



UNIVERSITÉ PIERRE MARIE CURIE
MASTER 1 INFORMATIQUE
SPÉCIALITÉ SAR

Projet d'interface graphique pour la logique en Licence 3

RAPPORT

Encadrement :

Mme Béatrice Bérard

Mr Mathieu Jaume

Mme Bénédicte Legastelois

Réalisation :

DIALLO Ousmane 3

KATUITSHI-NTUMBA Jean-Marc

Année universitaire : 2015 - 2016

Table des matières

Remerciements	2
Introduction	3
1 Phase d'analyse	5
1.1 Problématiques	5
1.2 Diagramme de use case	6
1.3 Fiches détaillées	6
1.4 Diagramme de classe	7
1.5 Diagramme de séquence	8
1.5.1 Initialisation de l'environnement et ajout d'une fleur .	8
1.5.2 ajout, analyse et évaluation d'une formule	8
2 Phase de conception	10
2.1 Diagramme de classe	10
2.2 Diagramme de composants	12
2.3 Diagramme de structure interne	13
3 Phase de Développement du projet	14
3.1 Composants environnement et formule	14
3.2 Composant analyseur syntaxique	14
3.3 Composant de communication java-python	16
3.3.1 Côté java	16
3.3.2 Serveur python	16
Conclusion	17
.1 Fiches détaillées	18

Remerciements

Nous tenons à remercier Mme Béatrice Berard, Mr Mathieu Jaume et Mme Bénédicte Legastelois pour leur encadrement tout au long de ce projet. Nous tenons aussi à remercier tous nos professeurs qui ont contribué à notre formation.

Introduction

Le projet **Interface graphique pour la Logique en Licence 3** consiste à développer un outil dynamique et robuste afin d'améliorer son enseignement. 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 suivi par Mme Béatrice Berard, Mr Mathieu Jaume et Mme Bénédicte Legastelois.

Cette interface sera un outil de travail convivial pour les étudiants lors des séances de travaux pratiques. Elle comprend un espace de création d'environnement qui peut contenir un ensemble d'objet (fleurs), un espace de création d'un ensemble de formules de la logique du 1er ordre et un ensemble de fonctionnalités associées permettant de sauvegarder/restaurer, analyser et évaluer ces formules.

Il s'agit d'expérimenter, selon un environnement bien défini et un ensemble de formules, les règles de la logique du 1er ordre. Les objets à manipuler sont des fleurs placées dans un jardin.

Les propriétés d'une fleur sont :

- **espèce** : Rose, Tulipe, Pâquerette
- **taille** : Petite, Moyenne, Grande
- **couleur** : Rose, Rouge, Blanche
- **Nom** : a, b, c, d,..., t, None(si pas de nom)

Le jardin est un ensemble de coordonnées(x, y) sur lesquelles on place des fleurs.

Par exemple les assertions suivantes seront exprimées comme suit :

"Toutes les roses blanches sont à l'est de toutes les tulipes"

$\forall x, \forall y ((Rose(x) \wedge est_blanche(x) \wedge Tulipe(y)) \Rightarrow alestde(x, y)).$

"Toutes les fleurs dans le jardin sont rouges" :

$\forall x (est_rouge(x)).$

Une fois l'interface lancée, l'utilisateur pourra :

- ajouter une fleur dans le jardin,
- saisir un ensemble de formules,
- lancer l'analyse et/ou l'évaluation de ces formules dans cet environnement.

Le résultat, (OK, NOK) sera affiché pour l'analyse, (True, False) pour l'évaluation.

Le résultat OK pour dire que la formule est syntaxiquement correcte NOK

sinon, True pour dire que l'assertion est vraie, False sinon.

Dans les prochains chapitres nous aborderons les différentes étapes **d'analyse**, de **conception**, de **développement** ainsi qu'une **conclusion** à la fin de ce document.

1 Phase d'analyse

1.1 Problématiques

Comme dit dans l'introduction, il y a un ensemble de fonctionnalité à implémenter suivant les différents composants du projet : l'environnement, les formules, l'analyseur syntaxique.

Pour la gestion de l'environnement, les principales fonctionnalités sont :

- initialisation du jardin (vide au départ)
- sauvegarde
- restauration
- impression
- ajout d'une fleur
- suppression d'une fleur
- transmission du jardin au moteur d'évaluation

Pour la gestion des formules

- saisie
- sauvegarde
- restauration
- impression
- transmission au moteur d'analyse/d'évaluation

l'analyseur syntaxique

- analyse syntaxique de la formule
- renvoi d'un arbre syntaxique

Quant au moteur d'évaluation, il est fourni sous la forme d'un script python.

Pour mieux appréhender le sujet, nous commencerons par faire une analyse de l'ensemble de ces fonctionnalités que nous présenterons grâce aux différents diagrammes d'analyse :

- **Diagramme de use case**
- **Fiches détaillées**
- **Diagramme de classe**
- **Diagramme de séquence**

1.2 Diagramme de use case

Ce diagramme nous présente l'ensemble des fonctionnalités ainsi que les différents acteurs de notre système. Cela est illustré grâce à la figure 1.

