



Université de Jijel
Faculté des sciences exactes et d'informatique
Département d'informatique
Classe: 2^{ème} Master SIAD



Cours en Système d'information, méthodes avancées

Chapitre 6: Conception

(générique / préliminaire/ détaillée)

Présenté par: Dr. D. Boukraa
Maître de conférences en informatique
boukraa.jimdofree.com
2020-2021

- ❖ Connaître les aspects à considérer dans la phase de conception
- ❖ Savoir croiser la solution fonctionnelle et la solution technique au niveau de la conception préliminaire
- ❖ Savoir traduire le diagramme de classes en schéma relationnel

1. Conception générique

- ❖ Développer la spécification technique proposée durant la capture des besoins techniques tout en restant indépendant des besoins fonctionnels.
- ❖ Eléments considérés
 - ❖ Elaboration du modèle de conception à base de frameworks
 - ❖ Organisation du modèle de conception à l'aide de packages
 - ❖ Elaboration du modèle d'exploitation

1. Conception générique

1. 1. Elaboration du modèle de conception à base de classes techniques

Identifier les classes techniques selon le modèle de développement logiciel adopté (5 couches, MVC...)

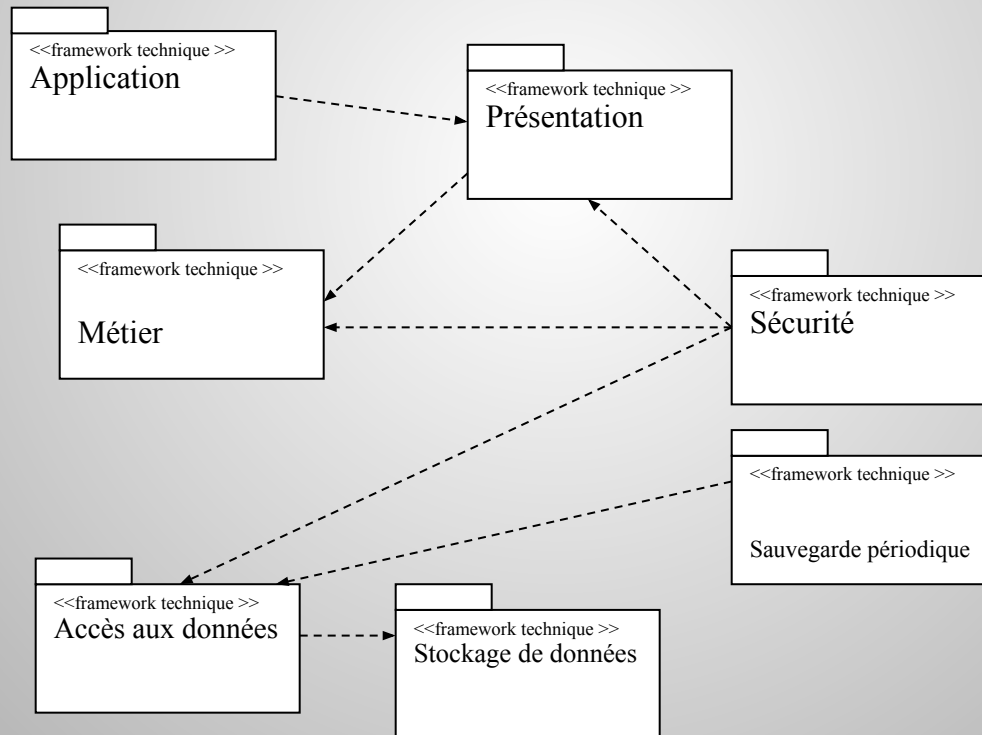
Exemple: dans le modèle à base de 5 couches, on identifie les classes techniques suivante par couche

- Présentation: fenêtre, panneau, bouton, ...
- Application: listes d'objets, règles...
- Métier: objet métier, ...
- Accès aux données: N-uplet, requête, ...
- Stockage de données: Table, clé primaire, ...

