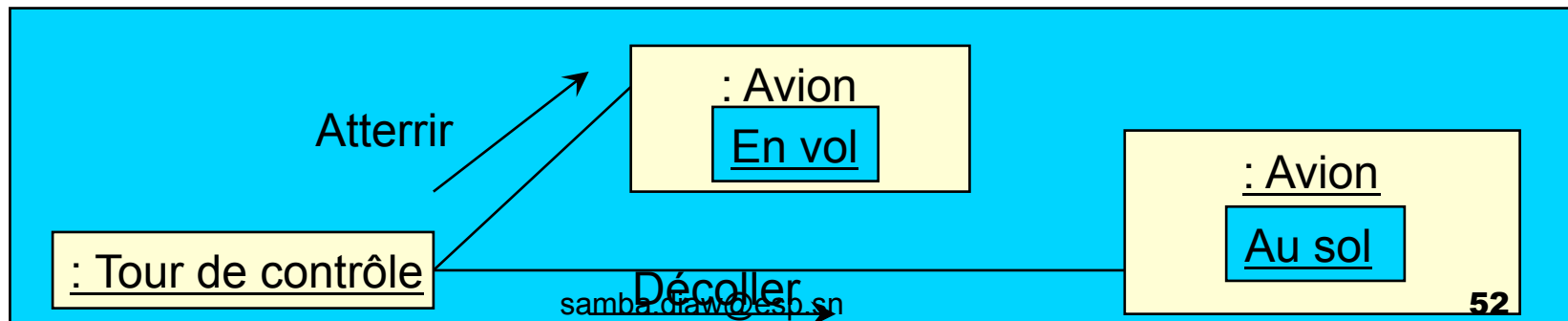
- Regroupe toutes les compétences d' un objet et décrit les actions et les réactions de cet objet
- Chaque atome (partie) de comportement est appelé opération
 - Les opérations d' un objet sont déclenchées suite à une stimulation externe, représentée sous la forme d' un message envoyé par un autre objet
 - L' état et le comportement sont liés

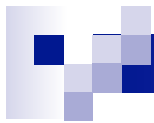
■ Objet

□ Exemple

■ Le comportement

- Le comportement à un instant donné dépend de l'état courant et l'état peut être modifié par le comportement
 - Il n'est pas possible de faire atterrir un avion que s'il est en train de voler: le comportement *Atterrir* n'est valide que si l'information *En vol* est valide
 - Après l'atterrissage, l'information *En vol* devient invalide et l'opération *Atterrir* n'a plus de sens





■ Objet

- Caractéristique fondamentales d' un objet

- **L' identité**

- Chaque objet possède une identité qui caractérise son existence propre
 - L' identité permet de distinguer tout objet de façon non ambiguë, indépendamment de son état
 - Permet de distinguer deux objets dont toutes les valeurs d' attributs sont identiques
 - deux pommes de la même couleur, du même poids et de la même taille sont deux objets distincts.
 - Deux véhicules de la même marque, de la même série et ayant exactement les mêmes options sont aussi deux objets distincts.



■ Objet

■ Caractéristique fondamentales d' un objet

□ L' identité

- L' identité est un concept, elle ne se représente pas de manière spécifique en modélisation. : chaque objet possède une identité de manière implicite
- En phase de réalisation, l' identité est souvent construite à partir d' un identifiant issu naturellement du domaine du problème.
- Nos voitures possèdent toutes un numéro d' immatriculation, nos téléphones un numéro d' appel



Le Diagramme de classes d'UML

**Représentation de la structure statique en
termes de classes et relations**



■ Classe

□ Définition

- Une classe décrit une abstraction d'objets ayant
 - Des propriétés similaires
 - Un comportement commun
 - Des relations identiques avec les autres objets
 - Une sémantique commune
- Par exemple Etudiant (resp. Filière, resp. Cours) est la classe de tous les étudiants (resp. filières, resp. cours)