Notre système à un acteur principal qui est l'étudiant représenté par le personnage de gauche. Les fonctionnalités sont celles qui sont citées dans la problématique : création, sauvegarde, restauration, impression du jardin et des formules ainsi que l'évaluation de ces formules représentées dans les cercles suivant le commentaire de droite dans le rectangle. L'outil de droite représente notre moteur d'évaluation qui est externe à notre projet.

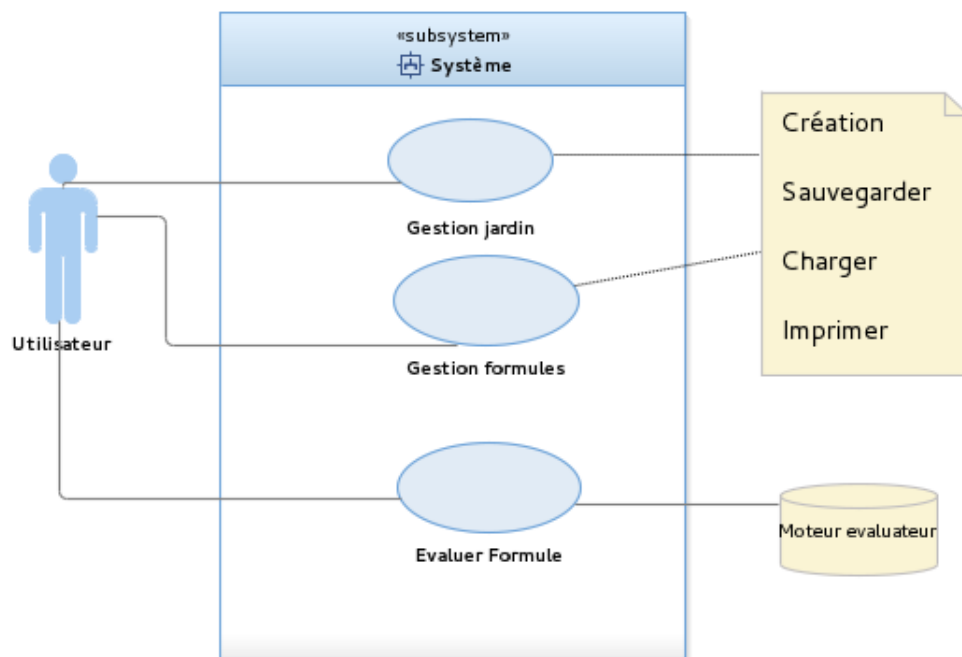


FIGURE 1 – Diagramme de use case

1.3 Fiches détaillées

Les fiches élaborées expliquent les différentes interactions qu'effectue un acteur sur le système afin de répondre à son besoin. Nous avons pris à titre

d'exemple ces quelques 9 cas parmi tant d'autres :

- cas 1 : Initialisation de l'environnement
- cas 2 : Sauvegarder l'environnement
- cas 3 : Charger d'un environnement
- cas 4 : Imprimer un environnement
- cas 5 : Créer une formule
- cas 6 : Charger un ensemble de formules
- cas 7 : Sauvegarder d'un ensemble de formules
- cas 8 : Imprimer un ensemble de formule
- cas 9 : Évaluer une formule

Voir le fichier annexe pour leur description.

1.4 Diagramme de classe

Comme le diagramme de classe présente une abstraction du système, nous avons les classes Jardin, Fleur, Coordonnées pour l'environnement ; Formule pour la gestion des formules.

Toutes les propriétés associées sont définies par des énumérations (taille, couleur, espèce, nom/constante, connecteur, variable et quantificateur, etc). Il est présenté par La figure 2.

