
Un gestionnaire de dialogue oral arabe Homme-machine

Réalisation et évaluation

Younès Bahou — Amine Bayoudhi — Lamia Hadrich Belguith

Laboratoire LARIS – MIRACL

Faculté des Sciences Economiques et de Gestion de Sfax

B.P. 1088, 3018 - Sfax – TUNISIE

Téléphone (216) 74 278 777, Fax (216) 74 279 139

bahou_younes@yahoo.fr, bayoudhi.amine@gmail.com, l.belguith@fsegs.rnu.tn

RÉSUMÉ. Dans le présent papier, nous proposons un gestionnaire de dialogue oral arabe Homme-machine. Ce travail entre dans le cadre de la réalisation du serveur vocal interactif SARF (Bahou et al., 2008) qui offre des renseignements sur le transport ferroviaire tunisien en langue arabe standard moderne.

Le gestionnaire de dialogue, que nous proposons, se base sur une approche structurale et utilise une grammaire formelle transformée en un automate fini déterministe afin d'assurer le suivi et le maintien du dialogue avec l'utilisateur. Les résultats d'évaluation de ce gestionnaire de dialogue sont très encourageants. Ainsi, les taux de réussite des tâches et d'identification des actes de dialogue sont respectivement de 89.65% et 98.31%, le temps moyen d'exécution des tâches est de 0.0779 secondes et les dialogues sont maintenus avec une moyenne de 13 tours de parole par dialogue.

ABSTRACT. In this paper, we propose an Arabic Human-machine oral dialogue manager. This work enters in the context of SARF system (Bahou et al., 2008) an interactive vocal server that provides information on Tunisian railway using modern standard Arabic.

The proposed dialogue manager is based on a structural approach and uses a formal grammar transformed into a deterministic finite automaton to monitor and maintain the dialogue with the user. The evaluation results of this dialogue manager are very encouraging. Thus, the rates of tasks success and of dialogue acts identification are respectively 89.65% and 98.31%, the average time of performance is 0.0779 seconds and the dialogues are held with an average of 13 turns of speech through dialogue.

MOTS-CLÉS : gestionnaire de dialogue Homme-machine, dialogue oral arabe, approche structurale, modèle de tâche, modèle de dialogue.

KEYWORDS: Human-machine dialogue manager, Arabic oral dialogue, structural approach, task model, dialogue model.

1. Introduction

Le domaine du dialogue oral Homme-machine intéresse encore et de plus en plus les chercheurs. Ils visent à améliorer les solutions proposées et de mettre en œuvre des Systèmes de Dialogue Oral Homme-Machine (SDOHM) robustes et efficaces. Le présent travail entre dans le cadre de ces recherches et porte sur la réalisation et l'évaluation d'un gestionnaire de dialogue oral Homme-machine. Ce gestionnaire est capable d'assurer le suivi et le maintien du dialogue avec les utilisateurs même dans les cas d'incidence et de rupture du dialogue. Cette robustesse est obtenue grâce à la méthode sur laquelle se base ce gestionnaire de dialogue. En effet, la méthode que nous proposons dans ce papier se base sur l'approche structurée et se compose d'un modèle de tâche et d'un modèle de dialogue.

Ce travail entre dans le cadre de la réalisation du serveur vocal interactif SARF (Bahou *et al.*, 2008) offrant des renseignements sur le transport ferroviaire tunisien en langue arabe standard moderne. Nous visons, à travers le présent travail, à appliquer et à tester l'approche structurée dans notre méthode de gestion de dialogue en arabe.

2. La gestion du dialogue oral Homme-machine

La gestion du dialogue occupe une place centrale dans un SDOHM et peut être considérée comme le contrôleur principal d'un tel système. Elle reçoit en entrée la structure sémantique engendrée par le système de compréhension et génère en sortie les réponses adéquates à l'utilisateur, à partir des informations extraites de la base de données de l'application. Elle utilise pour ce fait deux types de modèles : un modèle de tâche et un modèle de dialogue illustrés par la figure suivante :

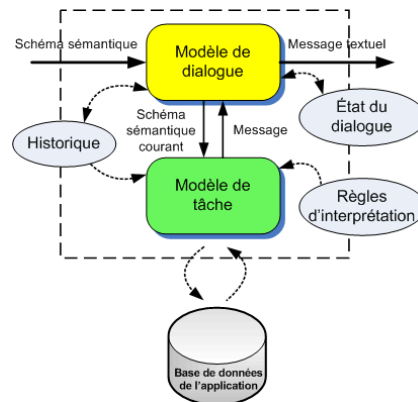


Figure 1. Architecture fonctionnelle d'un gestionnaire de dialogue

2.1. Modélisation des tâches

Une tâche est définie comme étant la réalisation d'un but dans un contexte et selon une procédure dont la représentation et la précision varient en fonction des objectifs dirigeant l'analyse (Caelen et Xuereb, 2007). Nous entendons par *modèle de tâche* la modélisation des différentes connaissances d'une application donnée. Ces connaissances peuvent être les objets ou les concepts manipulés, les relations qu'ils entretiennent entre eux, les règles de déduction qui permettent d'élaborer de nouvelles connaissances, la description d'exécution d'une tâche comme la description des scénarios et la modélisation des objectifs à atteindre. Dans la littérature, diverses approches de modélisation des tâches ont été appliquées dans plusieurs domaines. Elles sont répertoriées en trois approches principales à savoir (Minker et Bennacef, 2004) :