■ Classe

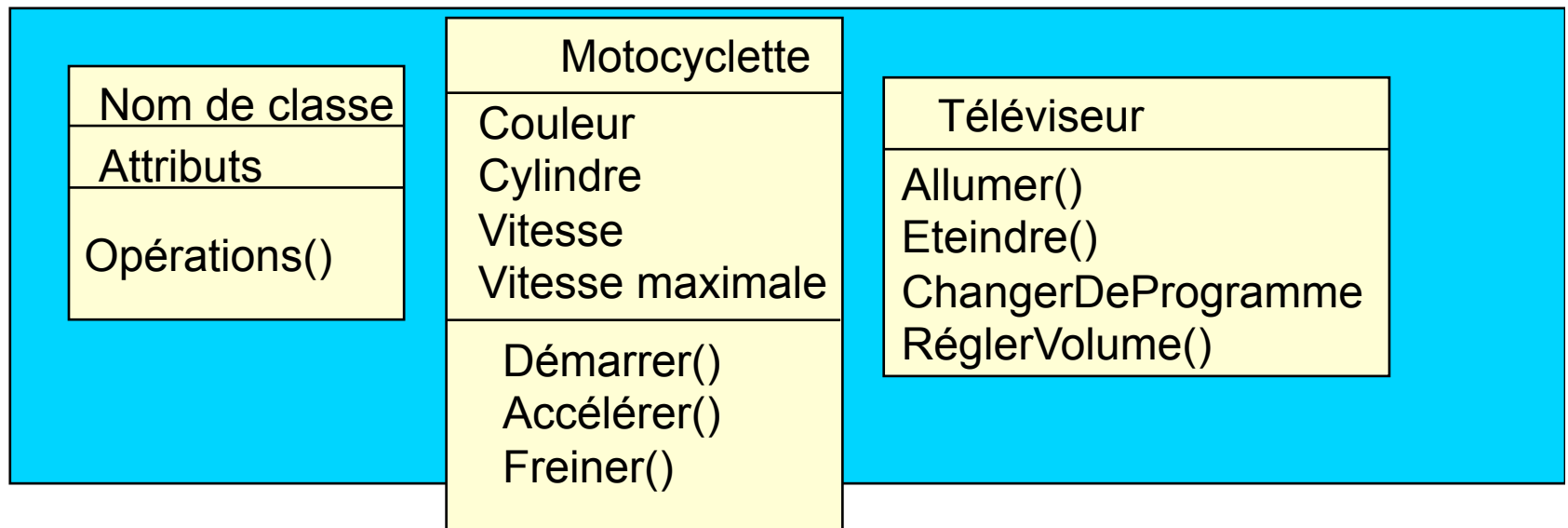
□ Caractéristiques d'une classe

- Un objet créé par (on dit également appartenant à) une classe sera appelé une *instance de cette classe* ce qui justifie le terme ``variables d'instances ' '
- les valeurs des *variables d'instances* sont propres à chacune de ces instances et les caractérisent
- Les généralités sont contenues dans la classe et les particularité sont contenues dans les objets
- Les objets sont construits à partir de la classe, par un processus appelé instanciation : tout objet est une instance de classe
- Nous distinguons deux types de classes
 - Classe concrète : peut être instanciée
 - Classe abstraite : est une classe qui ne donne pas directement des objets et ne contient que des méthodes abstraites.

■ Classe

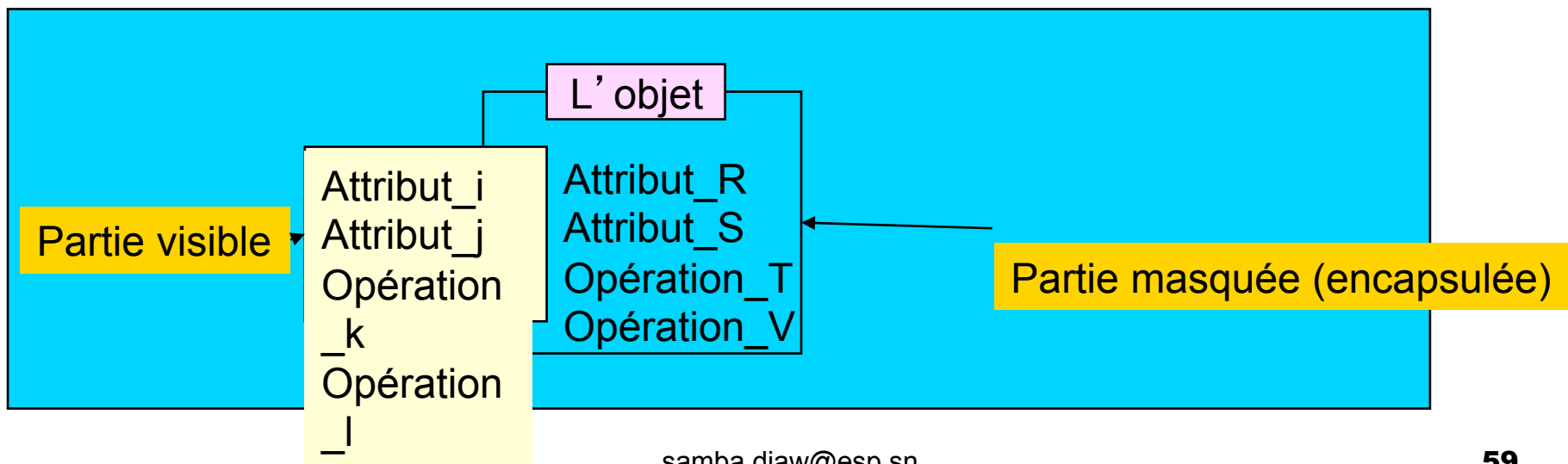
□ Représentation graphique d'une classe en UML

- Chaque classe est représentée sous la forme d'un rectangle divisé en trois compartiments
- Les compartiments peuvent être supprimés pour alléger les diagrammes



■ Encapsulation

- Consiste à masquer les détails d'implémentation d'un objet en définissant une interface
- L'interface est la vue externe d'un objet, elle définit les services accessibles (offerts) aux utilisateurs de l'objet.
- Est la séparation entre les propriétés externes, visibles des autres objets, et les aspects internes, propres aux choix d'implantation d'un objet.





