

Méthodes Orientées Objets : une introduction à UML

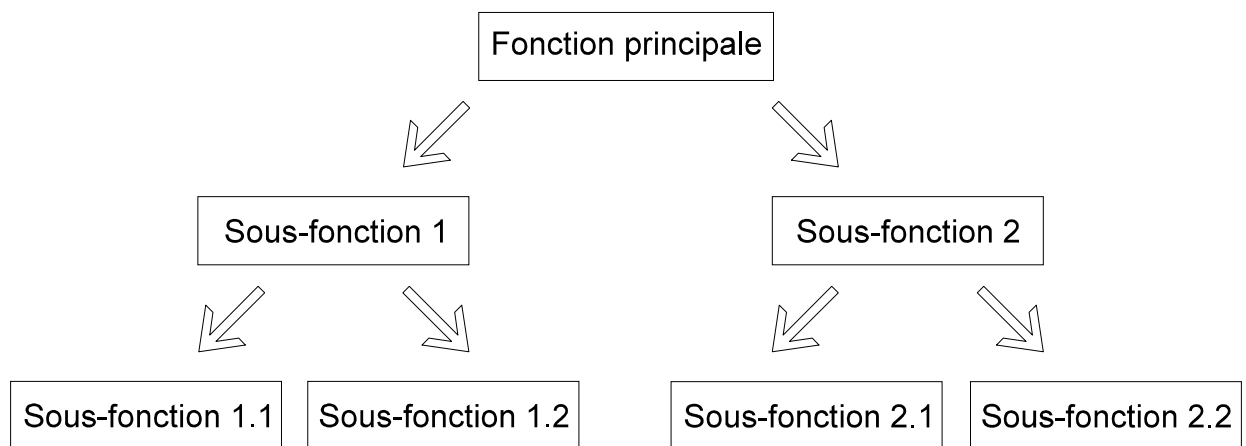
1. Les méthodes de développement de logiciels: Historique

Le Génie logiciel a vu fleurir plusieurs méthodes de développement. Certaines de ces méthodes sont reprises et utilisées par d'autres ingénieries. Par exemple la méthode SADT est utilisée dans la conception de produits et les systèmes de production. L'objectif ici est de faire un rapide survol des différentes classes de méthodes d'analyse et de conception de logiciels tout en les situant dans le temps.

- Méthodes de décomposition fonctionnelle (Années 70) : Verbe

Se sont des méthodes qui mettent l'accent sur les fonctionnalités que le système doit proposer à ses utilisateurs. SADT et SART constituent deux exemples de cette classe de méthodes, dont les caractéristiques sont :

- Décomposition du système par les fonctions ou les actions qu'il doit réaliser
- Se concentrer sur les traitements
- Les modules composant l'architecture du système = unité de traitement ou sous-programme
- Pas de masquage d'information,



- Méthodes systémiques (Années 80) : Sujet

Grâce aux travaux de Chen (1976) qui propose une notation pour construire un modèle entité-relation en vue de faire l'analyse des systèmes d'information, la modélisation effective des données est devenue une part importante de l'analyse des systèmes. Cependant, les traitements demeurent séparés des données qu'ils manipulent et sont modélisés séparément de celles-ci.

-Emergence des méthodes orientées objets (1990-1995) : Sujet + Verbe

Ces méthodes proposent d'analyser et de modéliser les systèmes en considérant de manière unifiée les données et les traitements qui leurs sont associés. Dans ce cadre, plusieurs méthodes ont été proposées : Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE...Aucune méthode ne s'est réellement imposée sur le marché. Cette période a vu un succès notable pour tout ce qui est orientée objet, aussi bien au niveau des méthodes que celui des langages de programmation.

-Unification et normalisation des méthodes orientées objets (1995-1997)

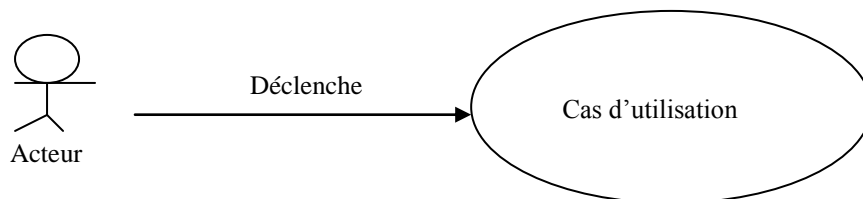
A l'origine, un travail coopératif et de synthèse entre les détenteurs des trois méthodes dominantes: OOD (G. Booch,), OMT (J. Rumbaugh), OOSE (I. Jacobson) a abouti à la proposition d'une première version d'UML (Unified Modeling Language) en 1995. Depuis Janvier 1997, d'autres révisions de l'UML ont été développées et proposées à l'OMG (Object Management Group) pour que ce langage soit reconnu comme une norme de modélisation orientée objet. Maintenant, On peut dire que ce langage constitue un standard incontournable.