- *Approche fonctionnelle* dans laquelle la représentation et la gestion de tâches mettent en valeur les diverses fonctionnalités du système dans lequel ces tâches opèrent. Ce type d'approche favorise la description des fonctions dont dispose l'utilisateur.
- *Approche structurelle* centrée sur la tâche et orientée vers la résolution de problèmes. Elle décrit les tâches par leurs objectifs et leurs méthodes de résolution. Pour résoudre un problème, elle cherche à décomposer les différentes actions nécessaires conduisant à sa résolution.
- *Approche mixte* combinant les deux approches précédentes. En effet, elle permet d'une part la représentation des tâches en vue de réaliser diverses fonctions dans le cadre d'un dispositif donné, et d'autre part la représentation des actions adaptées à la résolution de problèmes.

2.2. Modélisation du dialogue

La gestion de dialogue cherche à assurer une communication fiable et efficace avec l'utilisateur. Cette communication, pour atteindre ces qualités, doit être bien entretenue et bien modélisée, problème majeur dans la gestion du dialogue. Plusieurs approches ont été citées dans la littérature, nous proposons de les classifier en quatre approches à savoir (Bahou *et al.*, 2009a) :

- *Approche structurelle* qui prend son origine du modèle genevois (Roulet *et al.*, 1991) à partir duquel plusieurs travaux ont été inspirés. Parmi ces travaux, nous citons ceux de Luzzati (Luzzati, 1989), de Rosset et Lamel (Rosset et Lamel, 2001) et de Minker et Bennacef (Minker et Bennacef, 2004). Cette structure décrit autour d'elle des unités hiérarchiques plus complexes entretenant des relations fonctionnelles. La définition de ces deux niveaux d'analyse hiérarchique et fonctionnelle donne naissance à une grammaire hors-contexte décrivant les régularités des échanges dans le dialogue.

– *Approche orientée plan*, inspirée de la théorie de planification (Nilson, 1980), s’est intéressée aux interlocuteurs produisant le dialogue. Elle considère que chaque interlocuteur s’engageant dans la conversation a un comportement rationnel qui l’amène à établir des plans lui permettant d’atteindre ses buts. L’idée centrale de l’approche orientée plan consiste à préétablir l’ensemble des plans possibles de la tâche puis à reconnaître le plan de l’interlocuteur parmi ces plans. Parmi les travaux effectués autour de cette approche nous citons ceux de Rickel *et al.* (Rickel *et al.*, 2001) et de Sidner *et al.* (Sidner *et al.*, 2003).

– *Approche intentionnelle* est en réalité une extension de l’approche orientée plan. Seulement, elle a la faveur d’ajouter un niveau cognitif à la modélisation du dialogue car il ne suffit pas de viser l’organisation de la tâche et du discours pour bien adapter la réponse, mais il faut prendre en considération les intentions de l’interlocuteur. Ainsi, cette approche se développe autour de la logique intentionnelle évoluée avec l’apparition des modèles BDI (*Belief, Desire, Intention*) en intelligence artificielle. Les premiers travaux dans ce courant sont ceux de Cohen, Perrault et Allen (Cohen et Perrault, 1979 ; Allen et Perrault, 1980) qui sont ensuite repris et raffinés par Sadek (Sadek, 1996).

– *Approche conventionnelle* basée sur la théorie des jeux de dialogues conventionnels (Husltin, 2000). Cette théorie suppose que chacun des interlocuteurs est engagé dans un jeu social dont les tours de parole représentent les coups. Le problème peut se ramener ainsi à la gestion des buts qui sont traités selon des stratégies prenant en considération l’état des interlocuteurs et l’état du dialogue. Parmi les travaux les plus persistants dans cette approche, nous citons ceux de Rouillard (Rouillard, 2000) et de Nguyen (Nguyen, 2005).

3. Méthode proposée pour la gestion du dialogue en arabe

Dans cette section, nous proposons une méthode pour la gestion du dialogue oral Homme-machine en arabe basée sur une approche structurale. En effet, cette approche est plus adaptée aux systèmes de recherche d’information grâce à sa souplesse, son adaptation à la complexité de la conversation et sa robustesse lors de la modélisation des différentes situations du dialogue.

La méthode que nous proposons permet de gérer l’interaction avec les utilisateurs et de répondre à leurs demandes d’information selon les spécifications reliées au domaine d’application (*i.e.*, le transport ferroviaire tunisien) et à la langue utilisée (*i.e.*, l’arabe standard moderne). Cette méthode repose principalement sur deux modèles de base (Bahou *et al.*, 2009a) : le modèle de tâche et le modèle de dialogue (voir figure 2). Le premier modèle permet de compléter et vérifier l’incohérence des structures sémantiques représentant les sens utiles des énoncés, de générer une requête vers l’application, de récupérer le résultat et de formuler une réponse à l’utilisateur en langage naturel. Quant au modèle de dialogue, il assure l’avancement du dialogue avec l’utilisateur et l’identification de ses intentions.

L'interaction entre ces deux modèles est assurée grâce à un contexte du dialogue permettant le suivi et la mise-à-jour de l'historique du dialogue.