■ Encapsulation

- Présente un double avantage
 - facilite l'évolution d'une application car elle stabilise l'utilisation des objets : on peut modifier l'implémentation des attributs d'un objet sans modifier son interface
 - garantit l'intégrité des données, car elle permet d'interdire ou de restreindre l'accès direct aux attributs des objets (utilisation d'accesseurs)



Règle de visibilité

+ Attribut public
Attribut protégé
- Attribut privé

+ Opération publique()
Opération protégée()
- Opération privée()

Salarié

+ nom
age
- salaire

+ donnerSalaire()
changerSalaire()
- calculerPrime()

■ Encapsulation

- Il est possible d'assouplir le degré d'encapsulation au profit de certaines classes utilisatrices bien particulière en définissant des niveaux de visibilité
- Les trois niveaux distincts d'encapsulation couramment retenus sont:
 - Niveau privé : c'est le niveau le plus fort; la partie privée de la classe est totalement opaque
 - Niveau protégé : c'est le niveau intermédiaire ; les attributs placés dans la partie protégée ne sont visibles que par les classes dérivées de la classe fournisseur. Pour toutes les autres classes, les attributs restent invisibles
 - Niveau publique : ceci revient à se passer de la notion d'encapsulation et de rendre visibles les attributs pour toutes les classes

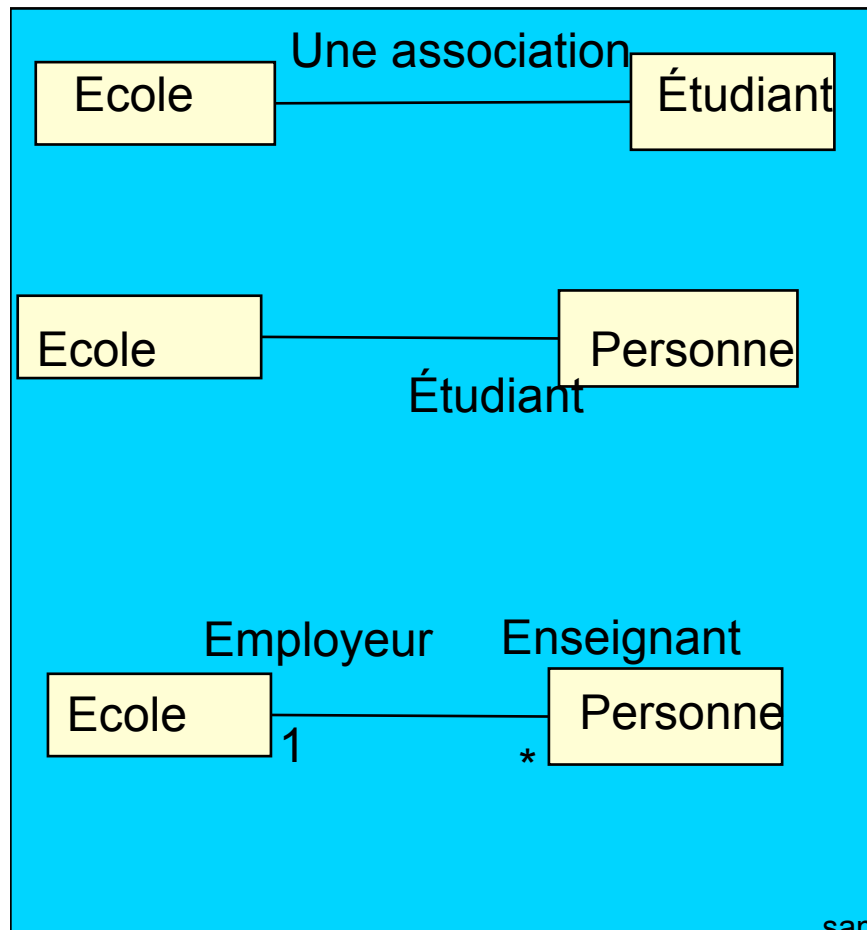


Attributs

- Une classe peut contenir des attributs
- La syntaxe d'un attribut est :
visibilité nom : type
- La visibilité est:
 - '+' pour public
 - '#' pour protected
 - '-' pour private
- UML définit son propre ensemble de types
 - Integer, real, string, ...
- Un attribut peut être un attribut de classe, il est alors souligné.
- Un attribut peut être dérivé, il est alors préfixé par le caractère '/'