2. Diagrammes UML : une vue synthétique

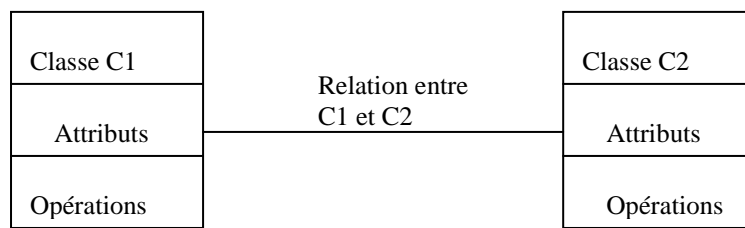
De manière générale, les développeurs utilisent des représentations graphiques pour décrire et communiquer leurs modèles. De telles représentations définissent le cadre de réflexion et d'analyse dans lequel pourra s'inscrire chaque membre de l'équipe de développement individuellement. UML (Unified Modeling Language) est un langage standard de modélisation de logiciels. Il propose plusieurs diagrammes, chacun permet de représenter la modélisation d'un aspect du système à développer. Nous donnons une vue synthétique des différents types de diagrammes proposés par UML avant de les aborder en détail dans la suite de ce cours.

➤ Diagrammes des cas d'utilisation :

représentation et structuration des besoins fonctionnels des utilisateurs : il décrit les rôles des utilisateurs du système (acteurs) et comment ces rôles se servent du système.



➤ Diagrammes de classes : représentation de la structure du système en termes de classes et de relations entre classes

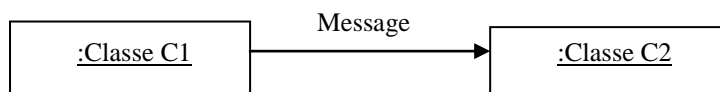


➤ Diagrammes d'objets : représentation des objets (instances des classes du modèle précédent) et de leurs liens (instances des relations du modèle précédent)



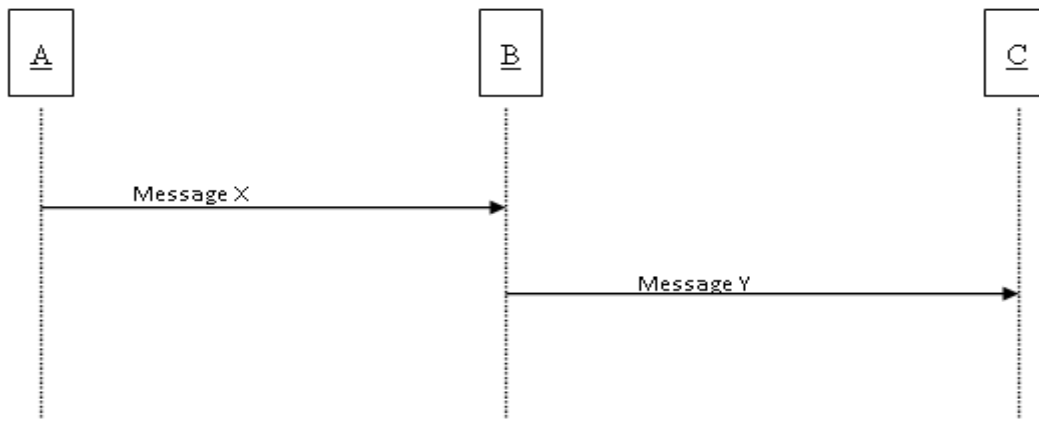
Remarque : :Client est un objet de la classe Client

➤ Diagrammes de collaboration : représentation spatiale des objets, des liens et des interactions



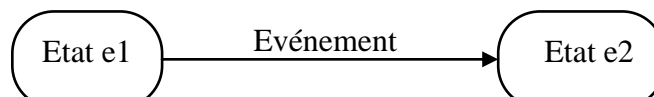
Remarque : Classe C1 (respectivement Classe C2) signifie un objet de la classe Classe C1 (respectivement de la classe Classe C2)

