

Génie Logiciel

Chapitre 6:

Autres notions et diagrammes UML

Niveau: 3^{ème} année License informatique

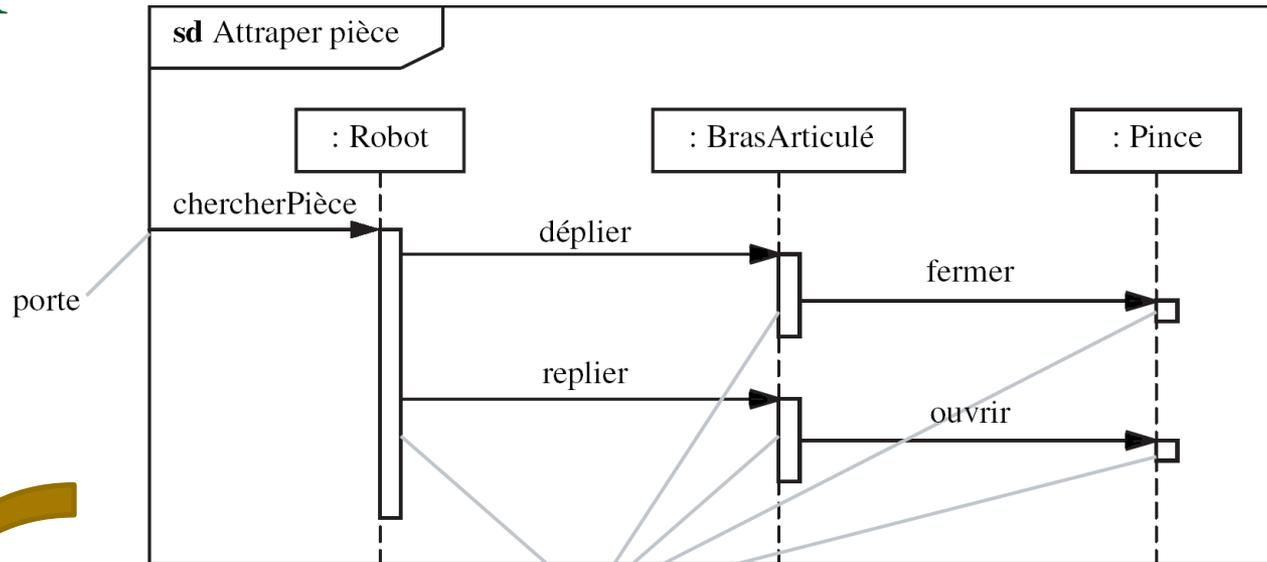
Année: 2022/2023

Diagrammes de communication

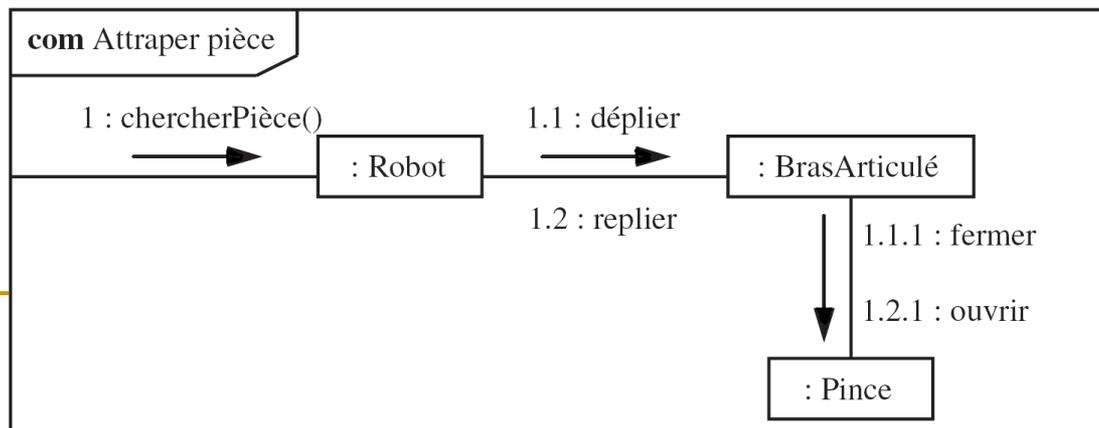
Diagrammes d'interaction

- Les diagrammes de communication et les diagrammes de séquences sont deux types de **diagramme d'interaction**
 - Un **diagramme de séquence** montre des interactions sous un angle temporel, en mettant l'emphase sur le séquençement temporel de messages échangés entre des lignes de vie
 - Un **diagramme de communication** montre une représentation spatiale des lignes de vie.
 - Ils représentent la même chose, mais sous des formes différentes.
- A ces diagrammes, UML 2.0 en ajoute un troisième : le **diagramme de timing**.
 - Son usage est limité à la modélisation des systèmes qui s'exécutent sous de fortes contraintes de temps, comme les systèmes temps réel.

