

HADL

CHENAIS SEBASTIEN – WOLLENBURGER ANTOINE

Table des matières

Introduction	2
Niveau M2	2
Description d'HADL.....	2
Diagramme d'HADL.....	5
Diagramme global	5
Explication du diagramme	5
Niveau M1	6
Première approche	6
Seconde approche	6
Conclusion	8

Introduction

Le but de ce projet est de construire un langage de description d'architecture maison : HADL.

L'application se fera sur un modèle de composant client-serveur, avec un serveur contenant un gestionnaire de connexions, un gestionnaire de base de données et un gestionnaire de sécurité.

Dans un premier temps, il vous sera décrit le méta modèle de niveau M2 (architecture d'infrastructure) puis son implémentation au niveau M1 (architecture d'application). L'étape M0 ne sera pas décrite ici, mais est déduite du M1.

Niveau M2

Ce niveau permet de définir le langage architectural, ses mécanismes et ses composants. Nous avons utilisé UML pour décrire HADL.

Description d'HADL

Les éléments nécessaires à ce langage sont :

- Composant
 - Port/service
- Connecteur
 - Glue
 - Rôle
- Configuration
 - Port/service
- Lien
 - Binding
 - attachment

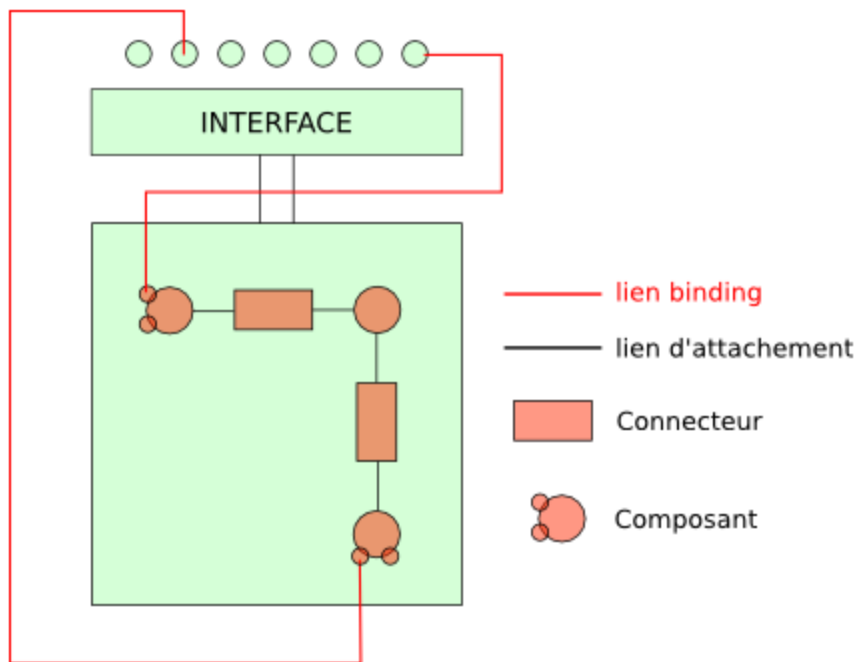


Figure 1- représentation des composants d'HADL

HADL est aussi fourni avec les mécanismes suivants :

- composition : le fait qu'un composant en utilise un autre
- héritage : le fait qu'un composant prenne les caractéristiques d'un autre
- raffinement : le fait de pouvoir faire une mise au point sur un composant pour avoir plus d'information