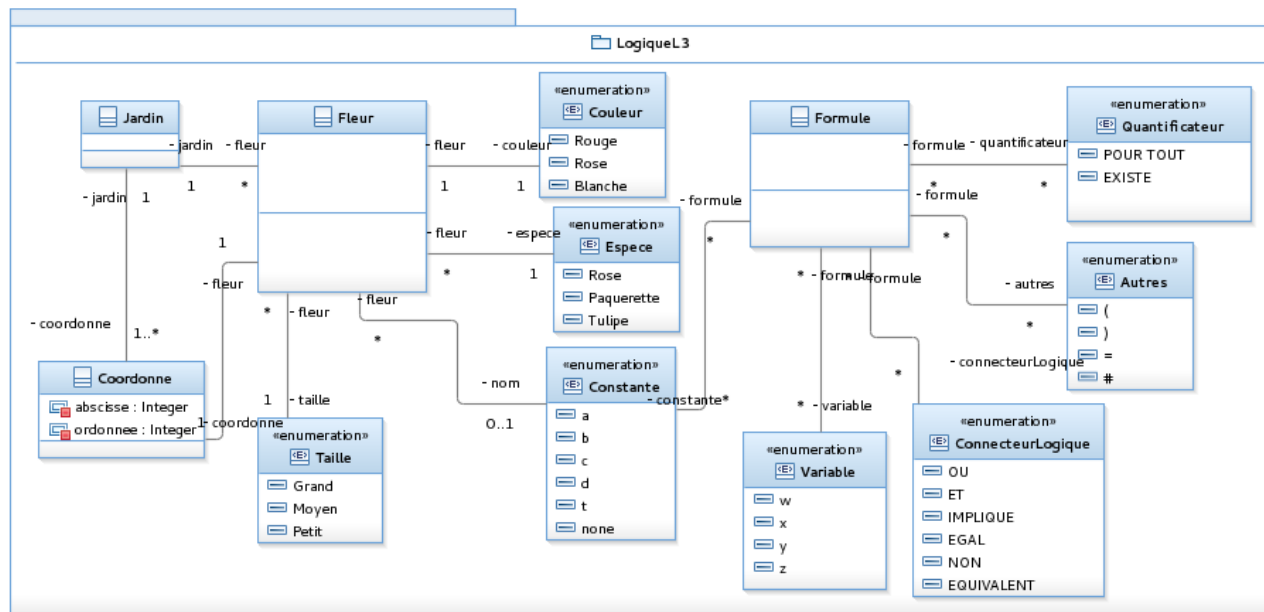


FIGURE 2 – Diagramme de classe

1.5 Diagramme de séquence

Dans cette étape nous vous présentons quelques diagrammes de séquence parmi tant d'autres

- **Initialisation de l'environnement**
- **Ajout,Analyse et évaluation d'une formule**

1.5.1 Initialisation de l'environnement et ajout d'une fleur

Ce digramme présente les différentes interactions lors de l'initialisation de l'environnement,figure 3 suivante.

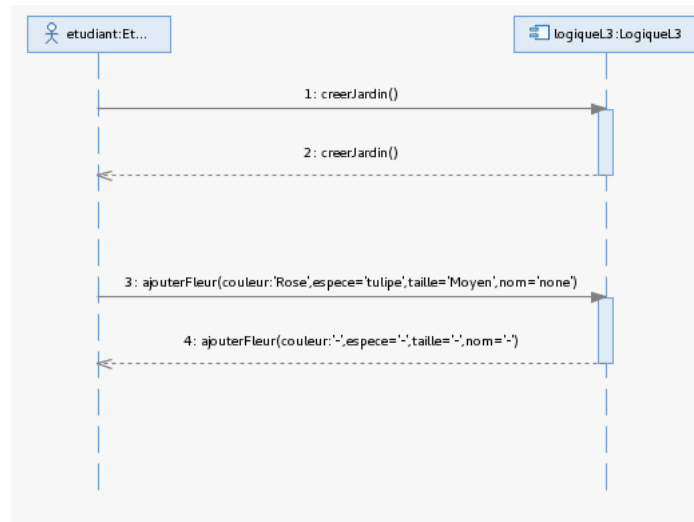


FIGURE 3 – Initialisation de l'environnement avec ajout d'une Tulipe rose

1.5.2 ajout, analyse et évaluation d'une formule

Ce digramme présente les différentes interactions lors d'un ajout, analyse et évaluation d'une formule dans un environnement. Comme le montre la figure 4 suivante.