Equivalence avec un diagramme de séquence



spécifications de l'exécution des méthodes



Diagrammes de composants et de déploiement

Composant

- Un **composant** logiciel est une unité logicielle autonome au sein d'un système global ou d'un sous-système.
- C'est un classeur structuré particulier, muni d'une ou plusieurs interfaces requises ou offertes.

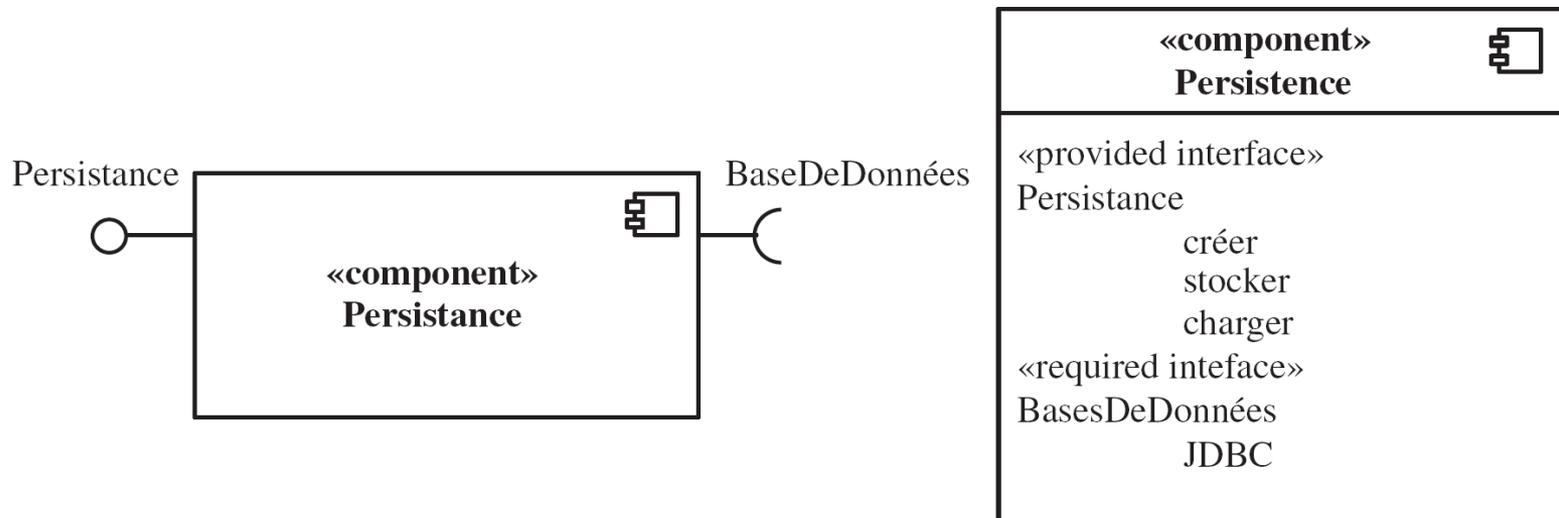
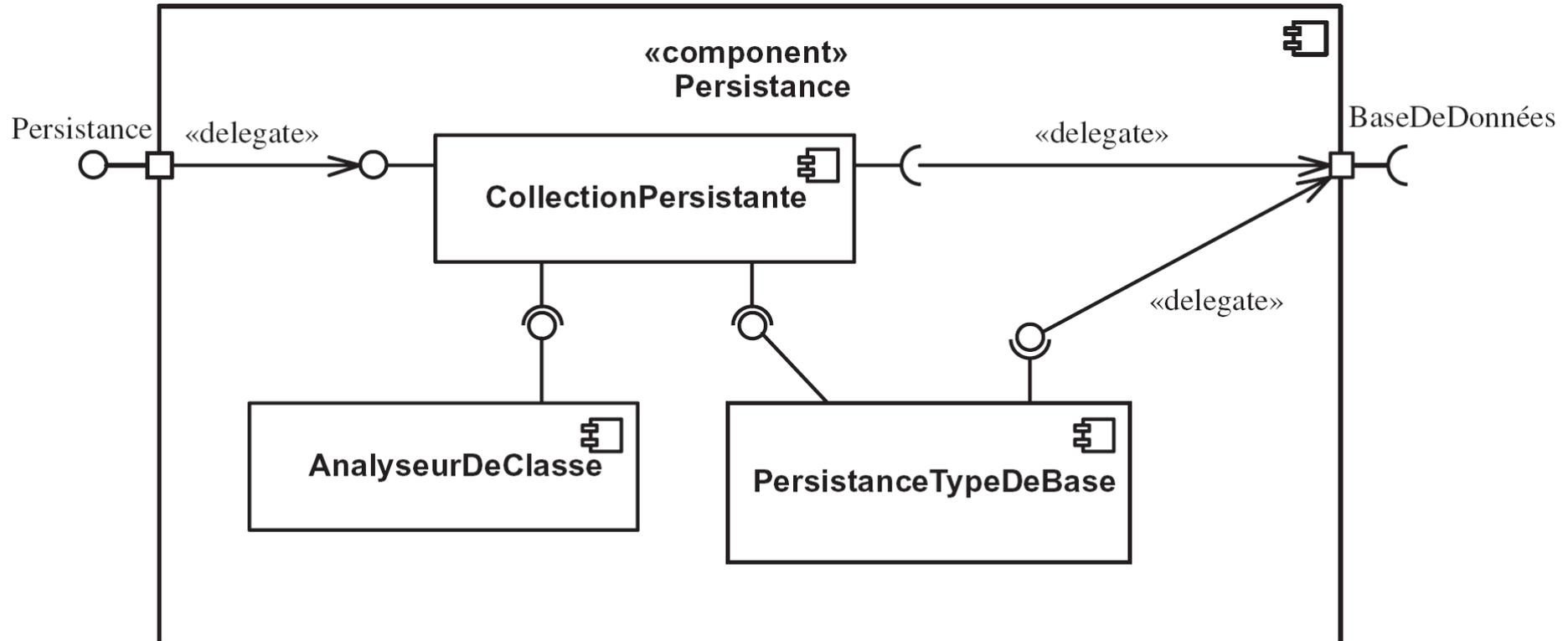


