

R5.A.08
Qualité de développement

I.Borne
(isabelle.borne@univ-ubs.fr)

IB- R5.A.08

1/23

1 . Architecture logicielle

« An architecture is the set of significant decisions about organization of a software system, the selection of the structured elements and their interfaces by which the system is composed, together with their behavior as specified in the collaboration among those elements , the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style that guides this organization – these elements and their interfaces, their collaborations, and their composition. »

G.Booch, J.Rumbaugh,I.Jacobson : The UML modeling language user guide

IB- R5.A.08

3/23

COURS n°1

- ♦ 1. Architecture logicielle
 - ♦ Définition
 - ♦ Vues
- ♦ 2. Modélisation d'une architecture (1ère partie)
 - ♦ Rappels packages
 - ♦ Composants logiciels
 - ♦ Diagramme de composants en UML

IB- R5.A.08

2/23

Vers une définition

Une architecture logicielle est une représentation abstraite d'un système. C'est la structure des structures d'un système.

Elle comprend :

- ♦ des composants logiciels ;
- ♦ les propriétés externes visibles de ces composants ;
- ♦ les relations entre ces composants.

Les propriétés externes visibles font référence aux hypothèses sur un élément que peuvent faire les autres éléments, telles que les services fournis, les caractéristiques de performance, l'utilisation des ressources partagées etc

IB- R5.A.08

4/23

Pourquoi développer une architecture logicielle ?

- ♦ Pour permettre à tous de mieux comprendre le système.
- ♦ Pour permettre aux développeurs de travailler sur des parties individuelles du système.
- ♦ Pour préparer les extensions du système.
- ♦ Pour faciliter la réutilisation.

IB- R5.A.08

5/23

Utilité d'une architecture logicielle (Garlan 2000)

Compréhension : vue de haut niveau

Réutilisation : identification des éléments réutilisables

Construction : plan de haut-niveau du développement et de l'intégration des modules.

Evolution : les points où un système peut être modifié et étendu.

Analyse : analyse plus approfondie de la conception, analyse de la cohérence, analyse des dépendances

Gestion : gestion générale, identification des dépendances

IB- R5.A.08

6/23

Qu'est-ce que la description d'une architecture logicielle ?

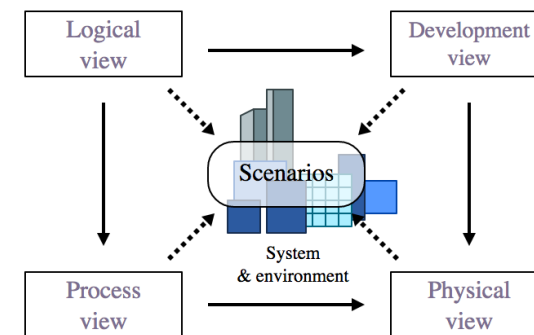
- ♦ Décrire l'organisation générale d'un système et sa décomposition en sous-systèmes ou composants
- ♦ Déterminer les interfaces entre les sous-systèmes
- ♦ Décrire les interactions et le flot de contrôle entre les sous-systèmes
- ♦ Décrire également les composants utilisés pour implanter les fonctionnalités des sous-systèmes
 - ♦ propriétés de ces composants
 - ♦ contenu (e.g., classes, autres composants)
 - ♦ machines ou dispositifs matériels sur lesquels ces modules seront déployés

IB- R5.A.08

7/23

Modéliser un système logiciel

4+1 modèles de vue de P.Kruchten



IB- R5.A.08

8/23

Vue logique d'une architecture

- ♦ Vue logique : description logique du système décomposé en sous-systèmes.
- ♦ Permet de répondre à des questions du type :
 - ♦ Quels sont les principaux composants et leurs relations ?
 - ♦ Quels sont les principaux entrepôts de données ?
 - ♦ Quels sont les protocoles d'interactions donc les besoins en infrastructure d'exécution ?
 - ♦ Quels sont les chemins critiques et sensible ?
 - ♦ Quel est le niveau de couplage et de cohésion ?
 - ♦ Etc ..

IB- R5.A.08

9/23

Vue de développement (implémentation)

- ♦ Illustre un système à partir de la perspective du programmeur et concerne la gestion du logiciel.
- ♦ Cette vue est aussi connue comme la vue d'implémentation.
- ♦ Décrit l'implémentation physique du système en termes de composants et de connecteurs.
- ♦ Pour cela elle utilise le diagramme de composants.