1. Conception générique

1. 2. Organisation du modèle de conception à l'aide de packages

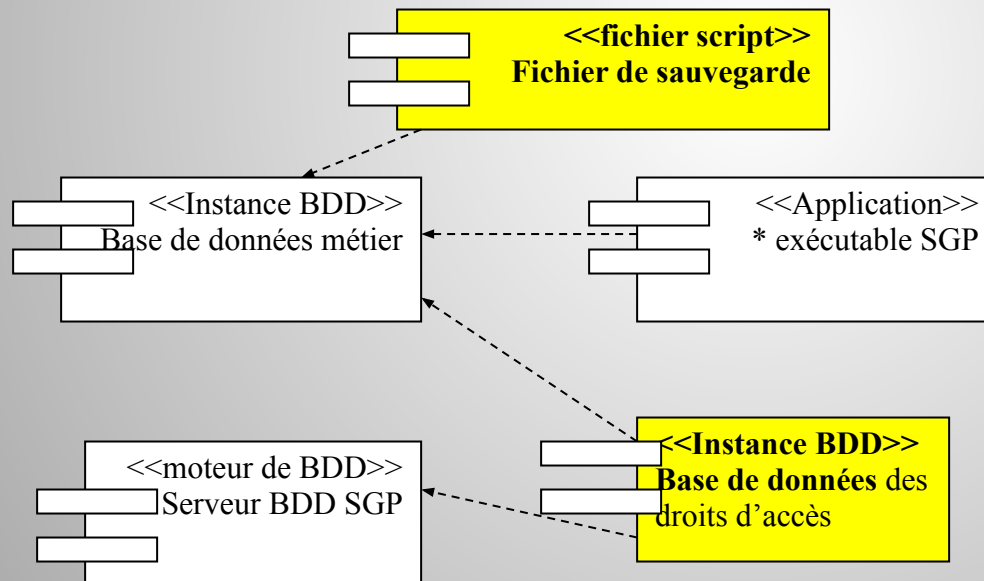
Structurer les classes techniques en frameworks (packages stéréotypés)



1. Conception générique

1. 3. Elaboration du modèle d'exploitation

- Inclure les composants nécessaires à la satisfaction des besoins techniques (ici: fichier de sauvegarde et BDD des droits d'accès)



2. Conception préliminaire

- ❖ Étape consistant à tenir compte des besoins exprimés dans la phase de capture des besoins fonctionnels dans la conception du logiciel.
- ❖ Adaptation de la conception générique en terme de modèle de composants et de configuration logicielles aux métier et domaine du système à développer.
- ❖ Première étape de la fusion qui sera suivie par le détail de conception plus tard.

2. Conception préliminaire

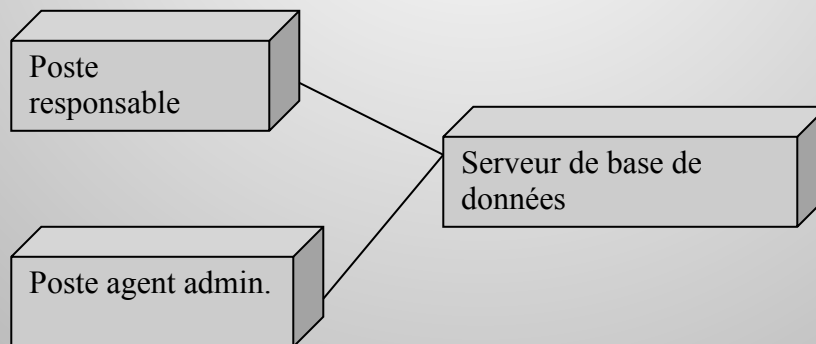
- ❖ Éléments considérés
 - ❖ Concevoir le déploiement
 - ❖ Concevoir le modèle d'exploitation
 - ❖ Concevoir le modèle logique

2. Conception préliminaire

2. 1. Concevoir le déploiement

- A cette étape là, on introduit la notion de poste de travail. Un poste de travail représente un ou plusieurs acteurs localisés sur une machine de même type.
- Personnaliser le modèle de déploiement en spécifiant le rôle de chaque élément