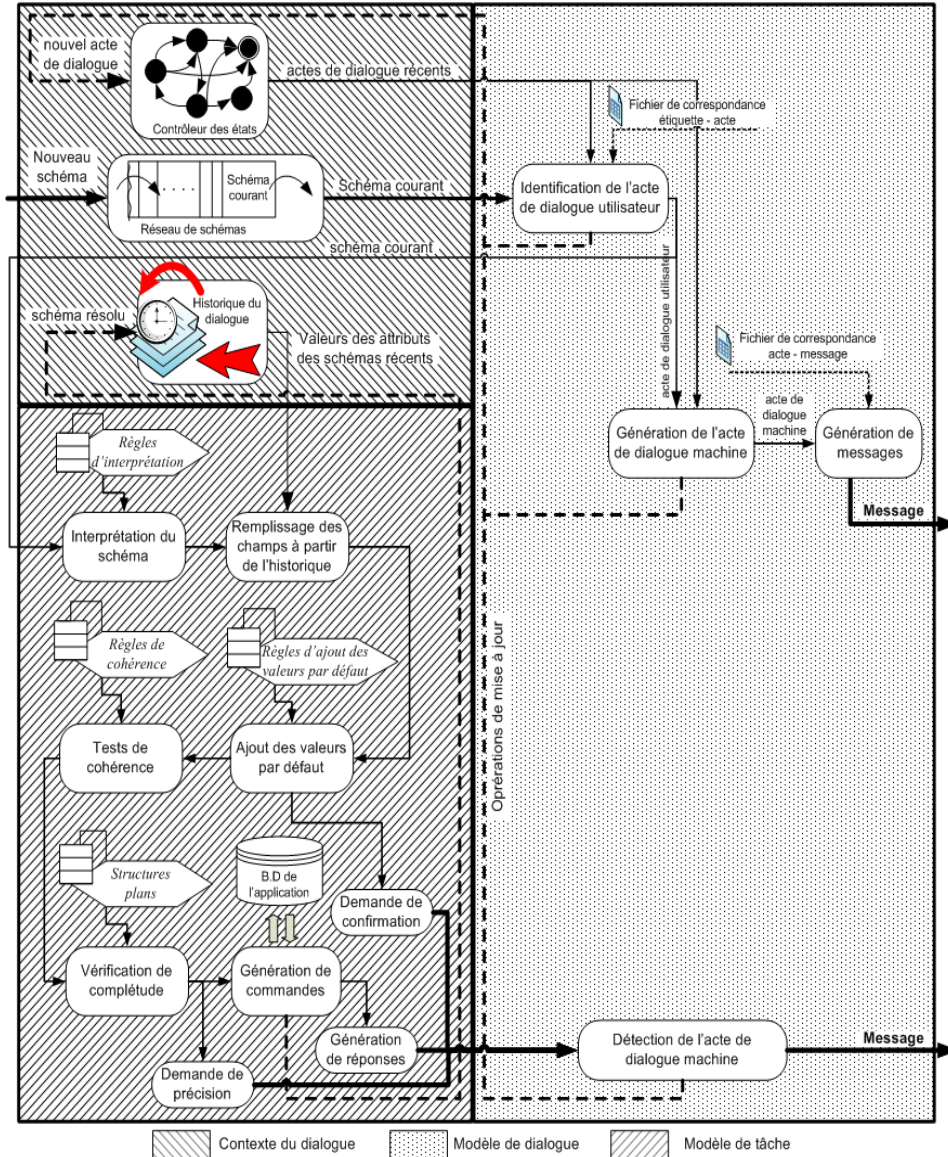


Figure 2. Étapes de la méthode proposée pour la gestion du dialogue oral arabe

3.1. *Modèle de tâche*

Le modèle de tâche est chargé de traiter tous les schémas sémantiques dont les actes de dialogue sont relatifs à la tâche. Son rôle principal est de compléter les structures sémantiques générées par le système de compréhension (Bahou *et al.*, 2008 ; Bahou *et al.*, 2009b ; Hadrich Belguith *et al.*, 2009) et de vérifier sa cohérence afin d'en extraire les informations nécessaires à la génération des requêtes vers le système d'information. Ensuite, il utilise les informations retournées pour formuler des réponses vers l'utilisateur en langage naturel. Pour cela, le modèle de tâche utilise deux principales ressources à savoir, les *structures tâches* et les *structures plans* (Minker et Bennacef, 2004). Les structures tâches contiennent les règles relatives à l'interprétation, la formulation des commandes et la génération de réponses. Quant aux structures plans, elles regroupent les règles de cohérence et les règles de complétude des schémas.

Le modèle de tâche que nous proposons est composé de six étapes à savoir (Bahou *et al.*, 2009a) :

– *Interprétation de la structure sémantique* : le schéma sémantique généré par le système de compréhension peut contenir plusieurs expressions qui nécessitent des reformulations pour qu'elles soient acceptables par le Système de Gestion de Base de Données Relationnelles (SGDB-R). Ainsi, l'étape d'interprétation du schéma s'avère nécessaire pour reformuler et standardiser ces expressions. Cette étape utilise un ensemble de règles d'interprétation dégagées à partir du corpus d'étude. Ces règles sont regroupées dans les *structures tâches*.

