

Rapport de Bureau d'études

Dialogue Oral Homme-Machine

Janvier 2018

3ASRI - S9 - 2017/2018

Yoann FLEYTOUX
Chris IBRAHIM

À l'attention de Mme. Isabelle Feranné

Table des matières

Table des matières	2
Définition du problème	3
Unités sémantiques utilisés	3
Modèle de langage	3
Choisir action :	4
Ajouter rendez-vous :	5
Supprimer rendez-vous :	6
Consulter rendez-vous :	7
Modifier rendez-vous :	8
Stratégies d'interaction	9
Stratégies de confirmation	9
Problèmes rencontrés	9
Exemples d'exécution	10

Définition du problème

Le but de ce TP/BE est de finaliser la version 1 de l'application de dialogue pour la gestion prise de rendez-vous réalisée dans les TPs précédents, et de développer la version 2. La version 2 permettra à l'utilisateur de réaliser 4 choix d'actions différentes :

- **ajouter** un nouveau rendez-vous, qui correspond à la fonction réalisée dans la version 1 de l'application
- **consulter** les rendez-vous selon un ou plusieurs critères
- **modifier** un rendez-vous selon un ou plusieurs critères
- **supprimer** un rendez-vous selon un ou plusieurs critères

Le but est de recueillir pendant ces scénarios des informations afin de générer en bout de fil une requête *SQL* qui serait envoyée à une base de données *SQL*.

Unités sémantiques utilisés

La plupart des concepts utilisés ou slots utilisés dans l'application V2 sont les mêmes que ceux de la V1. Néanmoins certains concepts ont été rajoutés pour pouvoir générer les requêtes *SQL*.

Les différents slots correspondant aux différentes informations sont les suivantes :

- *num_abonne* contient le numéro d'abonné
- *date_rdv* contient la date du rendez-vous
- *horaire_rdv* contient l'horaire du rendez-vous
- *lieu_rdv* contient le lieu du rendez-vous
- *objet_rdv* contient l'objet du rendez-vous
- *periode_rdv* (matin, midi, après-midi) utilisé dans **consulter rendez-vous**
- *confirm* permet de confirmer un slot
- *orientation* utilisé dans **choix rendez-vous** pour indiquer l'action choisie

Selon les phrases dites, (voir fichier ***orientation_grammaire.grxml***) , le slot orientation aura pour valeur possible de tag *COMMAND* les valeurs suivants, lesquels constitueront une partie de la requête *SQL* qui sera générée en fin de scénario :

- *insert* : signifie ajouter un rendez-vous
- *update* : permet de modifier un rendez-vous
- *select* : permet de consulter un rendez-vous
- *delete* : permet de supprimer un rendez-vous

Modèle de langage

Voici comment interagissent les différents sous-dialogues des fichiers .vxml en fonctions des réponses de l'utilisateur.

