

Conception des systèmes d'information

Chapitre 6 : diagrammes de composants et de déploiement

Dr. Mariem MAHFOUDH

mariem.mahfoudh@gmail.com

2 LSI-ADBD, ISIMS, 2023-2024



Ce cours a été construit en se basant sur les références suivantes :

- ▶ Livre "UML 2 pratique pour la modélisation", Benoît Charroux, Aomar Osmani et Yann Thierry-Mieg
- ▶ Livre "UML 2 pour la pratique", Pascal Roques
- ▶ Livre "Modélisation Objet avec UML", Pr. Pierre Alain Muller
- ▶ Cours "Conception des systèmes d'information", Pr. Faiez Gargouri
- ▶ Cours "Analyse, Conception Objet", Stephane Galland

1 Diagrammes de composants

- Définition
- Concepts de diagramme de composant

2 Diagrammes de déploiement

- Définition
- Concepts de diagramme de déploiement

Le diagramme de composants :

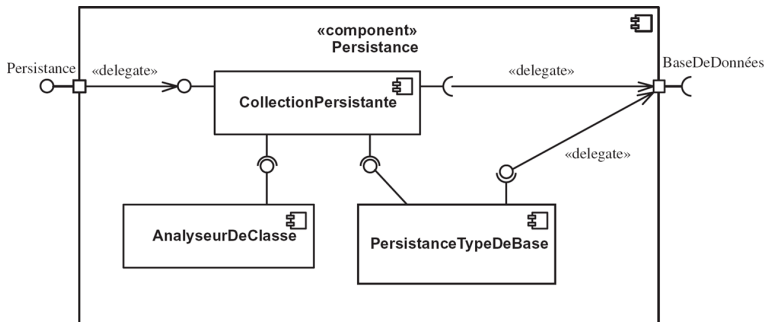
- ▶ représente l'**architecture logicielle** du système ;
 - ▶ présente une **vue statique** de l'implémentation du système illustrant les **choix de réalisation** ;
 - ▶ décrit le système modélisé sous forme de **composants** réutilisables ;
 - ▶ est utilisé pour décrire les **choix d'implémentation** et les dépendances de compilation et d'implémentation entre les composants du système.
-
- ▶ Un diagramme de composant décrit la façon selon laquelle les **composantes logicielles** seront réalisées.
 - ▶ Il est destiné aux **réalisateurs (développeurs)**.

Diagramme de composants

Concepts de diagramme de composant

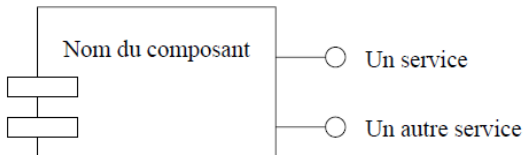
Les diagrammes de composants sont composés :

- ▶ des **composants** : des descriptions des implémentations du système,
- ▶ des **modules** : des groupes d'implémentations,
- ▶ des **dépendances** : des relations entre les diverses implémentations.

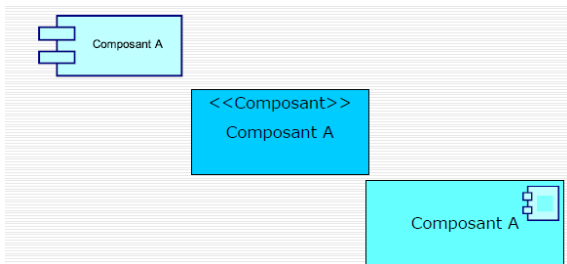


Composant

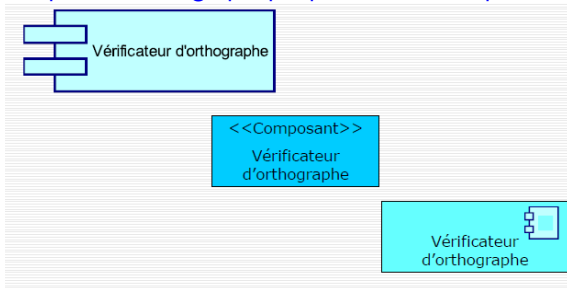
- ▶ Élément physique représentant une partie de l'implémentation du système qui peut être un :
 - code (source, binaire ou exécutable),
 - script, fichier de commande,
 - fichier de données, table, ...
- ▶ Un composant implante **des services utilisables** par d'autres composants.



Concepts du diagramme de composants : le composant



Représentation graphique possible de composant

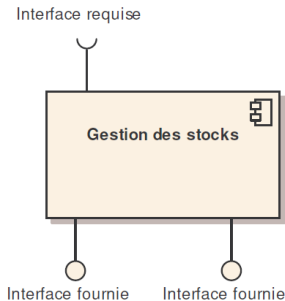


- ▶ UML propose des **stéréotypes de composants** :
 - << *document* >>: un document quelconque;
 - << *exécutable* >>: un programme qui peut s'exécuter sur un noeud (cf. diagrammes de déploiement);
 - << *fichier* >>: un document contenant du code source ou des données;
 - << *bibliothèque* >>: une bibliothèque statique ou dynamique;
 - << *table* >>: une table d'une base de données relationnelle.
- ▶ En général, un composant représente l'implantation d'une classe.

