

DE UML AUX BASES DE DONNEES

Corina CIOBANU^{1*}, Anastasia GHERMAN¹

¹Universitatea Tehnică a Moldovei, Facultatea Calculatoare, Informatică și Microelectronică, Departamentul Inginerie Software și Automatică, grupa FI-191, Chișinău, Republica Moldova

*Auteur correspondant: Ciobanu Corina ciobanu.corina@isa.utm.md

Résumé. *Comme n'importe quel type de projet, un projet informatique nécessite une phase d'analyse, suivi d'une étape de conception. Pour mieux comprendre un projet nous utilisons des modèles et des diagrammes, qui sont des représentations d'une application du monde réel. Les modèles offrent une vue abstraite du système, alors que les différents diagrammes en proposent des représentations concrètes. Les modèles UML (Unified Modeling Language) représentent les systèmes selon différents niveaux de détail. Les modèles UML contiennent des éléments de modèle, tels que des acteurs, des cas d'utilisation, des classes et des packages, et un ou plusieurs diagrammes qui montrent une perspective spécifique d'un système. Un modèle peut également contenir d'autres modèles plus détaillés. La représentation visuelle d'un système fournie par des diagrammes UML peut fournir une vue à la fois complète et détaillée du concept et de la conception d'une application.*

Les mots clé : *modélisation, diagrammes, étapes de la modélisation, types de diagrammes UML, base de données, structure de système, applications orientées objet.*

Introduction

« UML (Unified Modeling Language) est devenu le langage standard de modélisation des besoins des entreprises et des logiciels applicatifs. Néanmoins, UML ne se limite pas à la modélisation d'applications orientées objet. »[1] En fait, UML s'adapte très facilement aux circonstances. Il permet un grand nombre de modélisations, créant des modèles pour comprendre les processus, les enchaînements d'événements, les séquences de requêtes, les applications, les bases de données, les architectures. Alors, il peut représenter un grand nombre de modèles graphiques différents il est possible d'inclure toute la conception d'une application et de la base de données grâce à un seul langage partagé par tous. UML permet la représentation schématique de la structure d'un système. Il est plus facile de comprendre un dessin, que de lire un grand nombre de paragraphes de texte décrivant la même information. En outre, cela permet de créer des liens entre les équipes de développement, sans ignorer les composants importants. Lorsque des changements se produisent, les diagrammes UML peuvent être mis à jour sans affecter l'efficacité du travail. Ainsi, l'utilisation d'UML ne se limite pas à une simple conception d'une base de données. Ici, il doit être élevé à un niveau supérieur pour créer une structure de système.

Etapes de la modélisation et notions UML

Le développement du logiciel comporte plusieurs étapes illustrées dans la figure 1. «Chaque niveau de modélisation montre les activités majeures pratiquées et les éléments clés d'UML qui soutiennent cette activité.» [2] Nous pouvons distinguer deux types de modèles : logique et physique. Le premier type se compose d'entités et d'attributs et comprend des relations entre les différentes entités. Il intègre des éléments constitutifs mais n'est pas lié à un logiciel ou à un système de gestion de base de données particulier.

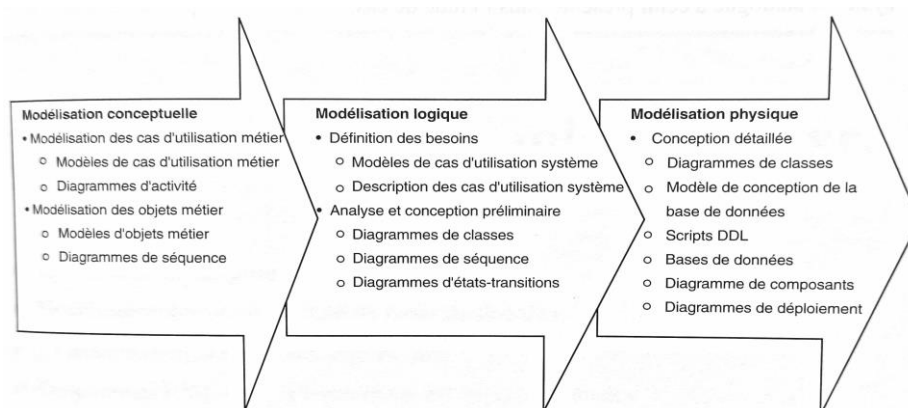


Figure 1. Etapes de la modélisation et notion UML correspondantes

Le modèle physique ou de déploiement fournit un modèle détaillé de la façon dont les composants seront déployés dans l'infrastructure du système. Il détaille les capacités du réseau, les spécifications du serveur, les exigences matérielles et d'autres informations liées au déploiement du système proposé. En plus de la modélisation proprement dite, le travail d'équipe doit également être pris en compte. "Nous avons besoin d'une méthode que nous pouvons adapter pour faire ce que nous voulons quand nous le désirons et avec les éléments de son choix. UML offre la possibilité de travailler en équipe même si chaque personne travaille sur différentes parties des applications. Il existe de nombreux outils comme Rational Rose, qui peuvent gérer le cycle de vie en utilisant UML." [3]. En fait, n'importe quel outil nous utilisons, la structure des diagrammes UML reste la même. Les diagrammes peuvent être utilisés pour collecter les besoins, décrire le déploiement et tout ce qui se situe entre les deux. Dans le tableau 1, nous voyons une brève description d'UML.