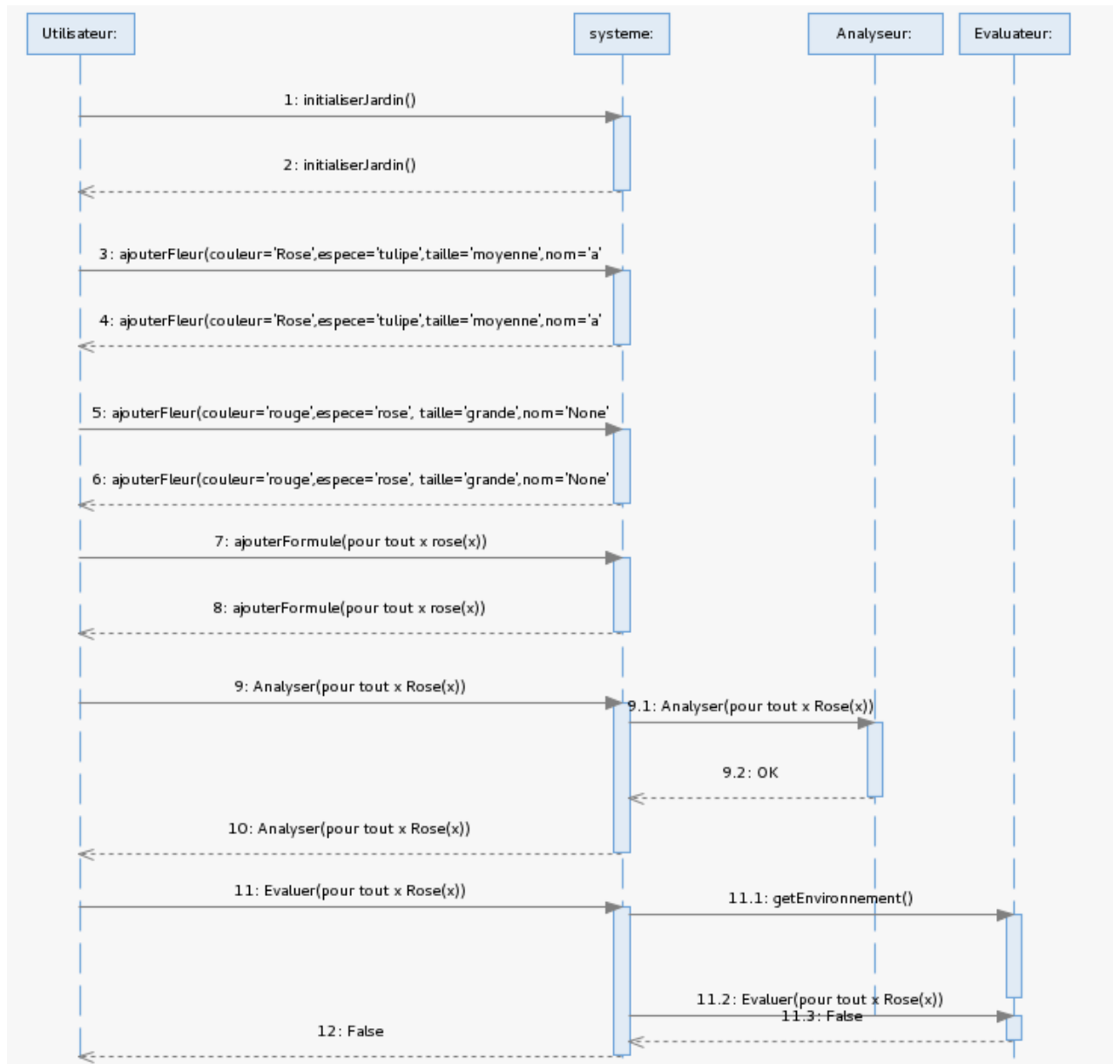


FIGURE 4 – ajout, analyse et évaluation

2 Phase de conception

Dans cette étape, nous parlerons de la conception, nous présenterons les différents composants ainsi que la structure interne du projet.

- Diagramme de classe
- Diagramme de composant
- Diagramme de structure interne

2.1 Diagramme de classe

Ce diagramme nous présente les différentes classes, énumérations et interfaces implémentées.

Le package pFormule contient le composant formule qui contient une classe Formule implémentant une interface IFormule.

Le package pEnvironnement contient les différentes classes et énumération nécessaire pour la gestion de l'environnement : Jardin qui implémentent l'interface IEnvironnement, Fleur qui implémentent l'interface IFleur (figure 5)

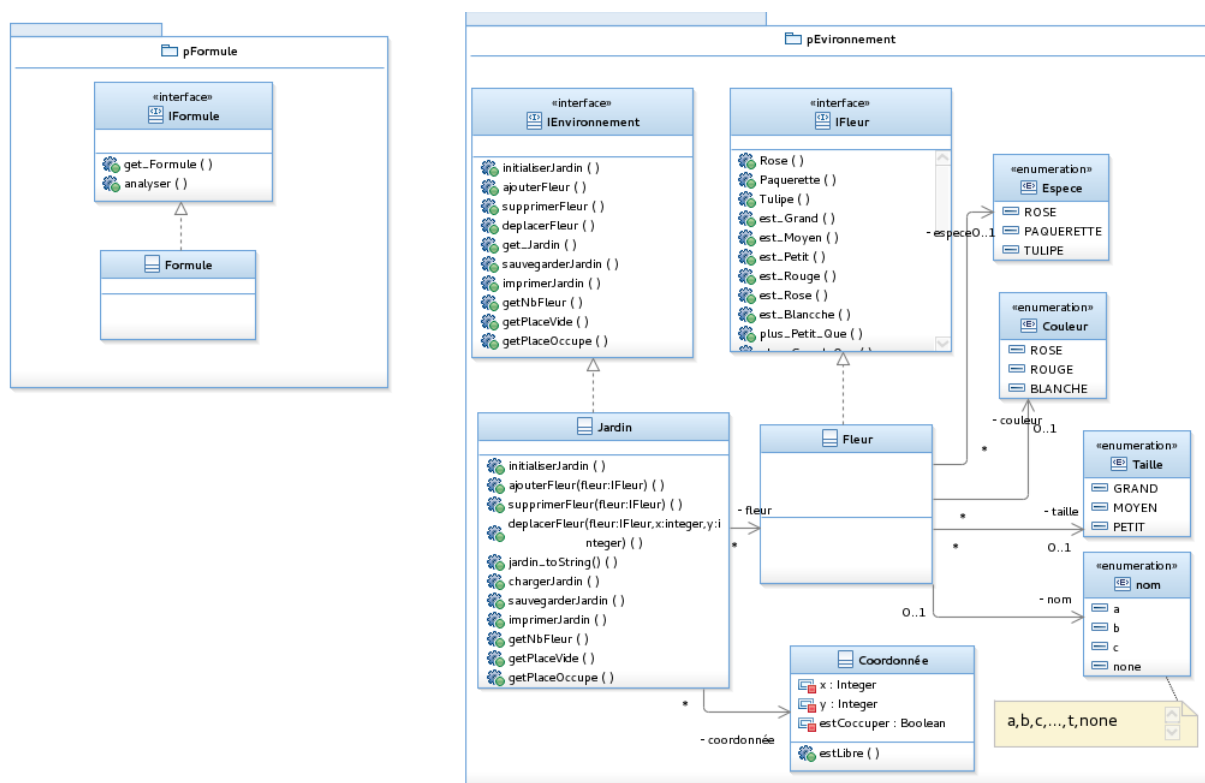


FIGURE 5 – Diagramme de classe niveau conceptuel

Le package cAnalyseur correspond à notre analyseur syntaxique (figure 6).