– *Instanciation des attributs à partir de l'historique* : étant donné la continuité de l'aspect dialogique, l'utilisateur peut omettre quelques détails qu'il a déjà mentionnés dans ses interventions précédentes. Ce qui explique la présence d'ellipses et d'anaphores au niveau de l'énoncé. Par conséquent, certains attributs peuvent avoir des valeurs vides au niveau du schéma sémantique. Cette étape vise donc à résoudre les ellipses et les anaphores par l'utilisation de l'historique du dialogue afin de rechercher des instances possibles aux attributs vides. Dans le cas où la recherche est fructueuse, l'instance trouvée est automatiquement accordée à l'attribut en question.

– *Instanciation des attributs avec des valeurs par défaut* : une deuxième raison qui peut expliquer l'omission de certains détails par l'utilisateur, est que ce dernier considère ces détails comme étant évidents. Cette étape permet d'instancier certains attributs par des valeurs par défaut. Ces attributs et leurs valeurs sont recensés à partir du corpus d'étude et sont stockés dans les *structures tâches*.

– *Vérification de la cohérence* : la vérification est nécessaire pour valider la cohérence des informations apportées par le schéma sémantique. Elle permet d'alerter l'utilisateur en cas d'erreurs de cohérence au niveau de sa demande. La détection de ces erreurs se fait grâce à un ensemble de règles de cohérence recensées

à partir du corpus d'étude et stockées dans les *structures plan*. En cas de présence d'incohérences, des demandes de confirmation seront générées à l'utilisateur pour corriger ses informations.

– *Vérification de la complétude* : cette étape permet de vérifier l'existence, dans le schéma courant, de toutes les informations nécessaires permettant la génération d'une requête valide vers l'application. Dans le cas échéant, des demandes de précision seront générées pour l'utilisateur afin qu'il puisse compléter les informations oubliées. Pour cela, chaque tâche est dotée d'un plan contenant une liste ordonnée des attributs nécessaires et des messages de demande de précision des attributs omis par l'utilisateur.

– *Génération de réponses* : une fois le schéma sémantique complété, une requête SQL est générée vers la base de données de l'application afin de dégager les informations recherchées par l'utilisateur. Ces informations sont ensuite reformulées sous forme de réponses en langage naturel et envoyées à l'utilisateur. Notons que les interventions de la machine peuvent être aussi sous forme de demandes de confirmation qui ne sont pas directement communiquées à l'utilisateur mais qui passent à travers le modèle de dialogue que nous exposons dans la section suivante.

3.2. *Modèle de dialogue*

Le modèle de dialogue permet de mieux comprendre les interventions de l'utilisateur en identifiant ses intentions et ses buts à atteindre. Il permet aussi de gérer le dialogue et d'assurer son avancement en prévoyant les solutions convenables pour les situations critiques comme la rupture de dialogue. Il assure également la communication avec le modèle de tâche afin de récupérer les demandes de confirmation et de précision ainsi que les réponses à générer vers l'utilisateur.

Le modèle de dialogue est composé de quatre étapes à savoir :

– *Identification des actes de dialogue* : la structure sémantique provenant du système de compréhension représentant un énoncé de l'utilisateur, peut contribuer à plusieurs interprétations pouvant mener à des résultats complètement différentes. A titre d'exemple, l'énoncé « القطار السريع » [AlqTAr AlsryE] (le train express) peut être interprété comme étant une réponse ou une demande de confirmation selon la situation dialogique. Ainsi, le gestionnaire de dialogue doit non seulement prendre en compte les énoncés produits par l'utilisateur, mais il doit aussi prévoir ses intentions. Tout énoncé est alors considéré comme un acte de langage contenant trois composantes à savoir, le locutoire (le dire), l'illocutoire (le faire) et le perlocutoire (l'effet produit sur l'interlocuteur). Dans le dialogue Homme-machine, l'acte de langage est appelé *acte de dialogue*. Sa composante perlocutoire perd son importance étant donné que nous sommes face à une machine. La composante locutoire est traitée par les systèmes de reconnaissance et de compréhension. Quant à la composante illocutoire, elle est identifiée dans cette étape. En effet, l'identification des actes de dialogue repose en grande partie sur le repérage des

marqueurs linguistiques et des indices pragmatiques dans l'énoncé. Étant donné que ces marqueurs et ces indices ne sont pas toujours présents dans un énoncé, l'identification des actes de dialogue se base aussi sur une grammaire formelle construite à partir du corpus d'étude (voir figure 3 de la section 4).

– *Génération et détection de l'acte de dialogue machine* : selon l'acte de dialogue utilisateur identifié, le modèle de dialogue décide le traitement que doit subir la structure sémantique provenant du module de compréhension. Si l'acte de dialogue est un acte qui concerne l'évolution du dialogue alors le modèle de dialogue génère l'acte machine correspondant en utilisant les règles de la grammaire formelle. Cette opération est suivie par une opération de mise-à-jour effectuée au sein du contexte de dialogue chargé de suivre l'avancement de la structure du dialogue par le biais d'un automate fini déterministe. Dans le cas contraire (*i.e.*, l'acte de dialogue identifié concerne la tâche) la structure sémantique est envoyée vers le modèle de tâche pour traitement. Après ce traitement, un message sera communiqué au modèle de dialogue sous forme de réponse, de demande de confirmation ou de demande de précision. Ainsi, le rôle du modèle de dialogue consiste à détecter dans ce message la nature de l'acte de dialogue puis d'effectuer les mises-à-jour nécessaires dans le contexte de dialogue.