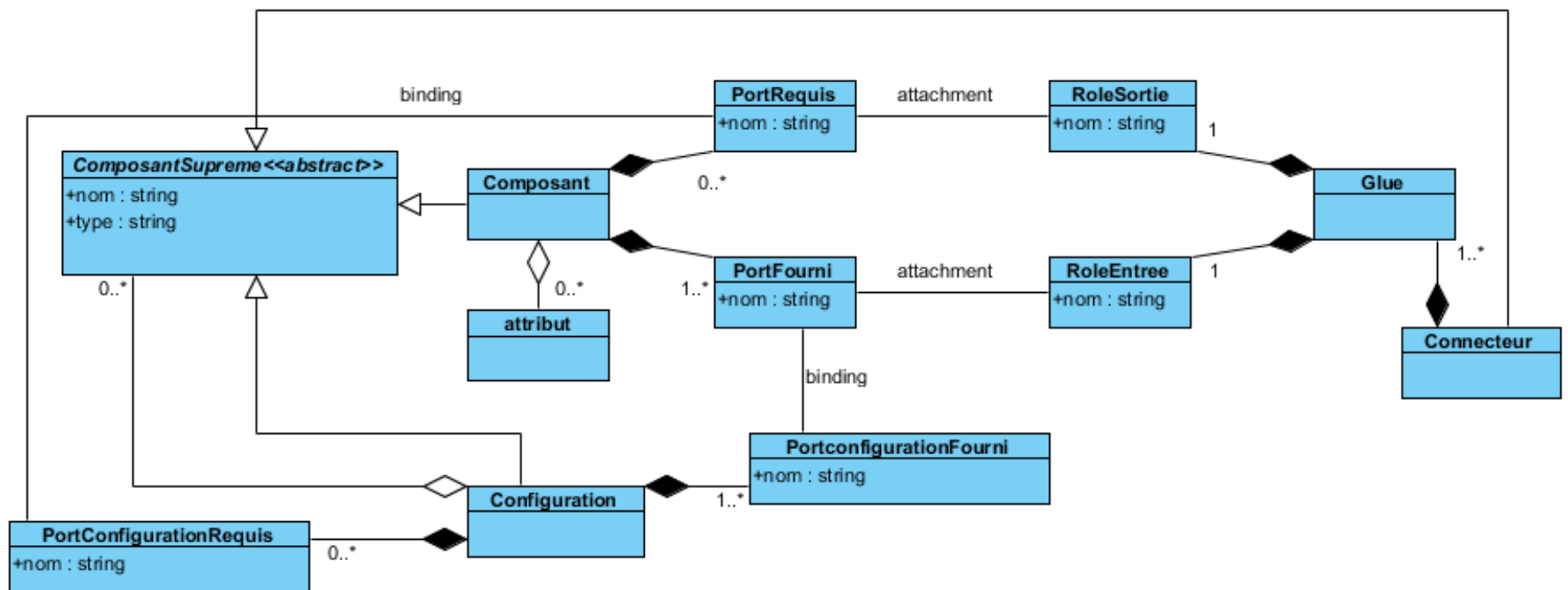


Figure 2 - Diagramme UML d'HADL

Diagramme d'HADL

Diagramme global

Cf. figure 2

Explication du diagramme

Les différents éléments d'HADL sont représentés ici : le composant avec ses ports/services, le connecteur avec sa glue et ses rôles et la configuration avec ses ports/services.

La configuration

Une configuration peut être composée de connecteurs, de composants et de configurations. Ces composites sont représentés par la classe abstraite `ComposantSupreme` et l'agrégation faible entre la configuration et cette classe abstraite.

Les ports d'une configuration ne peuvent être liés qu'au port du même type d'un composant. C'est-à-dire qu'un port fourni d'une configuration ne pourra être lié qu'à un port fourni d'un composant. Le lien sera alors un lien dit de « binding ». Une configuration possède au moins un port fourni.

Du fait de la présence d'au moins un port fourni, la configuration possède au moins un composant.

Le composant

Le composant hérite de `ComposantSupreme`. Il possède au moins un port fourni et peut posséder des ports requis. Un port fourni peut être lié à un rôle en entrée d'un connecteur par un lien dit « d'attachment » ou à un port fourni de configuration comme expliqué précédemment. Un port requis peut être lié à un rôle de sortie d'un connecteur ou à un port requis de configuration.

Un composant peut être composé d'attributs.

Le connecteur

Le connecteur est composé d'au moins une glue. Chaque glue est composée d'un rôle en entrée et d'un en sortie. Les rôles peuvent se lier aux ports des configurations comme expliqué précédemment.