Concepts du diagramme de composants : les dépendances

Interface de composant

- ▶ Les liens entre composants sont spécifiés à l'aide de dépendances entre leurs interfaces.
- ▶ L'interface fournie d'un composant est une interface qu'il est capable de mettre en oeuvre.
- ▶ L'interface requise d'un composant est une interface dont il a besoin pour fonctionner.



Dépendance

- ▶ Représente les relations de dépendances entre composants/modules.
- ▶ Illustre l'utilisation des **services** d'un composant par un second composant.

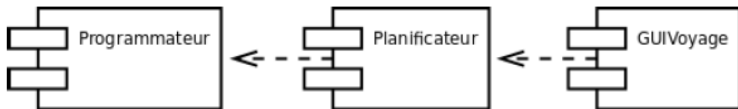
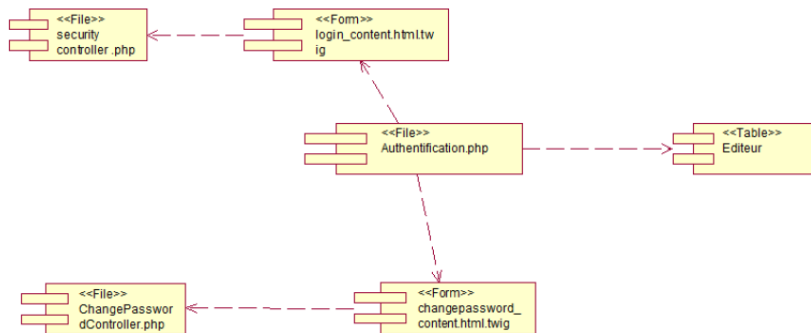


Diagramme de composants

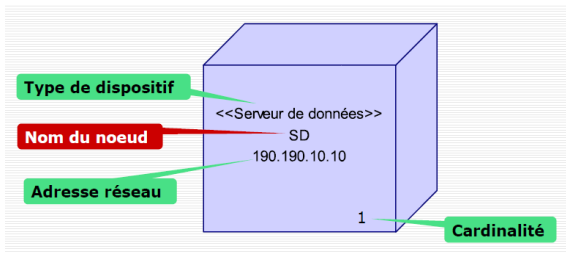


Exemple d'une application web : diagramme de composants relatif au cas d'utilisation << S'authentifier >>

- ▶ Représente la **structure physique** du système informatique et la répartition des composantes logicielles sur ce système.
- ▶ Constitue un moyen pour représenter les différentes architectures selon laquelle le nouveau logiciel peut être déployé :
 - Architecture centralisée ;
 - Architecture client/serveur à deux niveaux ;
 - Architecture client/serveur à trois niveaux;
 - Architecture client/serveur à n niveaux.
- ▶ L'environnement d'exécution ou les ressources matérielles sont appelés **noeuds**.
- ▶ Les parties d'un système qui s'exécutent sur un noeud sont appelées **artefacts**.

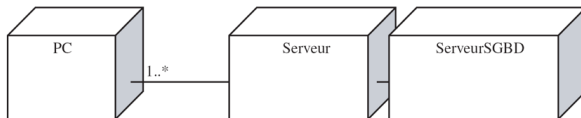
Diagramme de déploiement : les noeuds

- ▶ Chaque dispositif matériel est représenté par un noeud.
- ▶ Un noeud se représente par un cube dont le nom respecte la syntaxe des noms de classes.
- ▶ Un noeud est décrit par :
 - Un nom (obligatoire) ;
 - Un stéréotype (serveur, client, imprimante, routeur, ...) ;
 - Une adresse réseau ;
 - Une cardinalité.



Les associations : représentation

- ▶ Les noeuds sont inter-reliés par des associations.
- ▶ Chaque association entre deux noeuds représente l'existence d'un support de communication entre eux.
- ▶ Pour chaque association, on peut préciser :
 - Le nom ;
 - Le rôle de chaque côté ;
 - La cardinalité.



Les artefacts

- ▶ 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 flèche 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.

Diagramme de déploiement

Une application web d'une architecture 3-tiers

- ▶ La partie << Terminal Client >> représente le poste de l'utilisateur ainsi que son navigateur.
- ▶ La partie << Serveur d'application >> correspond au serveur web Apache traitant toutes les requêtes.
- ▶ La troisième partie << Serveur de la base de données >> comprend le système de gestion de la base de données << MYSQL >>.