■ Les relations entre les classes

□ Associations

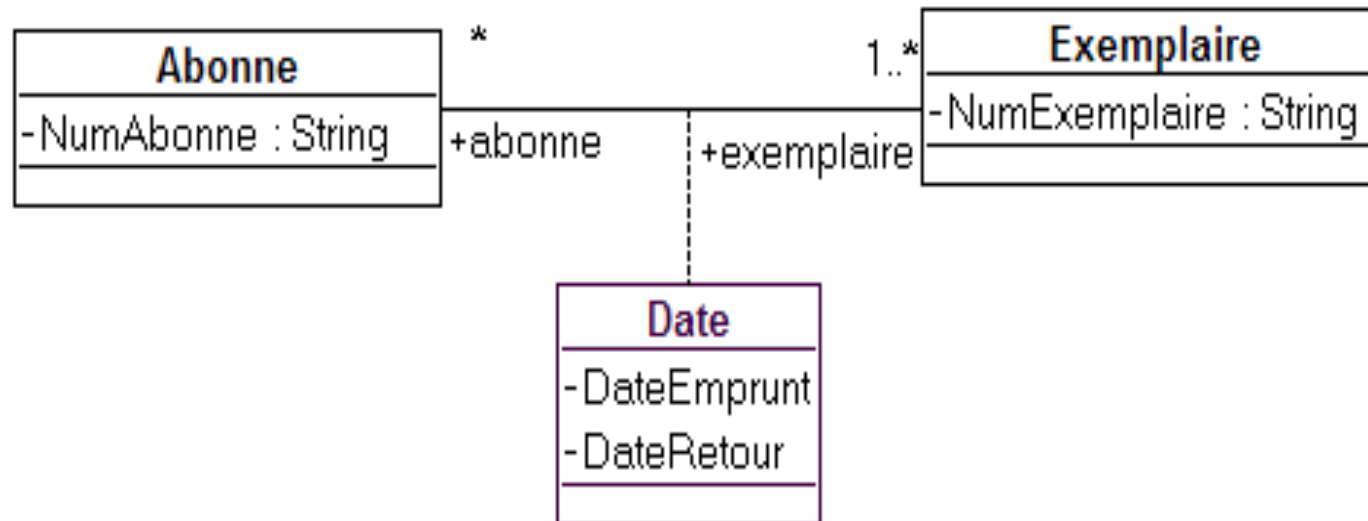


- Représentent les relations structurelles qui existent entre objets de différentes classes.
- Est une abstraction des liens qui existent entre les objets instances des classes associées
- La plupart des associations sont binaires
- Une association est caractérisée par:
 - un nom, qui peut être omis notamment quand les rôles des classes sont spécifiés,
 - un rôle pour chacune des classes qui participent à l'association. Le rôle d'une classe, écrit à l'extrémité du lien d'association, décrit la manière dont cette classe est ``vue" par l'autre classe.
 - une multiplicité

Relations entre classes

■ Classe d'association

- Une classe-association est une association qui est aussi une classe.
- Les classes-associations sont utilisées lorsque les associations doivent porter des informations





■ Les relations entre les classes

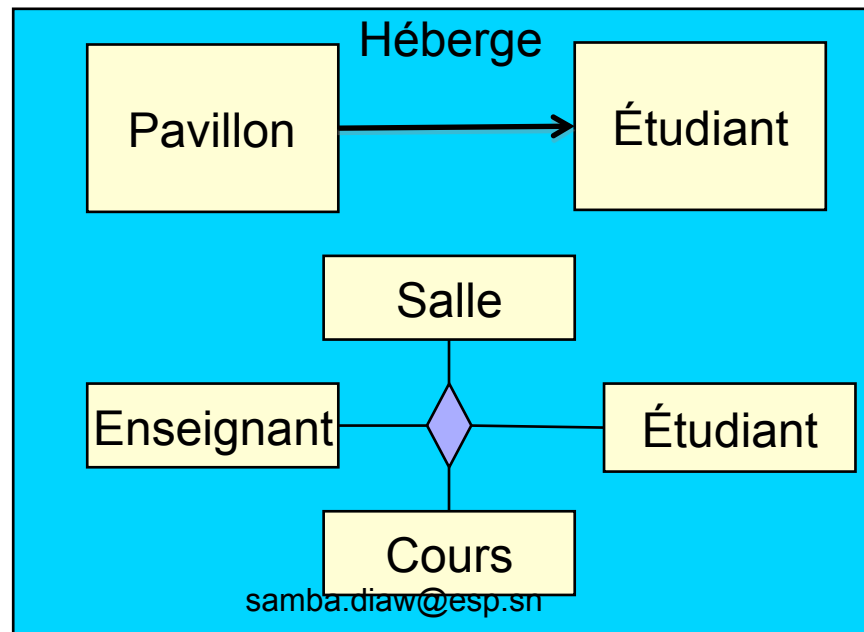
□ Multiplicité

- Chaque extrémité d'une association peut porter une indication de multiplicité (nombre d'occurrences) qui montre combien d'objets de la classe considérée peuvent être liés à un objet de l'autre classe.

1	Un et un
0..1	Zéro ou un
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	D' un à plusieurs

■ Les relations entre les classes

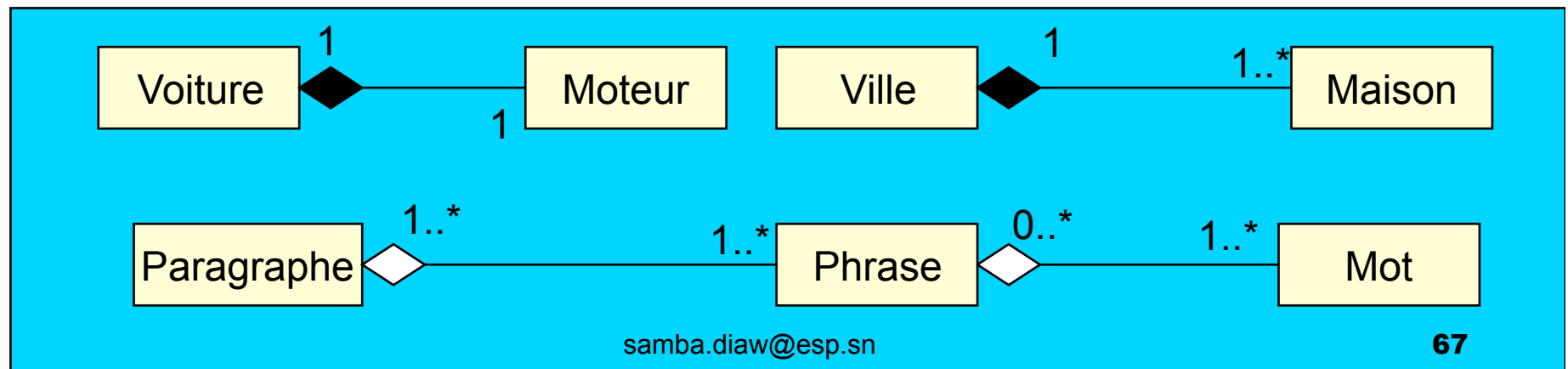
- Association n-aire et navigabilité
- Le nom de l'association peut être suivi d'une flèche de direction qui précise le sens de lecture (on dit que l'association est navigable)
- Il existe des associations ternaires (ou d'ordre supérieur) qui sont représentées par un diamant relié à chacune des classes participant à la relation.