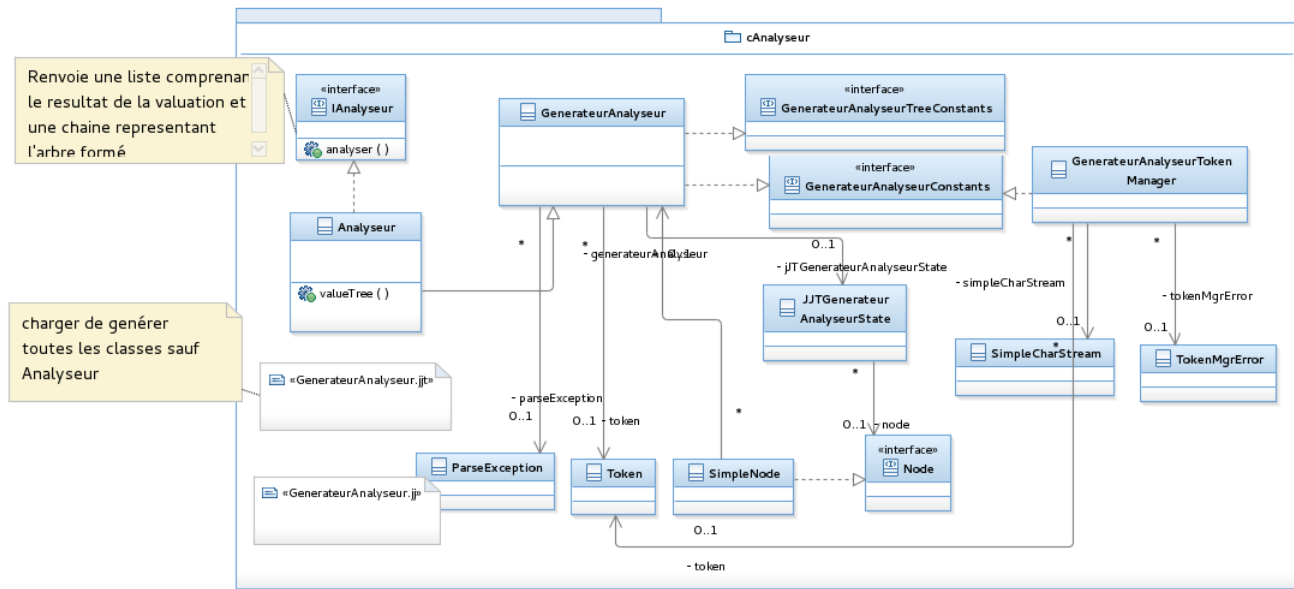


FIGURE 6 – diagramme de classe du composant analyseur syntaxique des formules

2.2 Diagramme de composants

Ce digramme présente les différents composants de notre application et les interfaces que ces composants implémentent.

Les différents composants :

- CIHM pour l'interface graphique utilise l'interface des autres composants
- CEvaluateur pour l'évaluation des formules fournies implémente IEvaluateur
- CEnvironnement pour l'ensemble des fonctionnalités de gestion du jardin implémente IEnvironnement
- CFormule pour les formules implémente IFormule
- CAnalyseur pour l'analyse syntaxique implémente IAnalyseur

Une illustration du diagramme à la figure 7.

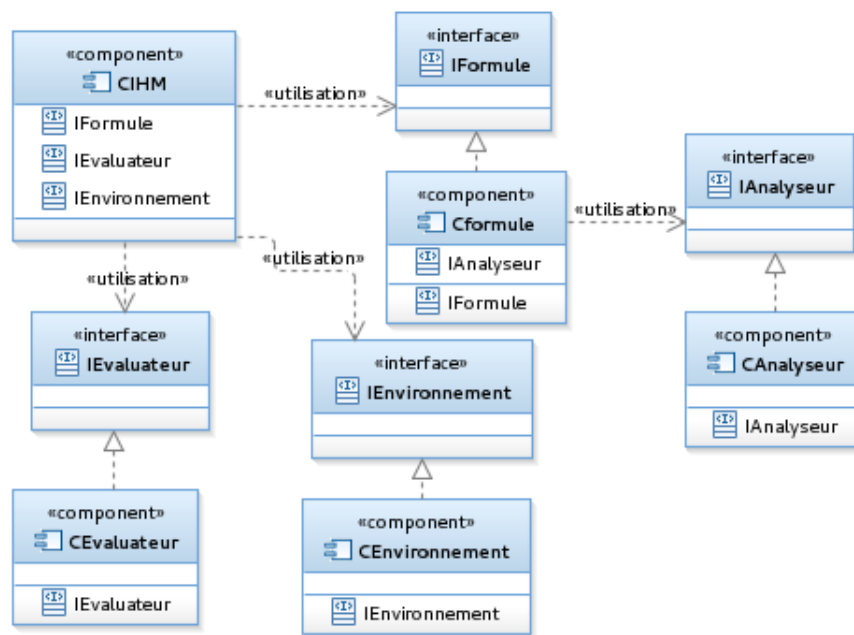


FIGURE 7 – Diagramme de composant

2.3 Diagramme de structure interne

Ce diagramme montre de façon précise les différentes communications entre les composants.

Par exemple le composant CIHM a besoin des composants CEvaluateur, CEnvironnement et CFormule pour fonctionner ; soit c'est pour créer un environnement ou évaluer un ensemble de formules dans un environnement.