Tableau 1

Description des diagrammes UML

Diagramme	Description
Cas d'utilisation	Le diagramme de cas d'utilisation, est un modèle des fonctions souhaitées pour le système et de l'environnement dans lequel s'exercent les processus métier. Ce modèle sert de contrat entre le client et les développeurs.
Interaction	Les diagrammes d'interaction sont soit des diagrammes de séquence, soit des diagrammes de collaboration, chacun d'eux montrant les interactions d'objets s l'intérieur du système. Ils peuvent servir à comprendre les requêtes qui affecteront la base de données
Activité	Les diagrammes d'activité montrent le déroulement d'un processus. Ils peuvent être utilisés pour donner une vue de haut niveau d'un processus métier et de la façon dont il opère.
Etats-transitions	Un diagramme d'états-transitions met en évidence le comportement dynamique du système ou d'objets appartenant au système.
Classe	Les diagrammes de classes sont des modèles logiques qui montrent la structure élémentaire du système.
Base de données	Le diagramme de base de données décrit la structure de la base de données et comprend les tables, les colonnes, les contraintes, etc.
Composants	Les diagrammes de composants montrent le stockage physique de la base, comprenant le SGBD, les espaces de tables et les partitions, des applications ainsi que les interfaces utilisées pour accéder à la base.
Déploiement	Les diagrammes de déploiement montrent la configuration matérielle utilisée pour la base de données et pour les applications

De la modélisation à la conception de bases de données

Comprendre une entreprise et son métier est une tâche très difficile. Pour construire le modèle d'entreprise on doit rassembler des informations sur la façon dont chacun, depuis l'employé des expéditions jusqu'au PDG, pense que l'entreprise fonctionne et effectue son travail.

La meilleure façon de comprendre l'information recueillie auprès des nombreux participants est de modéliser leurs descriptions. La première étape consiste à construire les diagrammes de cas d'utilisations. Il y a quelques concepts que nous distinguons à cette étape :

- Acteur – Personne ou système externe qui présente une interface avec le système ;
- Cas d'utilisation – Suite complète d'actions initiée par un acteur, que le système exécute pour apporter un bénéfice à cet acteur ;
- Diagramme de cas d'utilisation – Diagramme montrant les cas d'utilisation et leurs relations avec les acteurs et les autres cas d'utilisations.

Aussi, certains symboles sont connus que nous introduisons dans la figure 2.

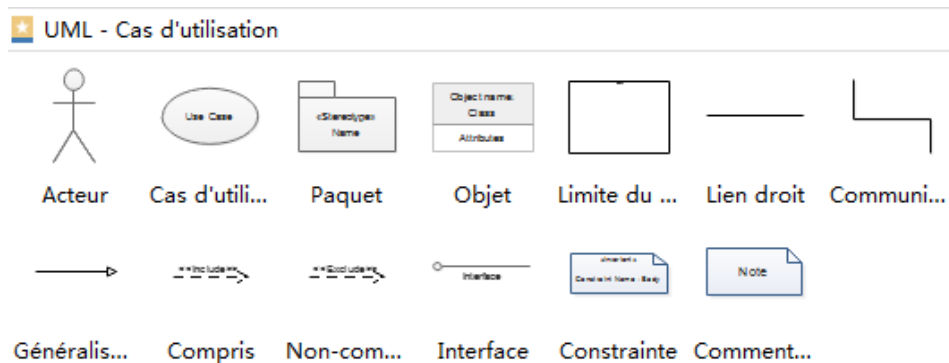


Figure 2. Eléments usuels des diagrammes de cas d'utilisation

La seconde étape de modélisation est la définition des besoins. Nos buts en définissant les besoins vont être de :

1. Etablir la portée du système que nous allons construire ;
2. Acquérir une compréhension détaillée des capacités que devra posséder le système.

Ce n'est qu'après avoir décrit le système plus en détail que nous pouvons passer à l'analyse et conception préliminaire.

