



HAL
open science

Le dialogue naturel comme un service dans un contexte multi-applicatif

Bertrand Gaiffe, Frédéric Landragin, Matthieu Quignard

► **To cite this version:**

Bertrand Gaiffe, Frédéric Landragin, Matthieu Quignard. Le dialogue naturel comme un service dans un contexte multi-applicatif. 2004, pp.57-66. halshs-00137052

HAL Id: halshs-00137052

<https://shs.hal.science/halshs-00137052>

Submitted on 16 Mar 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Le dialogue naturel comme un service dans un contexte multi-applicatif

Bertrand Gaiffe, Frédéric Landragin, Matthieu Quignard
LORIA — UMR 7503
Campus scientifique — BP 239
54506 Vandœuvre-lès-Nancy CEDEX
{gaiffe, landragi, quignard}@loria.fr

Mots-clefs – Keywords

Dialogue coopératif, langage naturel, terrain commun, référence, actes de langage, agents.
Collaborative dialogue, natural language, common ground, reference, speech acts, agents.

Résumé – Abstract

Rendre un système de dialogue générique vis à vis d'applications amène à considérer ces applications comme des agents rationnels. Dans cet article, nous défendons l'idée que le modèle de dialogue entre agents rationnels diffère du modèle de dialogue naturel aux être humains. De ce fait, nous justifions la nécessité d'un système de dialogue inséré entre humain et applications même intelligentes. Le rôle de ce système de dialogue va au delà de la simple traduction logique des énoncés en langage naturelle. Nous illustrons notre propos avec le système mis en œuvre dans le projet européen OZONE.

The design of generic dialogue systems may lead to consider applications as rational agents. In this article, we claim that a dialogue model for rational agents may not be suitable for interacting 'naturally' with human beings. We thus justify the need for a dialogue system to be inserted between the user and the applications even though those are to be intelligent. We give an illustration of that kind of dialogue with an example of interaction with the OZONE multimodal dialogue system.

1 Introduction

Nous nous intéressons à la conception de systèmes de dialogue homme-machine intelligents, capables de comprendre le sens et d'interpréter dans un contexte communicatif et applicatif des énoncés en langage naturel, capables de tenir avec l'utilisateur un dialogue suivi faisant preuve de collaboration, capables de détecter l'application concernée et de gérer simultanément plusieurs applications. Pour une communication naturelle entre l'utilisateur et le système, tous les aspects de la communication humaine sont à prendre en compte : identification de la signification d'un énoncé, identification de l'implicite (sous-entendus, présupposés), résolution des références, gestion des inférences, des actes de langage, exploitation des gestes effectués conjointement avec les mots, etc. Les systèmes actuels sont dédiés à une tâche particulière (ou à un type de tâche contraint), et ne sont que très partiellement réutilisables (Denecke, Waibel, 1999).

Notre objectif est de définir un modèle d'architecture permettant une séparation claire entre les capacités liées au dialogue en langage naturel, et les spécificités des tâches applicatives. Une application doit pouvoir utiliser le gestionnaire de dialogue comme un service pour communiquer avec l'utilisateur. Ceci pose le problème de l'auto-description de l'application pour être gérée par ce service. Si nous inversons la perspective, cela correspond à abstraire d'un système de dialogue ce qui a trait à l'application.

Dans une perspective de dialogue naturel multi-applicatif le but de l'utilisateur est très peu contraint par le contexte de tâche, et il incombe au gestionnaire de dialogue de l'identifier et de négocier avec l'utilisateur sa réalisation, voire de le co-construire avec lui (le système doit alors être dialoguant avant d'être exécutant). Par ailleurs, nous défendons ici l'idée que l'application a tout intérêt à prendre la forme d'un agent communiquant au sens de la *Foundation for Intelligent Physical Agents* (FIPA). La section 2 s'attache ainsi à spécifier les capacités des applications en les abstrayant du système de dialogue. Le problème qui se pose alors est de faire dialoguer un utilisateur humain avec des agents de ce type. Nous décrivons ainsi dans la section 3 les rôles du gestionnaire de dialogue. Nous en tirons des conclusions d'architecture que nous illustrons dans la section 4 avec une première expérimentation dans le cadre du projet européen OZONE. Nous donnons enfin des perspectives pour continuer dans la voie consistant à rendre le gestionnaire de dialogue plus indépendant des applications.

