Formal Software

Master ALMA 2eme année

Projet : Réalisation d'une application à automates

Encadrant : C.Jard

A.Marguerite
R.Rincé

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03, FRANCE

Table des matières

1 Introduction

1.1 Présentation du sujet

L'objectif de ce projet/TP est de réaliser un protocole de transfert de messages entre deux machines avec des listes des canaux de communication *First In First Out*. La démarche proposée par l'énnonce propose les étapes suivantes :

1. Modéliser ce protocole avec l'outil Roméo.
2. Construire le graphe des marquages.
3. Spécifier en logique temporelle :
   le protocole peut toujours revenir à l'état initial.
4. Expliciter les propriétés vraies.
5. Programmer une implémentation :
   Une machine est pilotée par l'utilisateur, l'autre par le programme.
6. Tests :
   rendre observable les différentes transitions.

2 Implémenation du protocole

2.1 Solutions choisies

Le langage choisi est JAVA. Les motivations de ce choix sont les suivants : la facilité ainsi que la mise à disposition par JAVA de mointeurs. Effet, la classe `UnicastRemoteObject` propose les objets *RMI* (Remote Method Invocation). Il s'agit d'objets « distants » auquels des appels de méthodes peuvent être effectués comme si l'objet était local. De plus l'implémentation des objets `RMI` utilise les moniteurs. Ce mécanisme garantissant l'emploie de cannaux de type `FIFO` est appoprié pour le protocole de ce projet.

2.2 Implémentation

2.2.1 Partie serveur

La partie « serveur » est implémentée par la classe `IA`, la figure 2.1 illustre que cette classe hérite de l'inferface `IIA`. Cet héritage est indispensable pour l'utilisation d'objet `RMI`. Les signatures des méthodes que l'interface contient, sont les moyen pour le client de dialoguer avec le serveur. Bien sur le client doit avoir la connsaissance de cette inerface.

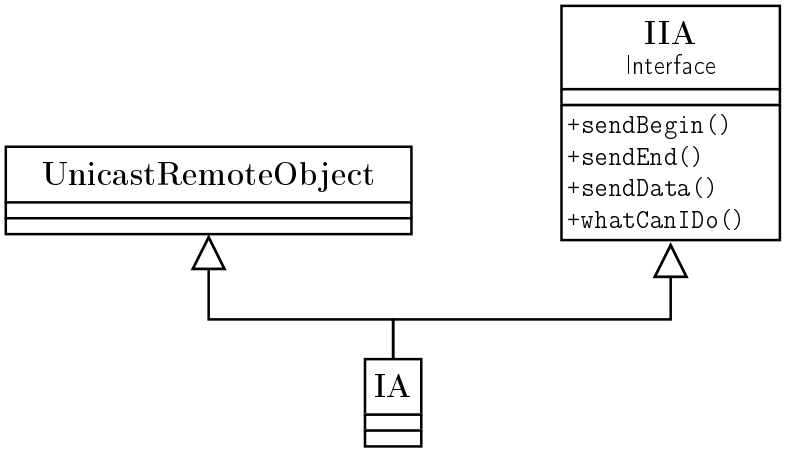

FIGURE 2.1 – Partie serveur

3 Conclusion

À ce stade, la modélisation de notre méta-modèle (*cf.* chapitre **??**) nous a permis de générer un modèle pour notre application *Clients-Serveur* (*cf.* chapitre **??**). Par la suite il sera nécéssaire de passer à l'étape 3 présentée au chapitre 1. C'est à dire d'implémenter le méta-modèle M2 en JAVA.

La création de l'application *Clients-Serveur* repose sur l'héritage des classes du méta-modèle. Enfin l'instanciation concrète de ses classes achèvera l'implémentation de nottre application.

Table des figures