



UNIVERSITÉ DE NANTES

Projet CSA

Auteur :
Thibaut PICHAUD

Auteur :
Julien OUVRARD

pour novembre 2015

Table des matières

1	Introduction	2
2	Analyse, specs et concept	2
2.1	Meta-Modèle M2	2
2.1.1	Le composant	2
2.1.2	Le connecteur	2
2.1.3	La configuration	2
2.2	Modèle M1	3
2.3	Implémentation	3
3	Vue du méta-modèle M2	4
4	Vue du modèle M1	4
5	Implémentation de l'H-ADL	4
6	Conclusion	5
7	Annexes	6

1 Introduction

Dans le cadre du module CSA (Component and Software Architecture), nous avons eu à réaliser un H-ADL (Home ADL), en proposant un méta-model M2, un modèle M1 et une implémentation de ceux-ci. Pour se faire, nous nous sommes basés sur les définitions de base d'un ADL et le modèle de représentation UML pour modéliser nos systèmes. Ce projet est scindé en deux parties, la première comprenant l'analyse, les specs et le concept de notre H-ADL. La deuxième, plus axée sur la programmation, est l'implémentation de l'H-ADL grâce à un exemple de connection Client/Serveur.

2 Analyse, specs et concept

2.1 Meta-Modèle M2

La couche objet M2 correspond à un méta-modèle. Elle est constituée de trois composants principaux, qui sont :

- Un composant
- Un connecteur
- Une configuration

2.1.1 Le composant

Un composant est une unité de composition qui spécifie une ou plusieurs fonctionnalités. Ils sont préfabriqués, pré-testés, s'auto-contiennent, et sont indépendants entre eux. Une de leurs autres distinctions est qu'ils sont configurables, et un mécanisme d'introspection les rends facile à utiliser. Ces composants fonctionnent en fournissant des services avec l'aide de besoins spécifiques et propre à eux-mêmes. Enfin, un composant a trois dimensions : son domaine(connexe, etc...), son mode d'expression(texte, graphique...) et son niveau d'abstraction(architecture détaillée, architecture haut niveau...).

2.1.2 Le connecteur

Un connecteur est une interaction entre deux ou plusieurs composants. Il en existe trois types :

- Le connecteur implicite
- Le connecteur explicite
- Le connecteur énuméré

2.1.3 La configuration

Une configuration est un graphe de composants et de connecteurs qui définit la façon dont ils sont reliés entre eux. Cette notion permet de vérifier plusieurs points

dans le méta-modèle :

- Si les composants sont bien reliés
- Si les interfaces de ces composants s'accordent
- Si leurs connecteurs permettent une communication correcte

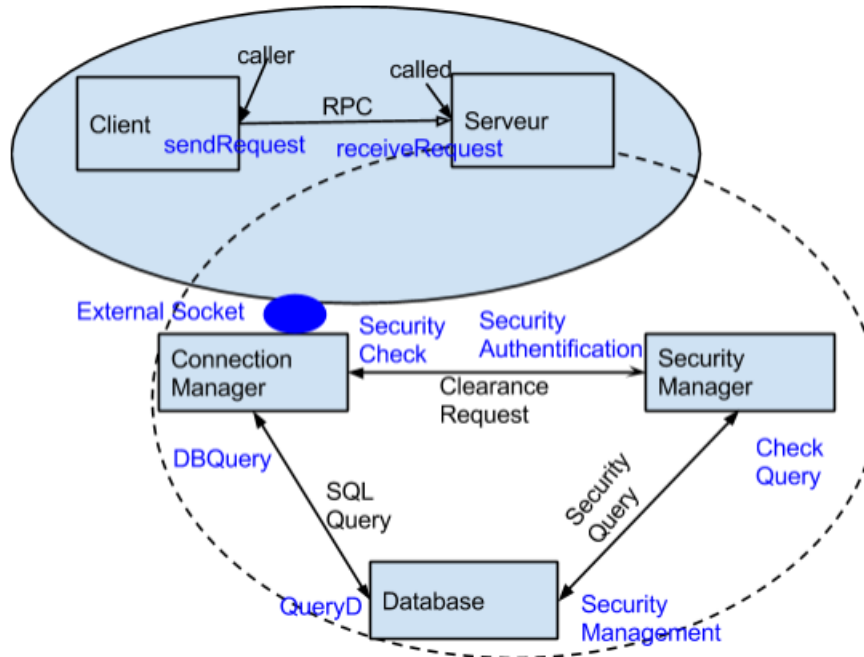
Une configuration permet aussi :

- d'insérer un verrou
- de connaître le potentiel de performance
- de connaître le potentiel de sécurité
- de connaître le potentiel de fiabilité

2.2 Modèle M1

Le modèle M1 est une spécification du méta-modèle M2, il permet de concrétiser les concepts créés dans le M2. Les classes utilisées dans cette couche sont donc des classes concrètes et non abstraites. Par exemple, on peut voir que dans le M2, les composants utilisent des interfaces, mais concrètement ces interfaces sont des ports ou des services. Dans le M1 nous n'auront donc pas affaire à des instances de la classe *interface* mais à celle de *port*.