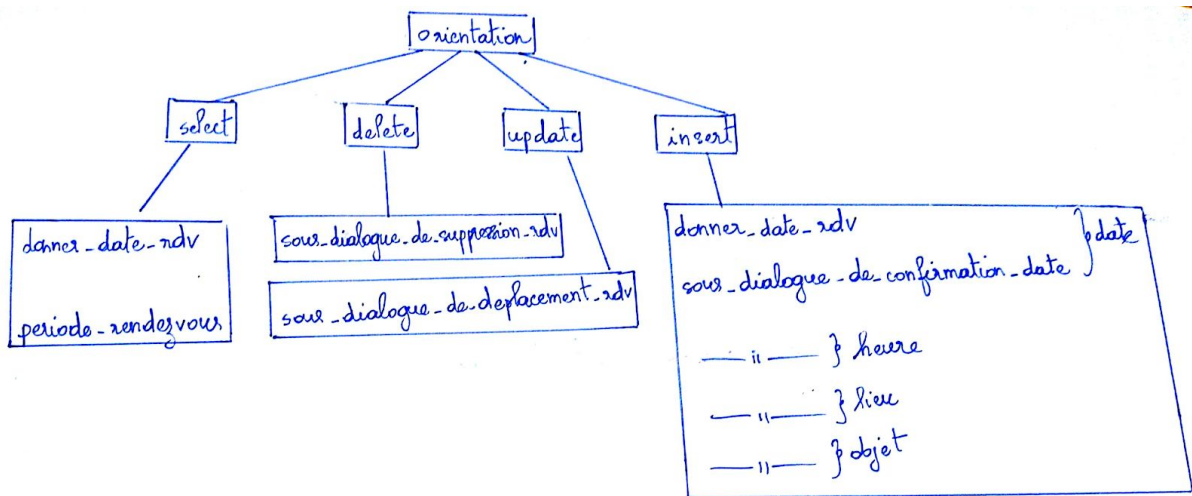


Figure 1: Schéma hiérarchique des sous-dialogues utilisés dans le système entier

Notons également que dans les sous-parties suivantes, les diagrammes illustrant les modèles de langage ne montrent pas toutes les phrases qui seront dites par le système dans les sous-dialogues mais elles décrivent seulement brièvement leur déroulement. De plus, on n'illustrera pas non plus dans les diagrammes la capacité de l'utilisateur de quitter et de demander de l'aide à tout moment.

Les dialogues réalisés sont formés à partir de l'imbrication et de l'enchaînement de différents sous-dialogues ayant une structure similaire. Nous mettrons en évidence à la fin de chaque modèle de langage, le sous-dialogue principal où l'on retourne à la fin du scénario.

Choisir action :

Le fichier **orientation.vxml** sert de "menu principal" en permettant de basculer d'un scénario à l'autre selon les choix de l'utilisateur et en envoyant la commande SQL qui correspond à chaque choix de l'utilisateur et à ses requêtes.

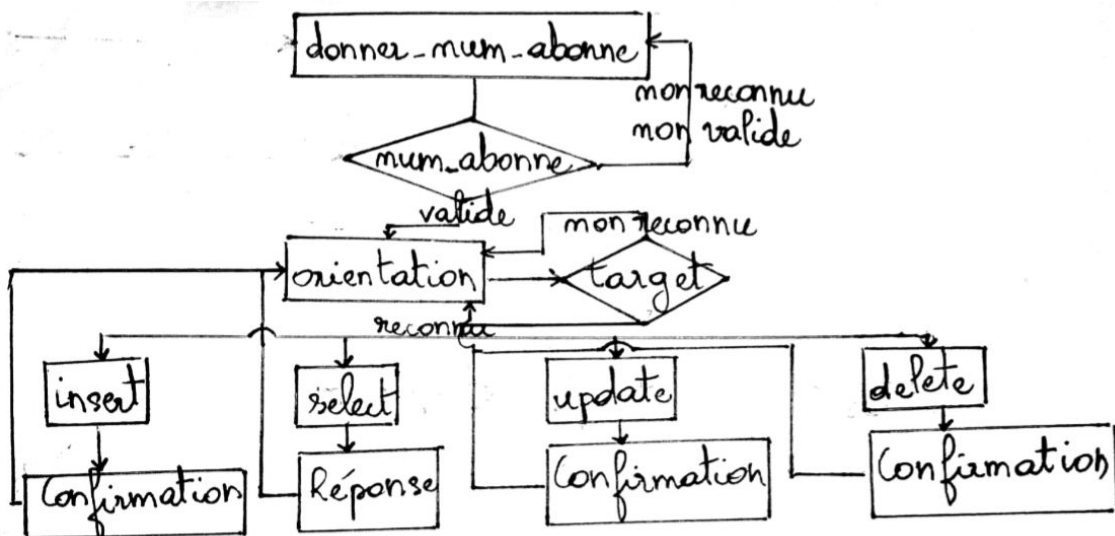


Figure 2 : Schéma modèle de langage choisir action

Chaque confirmation ou réponse générée suites aux différents scénarios est obtenue à l'aide d'un hypothétique retour de requête SQL générée à partir des informations recueillies par le système à partir des réponses de l'utilisateur. Après ces différentes confirmations et réponses possibles, on retourne dans le scénario principale en revenant au formulaire **orientation** où l'utilisateur peut à nouveau effectuer une requête (voir fin Exemples 1,2,3,4).