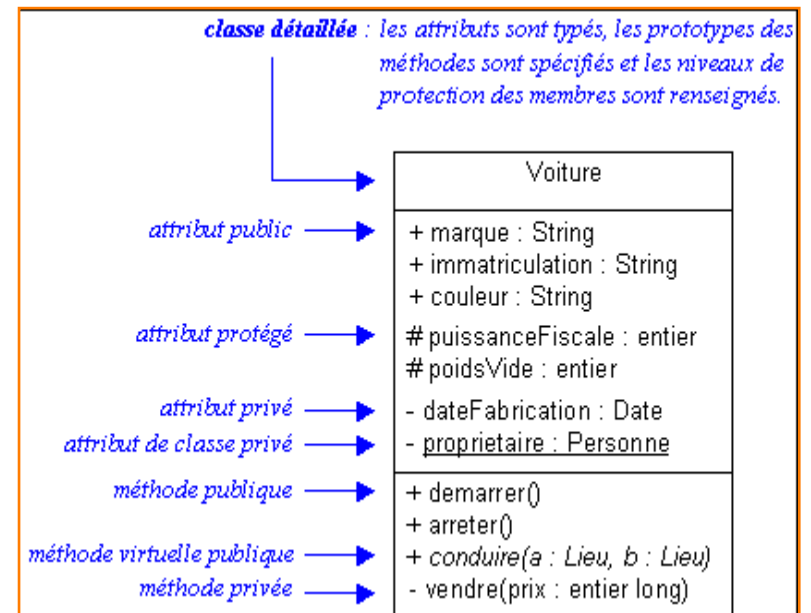
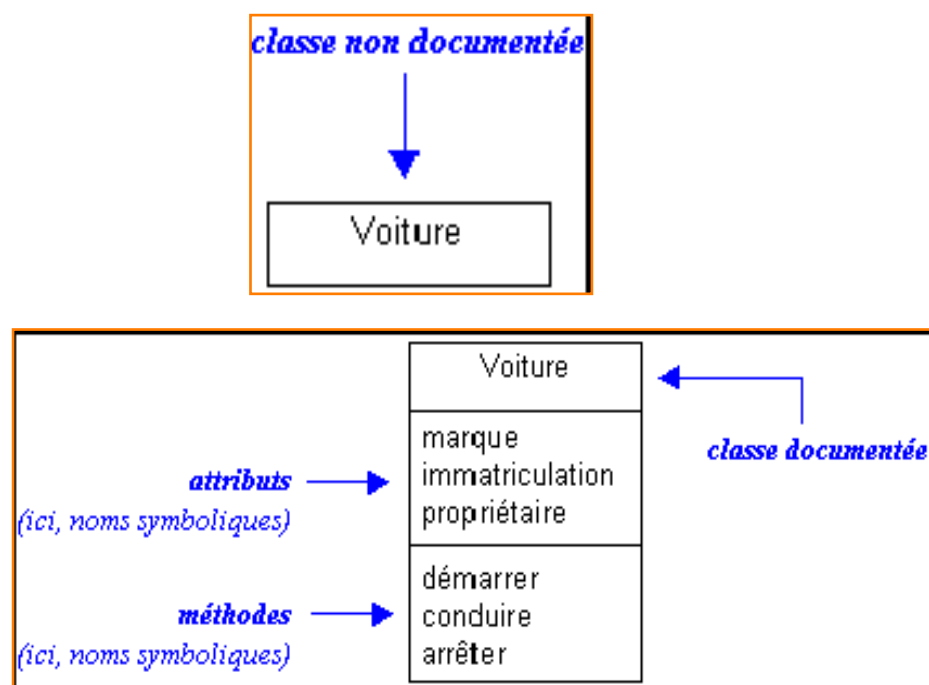
■ Les relations entre les classes

□ La composition et l'agrégation (shared)

- UML propose deux sémantiques de contenance : une sémantique de contenance faible, dite d'agrégation, qui permet de préciser que les éléments contenus peuvent être partagés entre plusieurs conteneurs,
- et une sémantique de contenance forte, dite composition, qui permet de préciser que les éléments contenus ne peuvent être partagés entre plusieurs conteneurs.
- Du point de vue graphique, la relation de contenance se représente à l'aide d'un losange sur l'extrémité. Le losange est blanc pour l'agrégation et noir pour la composition.



