RAPPORT DE PROJET

Compilation II

Youness DENDANE
Maxens MANACH

M1 Informatique - Parcours International

Parties du projet

La gestion du projet à été séparée en 5 grandes parties :

- Squelette du programme
- Déclaration des appareils
- Déclaration des interfaces
- Déclaration des scénarios
- Déclaration des commandes

Dans un premier temps, le lex à été conçu pour reconnaître tous les symboles du pseudo langage.

Ensuite la grammaire du .cup à été établie, suivi de la gestion des erreurs.

Puis le comportement de chacune des grandes parties à été traitée avec des actions java afin de produire du code dans deux fichiers distincts : CMaisonUser.java et HabitatSpecifique.java

La quasi-intégralité des mécaniques proposées par le pseudo-langage sont opérationnelles et correctement produites en java : squelette, déclarations, scénarios, …

Les scénarios étant la partie la plus compliquée du traitement, une fonctionnalité n'a pas pu être traitée à temps : Le transcript des scénarios dans CMaisonUser. Nous n'avons pas été en mesure de traduire le code des scénarios inscrits dans HabitatSpécifique vers le fichier CMaisonUser dans les temps. A la place, nous avons inscrit des chaînes vides dans le code du fichier (“”).

En revanche, la gestion des instructions (action, association, programmation, messages, boucles, si, sinon, …) est opérationnelle.

Enfin, la gestion des erreurs sémantiques n'a pas été traitée

Grammaire : voir fichier joint Grammaire.txt

Choix de conception

Au niveau de la conception, nous avons essayé de suivre au mieux les principes de la POO. Nous avons deux classes distinctes :
Classe "Ecriture" : cette classe est utilisée afin de permettre la création et la fermeture des fichiers "HabitatSpecific" et "CMaisonUser".

Classe "ProductionCode" : C'est la classe mère, celle qui contient toutes les méthodes pour par exemple : créer une boucle, une condition, une déclaration, définition…

On dispose de deux chaînes de caractères principales : ***cMaisonUser*** et ***habitatSpecific***.

Au fur et à mesure dans le programme, nous allons concaténer ces chaînes, jusqu'à avoir à la fin, le programme java complet. Ces chaînes sont les chaînes de plus haut niveau, c'est à dire que c'est le contenu de celles-ci qui sera directement manipulé par la classe "Ecriture" et donc, écrit dans les programme finaux.

Mais, nous disposons aussi d'autres chaînes locales, ayant une seule portée comme le cas de l'écriture de scénarios. Pour ce faire, nous utilisons une autre chaîne indépendante des autres, qui aura pour seul but, d'avoir comme contenu le code java du scénario.

Dès que l'interpréteur entrera au niveau du bloc de scénario, cette chaîne va être concaténée au fur et à mesure.
Puis à la fin d'un scénario, une méthode ("terminerScenario()") va être appelée pour faire deux choses :

- ajout d'une instance de type scénario dans le fichier CMaisonUser avec le bon nom de scénario mais sans la chaîne spécifique (2ème paramètre)
- écriture dans le fichier HabitatSpecific du scénario

En plus de ces chaînes, nous utilisons aussi d'autres attributs permettant de gérer des cas; comme par exemple si on se trouve dans une condition.

Il existe aussi, en dehors de ces classes, des variables globales déclarées qui nous permettent une réutilisation de variables. Par exemple, lors de la déclaration d'un **CEnsemble**, nous avons besoin de réutiliser le même nom de variable d'un ensemble pour pouvoir ajouter des appareils à celui-ci.
Nous les utilisons aussi par exemple afin de déterminer le type java d'un appareil et quel est le type de ce dernier.

Les variables globales nous permettent de nous passer du typage des non terminaux et, aussi, passer outre le problème de n'avoir qu'un seul retour par non terminal.
(RESULT = "chaine").

Résumé des programmes obtenus

Voici les résumés obtenus lors de la production de code des fichiers ex1ok, ex_simple et exemple_projet_domus.

```
manachma@localhost ~/S1/Compilation2 $ java parser < ex_simple.txt
Nombre d'appareils déclarés : 6
Nombre d'interfaces déclarées : 6
Nombre de scénarios déclarés : 6 {bonjour,bonjour2,soiree,fincafe,soiree_musique,verif}
Scénarios programmés :  {soiree_musique,soiree}
Associations Interfaces/Scénariis:
b1 -> bonjour
t1 -> bonjour
zap -> bonjour
b2 -> bonjour2
b3 -> soiree
b3 -> fincafe
b3 -> verif
b4 -> soiree_musique

manachma@localhost ~/S1/Compilation2 $ java parser < exemple_projet_domus.txt
Nombre d'appareils déclarés : 12
Nombre d'interfaces déclarées : 4
Nombre de scénarios déclarés : 7 {bonjour,soiree,soiree_musique,depart,noel1,noel2,noel3}
Scénarios programmés :  {soiree,soiree_musique}
Associations Interfaces/Scénariis:
b1 -> bonjour
b2 -> depart
t1 -> depart
c1 -> noel1
c1 -> noel2
c1 -> noel3

manachma@localhost ~/S1/Compilation2 $ java parser < ex1ok
Nombre d'appareils déclarés : 24
Nombre d'interfaces déclarées : 6
Nombre de scénarios déclarés : 6 {bonjour,soiree,soiree_musique,depart,test,test2}
Scénarios programmés :  {test2,soiree,soiree_musique}
Associations Interfaces/Scénariis:
b1 -> bonjour
tel1 -> test
tab1 -> test
zap -> test
b2 -> depart
t1 -> depart
```