Ajouter rendez-vous :

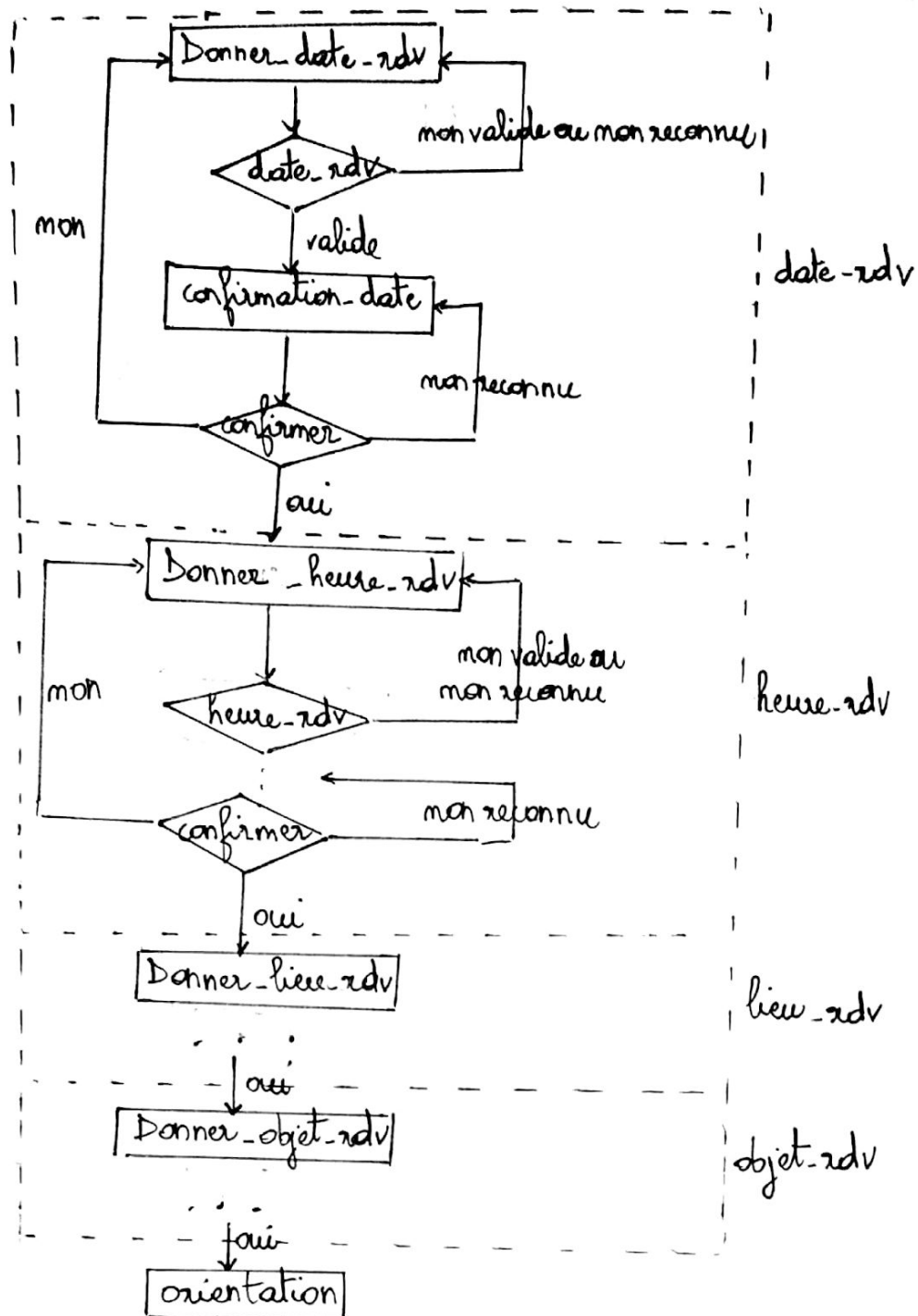


Figure 3 : Schéma modèle de langage ajouter rendez-vous

Pour chaque information du rendez-vous nécessaire au système (date, heure, lieu, objet), dans ce scénario (voir fichier *insert.vxml*), le système demande chaque fois à l'utilisateur de donner l'information, puis de confirmer. A la fin du scénario, c'est-à-dire après la dernière confirmation de l'utilisateur, le système sort du scénario **ajouter rendez-vous** pour revenir au choix de l'action en récapitulant toutes les informations recueillies (sortie pas montrée dans le diagramme, mais visible en fin Exemple 1.2). La requête SQL correspondante serait celle-ci :

```
INSERT INTO rendez-vous(jour,mois,annee,heure,minute,lieu,objet,num_abonne) VALUES
(date_rdv.nb_jour, date_rdv.nb_mois, date_rdv.nb_annee, horaire_rdv.h, horaire_rdv.min,
lieu_rdv.lieu , objet_rdv.objet, num_ab.nb)
```

Supprimer rendez-vous :