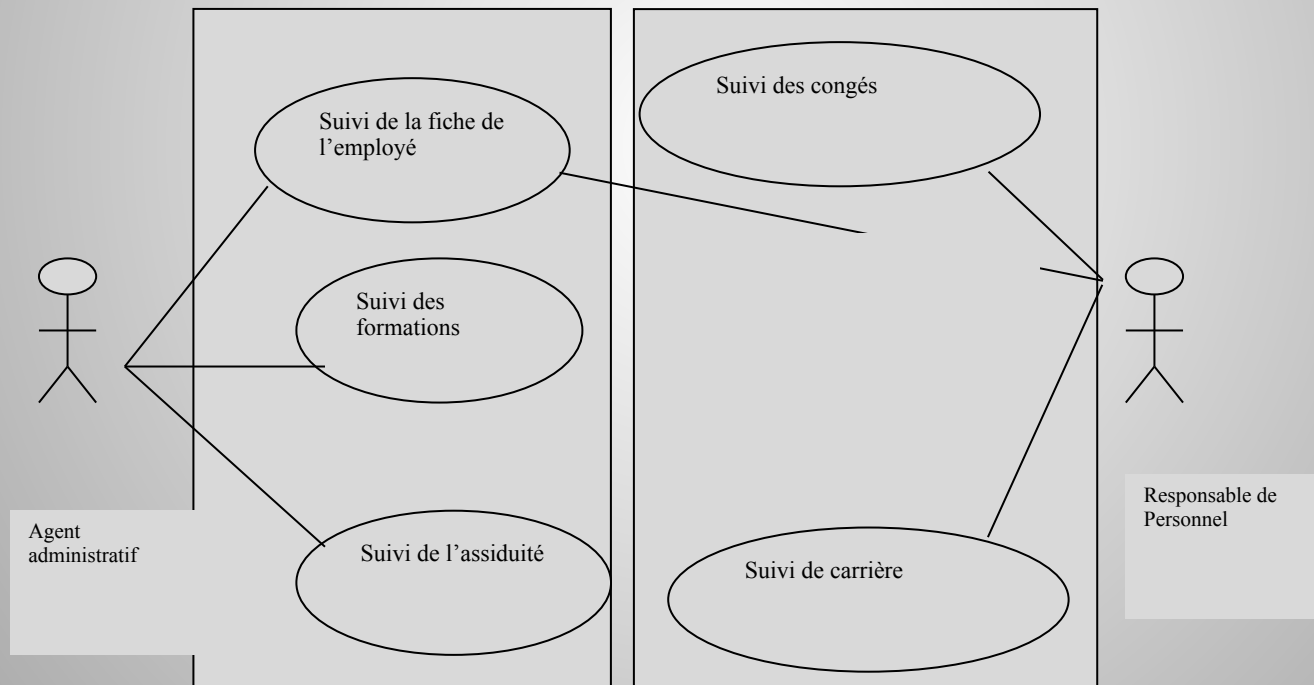
Exemple:



2. Conception préliminaire

2. 2. Concevoir le modèle d'exploitation

- ❖ Découpage du systèmes en applications

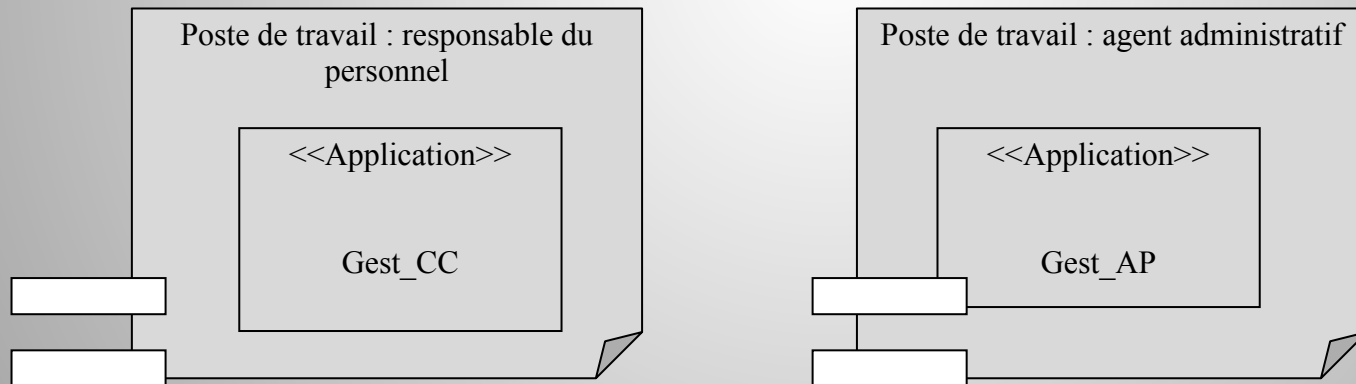


2. Conception préliminaire

2. 2. Concevoir le modèle d'exploitation

- ❖ Découpage du systèmes en applications

Spécifier l'emplacement de chaque application par rapport au modèle de déploiement

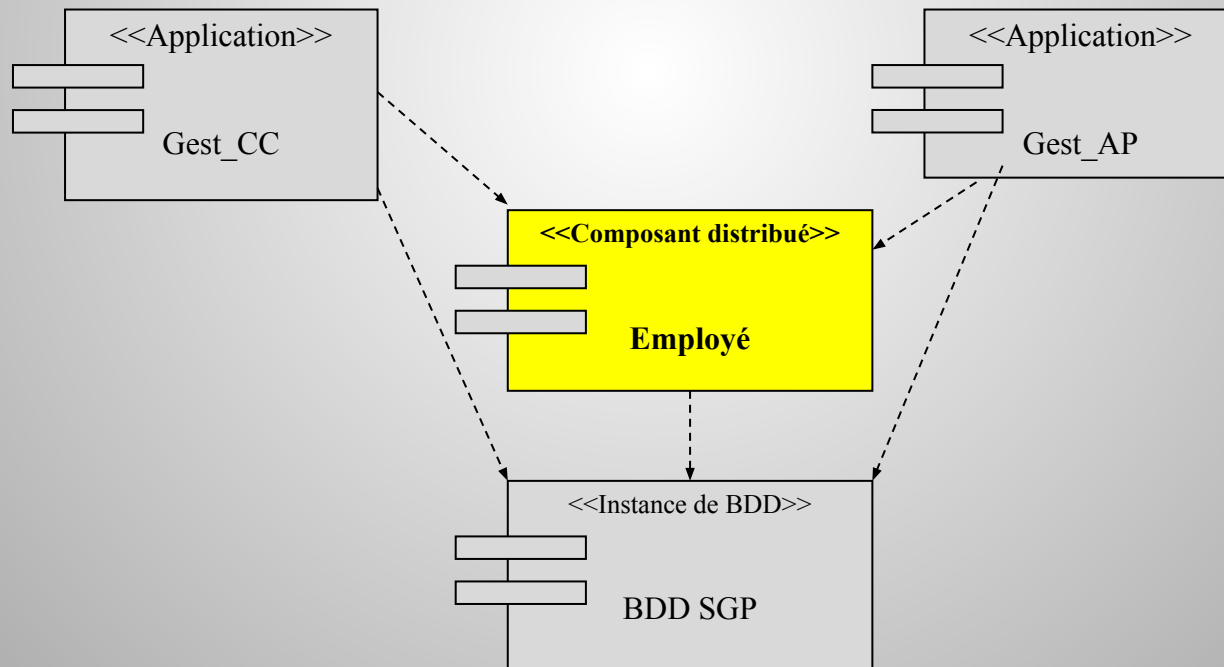


2. Conception préliminaire

2. 2. Concevoir le modèle d'exploitation

- ❖ Identification des composants distribués

Spécifier l'utilisation des composants distribués prévus dans l'étape de capture des besoins techniques



2. Conception préliminaire

2. 2. Concevoir le modèle d'exploitation

- ❖ Identification des interfaces IHM par application

| Vue IHM | Description |
|--------------------|--|
| Edition Employé | Editer les informations d'un employé. Création, modification, suppressions... |
| Sélection diplôme | Sélection d'un diplôme à assigner à un employé |
| Edition diplome | Ajout d'un nouveau diplôme avec les différentes informations |
| Sélection Employé | Sélection d'un employé à partir d'une liste pour effectuer des traitements le concernant |
| Période | Sélection d'une date à partir d'un calendrier |
| Sélection fonction | Sélection d'une fonction |
| Edition fonction | Edition d'une formations : création, modification... |
| Sélection service | Sélection d'un service |
| Edition service | Edition d'un service : création, modification ... |

2. Conception préliminaire

2. 3. Concevoir le modèle logique

Cette étape consiste donc à procéder à un découpage en catégories de conception:

On prend tour à tour chaque composant d'exploitation (applications, composants métier distribués, instance de base de données) d'une part, et chaque catégorie d'analyse d'autre part et de la projeter sur les framework abstraits qui représentent les couches logicielles.