2 Abstraction des applications du système de dialogue

Une des thématiques du projet européen *Offering an Open and Optimal roadmap towards consumer oriented ambient intelligence* (OZONE) est la conception d'interfaces homme-machine multi-plateformes et multi-applications. Dans le cadre de ce projet, nous avons été amenés à réfléchir sur une architecture de système de dialogue homme-machine où la gestion du dialogue est clairement séparée des applications : il s'agit de définir une architecture multi-agents dans laquelle le traitement automatique du langage naturel, depuis la reconnaissance de la parole (et éventuellement des gestes de désignation produits par l'utilisateur) jusqu'à l'interprétation linguistique contextuelle, constitue un service pour les applications. Pour abstraire les applications du système de dialogue, il faut qu'elles s'auto-décrivent au système de dialogue. Une bonne solution est de considérer une application comme un agent communiquant (par exemple de type FIPA-ACL).

2.1 Des applications capables de dialogue

Comme le montrent (Denecke, Waibel, 1999), les applications de dialogue-homme appartiennent pour la plupart à des catégories réduites : serveur d'information ; réservation d'une chambre d'hôtel ou d'un billet de train ; manipulation d'objets dans une scène ; etc. Schématiquement, nous pouvons distinguer d'un côté les applications de commande, de l'autre côté les applications de renseignement. Idéalement, toute application devrait être mixte.

Dans une application de commande, l'utilisateur est placé dans une logique de « faire-faire » : à l'aide de commandes vocales, il contrôle les objets de l'application, par exemple des formes géométriques affichées à l'écran (Landragin, 2003). Grossièrement, on peut considérer une telle application comme une collection de fonctions à valeurs booléennes. On peut considérer que dans un tel contexte applicatif, le dialogue avec l'utilisateur vise à identifier, éventuellement après des sous-dialogues, quelle fonction de l'application doit être appelée, et avec quels paramètres. La description d'une telle application est essentiellement une description des fonctions élémentaires avec le type de leurs paramètres. Ceci ne suffit cependant pas. En effet, pour résoudre les références aux objets au cours du dialogue, le système doit être en mesure d'interroger l'application via des descriptions logiques de ces objets. Ainsi, résoudre l'expression référentielle « *le triangle vert* » peut amener à interroger l'application sur les x tels que $triangle(x)$ et $vert(x)$ ¹.

Dans les applications de requête d'information, les énoncés élémentaires ou sous-dialogues avec l'utilisateur donnent lieu à des requêtes de type *base de données* dans l'application. En retour, le système de dialogue doit verbaliser les résultats à l'utilisateur. Schématiquement, l'application se compose d'une unique fonction de requête dont le paramètre est une description logique de la requête. Cette fonction renvoie des descriptions logiques au système de dialogue.

Pour abstraire sur les deux types d'applications précédemment envisagées, une bonne solution est d'envisager le problème non pas sous l'angle d'un appel de fonctions, mais sous l'angle de requêtes analogues à des actes de langage. Ainsi, une application de commande peut se voir comme une application apte à réagir à des ordres, mais également à des questions telles que « quels sont les objets présents dans l'état courant ? ». Les applications de requêtes sont, elles, essentiellement de type *question-réponse*.

2.2 Les agents communicants

La communauté multi-agent connaît particulièrement bien ce problème de communication entre applications. Ce problème a donné lieu à une abondante littérature (Chaib-draa, Levesque, 1996; Kumar et al., 2000) et à des normes comme celle de la *Foundation for Intelligent Physical Agents* (FIPA), *Agent Communication Language* (ACL). Il est frappant de constater que la communication entre agents est directement fondée sur la théorie des actes de langage (Searle, 1969). Dans cette théorie, un énoncé se représente schématiquement sous la forme $F(p)$ avec F un marqueur de force illocutoire (assertion, question, ordre, etc.), et p un contenu propositionnel. Exactement sur le même modèle, un message entre agents se décrit par :

- un marqueur illocutoire (assertion, ordre, question) choisi dans une librairie normalisée ;

¹Bien entendu, ce type d'expression peut également être anaphorique, l'interrogation de l'application n'a de sens que pour les emplois référentiels des définis.