■ Documentation d'une classe

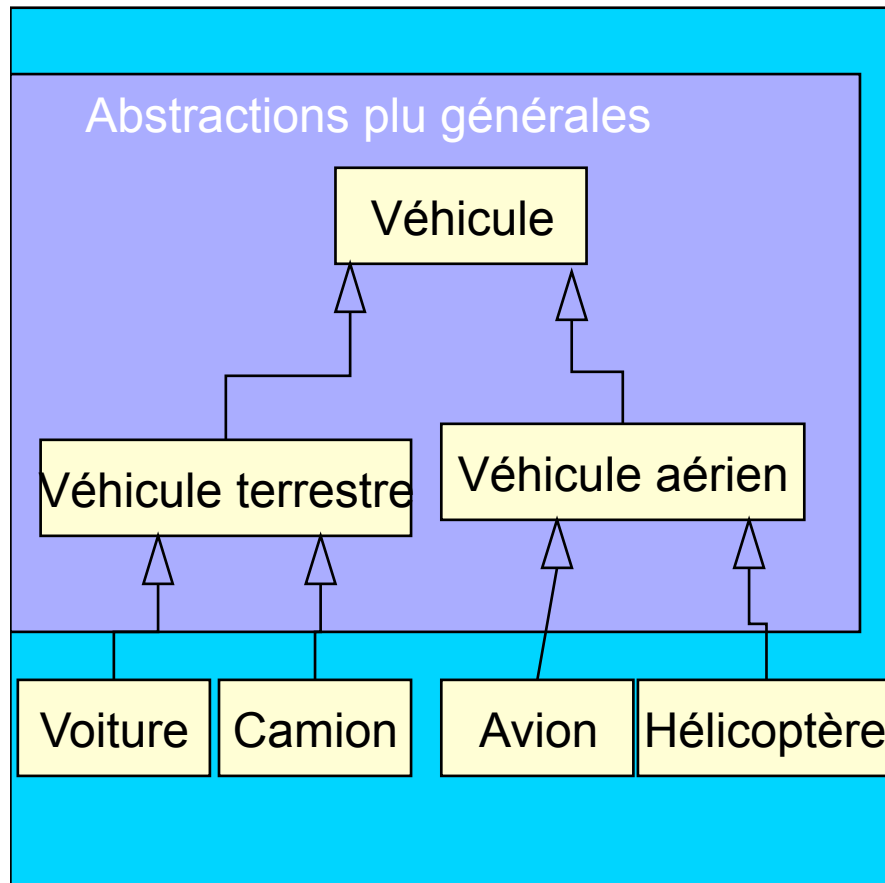


■ Les hiérarchies de classes

□ Généralisation et spécialisation

- La généralisation consiste à factoriser les éléments communs (attributs, opérations) d'un ensemble de classes dans une classe plus générale appelée super-classe

- Les classes sont ordonnées selon une hiérarchie ; une super-classe est une abstraction de ses sous-classes
- Un arbre (une hiérarchie) représentant une généralisation se détermine en partant des feuilles (et non pas de la racine) car les feuilles appartiennent au monde réel alors que les niveaux supérieurs sont des abstractions construites pour ordonner et comprendre

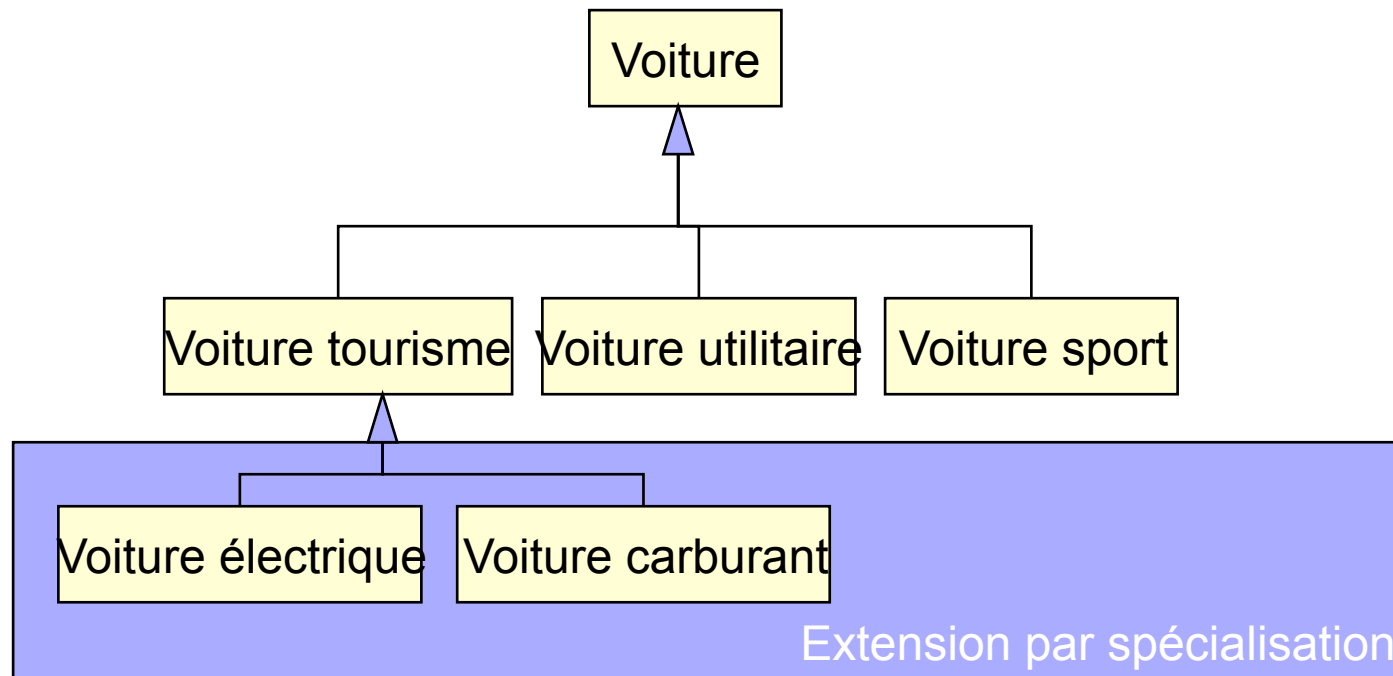


■ Les hiérarchies de classes

■ Généralisation et spécialisation

- La spécialisation permet de capturer les particularités d'un ensemble d'objets non discriminés par les classes déjà identifiées

- Les nouvelles caractéristiques sont représentées par une nouvelle classe, sous-classe d'une des classes existantes