Niveau M1

Première approche

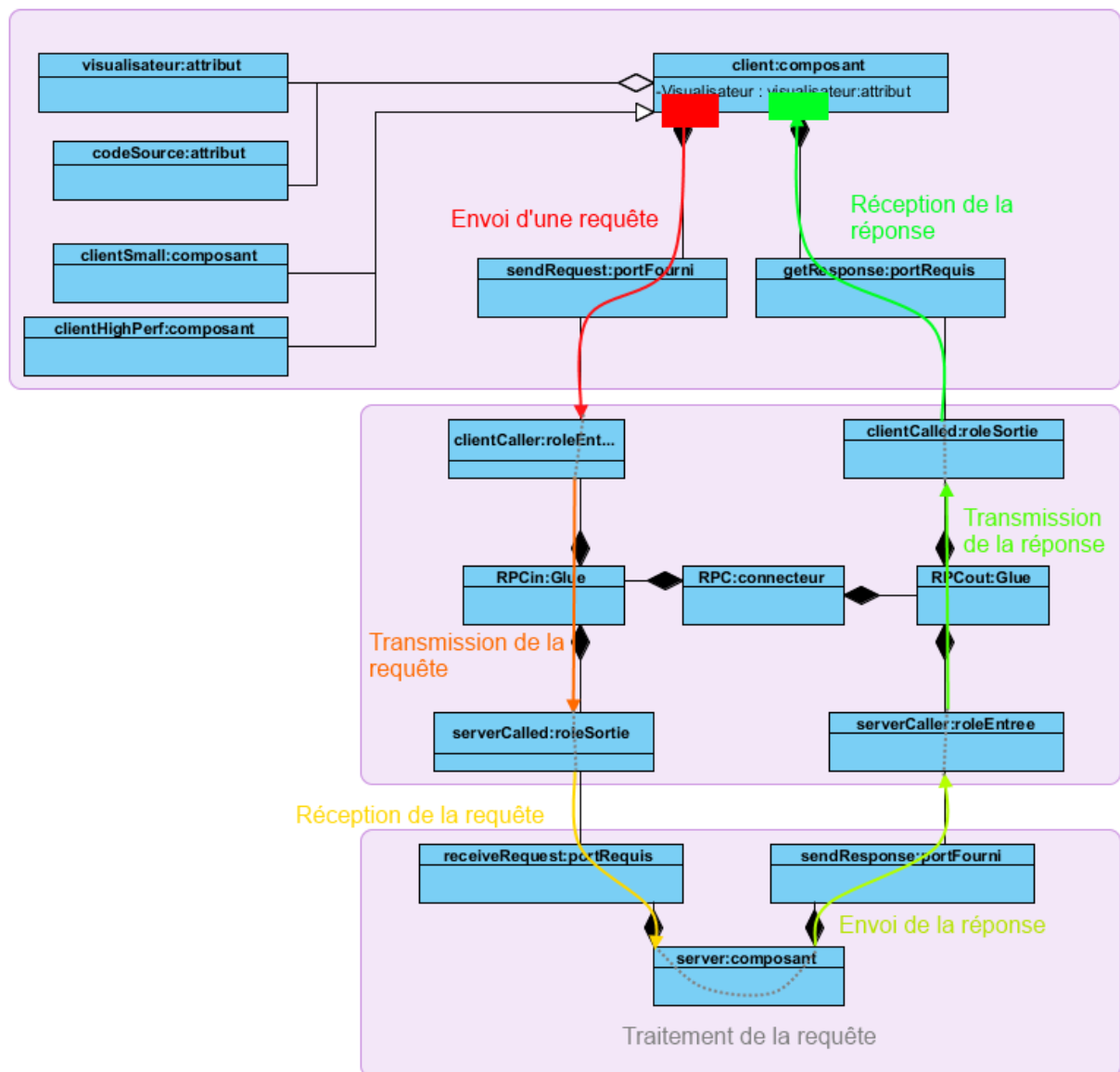


Figure 3 - M1 client-serveur sans raffinement

Dans cette première approche, nous avons modélisé le serveur comme un composant unique.

L'envoi d'un message au serveur par le client se déroule comme suit : le client émet un message, message relayé par le connecteur RPC au serveur. Ce message est ensuite traité par le serveur, qui envoie sa réponse au client via le connecteur RPC.