Le diagramme suivant montre les formulaires qui s'enchaînent à l'activation du sous dialogue ou scénario **supprimer rendez-vous** du fichier *supprimer_rdv_sous_dialogue.vxml* lequel est appelé par *delete.vxml*.

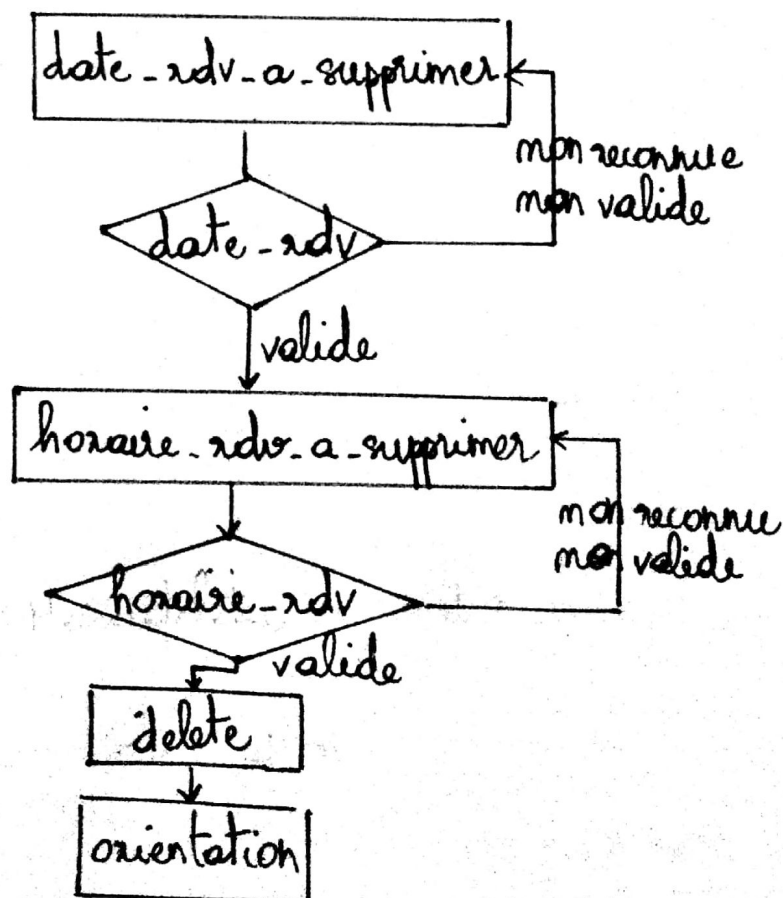


Figure 4 : Schéma modèle de langage supprimer rendez-vous

A la fin du scénario, c'est-à-dire après la dernière réponse de l'utilisateur (un horaire de rendez-vous valide), le système sort du scénario **supprimer rendez-vous** pour revenir au

formulaire **delete** puis au scénario **choisir action** en récapitulant l'action qui sera effectuée (phase pas montrée dans le diagramme, mais visible en fin Exemple 4). La requête SQL correspondante serait celle-ci :

```
DELETE * FROM rendez-vous WHERE num_ab.nb AND date_rdv_a_supprimer.nb_jour
AND date_rdv_a_supprimer.nb_mois
AND date_rdv_a_supprimer.nb_annee
AND horaire_rdv_a_supprimer.h
AND horaire_rdv_a_supprimer.min
```

Consulter rendez-vous :

Le diagramme suivant montre les formulaires qui s'enchaînent à l'activation du scénario **consulter rendez-vous** du fichier **select.vxml**.

