

Chapitre I: Introduction à la Conception Architecturale

Dr.Dalel Kanzari

I. Définitions

- ▶ Architecture:
Art de concevoir et de construire les éléments des constructions.
- ▶ Architecture informatique:
Art de concevoir et mettre en place la structure d'un système informatique (i.e. matériel et logiciel) en termes de composants et d'organisation de ses fonctions.
- ▶ Architecture logicielle:
Art de concevoir et réaliser un logiciel répondant aux spécifications prédéfinies (décrire le comment faire).

- ▶ architecte logiciel:
Métier chargé de concevoir et d'analyser le plan de construction d'un logiciel.

Le rôle principal est de définir une architecture (structuration) logicielle convenable répondant aux contraintes du système (spécifications)

I. Les concepts de l'architecture logiciel

L'architecture logicielle permet de :

- ▶ Décrire l'organisation générale d'un système et sa décomposition en sous-systèmes ou composants
- ▶ Décrire les composants utilisés pour implanter les fonctionnalités des sous-systèmes

❖ **Avantages:**

- **Compréhension:** compréhension des grands systèmes complexes (Abstraction des structures et des contraintes)
- **Réutilisation:** l'identification des éléments réutilisables,
- **Construction:** fourniture d'un plan de haut-niveau du développement et d'intégration des modules
- **Évolution :** mise en évidence des points où un système peut être modifié et étendu.

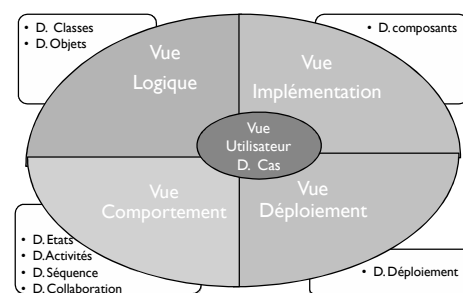
- **Analyse:** présentation d'une base d'analyse approfondie de la conception du logiciel (analyse de la cohérence, test de conformité, analyse des dépendance, etc)
- **Gestion:** contribution à la gestion générale du projet (agencement entre les différents composants, identification des dépendances et des délais, etc)

❖ **Critères de Qualité Logicielle**

- | | |
|--------------------|-------------------|
| ➤ Interopérabilité | ➤ Réutilisabilité |
| ➤ Portabilité | ➤ Extensibilité |
| ➤ Compatibilité | ➤ Efficacité |
| ➤ Validité | ➤ Autonomie |
| ➤ Vérifiabilité | ➤ Transparence |
| ➤ Intégrité | ➤ Composabilité |
| ➤ Fiabilité | ➤ Simplicité |
| ➤ Maintenabilité | |

II. La modélisation Architecturale d'un logiciel

❖ Les vue du modèle d'architecture



❖ Modélisation UML

- ▶ **Vue logique:** Description logique du système décomposé en sous-systèmes (modules + interface; UML : diagramme de paquetages)
- ▶ **Vue d'implémentation:** Description de l'implémentation (physique) du système logiciel en termes de composants et de connecteurs (UML : diagramme de composants)
- ▶ **Vue utilisateur:** Description des buts et objectifs des clients du système, et des besoins requis par la solution.

▶ **Vue de déploiement:**

Description de l'intégration et de la distribution de la partie logicielle sur la partie matérielle
(UML: diagramme combiné de composants et de déploiement)

▶ **Vue comportementale:**

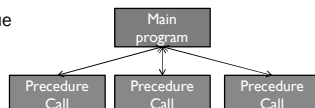
Description des aspects dynamiques du comportement du problème, sa solution d'interactions, et les divers collaborations entre les composants du système

III. Styles d'Architectures Logicielles

- ▶ Selon le niveau de granularité ou l'aspect du système souhaité, un système informatique peut utiliser plusieurs styles:

➤ **Architecture de décomposition fonctionnelle (main program & sub-routine):**

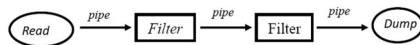
- ▶ Un programme principal appelle les composantes à tour de rôle
- ▶ Contrôle hiérarchique

➤ **Architecture Centrée sur les Données**

- Un composant central (serveur base de données) est responsable de la gestion des données
- Les composants périphériques (clients) utilisent le composant central (serveur de données)
 - ▶ Base de données
 - ▶ Blackboard
 - ▶ Datawarehouse

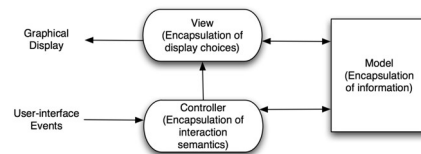
► Architecture des Flots de Données

- Plusieurs composants logiciels reliés entre eux par des flux de données
 - Par lots
 - Tuyaux et Filtres



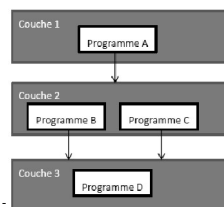
► Architecture par invocation implicite

- Orientée Événements
- Model-View-Controller



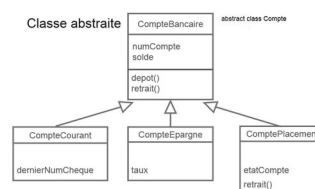
► Architecture Hiérarchique

- Organisation de composantes offrant des groupes de services en couches
 - Échanges limités aux composantes de la couche sous-jacente
- Machines virtuelles



► Architecture orienté objets

- Topologie et structure de contrôle arbitraire.
 - La durée de vie des instances peut varier
 - Présentation des relations statiques et dynamiques



► **Architecture orientée agents**

- Apprentissage des agents
- Comportement autonome des agents
- Communications inter-agents