– *Génération des réponses* : après détection de l'acte de dialogue machine, le modèle de dialogue doit générer le message textuel correspondant à cet acte. Pour cela, un fichier de correspondance *acte / message* est utilisé pour préciser le message à générer. Ce fichier regroupe un ensemble de paires « *acte de dialogue / message* » et permet de retrouver le message textuel convenant à un acte de dialogue donné.

3.3. Contexte de dialogue

Le contexte de dialogue est la partie du gestionnaire de dialogue responsable d'enregistrer les états du dialogue dans toutes ses différentes phases : avant le chargement de la structure sémantique dans le gestionnaire, au cours et à la fin de son traitement. C'est pourquoi, le contexte du dialogue est formé de trois parties chargées chacune d'intervenir dans une phase précise. Ces trois parties sont :

– *Le réseau de structures sémantiques* : c'est la partie du contexte permettant le stockage temporaire des structures sémantiques provenant du système de compréhension. Ces structures sont stockées en attendant d'être sélectionnées comme structure courante pour subir les traitements. L'opération de sélection est en réalité soumise à l'ordre d'arrivée de ces structures dans le réseau. C'est la raison pour laquelle, dans ce cas, une structure pile est généralement utilisée pour gérer la situation.

– *Le contrôleur des états* : il permet de recevoir les actes de dialogue utilisateur identifiés afin de mettre à jour l'état du dialogue à chaque chargement d'une nouvelle structure sémantique. Il permet aussi de sauvegarder l'acte de dialogue machine généré ou détecté par le modèle de dialogue. Il joue un rôle important dans

l'opération d'identification et de génération des actes de dialogue puisqu'il est doté d'un automate fini déterministe. Cet automate est utilisé pour implémenter la grammaire formelle chargée de modéliser la structure du dialogue.

– *L'historique de dialogue* : le rôle de l'historique de dialogue est de stocker les structures sémantiques complètes après leurs traitements afin de garder trace du déroulement du dialogue en cours. Ces traces sont très utiles dans le cas où le gestionnaire de dialogue reçoit des structures incomplètes dans les prochaines requêtes de l'utilisateur. Ainsi, le gestionnaire de dialogue fait recours à cet historique pour rechercher des instances possibles pour les valeurs manquantes dans les attributs de la structure sémantique.

4. Mise en œuvre de la méthode proposée

Pour la mise en œuvre de la méthode proposée, nous avons choisi le service des renseignements sur le transport ferroviaire comme domaine applicatif. Ainsi, nous avons réalisé un gestionnaire de dialogue capable de dialoguer en arabe standard moderne avec les utilisateurs tout en leur fournissant des informations sur ce domaine. Ce gestionnaire de dialogue est programmé en langage JAVA sous l'environnement JBuilder 2007. Avant de détailler l'architecture du gestionnaire de dialogue, nous présentons le corpus d'étude que nous avons utilisé.

4.1. Corpus d'étude et ressources requises

Les ressources linguistiques arabes sont quasi-absentes voire même indisponibles. C'est pour cette raison que nous étions amenés à construire notre propre corpus d'étude selon la technique du *Magicien d'Oz*. Ce corpus comporte 300 dialogues en arabe (environ 7590 énoncés) et de 11 heures d'enregistrement. Ce corpus nous a permis le recensement des actes de dialogues, des marqueurs linguistiques et l'ensemble des règles de réécriture du dialogue composant notre grammaire formelle. Les actes de dialogue sont illustrés par le tableau suivant :

Actes de dialogue	Symboles
Formalité d'Ouverture.	ForO
Formalité de Fermeture.	ForF
Relance	Rel
Demande d'Informations.	DInf
Demande de Précision.	DPr
Demande d'Explication	DExp
Demande de Confirmation	DConf
Réponse.	R

Tableau 1. Actes de dialogue

Les règles de réécritures (figure 3) représentent les différents enchainements possibles entre les constituants de la grammaire à savoir, les actes de dialogue et les sous-dialogues.

D	=	{SDForO}+ {SDInf}+ {SDForF}*
{SDInf}	=	{SDRel}* {SDDInf}+ {SDPr}* {SDR}
{SDRel}	=	Rel
{SDDInf}	=	DInf {SDConf}*
{SDPr}	=	DPr {SDConf}* {SDExp}* {SDR}
{SDExp}	=	DExp {SDConf}* {SDR}
{SDR}	=	R {SDConf}*
{SDConf}	=	DConf R
{SDForO}	=	ForO
{SDForF}	=	ForF
SDDInf : Sous-Dialogue Demande Information		SDExp : Sous-Dialogue Explication
SDPr : Sous-Dialogue Precision		SDR : Sous-Dialogue Réponse
SDConf : Sou-Dialogue Confirmation		SDInf : Sous-Dialogue Information
SDForO : Sous-Dialogue Formalité d'Ouverture		SDRel : Sous-Dialogue Relance
SDForF : Sous-Dialogue Formalité de Fermeture		D : Dialogue

Figure 3. Règles de réécriture de notre grammaire formelle

Afin de pouvoir utiliser cette grammaire, nous avons besoin de la transformer en un modèle d'implémentation. Puisqu'elle est transformable en grammaire régulière de type 3, nous pouvons retrouver un automate déterministe fini à partir de cette grammaire. La figure suivante représente la grammaire transformée en une grammaire régulière.