Diagramme de composant

- Les **diagrammes de composants** représentent l'architecture logicielle du système.
 - Les composants peuvent être décomposés en **sous-composants**, le cas échéant.
 - Les liens entre composants sont spécifiés à l'aide de **dépendances entre leurs interfaces**.
 - Le « **câblage interne** » d'un composant est spécifié par les **connecteurs de délégation**.
 - Un tel connecteur connecte un port externe du composant avec un port de l'un de ses sous-composants internes.

Exemple de modélisation d'un composant



Intérêt des diagrammes de composants

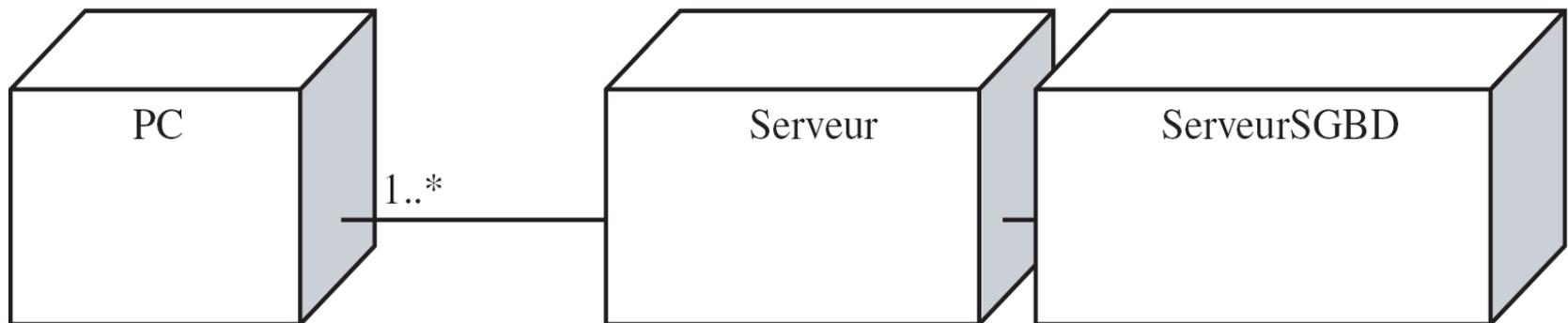
- Les diagrammes de composants permettent de:
 - Structurer une architecture logicielle à un niveau de granularité moindre que les classes
 - Les composants peuvent contenir des classes.
 - Spécifier l'intégration de briques logicielles tierces (composants EJB, CORBA, COM+ ou .Net, WSDL...) ;
 - Identifier les composants réutilisables.
- Un composant est un espace de noms qu'on peut employer pour organiser les éléments de conception, les cas d'utilisation, les interactions et les artefacts de code.