IB- R5.A.08

10/23

Vue des processus

- ♦ Concerne
 - ♦ les aspects dynamiques du système et explique les processus du système et comment ils communiquent ; elle cible le comportement du système au *runtime*.
 - ♦ La concurrence, la distribution, les intégrateurs, la performance, et l'échelle, etc.
- ♦ Les diagrammes UML utilisés pour représenter la vue des processus incluent le diagramme d'activité.

IB- R5.A.08

11/23

Vue physique (déploiement)

- ♦ Décrit le système selon le point de vue de l'ingénieur système.
- ♦ Concerne la topologie des composants logiciels sur la couche physique, ainsi que les connexions physiques entre ces composants.
- ♦ Elle est aussi connue comme la vue de déploiement.
- ♦ Les diagrammes UML incluent le diagramme de déploiement et des diagrammes combinés de composants et de déploiement.

IB- R5.A.08

12/23

Les scénarios

- ♦ La description d'une architecture est illustrée par un petit ensemble de cas d'utilisation, ou de scénarios qui deviennent une cinquième vue.
- ♦ Les scénarios décrivent des séquences d'interactions entre les objets et entre les processus.
- ♦ Utilisés pour identifier les éléments architecturaux et pour illustrer et valider la conception de l'architecture. Ils servent aussi de points de départ pour les tests d'un prototype d'architecture.
- ♦ Les diagrammes UML incluent le diagramme de cas d'utilisation.

2. Modélisation d'une architecture

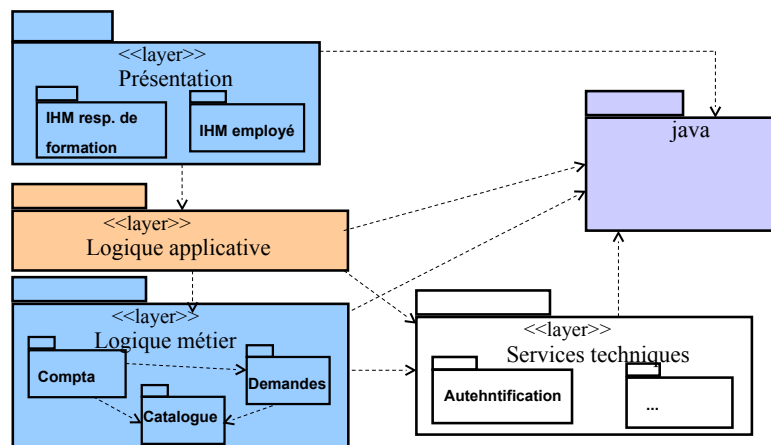
Tous les diagrammes UML peuvent être utilisés pour décrire les différents aspects d'un modèle architectural.

Trois diagrammes sont particulièrement utiles :

- ♦ diagramme de package
- ♦ diagramme de composants
- ♦ diagramme de déploiement

Diagramme de packages (rappel - voir cours R3.04)

Exemple de composition de packages pour une architecture en couche



Composants logiciels

« A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties. »
C.Szyperski

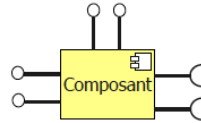
Les composants sont des spécifications d'unités fonctionnelles

- ♦ clairement définies
- ♦ sémantiquement cohérentes et compréhensibles

Développés ou acquis

Description des composants

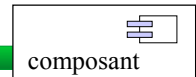
- Propriétés fonctionnelles
 - Services requis
 - Services fournis
 - Cycle de vie
- Contraintes
 - Type de communication
 - Ordonnement
- Propriétés non fonctionnelles
 - Performance, robustesse, ...