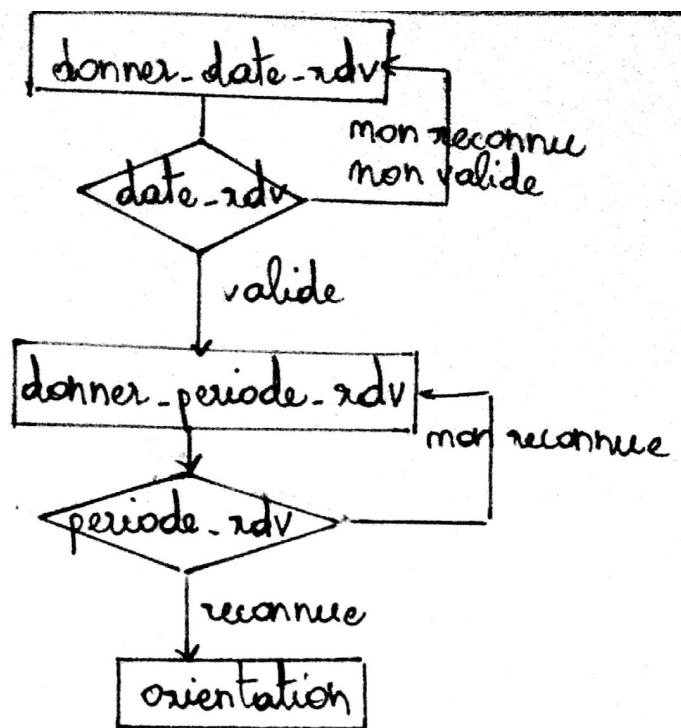


Figure 5 : Schéma modèle de langage consulter rendez-vous

A la fin du scénario, c'est-à-dire après la dernière réponse de l'utilisateur (un horaire de rendez-vous valide), le système sort pour revenir au **choix de l'action** en affichant les informations relatives au rendez-vous que recherchait l'utilisateur (sortie pas montrée dans le diagramme, mais visible en fin Exemple 2). La requête SQL correspondante serait celle-ci :

```
SELECT * FROM rendez-vous WHERE num_ab.nb
AND date_rdv_a_supprimer.nb_jour
AND date_rdv.nb_mois
AND date_rdv.nb_annee
```

Modifier rendez-vous :

Le diagramme suivant montre les formulaires contenues dans **deplacer_rdv_sous_dialogue.vxml** qui est appelé à l'activation du scénario **modifier rendez-vous** du fichier **update.vxml**.