Notions concernant cette étape :

- Diagramme de classe – Diagramme qui montre les classes, leur interrelation et leurs relations avec d'autres éléments de modélisations ;
- Association – Relation entre deux éléments de modélisation ;
- Généralisation – Relation entre des éléments de modélisation montrant qu'un élément (la sous-classe) et un „type de” autre élément (la superclasse) ;
- Agrégation – Relation entre des éléments de modélisation indiquant qu'un élément est une „partie de” un autre élément (l'agrégat) ;
- „Extends” – Stéréotype d'une relation indiquant un cas d'utilisation facultatif qui peut être exécuté en fonction d'un critère spécifique ;
- „Includes” - Stéréotype d'une relation indiquant un cas d'utilisation facultatif inséré dans un ou plusieurs autres cas d'utilisation à un ou plusieurs endroits ;
- Etats transitions – Diagramme qui montre le comportement dynamique d'une classe.

Dans la figure 3 nous pouvons voir les symboles utilisés pour construire les diagrammes de classes :

La troisième étape est la conception de la base de données. Le diagramme de classe constitue l'épine dorsale pour les bases de données. Nous devons réaliser une correspondance entre les attributs et les colonnes.

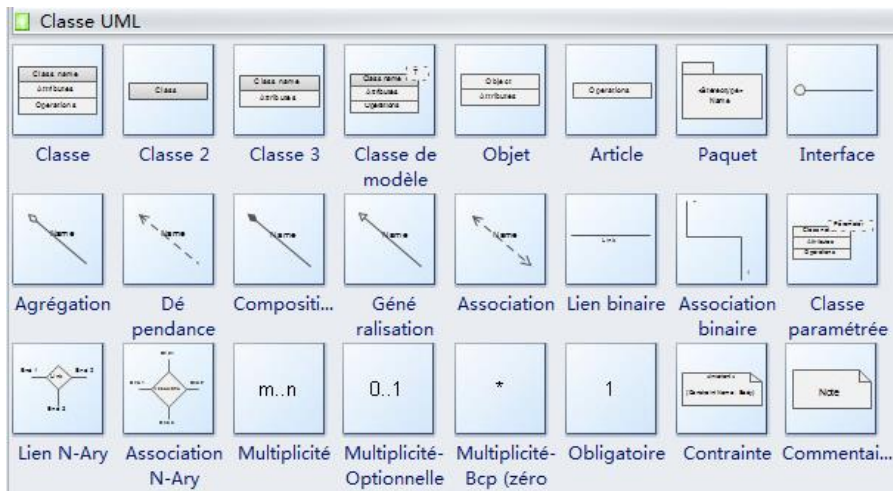


Figure 3. Eléments usuels des diagrammes de classes

Le nombre de table dans la base de données peut différer du diagramme de classe, surtout dans la relation plusieurs-à-plusieurs. Nous décrivons ci-dessous les éléments des diagrammes :

- Table – Groupement d’information dans une base de données concernant le même sujet ;
- Colonne – Composant d’une table ne contenant qu’un seul attribut de la table ;
- Clé primaire – Clé candidate retenue pour identifier les lignes dans une table ;
- Clé étrangère – Colonne ou ensemble de colonnes dans une table qui correspond à la clé primaire d’une autre table ;
- Vue – Table virtuelle qui, du point de vue d’utilisateur, se comporte exactement comme une table normale mais ne possède cependant pas d’existence propre ;
- Procédure stockée – Fonction programmée indépendante qui s’exécute normalement sur le serveur ;
- Domaines – Ensemble des valeurs valides pour un attribut ou une colonne.

Conclusion

En conclusion, UML peut être défini comme un langage de modélisation pour capturer les aspects architecturaux, comportementaux et structurels d'un système. Nous construisons des modèles pour communiquer la structure et le comportement souhaités de notre système. Nous construisons des modèles pour visualiser et contrôler l'architecture du système. Nous construisons des modèles pour mieux comprendre le système que nous construisons, exposant souvent des opportunités de simplification et de réutilisation. Et nous construisons des modèles pour gérer les risques L’objectif d’UML peut être défini comme un mécanisme de modélisation simple pour modéliser tous les systèmes pratiques possibles dans l’environnement complexe d’aujourd’hui.

Nous remercions Mme Daniela Istrati, lecteur universitaire au Département Génie Logiciel et Automatique, Université Technique de Moldova, pour l’aide à l’élaboration de cet article.

Bibliographie :

1. Eric J. Naiburg et Robert A.Maksimchuk „Conception de bases de données avec UML”, CAMPUSPRESS, 2002.
2. Article publié par edrawsoft.com sur la thème „Différents diagrammes UML – Objectif et utilisation” <https://www.edrawsoft.com/fr/uml-introduction.html>
3. Martin Seidl, Marion Scholz, Christian Huemer and Gerti Kappel „UML Classroom: An Introduction to Object-Oriented Modeling”, Springer Publishing 2015