■ Les hiérarchies de classes

■ Généralisation et spécialisation

- ☐ La généralisation et la spécialisation sont deux point de vue antagonistes du concept de classification; elle expriment dans quel sens une hiérarchie de classe est exploitée
- ☐ La généralisation ne porte aucun nom particulier ; elle signifie toujours : est un ou est une sorte de
- ☐ La généralisation ne concerne que les classes, elle n'est pas instanciable en liens
- ☐ La généralisation ne porte aucune indication de multiplicité



■ Les hiérarchies de classes

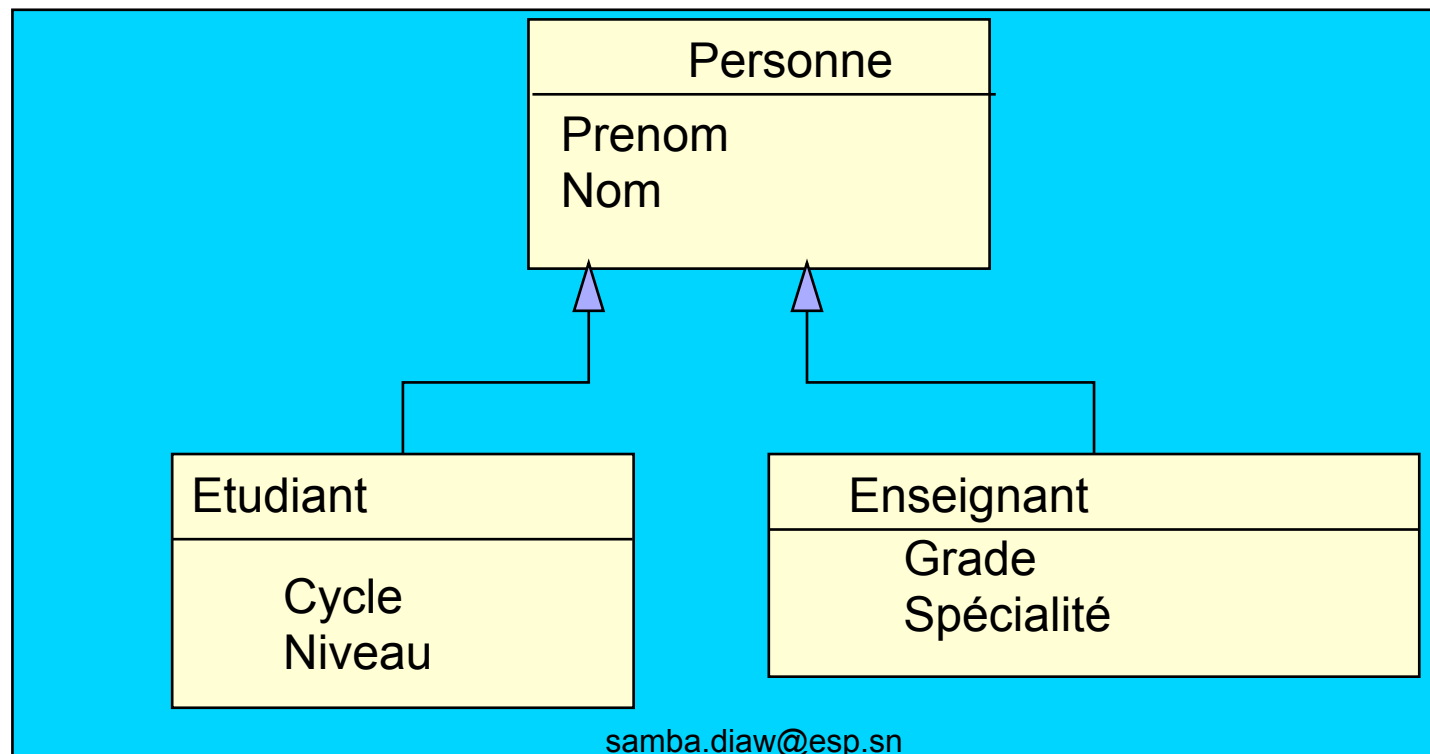
■ Généralisation et spécialisation

- La généralisation est une relation non réflexive : une classe ne peut pas dériver d'elle-même
- La généralisation est une relation non symétrique : si une classe B dérive d'une classe A, alors la classe A ne peut pas dériver de la classe B
- La généralisation est une relation transitive : si une classe C dérive d'une classe B qui dérive elle-même d'une classe A, alors C dérive également de A
- Les objets instances d'une classe donnée sont décrit par la propriétés caractéristiques de leur classe, mais également par les propriétés caractéristiques de toutes les classes parents de leur classe

