UPMC

Table des matières

1 Introduction **2**

2 Spécification Technique des Besoins et Exigences **2**
2.1 Besoins et Exigences du client 2

3 Solutions **6**
3.1 Architecture logicielle 6
3.2 Choix technologique 6
3.3 Délais et taches à réaliser 7
3.4 Organisation et gestion du projet 7

1 Introduction

Le projet **"Interface graphique pour la Logique en L3"** consiste à développer un outil dynamique et robuste pour améliorer l'enseignement de la logique en Licence 3. Ce projet nous a été soumis dans le cadre de L'UE PSAR du master 1 Informatique spécialité SAR. Il est sous la responsabilité de Mr Fabrice Kordon suivit par Mme Béatrice Berard,Mr Mathieu Jaume et Mme Bénédicte Legastelois.

2 Spécification Technique des Besoins et Exigences

2.1 Besoins et Exigences du client

2.1.1 Interpretation d'une formule de la logique des prédicats

Terme
Un terme est défini de façon inductive comme suit :

- Toute variable est un terme
- Toute constantes est un terme

Nous manipulerons en tout cinq (5) variables {v,w,x,y,z} et 20 constantes {a,b,c,...,t} dans notre projet.

Interprétation des termes :
Interpréter un terme, c'est lui associer une valeur appartenant à un domaine.

- Si le terme est une variable, c'est la valeur associée à la variable
- Sinon c'est la valeur globale résultant de l ?évaluation des différents arguments du terme

Interprétation des termes du projet :
Dans ce projet les domaines d'interprétation seront des jardins contenant des fleurs.
Les objets manipuler seront des fleurs dont les caractéristiques sont les suivantes :

- espèce = rose, paquerette, tulipe
- taille = grand, moyen, petit
- couleur = rouge, rose ,blanche
- nomune constante ou none s'il est anonyme

une fleur dans le jardin est représentée par un quintuplets {(x,y),e,t,c,n} où :

- (x,y) est sa position dans le jardin
- e : son espèce
- t : sa taille
- c : sa couleur
- n : son nom ou none si elle est anonyme

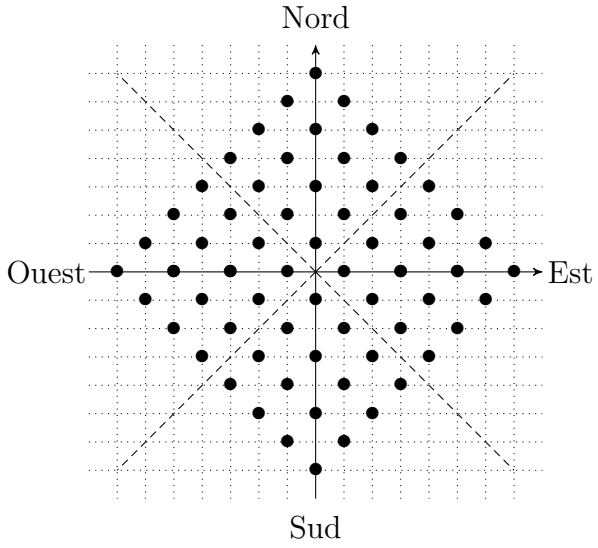

FIGURE 1 – Places d'un jardin

Construction du jardin

les coordonnées (x, y) sont des places du jardin

$$\text{Places} = \left\{ \begin{array}{c} (0,7), \\ (-1,6),(1,6), \\ (-2,5),(0,5),(2,5), \\ (-3,4),(-1,4),(1,4),(3,4), \\ (-4,3),(-2,3),(0,3),(2,3),(4,3), \\ (-5,2),(-3,2),(-1,2),(1,2),(3,2),(5,2), \\ (-6,1),(-4,1),(-2,1),(0,1),(2,1),(4,1),(6,1), \\ (-7,0),(-5,0),(-3,0),(-1,0),(1,0),(3,0),(5,0),(7,0), \\ (-6,-1),(-4,-1),(-2,-1),(0,-1),(2,-1),(4,-1),(6,-1), \\ (-5,-2),(-3,-2),(-1,-2),(1,-2),(3,-2),(5,-2), \\ (-4,-3),(-2,-3),(0,-3),(2,-3),(4,-3), \\ (-3,-4),(-1,-4),(1,-4),(3,-4), \\ (-2,-5),(0,-5),(2,-5), \\ (-1,-6),(1,-6), \\ (0,-7) \end{array} \right\}$$

Formule de la logique des prédicats :

Un Symbole de prédicat correspond au nom d'une propriété portant sur un ou plusieurs termes.

Exemple de prédicat :

(f1) d est une rose :
$Rose(d)$;

(f2) Toutes les fleurs sont des roses :
$\forall x\ Rose(x)$;

(f3) Il existe une rose :
$\exists x\ Rose(x)$;

(f4) Toute fleur blanche est plus petite qu'une fleur située à son est :
$\forall x\ (est\_blanc(x) \implies \exists y\ (plus\_petit\_que(x, y) \wedge a\_l\_est\_de(y, x)))$;

(f5) Toute fleur est à l'est ou à l'ouest, ou au sud, ou au nord :
$\forall x(a\_l\_est(x) \vee a\_l\_ouest(x) \vee au\_sud(x) \vee au\_nord(x))$;

(f6) Toutes les grandes fleurs sont rouges et il n'existe pas de fleur blanche au sud d'une fleur rouge :
$\forall x\ (est\_grand(x) \implies est\_rouge(x)) \wedge\ /\exists x(est\_blanc(x) \wedge \exists y\ (est\_rouge(y) \wedge au\_sud\_de(x, y)))$;

(f7) Il existe une fleur rouge au nord de la fleur g :
$\exists x\ (est\_rouge(x) \wedge au\_nord\_de(x, g))$

Implémentation des formules

Une variable est libre si elle n'est associée à aucun quantificateur (existentiel ,pour tout). Dans le cas du projet, les formules ne devront pas contenir des variables libres.

