

DIAGRAMME DE CLASSES

Quand un développeur (maître d'oeuvre) est amené à construire une application, il commence par étudier le besoin formulé par le client (maître d'ouvrage). Pour ce faire, il interroge les parties prenantes du projet (les différents utilisateurs, les managers, etc.) au cours d'entrevues, de réunions, etc. Il va consulter les documents existants (les factures, les bons de commandes, etc.). Ainsi, le développeur recueille beaucoup d'informations.

Il lui faut alors organiser ces informations : il doit éliminer les informations inutiles et comprendre les rapports qui existent entre les informations restantes.

Pour formaliser son travail, le développeur doit utiliser un langage. Un des langages le plus répandu est UML (*Unified Modeling Language*) qui fournit un ensemble de diagrammes. Parmi ces diagrammes, le plus utilisé est le diagramme de classes.

Le diagramme de classes permet essentiellement de documenter le projet et de faciliter la communication avec les autres membres du projet. On pourra aussi générer une partie du code source de l'application à partir des diagrammes de classes (soit manuellement soit automatiquement à l'aide d'un outil de génération de code).

Le développeur modélise les concepts du problème qu'il doit informatiser. Il est un peu comme un architecte qui conçoit des plans avant la construction d'un bâtiment. Ces plans permettent de communiquer avec le client, d'affiner les propositions. Ces plans serviront aussi aux différents corps de métiers (maçons, électriciens, etc.) qui interviendront dans la construction.

Remarque : on peut être amené à construire un diagramme de classes à partir du code source d'une application dans un but de documentation. On appelle cela la rétro-conception.

A priori, un diagramme de classes est plutôt destiné à être utilisé dans un contexte de programmation objet. En fonction du langage cible, on utilisera ou non certaines possibilités offertes par UML : par exemple, nous n'utiliserons pas l'héritage multiple car il n'existe pas en Java.

Nous verrons ultérieurement que nous pourrons utiliser les diagrammes de classes dans un contexte de bases de données relationnelles moyennant quelques aménagements.