- un contenu propositionnel dans un langage logique.

Lorsqu'un agent envoie un message à un autre agent, c'est dans un but particulier ; son acte de communication est directement lié à ce but et ce but met en jeu l'agent avec lequel il communique. Pour que cette communication ait une chance d'aboutir, un modèle est nécessaire.

Les actes de communication proposés par FIPA-ACL sont fondés sur une logique multimodale de croyances, de désirs et d'intentions (Sadek, 1990; Cohen, Levesque, 1995). Ainsi, un agent qui pose une question : ne connaît pas la réponse à cette question ; suppose que son interlocuteur connaît la réponse ; souhaite connaître cette réponse. Ces conditions qui motivent sa question sont également présupposées par son agent interlocuteur.

En résumé, les actes de langage entre agents sont fondés sur une logique de désirs, d'intentions et de croyances, qui permet de planifier la communication entre agents en termes de buts. Un agent est donc dans un monde homogène de buts à résoudre, soit par des actes classiques, soit par des actes de communication. La résolution via des actes de communication suppose un dialogue et permet d'envisager des échecs explicites. Par exemple, un agent peut refuser d'effectuer un ordre (ou de répondre à une question) tout simplement parce qu'il n'a pas les moyens d'effectuer l'ordre (ou qu'il ne connaît pas la réponse à la question).

2.3 Description d'une application

Dans les paragraphes précédents, nous n'avons évoqué que la partie illocutoire des messages entre agents. Bien entendu, le contenu propositionnel des messages est tout aussi important. Pour qu'un contenu propositionnel puisse être transmis entre des agents, il faut que ces agents s'accordent sur une syntaxe de construction des messages (par exemple la syntaxe du calcul des prédicats du premier ordre) et un vocabulaire terminal (les symboles non logiques).

Le second point relève directement d'une ontologie. Dans le cas d'une application de commande, une telle ontologie doit au moins contenir :

- les noms des actions et les types de leurs paramètres ;
- les types des objets (et au moins leurs propriétés modifiables par des actions) ;
- les effets des actions.

Etant donné que les types d'actes illocutoires sont définis dans une logique de croyances, désirs et intention, le raisonnement d'un agent, à l'émission ou à la réception d'un message, impose en particulier de manipuler des croyances de son agent interlocuteur sur la proposition émise. Un tel raisonnement n'est possible que si les agents s'accordent sur une telle ontologie, faute de quoi : soit la réponse sera systématiquement une incompréhension ; soit il faut envisager un dialogue de négociation de sens entre les agents (ce que nous n'envisagerons pas ici).

Conclusion partielle

Une bonne façon d'abstraire l'application d'un système de dialogue est donc de considérer cette application comme un agent capable de dialoguer. Un tel agent doit être prédictible quant à son comportement, et cette prédictibilité est bien prise en compte par une logique de désirs, d'intentions et de croyances. Le modèle du dialogue entre agents sous-jacent repose directement sur cette logique. Nous avons posé le problème du paramétrage d'un système de dialogue par son application via la définition de cette application comme un agent capable de communication.

Le dialogue naturel comme un service dans un contexte multi-applicatif

La question se pose alors du rôle dévolu à un gestionnaire de dialogue.

3 Problèmes liés au dialogue homme-machine spontané

Nous avons vu dans la section précédente que le modèle de dialogue correspondant à la communication entre agents intelligents relevait d'une logique d'états cognitifs. La question se pose alors de faire dialoguer un être humain avec un agent de ce type. Ceci pose immédiatement la question du rôle d'un gestionnaire de dialogue.