Le composant CEnvironnement n'a besoin d'aucun autre pour fonctionner, il est indépendant de tous les autres.

Tout cela est illustré à la figure 8.

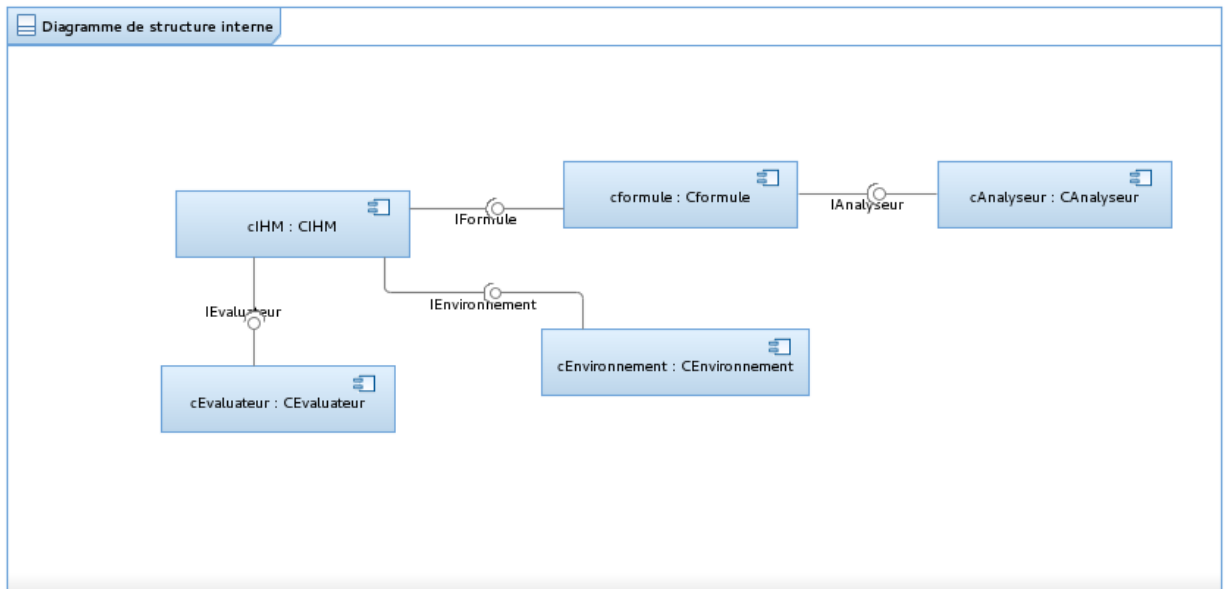


FIGURE 8 – Diagramme de structure interne

3 Phase de Développement du projet

Dans cette étape du projet, nous expliquons l'implémentation des différents composants du projet.

- **Composant environnement**
- **Composant formule**
- **Composant analyseur syntaxique**
- **Composant de communication java et python**

3.1 Composants environnement et formule

Le composant environnement est composé de la classe Jardin, Coordonnées et Fleurs.

C'est dans ce composant que les différentes fleurs peuvent être ajoutées pour construire notre environnement d'évaluation de formules.

Les différents checkbox permettent de faire le choix pour ajouter une fleur dans le jardin.

Le menu fichier, permet de sauvegarder, de charger un environnement.

Le composant formule, nous permet de saisir l'ensemble des formules à évaluer.

A l'aide du bouton Ajouter et du menu à onglet, il est possible d'ajouter d'autres formules.

Les deux composants sont présenté par la figure 9.

3.2 Composant analyseur syntaxique

L'analyseur syntaxique est le composant qui vérifie si les formules sont syntaxiquement correctes.

Il est constituée de plusieurs classes générées interagissant entre elles afin d'obtenir une vérification syntaxique et la construction de l'arbre syntaxique. l'outil javaCC génère ces classes par compilation d'un fichier "grammaire.jjt" dans lequel une grammaire est spécifiée (décrite en langage EBNF).

Il reçoit en paramètres d'entrée une formule sous forme d'une chaîne de caractère et renvoi en sortie un arbre de syntaxe si la formule est syntaxiquement valide, false sinon.