Seconde approche

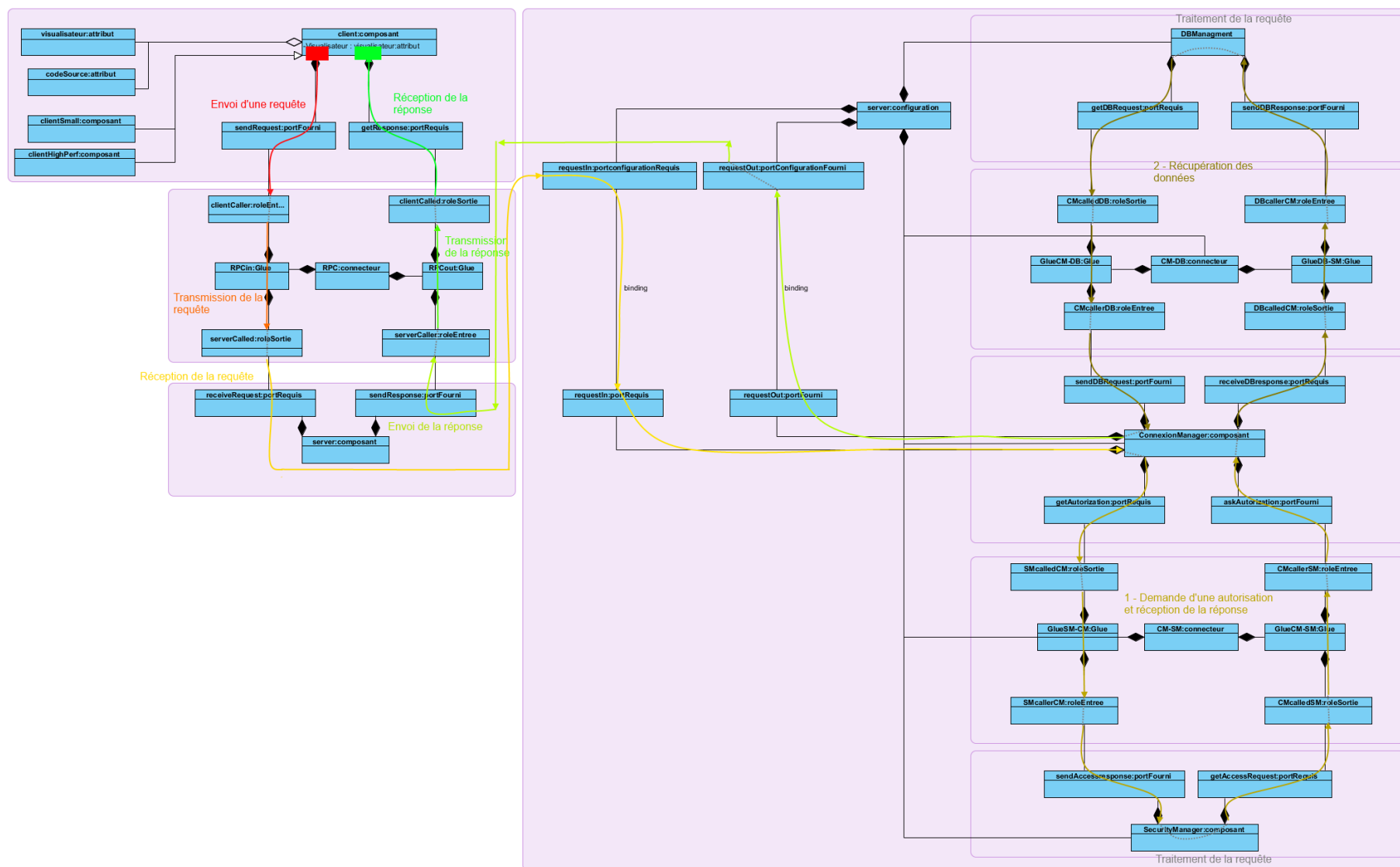


Figure 4 - M1 client-serveur raffiné

Le serveur est ici considéré comme une configuration, et lié au premier modèle via des liens de binding.

Le traitement interne au serveur du message se déroule en deux étapes :

- Tout d'abord, le connexion manager, qui a reçu le message par les liens de binding, demande une autorisation au security manager, via le connecteur cm-sm.
- Cette autorisation reçue, le connexion manager demande la récupération de données au DBmanager, via le connecteur cm-db.

Ensuite, les données récupérées par le connexion manager sont renvoyées au client, via les liens de binding et le connecteur RPC.

Conclusion

HADL permet de simplifier grandement la lisibilité d'un modèle sur une architecture complexe. De plus, l'architecture par composants permet la réutilisabilité ceux-ci.