IB- R5.A.08

17/23

Diagramme de composants UML



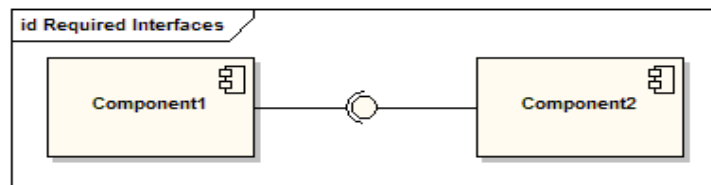
- ♦ En UML tous les éléments physiques (tels que les exécutables, les bibliothèques, les fichiers et les documents) sont modélisés avec des composants.
- ♦ Chaque composant contient l'implémentation de certaines classes du système pour lesquelles il spécifie des interfaces et implémente un service.
- ♦ Un composant « bien formé » ne dépend pas directement d'autres composants, mais des interfaces qu'ils supportent. Il est remplaçable par un autre qui implémente les mêmes interfaces.

IB- R5.A.08

18/23

Interfaces des composants

- ♦ Types d'interfaces :
 - Interface offerte ou fournie : définit la façon de demander l'accès à un service offert par le composant
 - Interface requise : définit le type de service requis par le composant



IB- R5.A.08

19/23

Quelques stéréotypes UML pour les composants

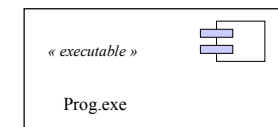
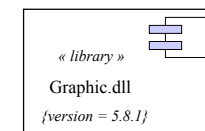
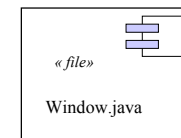
«executable» : composant binaire

«library» : bibliothèque d'objets statique ou dynamique

«table» : table d'une base de données

«file» : document contenant du code source ou des données

«document» : un document quelconque



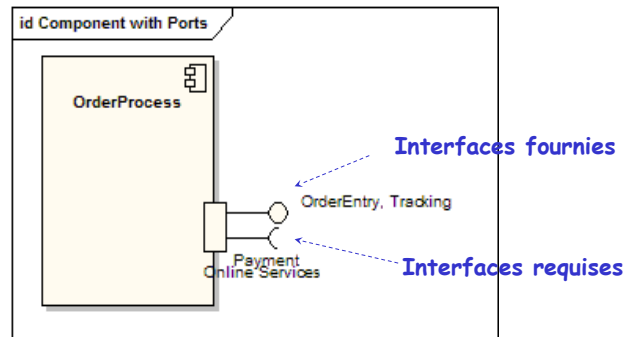
IB- R5.A.08

20/23

Ports d'un composant

C'est le point de communication du composant.

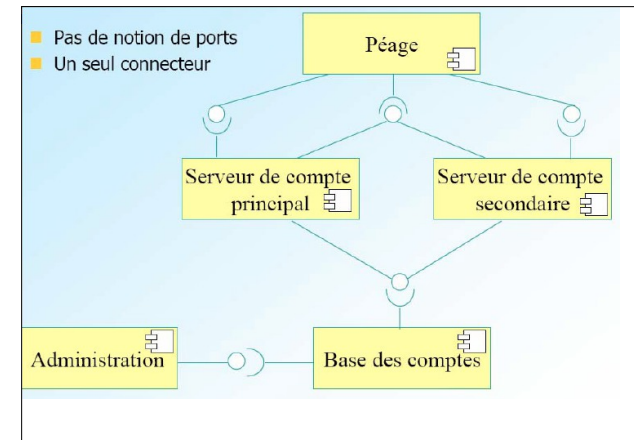
Une interface ou plusieurs interfaces peuvent être attachées à un port



IB- R5.A.08

21/23

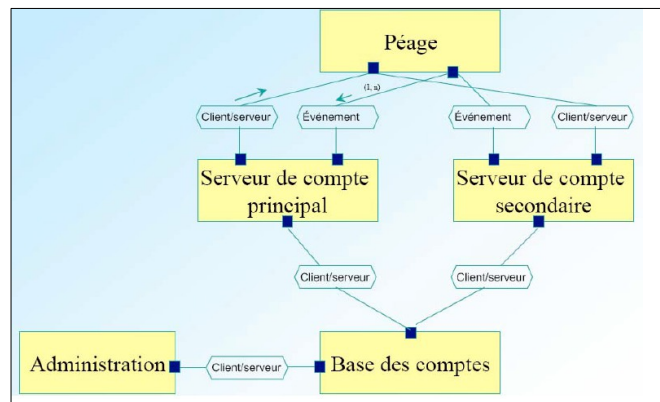
Exemple



IB- R5.A.08

22/23

Exemple de types de relation



IB- R5.A.08

23/23