

# Implémentation de la stratégie mixte

Islem BASSOUMI

November 25, 2023

## 1 Introduction

Dans le cadre de ce TP, l'objectif était de développer une stratégie d'interaction utilisateur plus flexible et dynamique. Cette amélioration permet au système de comprendre des phrases intégrant plusieurs informations simultanément, contrairement aux versions antérieures qui ne pouvaient traiter qu'une seule information à la fois. Par exemple, le système est désormais capable de traiter des requêtes complexes telles que : "Je souhaite réserver la salle 314 à la Maison de la Formation Jacques André pour le Programme Grande École, le lundi 1er octobre à 10 heures", ou "Demande de réservation de la salle 108 du bâtiment U3 à 14 heures, le dix octobre 2022". Cette avancée représente une étape significative dans l'amélioration de l'interaction utilisateur, rendant le système plus intuitif et efficace dans la compréhension et le traitement des demandes multiples et variées.

## 2 Implémentation de reservation.grxml

La stratégie mixte implémentée pour la reconnaissance de réservation de salles fonctionne comme suit :

- **Modularité des Règles** : Chaque composante de la demande (localisation, date, horaire, etc.) est gérée par une règle spécifique.
  - *Localisation* : Traitée par la règle référencée dans '`<ruleref uri="grammaire_salle.grxml#localisation"/>`'.
  - *Date* : Gérée par '`<ruleref uri="grammaire_dates_v3.grxml#Date"/>`'.
  - *Horaire* : Spécifiée dans '`<ruleref uri="grammaire_horaire_v1.grxml#Horaire"/>`'.
  - *Objet de la Réunion* : Défini par '`<ruleref uri="grammaire_objet_reunion.grxml#objet_reservation"/>`'.
- **Flexibilité de Séquençage** : La règle principale ('`<rule id="reservation">`') permet un séquençage flexible des informations.
  - Utilisation de '`<item repeat="1-">`' pour accepter des séquences variables d'éléments.
- **Intégration des Règles de Politesse** : Des éléments tels que "`s'il vous plaît`" et "`merci`" sont intégrés pour une interaction plus naturelle et polie.
  - Ces expressions peuvent apparaître ou non dans la demande, comme indiqué par '`<item repeat="0-1">`'.
- **Extraction et Stockage des Informations** : Chaque règle modulaire extrait des informations spécifiques et les stocke dans l'objet de sortie ('`out`').
  - Par exemple, la localisation est stockée dans '`out.salle`' et '`out.batiment`'.

Ceci illustre comment la stratégie mixte permet de traiter des demandes complexes contenant plusieurs types d'informations en une seule entrée.

### 3 Tests

Dans cette section, on va présenter plusieurs tests pour permettre la validation de notre grammaire reservation.grxml. Dans un premier temps, on va commencer par tester si notre grammaire accepte une informations à la fois et tester par la suite différentes combinaisons d'informations.

```
Enter an utterance or press Enter to finish:
je veux reserver la salle 101
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  salle: "101"
}
```

Figure 1: Test de l'information 'localisation'.

```
Enter an utterance or press Enter to finish:
je veux reserver le deux mars deux mille vingt trois
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  nb_jour: 2,
  num_mois: 2,
  nb_annee: 2023
}
```

Figure 2: Test de l'information 'date'.

```
Enter an utterance or press Enter to finish:
je veux reserver a midi trente
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  heure: 12,
  minute: 30
}
```

Figure 3: Test de l'information 'horaire'.

```
Enter an utterance or press Enter to finish:
je veux reserver pour cours
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  objet: "cours"
}
```

Figure 4: Test de l'information 'objet'.

```
Enter an utterance or press Enter to finish:
je souhaiterais reserver la salle 101 U3 pour un cours le lundi deux mars deux mille vingt trois a midi trente
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  salle: "101",
  batiment: "U3",
  objet: "cours",
  nb_jour: 2,
  num_mois: 2,
  nb_annee: 2000,
  heure: 12,
  minute: 30
}
```

Figure 5: Exemple 1 de plusieurs informations à la fois.

```
Enter an utterance or press Enter to finish:
je veux reserver pour une table ronde la salle 102 a treize heure dix lundi trois mai deux mille vingt deux
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  objet: "table ronde",
  salle: "102",
  heure: 13,
  minute: 10,
  nb_jour: 3,
  num_mois: 4,
  nb_annee: 2022
}
```

Figure 6: Exemple 2 de plusieurs informations à la fois.

```
Enter an utterance or press Enter to finish:
je veux reserver a dix sept heures la salle 102 du U1 pour un seminaire le deux avril
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  heure: 17,
  minute: 0,
  salle: "102",
  batiment: "du U1",
  objet: "seminaire",
  nb_jour: 2,
  num_mois: 3,
  nb_annee: 0
}
```

Figure 7: Exemple 3 de plusieurs informations à la fois.

```
Enter an utterance or press Enter to finish:
je souhaiterais le deux juin deux mille vingt trois reserver la salle 102 merci
Semantic result: {
  reservation: {
  },
  nb_jour: 2,
  num_mois: 5,
  nb_annee: 2000,
  heure: 20,
  minute: 3,
  demande: "reserver",
  salle: "102"
}
```

Figure 8: Exemple 4 de plusieurs informations à la fois.

```
Enter an utterance or press Enter to finish:
je veux reserver la salle 101 pour un cours merci
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  salle: "101",
  objet: "cours"
}
```

Figure 9: Exemple 5 de plusieurs informations à la fois.

```

Enter an utterance or press Enter to finish:
je souhaiterais reserver a seize heure moins le quart lundi deux juillet deux mille vingt trois pour une reunion merci
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  heure: 20,
  minute: 3,
  nb_jour: 2,
  num_mois: 6,
  nb_annee: 2000,
  objet: "reunion"
}

```

Figure 10: Exemple 6 de plusieurs informations à la fois.

```

Enter an utterance or press Enter to finish:
je veux reserver pour une reunion a midi la salle 101
Semantic result: {
  reservation: {
  },
  demande: "reserver",
  objet: "reunion",
  heure: 12,
  minute: 0,
  salle: "101"
}

```

Figure 11: Exemple 7 de plusieurs informations à la fois.

```

Enter an utterance or press Enter to finish:
je veux demander une reservation lundi deux mars deux mille six pour une reunion a midi douze
Semantic result: {
  reservation: {
  },
  demande: "demander une reservation",
  nb_jour: 2,
  num_mois: 2,
  nb_annee: 2000,
  heure: 12,
  minute: 12,
  objet: "reunion"
}

```

Figure 12: Exemple 8 de plusieurs informations à la fois.

## 4 Conclusion

Ce travail a présenté l'implémentation réussie d'une stratégie mixte . Grâce à cette approche, le système peut désormais traiter des demandes complexes, intégrant simultanément plusieurs informations. Les tests effectués ont validé l'efficacité de cette implémentation, démontrant sa capacité à comprendre diverses combinaisons de données.