Formules du projet

Il existe des prédicats unaire, binaires et ternaires :

- P1-prédicats unaire
- P2- prédicats binaires
- P3- prédicats ternaire

P1 = {est_rouge, est_rose, est_blanc, a_l_est, a_l_ouest, au_sud, au_nord , Rose, Paquerette, Tulipe, est$_g rand, est_m oyen, est_p etit$}

$P2 = \{a\_l\_est\_de, a\_l\_ouest\_de, au\_sud\_de, au\_nord\_de, même\_latitude, même\_longitude, plus\_grand\_que, plus\_petit\_que, même\_taille\_que, même\_couleur\_que\}$

$P3 = \{est\_entre\}$

Prédicat	Définition	Relation
Espèce	Rose, paquerette,Tulipe	Unaire
Taille	est_grand,est_moyen,est_petit	Unaire
Couleur	est_rouge,est_rose,est_blanc	Unaire
Position	a_l_est,a_l_ouest,au_sud,au_nord	Unaire
Comparaison de position	a_lest_de,a_l_ouest_de,au_sud_de	Binaire
	au_nord_de ,même_latitude, même_longitude	
Comparaison de taille	Plus_petit_que,plus_grand_que,même_taille_que	Binaire
Comparaison de couleur	même_couleur_que	Binaire
Égalité	=	Binaire
Comparaison de position	est_entre	Ternaire

Toutes les formules soumises à la vérification doivent ne sont pas censées contenir des variables libres (une erreur est donc déclenchée lors de l' évaluation d'une formule avec une variable libre).

3 Solutions

3.1 Architecture logicielle

Le projet est structuré en modules indépendants :

1-Module d'analyseur syntaxique des formules
L'analyseur syntaxique sera un module du projet qui s'occupera de la vérification syntaxique des différentes formules qui seront soumises à une évaluation éventuelle.
Une fois que l ?utilisateur (l'étudiant) saisit sa formule et demande une interprétation, nous l'appellerons en premier pour s'assurer que la formule est bien formée avant d'appeler notre interpréteur.
L'analyseur syntaxique sera un module du projet qui s'occupera de la vérification syntaxique des différentes formules qui seront soumises à une évaluation éventuelle.
Une fois que l'utilisateur (l'étudiant) saisit sa formule et demande une interprétation, nous l'appellerons en premier pour s'assurer que la formule est bien formée avant d'appeler notre interpréteur.

2-Module Interface graphique
Toute l'application sera présentée dans une interface conviviale, dans laquelle l'utilisateur aura la possibilité de construire ses propres formules, son propre environnement et demander l'évaluation de ses formules.
Il est évident que l'interface offrira un menu standard dans lequel il pourra sauvegarder/restaurer son environnement, ses formules dans/depuis le répertoire de son choix.
Notre choix du langage tient compte en premier lieu du temps de réponse et de la robustesse.

3-Module évaluant les formules(Moteur)
Nous disposons d'un module d'interprétation développé en python, il faudra faire communiquer les deux modules pour obtenir l'interprétation de(s) la formule(s).
Le module python attend deux paramètres :

- l'environnement bien défini
- la formule bien formée

3.2 Choix technologique

Comme le moteur est développé en python, nous partons de l ?idée de faire communiquer deux langages car, implémenter une interface graphique en python s ?avère fastidieux.

Resultats de nos recherches

Notre application sera implémentée en java, il faut trouver le middleware nécessaire pour la communication des deux langages.

Le choix du middleware dépend fortement :

— du temps de réponse
— de la prise en main

Faire Communiquer java et python

La Bibliothèque **Jython** permet d ?exécuter en Java un programme écrit en python, Il existe aussi jeep qui fait de même.
Désavantage : Il y a en effet une latence élevée de l ?interpréteur Jython.
CORBA, ICE et ses variantes sont de langage de définitions d ?interfaces (IDL) visant à faire communiquer deux entités (services, composants) écrit dans des langages différents ; Il s ?agit de mapper (traduire des éléments fournis par l ?IDL en éléments d ?un langage de programmation).
Nous utiliserons cette deuxième approche dans le cadre de ce projet.

3.3 Délais et taches à réaliser

L'ue PSAR est organisé comme suit :
— Rédaction d'un cahier de charge
— Réalisation du projet
— Rédaction d'un rapport
— Soutenance du projet
Comme tout projet cela est géré dans un temps bien définis.
— 01 Mars 2016 : Cahier de charges
— Rapport
— Soutenance du projet

3.4 Organisation et gestion du projet

Nous avons décidé pour chaque module de discuter et partager nos idées, avant de nous répartir le travail. Ensuite, après que chacun de nous ai fini, de travailler sur un composant de l ?application, ou sur une classe, nous passerons à la mise en commun et à la réalisation des différents tests à différent niveau d ?abstraction.

Gestion du projet
Dans le cadre du projet, afin de nous assurer de gérer au mieux les incohérences et les bugs, nous utiliserons l ?outil Git.
Gestion des documents dans le projet
Tous les documents que nous produirons dans le cadre du projet (cahier de charges, rapport d'avancement,Rapport final) serons édités en latex.