D	→	ForO SD1
SD1	→	ForO S D1 Rel SD2 DInf SD3
SD2	→	Rel SD2 DInf SD3
SD3	→	DConf SD4 DPr SD5 R SD6
SD4	→	R SD3
SD5	→	DConf SD7 DExp SD8 R SD3
SD6	→	DConf SD9 Rel SD2 DInf SD3 ForF SD10
SD7	→	R SD5
SD8	→	DConf SD11 R SD5
SD9	→	R SD6
SD10	→	ForF SD10
SD11	→	R SD8

Figure 4. Transformation en grammaire régulière

Puisque la grammaire est une grammaire régulière à droite, nous pouvons la transformer en un automate déterministe fini comme illustré par la figure 5.

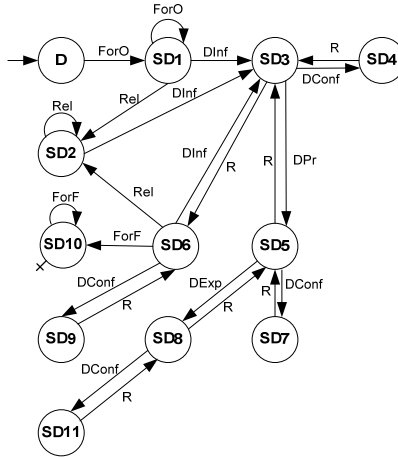


Figure 5. Transformation de la grammaire régulière en un automate fini déterministe

Ainsi, cet automate est capable de contrôler le parcours du dialogue et de donner des prévisions sur son évolution régulière. Cependant, il n'est pas capable de traiter les cas des ruptures et des incidences qui peuvent affecter le dialogue. C'est pourquoi, nous avons défini des règles supplémentaires pour ajouter des transitions vides afin d'éviter l'état du blocage de la machine. La figure suivante présente le nouvel automate après ajout des transitions vides.

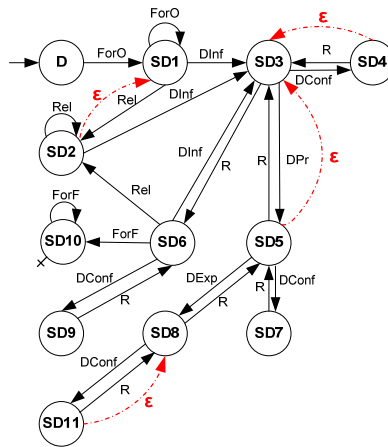


Figure 6. L'automate après ajout des transitions vides

4.2. Architecture détaillée du gestionnaire de dialogue

Le gestionnaire de dialogue est constitué de plusieurs composants et plusieurs ressources (figure 7) pour traiter le schéma sémantique et générer par la suite les réponses vers l'utilisateur.

Les composants du gestionnaire de dialogue sont :

- *Identificateur des actes de dialogue* : permet d'identifier les actes de dialogue de l'utilisateur à partir des schémas sémantiques. Pour cela, il utilise un fichier de correspondance *étiquette / acte* et un automate décrivant les transitions entre les différents états du dialogue.
- *Module complétif des schémas* : permet d'interpréter les valeurs des attributs des schémas, de compléter les attributs vides par des valeurs à partir de l'historique et avec valeurs par défaut. Pour cela, il utilise des règles recensées à partir de notre corpus d'étude et stockées dans des fichiers XML.
- *Vérificateur de la cohérence et de la complétude* : permet de vérifier la cohérence et la complétude des attributs du schéma sémantique. Pour cela, il utilise des règles de cohérence des informations et des règles de complétude des schémas recensées aussi à partir de notre corpus d'étude et stockées dans des fichiers XML.
- *Générateur des commandes* : permet de communiquer avec la base de données de l'application. Pour cela, il génère des commandes SQL afin de rechercher et de récupérer les informations demandées par l'utilisateur.
- *Générateur des actes de dialogue* : permet de générer les actes de dialogue machine. Pour cela, il utilise des règles de transition de l'automate et les résultats obtenus par les autres modules en aval.
- *Générateur des réponses* : permet de générer des réponses à l'utilisateur sous forme de messages textuels selon les actes de dialogue machine déjà générés et les informations retrouvées dans la base de données de l'application.