3. Conception détaillée

- Dernière étape avant le codage.
- A cette étape, on commence à raffiner les choix de conception en incluant les spécificités du langage ou environnement de développement.
- Etapes
 - Concevoir les classes
 - Concevoir les associations
 - Concevoir les attributs
 - Concevoir les opérations

3. Conception détaillée

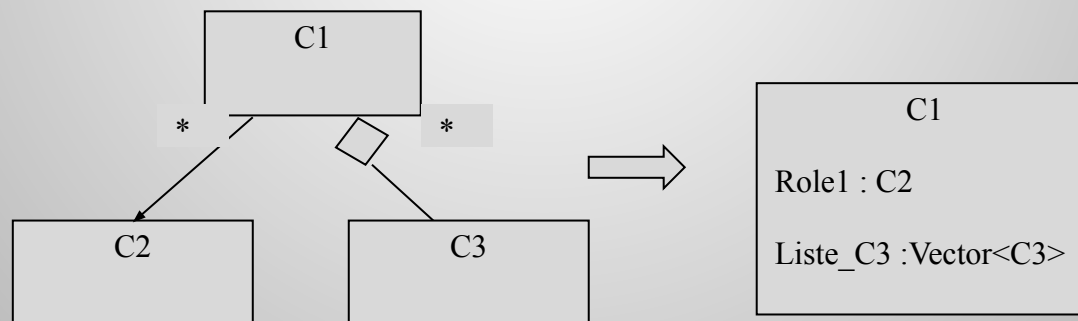
3. 1. Conception des classes

- Relativement systématique: les concepts de l'orienté-objet trouvent leur équivalence au niveau des langages de programmation, comme Java
- Si les concepts ne sont pas explicités dans le langage de programmation, les expliciter

3. Conception détaillée

3. 2. Conception des associations

- Lorsque le concept d'association n'est pas pris en charge par le langage de codage, procéder à la transformation de l'association. Cette transformation dépend du type et des multiplicités de l'association.



3. 2. Conception des associations

- Les multiplicités supérieures à 1 (attributs obligatoires)
- Les contraintes *lecture seule* et d'*ordonnement*
- La gestion des associations de type composition lors de la destruction de l'agrégat
- Les associations bidirectionnelles donnent lieu chacune à la copie des attributs et des opérations de part et d'autre.
- La classe d'association se transforme d'abord en deux associations puis chaque association sera gérée de sa part par les mêmes principes d'une association normale.
- transformer les relations n-aires en relations binaires.

3. Conception détaillée

3. 3. Conception des attributs

- Typage avec des types scalaires ou complexes.
- Prise en compte des contraintes de type lecture seule ou lecture écriture.
- Conservation de la visibilité de chaque attribut.
- Ajout, si nécessaire, des méthodes d'accès en écriture ou lecture *set<nom d'attribut>* ou *get<nom d'attribut>*
- Spécifier les méthodes de mise à jour des attributs dérivés lorsque les attributs de base changent de valeur.

3. 3. Conception des opérations

- Détailler l'algorithme nécessaire à sa mise en œuvre.

3. Conception détaillée

3. 2. Conception des associations

Exemple de la couche « accès aux données »

- Spécification de la transformation vers le modèle relationnel pour la mise en œuvre de la base de données

| Modèle objet | Modèle relationnel |
|--------------------------|---|
| Classe | Table |
| Attribut de type simple | Colonne |
| Attribut de type composé | Colonnes ou clé étrangère |
| Instance | T-uplet |
| OID | Clé primaire |
| Association | Clé étrangère ou table de liens |
| Héritage | Clé primaire identique sur plusieurs tables |