■ Les hiérarchies de classes

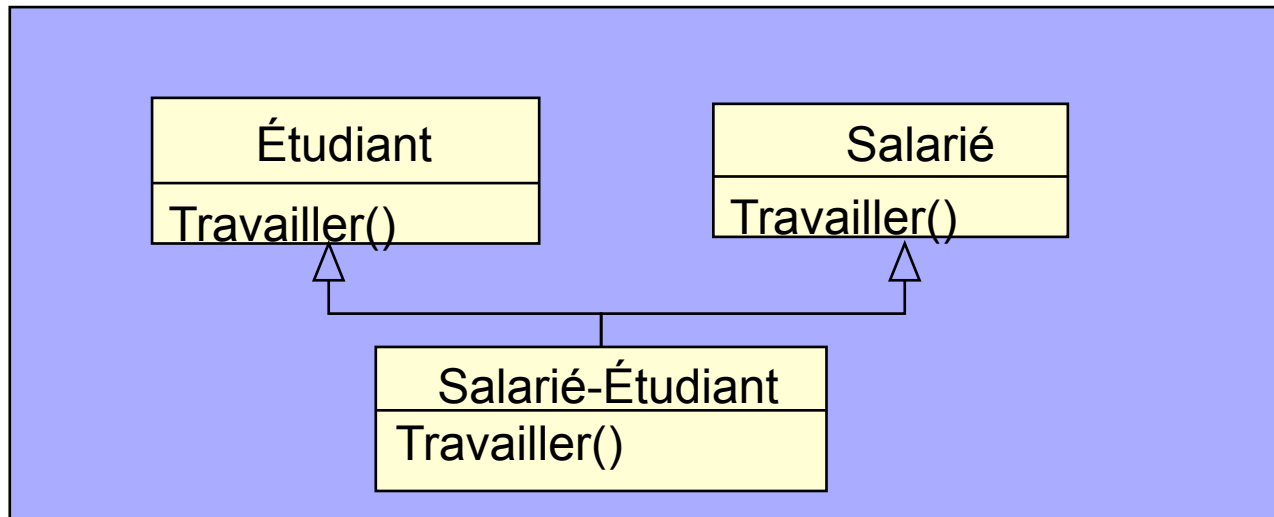
■ Généralisation et spécialisation

- Les propriétés d'une sous-classe englobent les propriétés de toutes ses super-classes
 - Ce qui est vrai pour un objet instance d'une super classe est vrai pour un objet instance d'une sous-classe



■ Les hiérarchies de classes

- Généralisation multiple :
- Elle existe entre arbres de classes disjoints
 - Une classe ne peut posséder qu'une fois une propriété donnée





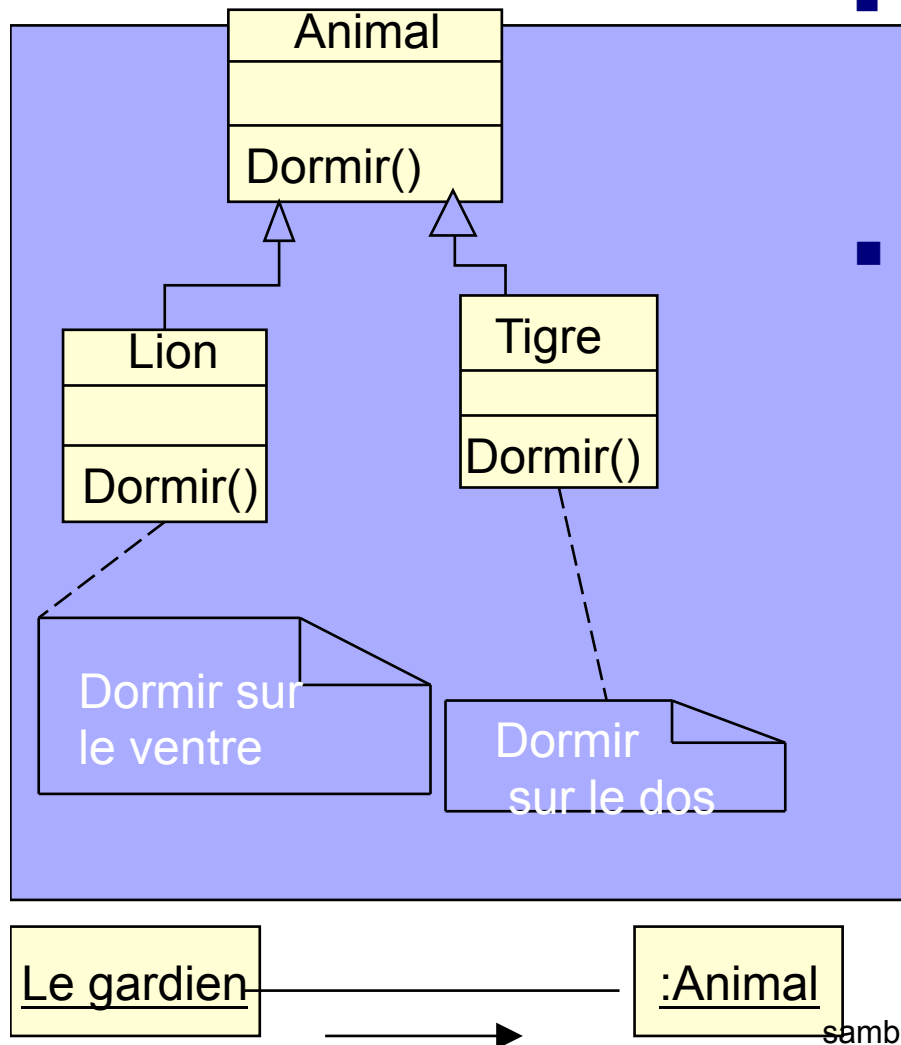
■ Les hiérarchies de classes

■ Polymorphisme : notion de surcharge

- Le polymorphisme signifie qu'une même opération peut se comporter différemment sur différentes classes

■ L'opération ***Dormir()*** agira de manières différentes sur un lion, un tigre ou un ours.

- Le polymorphisme signifie que les différentes méthodes d'une opération ont la même signature
 - Lorsque une opération est invoquée sur un objet, celui-ci « connaît » sa classe et par conséquent est capable d'invoquer automatiquement la méthode correspondante.



Exemple de diagramme de Classes