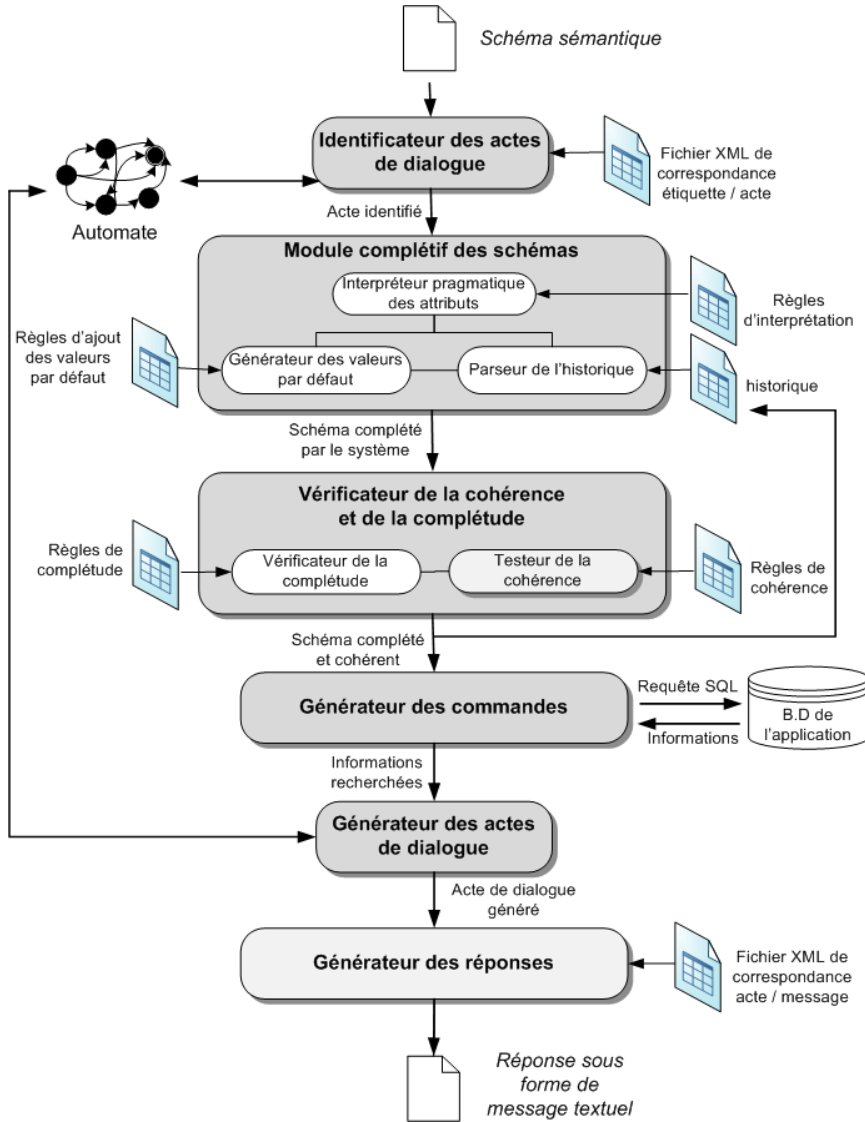


Figure 7. Architecture détaillée du gestionnaire de dialogue

5. Évaluation

D'après Roque et al. (Roque *et al.*, 2006), l'évaluation du gestionnaire de dialogue est une opération difficile à réaliser en raison de l'absence d'une métrique standard capable de mesurer la performance du gestionnaire de dialogue sans faire

Younès Bahou, Amine Bayoudhi, Lamia Hadrich Belguith

intervenir les autres composants du SDOHM. Ainsi, vu l'absence de mesures standardisées pour l'évaluation des gestionnaires de dialogue, nous avons procédé à l'évaluation de mener une évaluation du nombre moyen de tours de parole et du taux de réussite de la tâche. Nous avons également mené une évaluation portant sur d'autres critères. Le premier critère concerne *le taux d'identification des actes* (tableau 2). Ce critère traduit la réussite du gestionnaire de dialogue à identifier l'acte de dialogue de l'utilisateur. Le deuxième critère concerne *le taux de réussite des tâches* (tableau 2) qui permet de mesurer la capacité de la machine à terminer le dialogue avec satisfaction de l'utilisateur.

Taux d'identification des actes de dialogue	Taux de réussite des tâches
98.31%	89.65%

Tableau 2. *Évaluation selon l'identification des actes de dialogue et selon la réussite des tâches*

Nous avons aussi calculé le nombre moyen de tours de parole (tableau 3) pour évaluer l'efficacité des stratégies de dialogue à guider l'utilisateur vers la réalisation de ses buts.

Nb. minimal de tours de parole	Nb. maximal de tours de parole	Nb. moyen de tours de parole
7	19	13

Tableau 3. *Évaluation selon le nombre de tours de parole*

Par ailleurs, nous avons estimé le *temps moyen d'exécution*¹ qui s'avère intéressant surtout avec la diversité des ressources utilisées, les parcours répétitifs pour récupérer les règles à partir des fichiers XML ainsi que l'accès important à la base de données de l'application. Nous avons calculé le temps moyen d'exécution des tâches pour évaluer les temps d'accès à la base de données de l'application. Les résultats obtenus sont illustrés par le tableau suivant :

Tâche	Horaire	Prix	Durée	Type de train	Trajet
Temps moyen d'exécution (ms)	76.8	86.4	82.6	78.5	65.5

Tableau 4. *Évaluation selon le temps moyen d'exécution des tâches*

¹ Nous avons utilisé un ordinateur avec un processeur intel® core™ DuoT2250 (1.73 GHZ).

