

# Interface graphique pour la Logique en L3

## Présentation du rapport du projet de l'UE PSAR

Diallo Ousmane 3-Katuitshi-Ntumba Jean Marc

Encadrants:

Mme Béatrice Bérard

Mr Mathieu Jaume

Mme Bénédicte Legastelois

Université Pierre et Marie Curie  
Master 1 Informatique spécialité SAR

02 Mai 2016 15H40

# Problématique et contexte

Le projet **Interface graphique pour la Logique en Licence 3**

Objectif : développer un outil dynamique et robuste afin d'améliorer l'enseignement de la logique.

# Logique du 1er ordre

## Formules

$F ::= P(x_1, \dots, x_n) | \neg F | F_1 \wedge F_2 | F_1 \vee F_2 | F_1 \implies F_2 | \forall x F | \exists x F$   
où  $x_i$  est une variable ou une constante,  $P$  un prédicat,  $F_i$  des formules.

## Prédicats

Nous avons trois types de prédicat :

unaire : `Rose(a)`

binaire : `a_est_de(x,a)`

ternaire : `entre(x,y,b)`



# Analyse syntaxique de formules

formules

Variables Connecteurs Positions Couleurs

Espèces Taille Constantes

$\forall$   $\exists$   $1$   $=$   $\vee$   $\wedge$   $\Rightarrow$   $($

)

Formules

Ajouter Analyser Evaluer Tout

Rose(x) ☒ ok

Rose(ax) ☒ nok

$\exists x \text{ est\_entre}(x,y,z) \Rightarrow x=a$  ☒ ok

$a\_est(x) \forall x$  ☒ nok

$a\_est(a) \Rightarrow (\forall x \text{ Rose}(x))$  ☒ ok

# Evaluation de formule

logique du 1er ordre

Échier A propos

espece ☐ rose ☐ paquerette ☒ tulipe












Taille ☒ Petite ☐ Moyenne ☐ Grande

Couleur ☐ Rose ☒ Rouge ☐ Blanche

Nom Nom des fleurs g

action

jardin

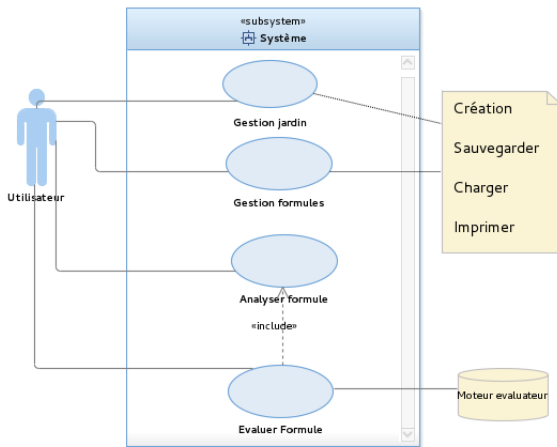
							
 g		 d					
				 a			
							
							

formules

Variables ☒ Connecteurs ☒ Positions ☒ Couleurs

Espece Taille Constantes

# Structure globale de l'application



Outils utilisés : eclipse, python, javaCC, Git, Latex, Beamer

# Composants du projet

Nous avons découpé le projet en 6 composants :

- Jardin : gestion du jardin.
- Formule : gestion des formules.
- Analyseur : analyseur syntaxique des formules.
- Moteur : moteur qui s'occupe de l'évaluation des formules (fourni en python).
- Connexion : gestion de la connexion entre notre application et le serveur python.
- Serveur : s'occupe de la gestion et du traitement des requêtes d'évaluation des formules.



# Jardin

## Jardin

- création du jardin
- sauvegarde du jardin
- chargement du jardin
- ajout d'une fleur dans le jardin
- déplacer une fleur dans le jardin
- supprimer une fleur dans le jardin

# Gestion des formules

## Formule

- création d'une formule.
- sauvegarde des formules.
- chargement des formules.

## Analyseur syntaxique

- insertion d'une formule à analyser
- analyse de la formule
- renvoi du résultat :
  - false si incorrecte
  - true et arbre syntaxique généré si correcte

# Communication java & python

## Connexion

- établissement de la connexion
- préparation du jardin à envoyer sur la socket
- envoi sur la socket du jardin
- préparation des formules à envoyer sur la socket
- envoi sur la socket les formules
- attente/réception du résultat

# Communication java & python

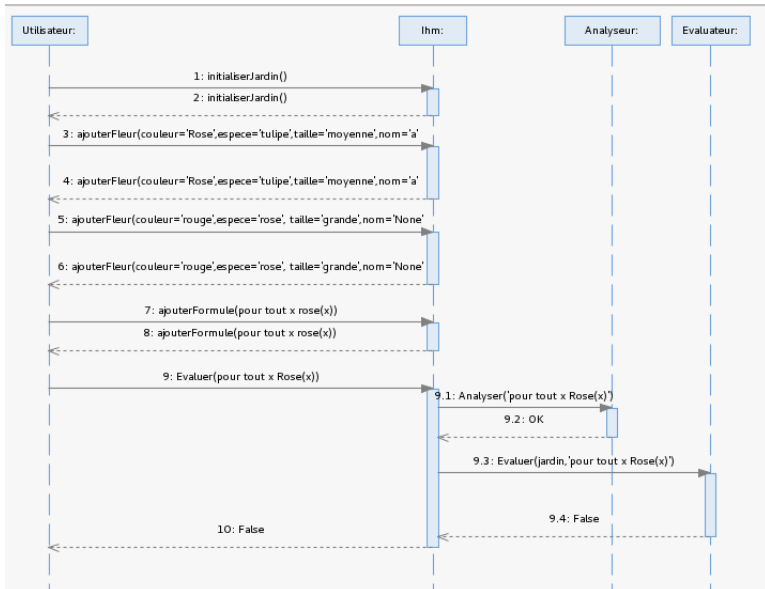
## Serveur python

- création de la socket de communication
- attente de connexion
- attente des requêtes d'évaluation
- lancement du moteur
- envoi du résultat sur la socket

## Moteur python

- évaluation d'un ensemble de formules dans un jardin.
- renvoie pour chaque formule :
  - true si elle est vraie
  - false sinon

# Diagramme de séquence



# Conclusion

## Objectif et Réalisations

- Objectif : développer une interface graphique pour l'enseignement de la logique en L3.
- Réalisations :
  - Une interface graphique en 2D
  - Un analyseur syntaxique de formules
  - Communication java & python

## Problèmes rencontrés

- Mise en place de l'interface en 3D.
- Communication java & python

## Perspectives d'amélioration

- Mise en place de l'application en 3D.
- Amélioration de l'analyseur syntaxique