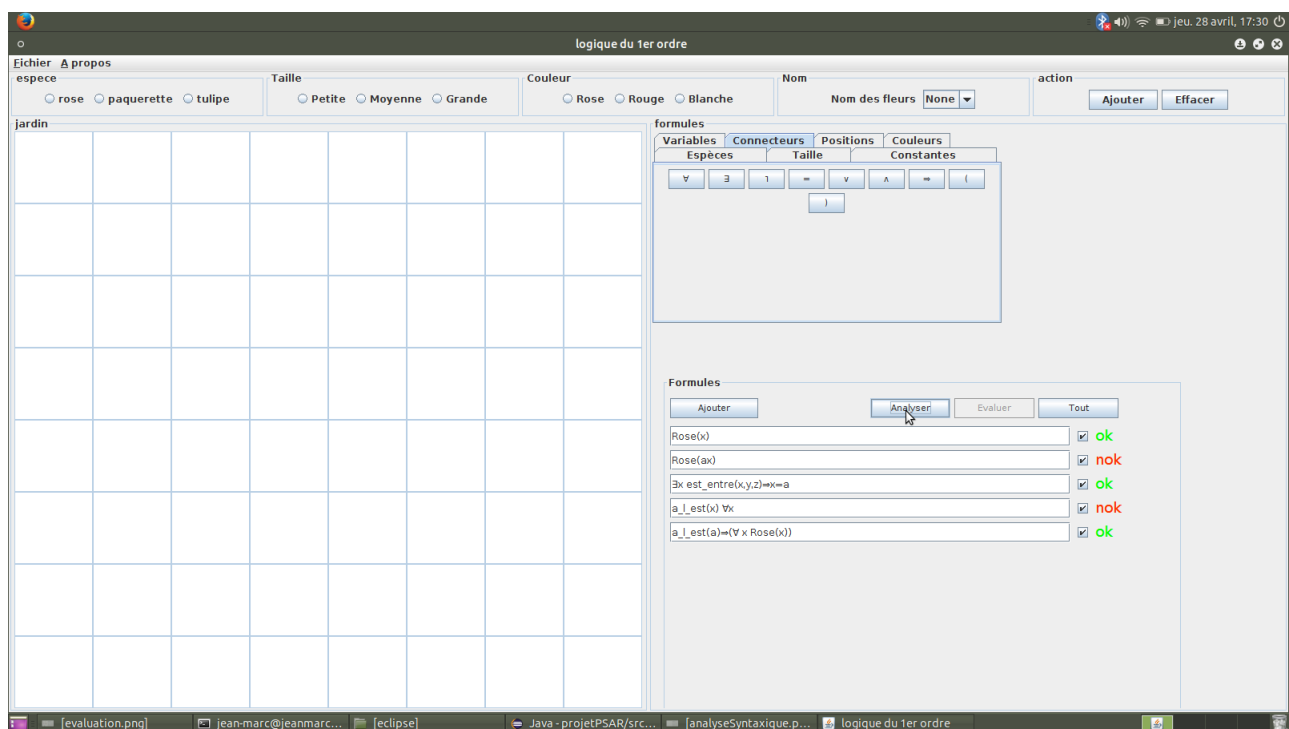


FIGURE 9 – composant environnement et formule

3.3 Composant de communication java-python

Notre application java communique avec le moteur python au moyen de sockets. Nous avons implémenté un programme java qui s'occupe de l'émission et de la réception des requêtes/réponses des clients java ; et un serveur python qui s'occupe de l'échange entre le moteur et notre application. Cette méthode de communication a été choisie pour son efficacité en terme de temps de réponses par rapport aux autres middleware que nous avons essayé tels que :

Jython : qui permet de faire communiquer une application java avec une application python, mais qui est très lent en terme de temps de réponse.

3.3.1 Côté java

Du côté de java, notre programme ouvrira une socket de connexion pour envoyer toutes nos formules et l'environnement d'évaluation au serveur python.

3.3.2 Serveur python

Pour communiquer avec le moteur fourni en python, ce composant sert de pont entre notre programme java et le moteur. Il permettra de lancer un serveur python qui sera en attente des requêtes.

Conclusion

Ce projet nous a permis de mettre en pratique les connaissances théoriques déjà apprises dans les différents UE de notre parcours.

Grâce à ce projet nous avons appris :

- **La gestion d'un projet informatique**
- **Latex**
- **beamer**
- **Git**

Comme toute œuvre humaine, nous avons été confronté à certaines difficultés dont entre autres :

- **communication python & java (résolu avec les sockets tcp)**
- **mise en place de l'interface graphique en 3D (réalisation en 2D)**

.1 Fiches détaillées

cas 1 : créer un environnement

Identification

Nom du cas : Créer un environnement

But : Décrire les étapes permettant à un utilisateur de créer un jardin via notre application

Acteur principal : Utilisateur

Acteur Secondaire : néant

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version : 1.0

Séquencement :

Le cas d'utilisation commence lorsque l'application est démarrée

Préconditions :

néant

Enchaînement nominal :

1. L'utilisateur choisit «créer jardin»
2. Le système affiche un jardin vide
3. l'utilisateur sélectionne « ajouter fleur »
4. le système ajoute la fleur à la position suivante dans le jardin

Post-conditions

Le jardin est visible sur l'interface graphique.

cas 2 : Sauvegarder un environnement

Identification

Nom du cas : Sauvegarde d'un environnement

But : Décrire les étapes permettant à un utilisateur de sauvegarder un jardin

Acteur principal : Utilisateur

Acteur Secondaire :

néant

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version : 1.0

Séquencement :

Le cas d'utilisation commence lorsqu'un jardin a été créé ou chargé

Préconditions :

Un jardin est créé ou chargé selon le cas d'utilisation usecase1

Enchaînement nominal :

1. L'utilisateur choisit «sauvegarder jardin»
2. Le système ouvre le menu de dialogue standard lui permettant de choisir le répertoire de sauvegarde
3. l'utilisateur choisit son répertoire de sauvegarde
4. l'utilisateur saisit le nom du fichier à sauvegardé
5. le système enregistre le jardin dans ce répertoire.