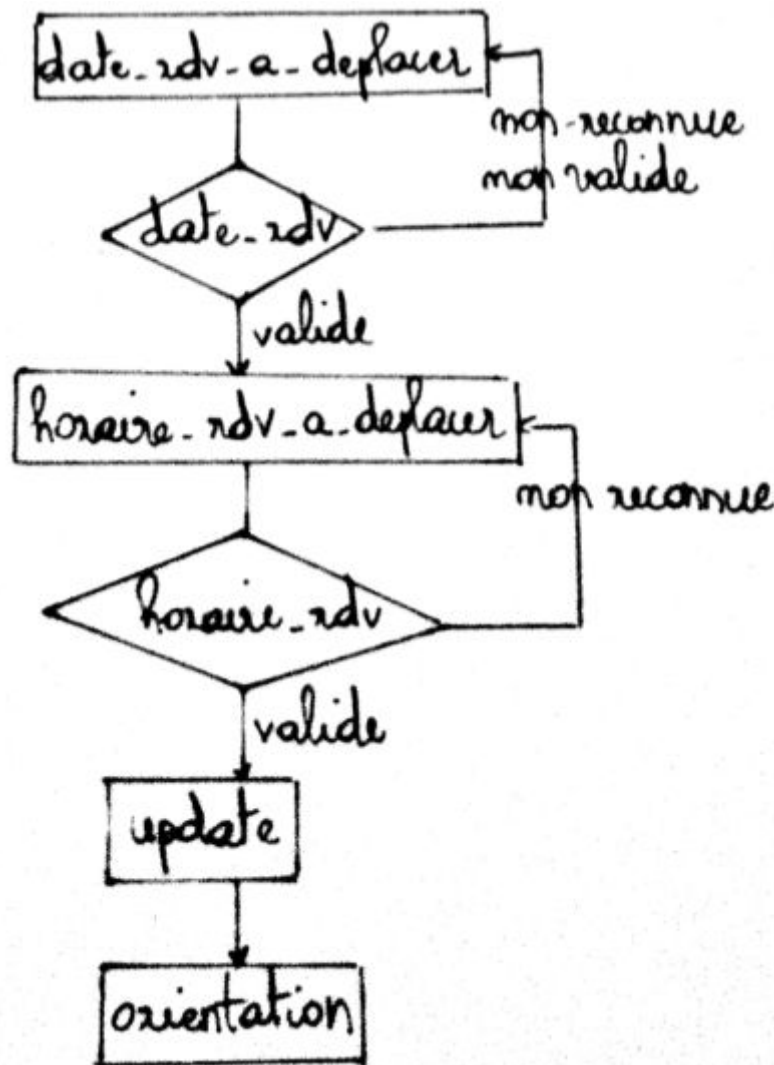


Figure 6 : Schéma modèle de langage modifier rendez-vous

Ainsi la fin du scénario, c'est-à-dire après la dernière réponse de l'utilisateur (un horaire de rendez-vous valide), le système sort pour revenir dans **update** puis au **choix de l'action** en affichant les informations relatives aux modifications du rendez-vous qu'a effectuées l'utilisateur (sortie pas montrée dans le diagramme, mais visible en fin Exemple 3). La requête SQL correspondante serait celle-ci :

```
UPDATE rendez-vous SET horaire_rdv_a_deplacer.h AND horaire_rdv_a_deplacer.min  
WHERE num_ab.nb
```

```
AND date_rdv_a_supprimer.nb_jour
```

```
AND date_rdv_a_deplacer.nb_mois
```

```
AND date_rdv_a_deplacer.nb_annee
```


Stratégies d'interaction

Comme stratégie d'interaction, le moteur de dialogue utilise une stratégie mixte où chaque fois, le système initiera l'échange en posant des questions à l'utilisateur, sauf dans certains cas spéciaux. En effet, à tout moment du dialogue, l'utilisateur peut demander de l'aide au système pour qu'il lui dise ce qu'il attend de lui ou demander de fermer l'application (voir fin Exemple 1.2). Ces dernières interactions ne sont pas mises en évidence dans les modèles de langages, de crainte de les surcharger mais sont bien possibles.

Stratégies de confirmation

Pour confirmer les réponses de l'utilisateur, les stratégies de confirmations implicites (voir Exemple 2) et explicites (voir Exemple 1) sont utilisés dans différentes parties de la conversation. Il s'agit donc d'une stratégie de confirmation mixte. Le but est de représenter le fait que le système ne demandera pas forcément de confirmation à l'utilisateur s'il a suffisamment confiance à la reconnaissance de parole, tandis que si la reconnaissance n'est pas suffisamment fiable, elle se fait implicitement en embrayant directement avec la question suivante.

Problèmes rencontrés

Des problèmes ont été rencontrés pendant l'exécution, lors des recherches des règles de grammaires à utiliser par le système, dans les parties du dialogue qui en avaient besoin. Cela conduisait à des plantages dans l'application qui pouvaient provenir simplement de collages dans le code d'instructions provenant de pdf.

Le système de dialogue ne remplit pas tous les points qui ont été exigés dans le cahier des charges notamment l'ajout d'une personne à voir pour le rendez-vous et le fait de pouvoir modifier toute information relative au rendez-vous en plus de la date et de l'horaire.