Enfin, nous avons calculé le temps moyen d'exécution des étapes de la méthode proposée pour évaluer les temps d'accès aux fichiers XML contenant les différentes règles. Les résultats obtenus sont illustrés par le tableau suivant :

Etape de la méthode	Interprétation	Historique	Valeurs par défaut	Cohérence	Complétude
Temps moyen d'exécution (ms)	14.31	4.09	5.5	0.16	3.57

Tableau 5. *Évaluation selon le temps moyen d'exécution des étapes*

6. Conclusion et perspectives

Dans ce papier, nous avons proposé un gestionnaire de dialogue oral permettant d'offrir des informations sur le transport ferroviaire tunisien en arabe standard moderne. Ce gestionnaire est basé sur une méthode qui fait partie de l'approche structurelle. Ainsi, elle repose sur deux modèles à savoir, le modèle de tâche et le modèle de dialogue. Le modèle de tâche permet d'interpréter et de compléter les demandes des utilisateurs, formuler les requêtes à communiquer à la base de données de l'application et générer les réponses à l'utilisateur. Le modèle de dialogue permet d'identifier des actes de dialogue, décider de la nature de l'intervention machine adéquate et assurer le suivi et l'avancement du dialogue en cas d'incidence ou de rupture. Les résultats de l'évaluation obtenus sont très encourageants. Ainsi, les taux de réussite des tâches et d'identification des actes de dialogue sont respectivement de 89.65% et 98.31%, le temps moyen d'exécution des tâches est de 0.0779 secondes et les dialogues sont maintenus avec une moyenne de 13 tours de parole.

Comme perspectives, nous envisageons d'étendre les deux modèles de dialogue pour couvrir certains problèmes non résolus par notre système de compréhension à savoir, le traitement des disfluences complexes et imbriquées, ainsi que le traitement des mots considérés comme hors-vocabulaire par le système de compréhension.

7. Bibliographie

- Allen J. F., Perrault C. R., « Analysing intention in dialogues », *Artificial Intelligence*, vol. 15, n° 3, 1980, p. 23-46.
- Bahou Y., Bayoudhi A., Hadrich Belguith L., « Gestion de dialogue oral Homme-machine en arabe », *Actes de la 16ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'09) – session poster*, Senlis, France, 24 – 26 Juin 2009a.
- Bahou Y., Hadrich Belguith L., Ben Hamadou A., « Disfluency and Out-Of-Vocabulary Word Processing in Arabic Speech Understanding », *Machine Translation Summit XII*,

Younès Bahou, Amine Bayoudhi, Lamia Hadrach Belguith

Third Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3), Ottawa, Ontario, Canada, 26 – 30 Août 2009b.

Bahou Y., Belguith Hadrach L., Ben Hamadou A., « Towards a Human-Machine Spoken Dialogue in Arabic », *LREC'08, Workshop HLT within the Arabic World: Arabic Language and local languages processing Status Updates and Prospects*, Marrakech, Maroc. 2008

Caelen J., Xuereb A., *Interaction et pragmatique: jeux de dialogue et de langage*, Hermès-Lavoisier, Paris, 2007.

Cohen P. R., Perrault C. R., « Elements of a plan based theory of speech acts », *Cognitive science*, vol. 3, 1979, p. 177-212.

Hadrach Belguith L., Bahou Y., Ben Hamadou A., « Une méthode guidée par la sémantique pour la compréhension automatique des énoncés oraux arabes », *International journal of information sciences for decision making (ISDM)*, vol N 35, septembre 2009.

Husltin J., Dialogue model for inquiry and transaction, Thèse de doctorat, Université de Twente, Pays-Bas, 2000.

Luzzati D., Recherches sur le dialogue homme-machine : modèles linguistiques et traitement automatique, Thèse d'Etat, Paris III, France, 1989.

Minker W., Bennacef S., *Speech And Human-Machine Dialog*, Kluwer Academic Publishers Group, Pays-Bas, 2004.

Nguyen H., Dialogue homme-machine : modélisation de multisessions, Thèse de doctorat, Université Joseph Fourier, France, 2005.

Nilson N. J., *Principles of artificial intelligence*, Morgan Kaufmann Publishers Inc, San Francisco, 1980.

Rickel J., Lesh N., Rich C., Sidner C. L., Gertner A., « Using a model of collaborative dialogue to teach procedural tasks », *Workshop on Tutorial Dialog Systems, AIED*, San Antonio, Texas, USA, 2001.

Roque A., Ai H., Traum D., « Evaluation of an Information State-Based Dialogue Manager », *Proceedings of Brandial 2006: The 10th Workshop on the Semantics and Pragmatics of Dialogue*, University of Potsdam, Germany, 2006.

Rosset S., Lamel L., « Gestionnaire de dialogue pour un système d'informations à reconnaissance vocale », *Actes de la 8^{ème} Conférence sur le Traitement Automatique des Langues Naturelles (TALN'01)*, Tours, France, 2001.

Rouillard J., Hyperdialogue sur Internet : Le système HALPIN, Thèse de doctorat, Université Grenoble I, France, 2000.

Roulet E., Auchlin A., Moeschler J., Rubattel C., Schelling M., *L'articulation du discours en français contemporain*, 3^{ème} éd., Peter Lang, Berne, 1991.

Sadek D., « Le dialogue homme-machine : de l'ergonomie des interfaces à l'agent intelligent dialoguant », *OFTA, ARAGO 18 Nouvelles interfaces homme-machine*, 1996, p. 277-321.

Sidner C. L., Lee C. H., Lesh N., « The Role of Dialog in Human Robot Interaction », *International Workshop on Language Understanding and Agents for Real World Interaction*, Sapporo, Japan, 2003.