Post-conditions

Le jardin est bien sauvegardé dans le répertoire choisi par l'utilisateur.

cas 3 : Charger un environnement**Identification**

Nom du cas : Chargement d'un jardin existant

But : Décrire les étapes permettant à un utilisateur de charger un jardin existant

Acteur principal : Utilisateur

Acteur Secondaire :

néant

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version : 1.0

Séquencement :

Le cas d'utilisation commence lorsque l'application est démarrée

Préconditions :

L'utilisateur dispose d'un jardin sauvegardé sur son répertoire

Enchaînement nominal :

1. L'utilisateur choisit «Charger
2. Le système ouvre le menu de dialogue standard lui permettant de choisir le répertoire où se trouve son jardin.
3. l'utilisateur choisit le jardin à charger et valide
4. le système charge le jardin dans l'interface graphique .

Enchaînement d'exception

E1 : L'utilisateur annule le chargement

L'enchaînement reprend au point 1 du cas nominal

Post-conditions

Le jardin sélectionné depuis son répertoire est bien chargé sur l'interface graphique.

cas 4 : Imprimer un environnement**Identification**

Nom du cas : Impression d'un jardin

But : Décrire les étapes permettant à un utilisateur d'imprimer un jardin

Acteur principal : Utilisateur

Acteur Secondaire :

néant

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version :1.0

Séquencement :

Le cas d'utilisation commence lorsque l'application est démarrée

Préconditions :

L'utilisateur dispose d'un jardin sur l'interface graphique

Enchaînement nominal :

1. L'utilisateur choisit «Imprimer jardin»
2. Le système ouvre le menu de dialogue standard lui permettant de choisir l'imprimante.
3. l'utilisateur choisit l'imprimante et valide.
4. le système lance l'impression.

Enchaînement d'exception

E1 : L'utilisateur annule l'impression

L'enchaînement reprend au point 1 du cas nominal

Post-conditions

L'utilisateur dispose sur papier imprimé son jardin

cas 5 : créer une formule

Identification :

Nom du cas : «créer une formule »

But : Décrire les étapes permettant à l'utilisateur de créer une formule

Acteur principal : Utilisateur

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version :1.0

Séquencement :

Le cas d'utilisation démarre lorsque l'application est démarrée

Préconditions :

néant

Enchaînement nominal :

1. L'utilisateur choisit « créer une formule »
- 2 . Le système ouvre un espace d'édition des formules

3. L'utilisateur saisit sa formule.

Post-conditions

L'espace d'édition des formules contient bien une formule

cas 6 : charger une formule

Identification :

Nom du cas : Charger une formule

But : Décrire les étapes permettant à l'utilisateur de charger une formule.

Acteur principal : Acteur

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version :1.0

Séquencement :

Le cas d'utilisation démarre lorsque l'application est démarrée

Préconditions :

néant

Enchaînement nominal :

1.L'utilisateur choisit « charger une formule »

2.Le système ouvre un menu déroulant contenant les fichiers des formules stockées en mémoire

3.L'utilisateur sélectionne le fichier contenant sa formule.

4.Le système affiche les différentes formules contenues dans le fichier

Post-conditions

L'espace d'édition des formules contient bien toutes les formules du fichier

cas 7 : Sauvegarder une formule

Identification :

Nom du cas : Sauvegarder une formule

But : Décrire les étapes permettant à l'utilisateur de sauvegarder une formule

Acteur principal : Acteur

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version :1.0

Séquencement :

Préconditions :

Il existe au moins une formule sur l'espace d'édition des formules.

Enchaînement nominal :

1. l'utilisateur choisit « sauvegarder une formule »

2. Le système ouvre une fenêtre permettant à l'utilisateur de saisir le nom et l'emplacement du fichier
3. l'utilisateur saisit un nom ,sélectionne l'emplacement du fichier et valide
4. le système enregistre le fichier.

Post-conditions

Un fichier contenant des formules est bien crée à l'emplacement choisi.

cas 8 : imprimer une formule

Identification :

Nom du cas : Imprimer une formule

But : Décrire les étapes permettant à utilisateur d'imprimer une formule

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version :1.0

Séquencement :

Préconditions :

Il existe au moins une formule sur l'espace d'édition des formules.

Enchaînement nominal :

- 1.l'utilisateur choisit « imprimer une formule »
2. le système affiche la fenêtre standard d'impression pour choisir l'imprimante
3. l'utilisateur confirme.
4. le système lance l'impression.

Post-conditions

La formule est bien imprimée sur papier

cas 9 : évaluer une formule

Identification :

Nom du cas : Évaluer une formule

But : Décrire les étapes permettant à l'utilisateur d'évaluer une ou plusieurs formules

Date de création : 04/03/2016

Date de mise à jour : 04/03/2016

Responsable : Ousmane Jean Marc

Version :1.0

Séquencement :

Préconditions :

Il existe au moins une formule sur l'espace d'édition des formules.

Enchaînement nominal :

- 1.l'utilisateur choisit «évaluer»

2. Le système lance l'analyseur syntaxique pour vérifier la ou les formules
3. Le système lance l'outil d'évaluation de formules
4. le système affiche le résultat.

Post-conditions

Le résultat de l'évaluation est bien visible à l'écran

Enchaînement alternatif :

A1 : Formule mal formée

A l'étape 2 du cas nominal si la formule est mal formée, le système demande à l'utilisateur de corriger sa formule.

L'enchaînement reprend au cas nominal 1.