➤ Diagrammes de séquence : représentation temporelle des objets et de leurs interactions

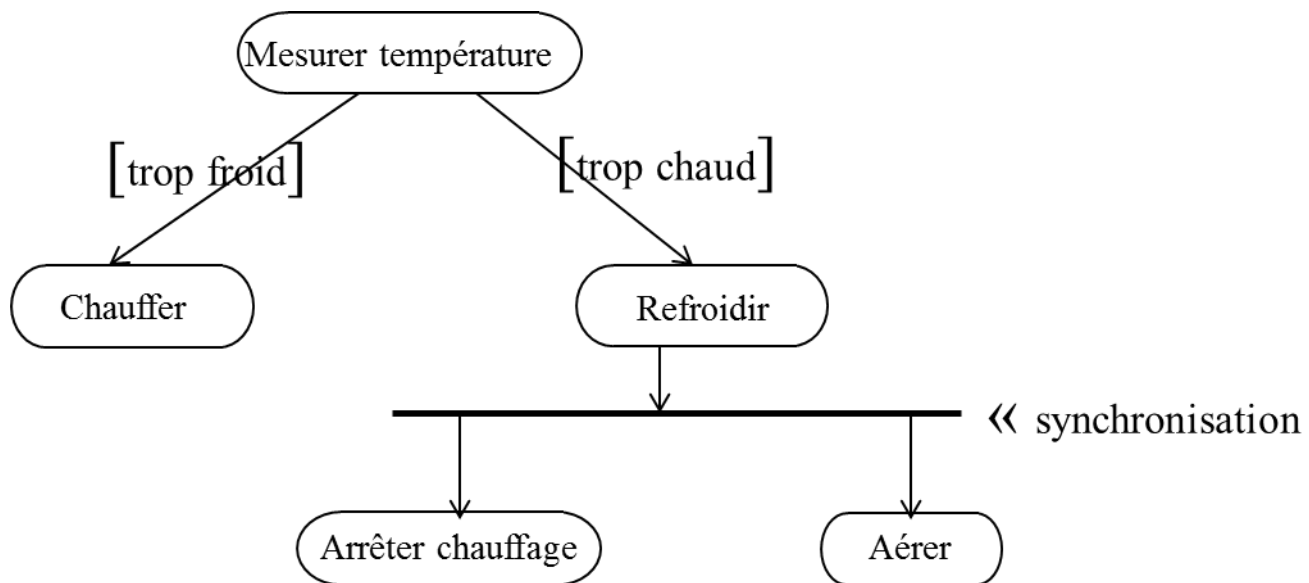


A, B, et C sont des objets des classes A, B et C

- **Diagrammes d'états-transitions** : représentation du comportement d'une classe en termes d'états et de transitions entre états.



- **Diagrammes d'activités** : représentation du comportement d'une opération en termes d'actions



- **Diagramme de composants**: représentation de l'architecture logicielle du système en termes de modules
- **Diagrammes de déploiement** : représentation du déploiement des composants sur les dispositifs matériels

Pour donner une idée sur le lien pouvant exister entre ces différents modèles lors de l'analyse et la conception d'un système, on peut dire qu'ils se complètent en répondant chacun aux trois questions fondamentales suivantes : **que doit faire le système ? sur qui il le fait ? quand et comment il le fait ?**. Précisons un peu plus à propos de ces questions qui caractérisent les trois aspects à analyser lors du développement d'un système logiciel :

- **Que doit faire le système** (quelles sont les fonctionnalités ou les services que le système doit proposer) pour satisfaire les besoins de ces utilisateurs (appelés acteurs) ?:
c'est le modèle des cas d'utilisations qui décrit la réponse à cette question en proposant pour chaque classe d'utilisateur un ensemble de services ou d'unités fonctionnelles appelés cas d'utilisation.
- **Sur qui il le fait ?** autrement dit, quelles sont les entités qui par leur coopération (on dit aussi interaction) vont devoir mettre en oeuvre les cas d'utilisation ?
Se sont le modèle de classes à partir duquel on fabrique le modèle d'objets qui décrit ces entités.
- **Quand et comment il le fait ?** quels sont les messages (qui s'échangent entre les objets du système) et les activités (les suites d'actions ou d'opérations) qu'ils déclenchent et à quel moment (aspect temporel)?
Se sont le modèle de séquence ou de collaboration (pour les échanges de messages), le modèle d'activités et le modèle de comportement (diagramme état/transition) qui spécifient la réponse à cette question.