Exemples d'exécution

```
C:\Optimtalk>ot_vxml_interpreter.exe D:\Documents\S9\DOHM_21_01_21_06h\orientati
on.vxml

OptimTalk VoiceXML interpreter (version 1.10)
=====
Copyright (c) 2004-2010 OptimSys, s.r.o., Czech Republic
http://www.optimsys.cz

Interpreting VoiceXML document file:/D:/Documents/S9/DOHM_21_01_21_06h/orientati
on.vxml

Quel est votre numero d'abonne ?
KEYB>le deux
Votre numero d'abonne est le
deux
TRACE VALEUR
2
Quelle est votre requete?
KEYB> je voudrais ajouter un rendez-vous
TRACE VALEUR
ajouter
COMMANDE BDD
SELECT * FROM rendez-vous WHERE num_client=
2
Pourriez-vous me donner la date de votre rendez-vous ?
KEYB> le lundi douze janvier deux mille treize
Votre rendez-vous est le
lundi douze janvier deux mille treize
TRACE VALEUR
nb_jour:
12
nb_mois:
1
nb_annee:
2013
Pouvez-vous me le confirmer?
KEYB> oui
la valeur recupere est True
TRACE Reponse
TRACE date rendez vous confirme
Pourriez-vous me donner l'heure de votre rendez-vous ?
KEYB> a huit heure deux
Vous souhaitez avoir un rendez-vous à
huit heure deux
Votre rendez-vous est a
huit heure deux
TRACE VALEUR heures:
8
minutes:
2
minutes totales:
482
Pouvez-vous me le confirmer?
KEYB> oui
la valeur recupere est True
TRACE Reponse
TRACE heure rendez vous confirme
Pourriez-vous me donner le lieu de votre rendez-vous ?
KEYB> a Paul Sabatier
Votre rendez-vous est a
Paul Sabatier
TRACE VALEUR lieu:
Paul Sabatier
text:
Paul Sabatier
Pouvez-vous me le confirmer?
KEYB> oui
la valeur recupere est True
TRACE Reponse
TRACE lieu rendez vous confirme
Pourriez-vous me donner l'objet de votre rendez-vous ?
```

Exemple 1.1 : Scénario ajout d'un rendez-vous

```

Pourriez-vous me donner l'objet de votre rendez-vous ?
KEYB> c'est pour une revue
Votre rendez-vous est pour
c est pour une revue
TRACE VALEUR objet
une revue
Pouvez-vous me le confirmer?
KEYB> oui
la valeur recupere est True
TRACE Reponse
TRACE objet rendez vous confirme
donner_date_rdv donner_lieu_rdv donner_objet_rdv Vous avez bien ajouter un rende
z-vous a
Paul Sabatier
pour
une revue
durant le
lundi douze janvier deux mille treize
a
huit heure deux
COMMANDE BDD INSERT INTO rendez-vous(jour,mois,annee,heure,minute,lieu,objet,num
_abonne) VALUES(
12
,
1
,
2013
,
8
,
2
,
Paul Sabatier
,
une revue
,num_client =
ERROR: Interpretation failed since an error occurred in call to VoiceXML interpr
eter.
VoiceXML interpreter reports: The following error occurred during the interpreta
tion: Prompt queue exceeded its maximum length.
C:\Optimtalk>

```

Exemple 1.2 : Scénario ajout d'un rendez-vous

```

C:\Optimtalk>ot_vxml_interpreter.exe D:\Documents\S9\DOHM_21_01_21_06h\orientati
on.vxml

OptimTalk VoiceXML interpreter <version 1.10>
=====
Copyright (c) 2004-2010 OptimSys, s.r.o., Czech Republic
http://www.optimsys.cz

Interpreting VoiceXML document file:/D:/Documents/S9/DOHM_21_01_21_06h/orientati
on.vxml

Quel est votre numero d'abonne ?
KEYB> le deux
Votre numero d'abonne est le
deux
TRACE VALEUR
2
Quelle est votre requete?
KEYB> je voudrais consulter mes rendez-vous
TRACE VALEUR
consulter
COMMANDE BDD
SELECT * FROM rendez-vous WHERE num_client=
2
Pourriez-vous me donner la date de votre rendez-vous ?
KEYB> le lundi douze janvier deux mille treize
J'ai entedu le
lundi douze janvier deux mille treize
TRACE VALEUR
12
1
2013
Quel periode souhaitez-vous consulter?
KEYB> le matin
TRACE Valeur
matin
Vous avez un rendez-vous le
lundi douze janvier deux mille treize
durant le
matin
a
8
heures
COMMANDE BDD SELECT * FROM rendez-vous WHERE num_client =
2
AND jour=
12
AND mois =
1
AND annee =
2013
Quelle est votre requete?
KEYB> aurevoir
Aurevoir!

C:\Optimtalk>