Architecture matérielle

- En dernier lieu, un système doit s'exécuter sur des ressources matérielles dans un environnement matériel particulier.
- UML permet de représenter un environnement d'exécution ainsi que des ressources physiques (avec les parties du système qui s'y exécutent) grâce aux **diagrammes de déploiement**.
- L'environnement d'exécution ou les ressources matérielles sont appelés « **nœuds** ».
- Les parties d'un système qui s'exécutent sur un noeud sont appelées « **artefacts** ».

Noeud

- Un **noeud** est une ressource sur laquelle des artefacts peuvent être déployés pour être exécutés.
- C'est un classeur qui peut prendre des attributs.
 - Un noeud se représente par un cube dont le nom respecte la syntaxe des noms de classes.
 - Les noeuds peuvent être associés comme des classes et on peut spécifier des multiplicités.

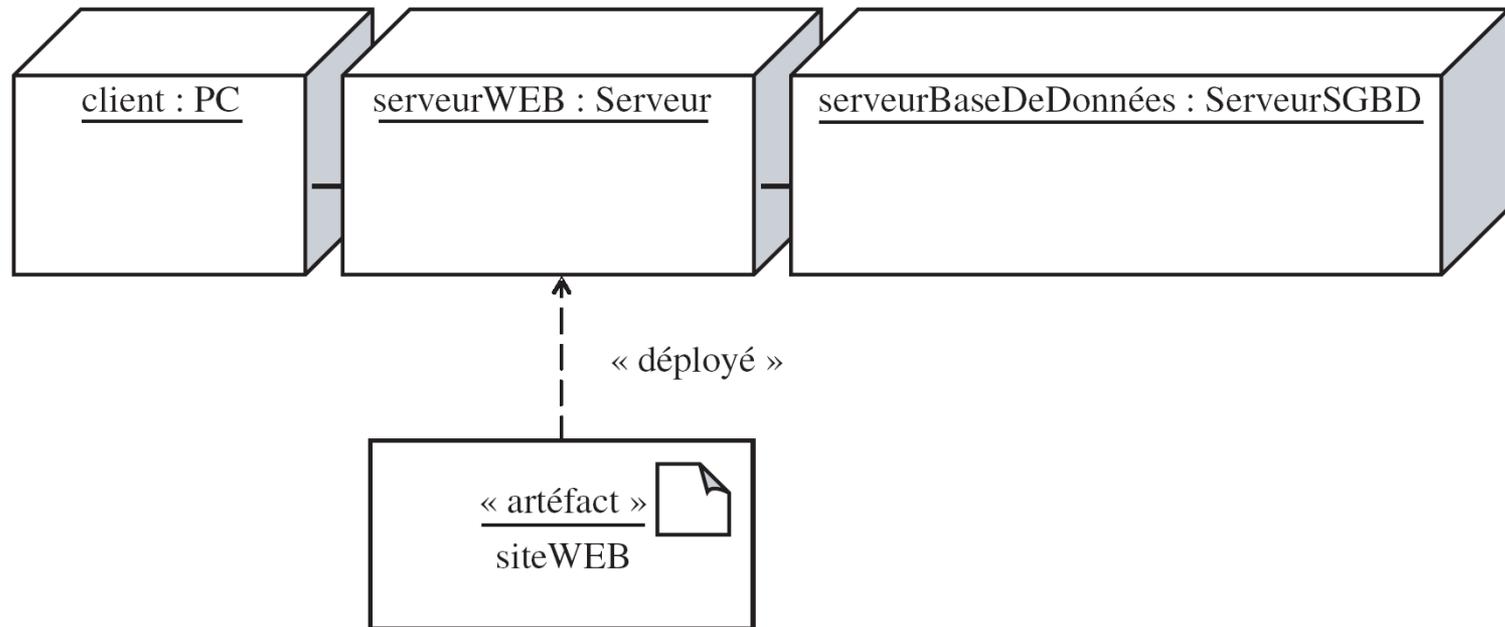


Artefact

- Un **artefact** est la spécification d'une entité physique du monde réel.
- Il se représente comme une classe par un rectangle contenant le mot-clé **artefact** suivi du nom de l'artefact.
 - Un artefact déployé sur un noeud est symbolisé par une èche en trait pointillé qui porte le stéréotype déployé et qui pointe vers le noeud.
 - L'artefact peut aussi être inclus directement dans le cube représentant le noeud.
- Plusieurs stéréotypes standard existent pour les artefacts : **document, exécutable, fichier, librairie, source.**

Instanciación de nodos y artefactos

- Los nombres de las instancias de nodos y artefactos están subrayados.



Exécution des composants

- *On utilise des flèches de dépendance pour montrer la capacité d'un nœud à prendre en charge un type de composant.*