2.3 Implémentation



3 Vue du méta-modèle M2

Les différents éléments (voir figure 1 en annexe) du méta-modèle sont :

Composant : Un composant est un élément générique du méta-modèle.

Configuration : Une configuration est un composant composite.

Connecteur : Un connecteur est un composant liant plusieurs composants.

Glue : Élément permettant de faire la jonction entre les rôles d'un connecteur.

Interface : Partie d'un composant effectuant la connexion avec ce dernier.

Service : Une "fonction" d'un composant appelée par un autre composant en passant par les ports.

Port : Interface de connexion d'un composant

Rôle : Interface de connexion d'un connecteur

PointConnexion : "Super classe" des ports et rôles

TypeConnexion : Permet de définir le type de la connexion du port ou du rôle

Propriété : Permet de spécifier un composant

ContrainteTechnique : Spécification d'un composant afin de lui donner des limites

4 Vue du modèle M1

Le diagramme M1 (voir figure 2) utilise pour chaque connecteur les rôles fournis et requis appelés respectivement *called* et *caller*, ainsi qu'une *glue*. On remarquera que le client est un *composant atomique* alors que le serveur est une *configuration*. Code couleur du schéma :

Violet : Ports (clair : Requis, foncé : Fournis)

Vert : Services (clair : Requis, foncé : Fournis)

Jaune : Composants atomique

Bleu : Connecteur

Gris : Glue

Orange : Role (foncé : Requis, clair : Fournis)

Rose : Configuration

5 Implémentation de l'H-ADL

L'implémentation de ce langage se fera dans une étape ultérieure au rendu de ce rapport.

6 Conclusion

Ce projet avait pour but de nous introduire dans l'univers de l'Architecture logicielle, en modélisant notre propre langage H-ADL. Pour ce faire, nous avons à créer nos propres méta-modèles et modèles. Ces objectifs ont été pour la plupart atteints. Les étapes successives de ce projet nous ont aidés à voir les relations entre les différentes couches objets, en particulier celles entre le M2 et le M1. Grâce à ces nouvelles connaissances, nous devrions réussir la prochaine étape de ce projet (l'implémentation de notre langage à l'aide d'une application Client/Serveur) haut la main.