```

Exemple 2 : Scénario select

```

C:\OptimTalk>ot_vxml_interpreter.exe D:\Documents\S9\DOHM_21_01_21_06h\orientati
on.vxml

OptimTalk VoiceXML interpreter (version 1.10)
=====
Copyright (c) 2004-2010 OptimSys, s.r.o., Czech Republic
http://www.optimsys.cz

Interpreting VoiceXML document file:/D:/Documents/S9/DOHM_21_01_21_06h/orientati
on.vxml

Quel est votre numero d'abonne ?
le deux
Votre numero d'abonne est le
deux
TRACE VALEUR
2
Quelle est votre requete?
KEYB> je voudrais deplacer un rendez-vous
TRACE VALEUR
deplacer
COMMANDE BDD
SELECT * FROM rendez-vous WHERE num_client=
2
Donner la date du rendez-vous que vous souhaiteriez deplacer?
KEYB> le lundi douze janvier deux mille treize
Merci! On vous change cela
la valeur recupere est lundi douze janvier deux mille treize
Donner maintenant l'horaire du rendez-vous que vous souhaiteriez deplacer?
KEYB> a huit heures deux
Merci! On vous change cela
la valeur recupere est huit heures deux
Vous avez donc maintenant un rendez-vous le
lundi douze janvier deux mille treize
a
huit heures deux
COMMANDE BDD UPDATE rendez-vous SET heure=
8
.minute=
2
WHERE num_client =
2
AND jour=
12
AND mois =
1
AND annee =
2013
Quelle est votre requete?
Je n'ai pas entendu. Pouvez-vous me preciser votre requete ?
Pourriez-vous me donner la date de votre rendez-vous ?
KEYB> *** USER HUNG UP ***

```

Exemple 3 : Scénario update

```

Copyright (c) 2004-2010 OptimSys, s.r.o., Czech Republic
http://www.optimsys.cz

Interpreting VoiceXML document file:/C:/Users/user/Documents/DOHM_21_01_21_06h/orientation.vxml

Quel est votre numero d'abonne ?
KEYB> douze
Votre numero d'abonne est le
douze
TRACE VALEUR
12
Quelle est votre requete?
KEYB> je souhaiterais annuler un rendez-vous
TRACE VALEUR
annuler
COMMANDE BDD
SELECT * FROM rendez-vous WHERE num_client=
12
Donner la date du rendez-vous que vous souhaiteriez supprimer?
KEYB> lundi deux mars deux mille seize
la valeur recupere est lundi deux mars deux mille seize
Merci! Donner maintenant l'heure du rendez-vous que vous souhaiteriez suprrimer?
KEYB> minuit
Merci! On vous change cela
la valeur recuperee est minuit
Etes-vous sur? C'est trop tot pour un rendez-vous.
Merci! Donner maintenant l'heure du rendez-vous que vous souhaiteriez suprrimer?
Je n'ai pas entendu. Pouvez-vous repeter ?
KEYB> six heures
Merci! On vous change cela
la valeur recuperee est six heures
Etes-vous sur? C'est trop tot pour un rendez-vous.
Merci! Donner maintenant l'heure du rendez-vous que vous souhaiteriez suprrimer?
KEYB> pouvez-vous m aider
Repondez en donnant une horaire complete (exemple: a huit heure deux) comprise entre 7h30 et 21h30
KEYB> midi quinze
Merci! On vous change cela
la valeur recuperee est midi quinze
Vous avez bien supprimé le rendez-vous du
lundi deux mars deux mille seize
a heure=
12
heures et minute=
15
minutes
COMMANDE BDD DELETE * FROM rendez-vous WHERE num_client =
12
AND jour=
2
AND mois =
3
AND annee=
2016
AND heure=
12
AND minute=
15
Quelle est votre requete?

```

Exemple 4 : Scénario delete