1. Introduction

Lors de ce chapitre, nous allons identifier les diagrammes de classes, de cas d’utilisation, les modèles de données réalisés pour mettre en œuvre l’architecture OCP SAFI proposée. La motivation fondamentale de la modélisation est de fournir une démarche antérieure afin de réduire la complexité du système étudié lors de la conception et d’organiser la réalisation du projet en définissant les modules et les étapes de la réalisation. Plusieurs démarches de modélisation sont utilisées. Nous adoptons dans notre travail une approche objet basée sur un outil de modélisation UML. En fait, UML (Unified Modleing Language) est un standard ouvert contrôlé par l’OMG, un consortium d’entreprises qui a été fondé pour construire des standards qui facilitent l’interopérabilité et plus spécifiquement, l’interopérabilité des systèmes orientés objet. UML est issu de l’unification de nombreux langages de modélisation graphique orientée objet. Il unifie à la fois les notations et les concepts orientés objets.

1. Conception de l’architecture
   1. Identification des diagrammes
   2. Pour modéliser l'architecture du problématique rencontrée, nous allons identifier deux diagrammes: ● Les diagrammes de cas d’utilisations représentent un intérêt pour l’analyse des besoins métier ce qui nous permettra de démarrer l’analyse orientée objet et identifier les classes candidates. ● Un diagramme de classes est une collection d'éléments de modélisations statiques (classes, paquetages...), qui montre la structure d'un modèle. Les classes sont liées entre elles par des associations. Une association permet d’exprimer une connexion sémantique bidirectionnelle entre deux classes.
   3. Conception des couches Cette phase consiste à enrichir la description du procédé, de détails d’implémentation afin d’aboutir à une description très proche d’un programme. Nous allons modéliser toute l'architecture en diagramme de cas d'utilisation, d'activités et de classes.

2.2.1. Diagramme de cas d'utilisation

Le diagramme suivant représente le cas d'utilisation de notre architecture (figure16). Il décrit le comportement du système du point de vue utilisateur. En effet, l’utilisateur va choisir les schémas XML source et cible, le système réalise le Matching entre les schémas pour avoir les valeurs de correspondances entre les entités des schémas. Ces valeurs de correspondances seront filtrées, on aura des valeurs de correspondances plus précises. Ensuite, l’utilisateur réalise le Mapping et le résultat sera généré en XQuery.

2.2.3 Diagramme de classes Le diagramme de classes représenté dans la figure18 suivante décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes.

3. Conclusion Dans ce chapitre, nous avons identifié les diagrammes de cas d’utilisation, d'activités et de classes pour faciliter la réalisation de notre prototype. Dans le chapitre suivant nous montrerons les étapes, plus en détails, que nous avons suivies pour implémenter et réaliser notre solution.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Chapitre 7 Réalisation

1. Introduction

L’implémentation est la phase la plus importante après celle de la conception. Le choix des outils de développement influence énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système. Dans ce chapitre, nous allons commencer par la citation des languages utilisés. Ensuite, on verra les captures