7 Annexes

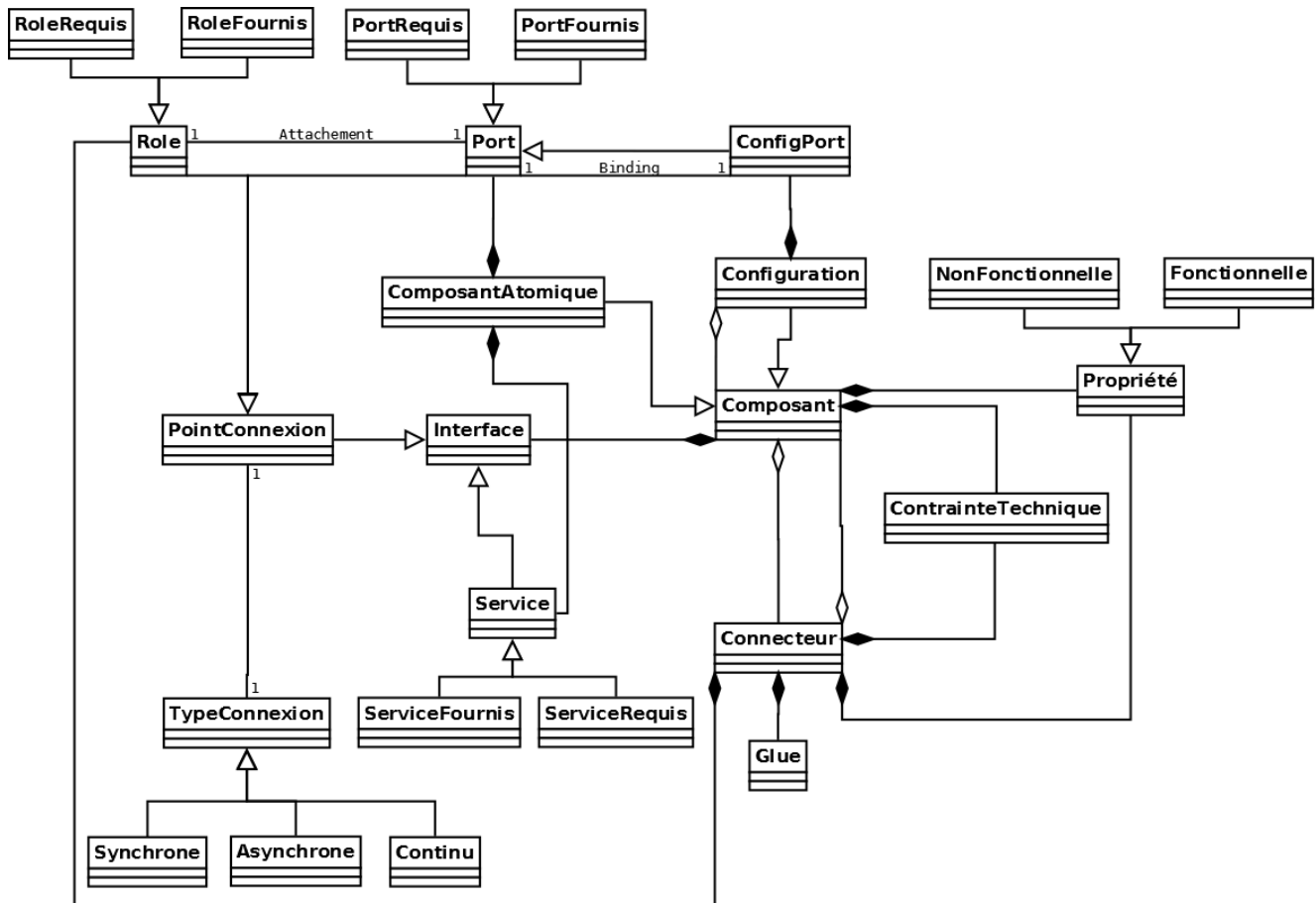


FIGURE 1 – Diagramme UML du M2

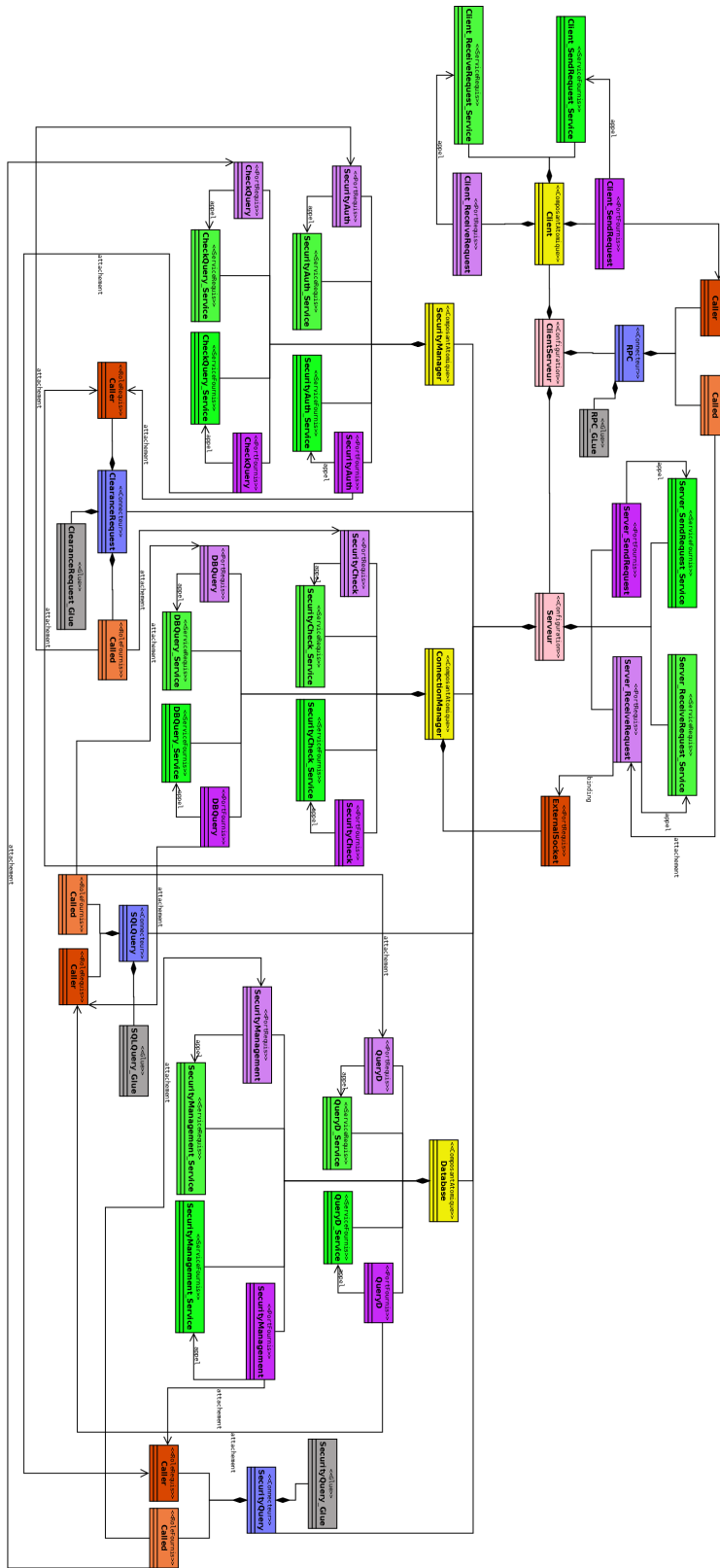


FIGURE 2 – Diagramme UML du M1