3.1 Structuration du dialogue comme moyen de résoudre la référence

Basiquement, un système de dialogue doit, à partir des énoncés d'un utilisateur, déduire les intentions de cet utilisateur. En supposant un système pleinement coopératif, il adopte alors ces intentions pour siennes, et planifie une résolution de ce but à l'aide d'actes de langage envers l'application. L'identification des buts de l'utilisateur pose le problème de la structuration du dialogue, ne serait-ce que pour résoudre la référence.

Le problème classique posé par une approche fondée sur la théorie des actes de langage est celui de la structuration du dialogue. Très directement, étant donné un dialogue de la forme : $F_1(p_1) ; F_2(p_2) ; \dots ; F_n(p_n)$, comment rendre compte en particulier de reprises anaphoriques de l'un des énoncés sur un autre.

Structuration du dialogue par les intentions : une première solution envisageable à ce problème relève de théories du même type que celle qui préside à la modélisation du dialogue entre agents intelligents. Dans ce type de théories, la structure du dialogue est supposée refléter les intentions des interlocuteurs (Grosz, Sidner, 1986). L'accessibilité référentielle dans ces théories est directement liée à la relation de but à sous-but. Dans la mesure où la structuration du dialogue est nécessaire au calcul des intentions, la seule solution est alors de préjuger de l'intention d'un énoncé pour structurer le dialogue, et de vérifier *a posteriori* que ce préjugé est valide. Des versions plus récentes de ces théories (Litman, Allen, 1990; Cohen, Levesque, 1995) s'appuient entièrement sur des logiques de type croyance, désir, intention, pour structurer le dialogue. Deux problèmes se posent alors. D'une part ces logiques modales, si elles doivent de plus intégrer le calcul référentiel, doivent être au moins des logiques modales du premier ordre et sont donc complexes. D'autre part, elles posent le problème de la gestion du terrain commun.

Dialogue et terrain commun : comme le remarquent (Asher, Lascarides, 2003), la gestion du terrain commun dans une théorie du dialogue fondée sur une logique intégrant la croyance est un problème extrêmement difficile. Si on considère qu'un système de dialogue doit aider l'utilisateur dans la réalisation d'une tâche, on pose nécessairement le problème d'une divergence de croyance entre le système et cet utilisateur. Typiquement, l'utilisateur croit quelque chose de faux, et le système doit l'aider en expliquant que cette croyance est fautive. Ceci peut correspondre à des exemples aussi simples que : « *déplace le triangle vert* » — « *il n'y a pas de triangle vert* » ; ou : « *efface ce fichier* » — « *vous n'avez pas le droit d'effacer un fichier qui ne vous appartient pas* ». Ce type d'exemples illustre la nécessité de différencier les croyances des interlocuteurs. Si on s'appuie sur une telle logique pour structurer le dialogue, on aboutit à la conclusion que dans le second exemple, le système ne peut pas dire « *vous n'avez pas le droit*

de l'effacer. Il ne vous appartient pas ». Si au contraire on se donne des axiomes permettant la croyance mutuelle, la gestion des divergences de croyances devient délicate. Il nous semble que l'analyse de (Asher, Lascarides, 2003) est parfaitement justifiée et conduit à considérer que la structuration du dialogue, c'est-à-dire du terrain commun, doit essentiellement se faire sur une base linguistique.

Les théories linguistiques du dialogue : une longue tradition de structuration du dialogue sur des bases plus linguistiques existe (Roulet et al., 1985). Si ces théories ne déniaient pas le rôle des intentions pour structurer le dialogue, elles s'accordent cependant pour ne pas fonder totalement cette structuration sur les intentions. Nous avons déjà indirectement mentionné plus haut la *Segmented Discourse Representation Theory* ou SDRT (Asher, Lascarides, 2003) comme une théorie du dialogue qui ne se fonde pas directement sur une modélisation des croyances et intentions. La SDRT vise à modéliser le dialogue sur la base de ce qui est dit. Elle fait usage pour cela d'actes de langage relationnels (question-réponse, contestation, etc.). Pour des raisons d'efficacité, ces relations de discours (et de dialogue) sont inférées dans une logique qui n'a que faiblement accès aux intentions et croyances d'une part (techniquement, elle n'a qu'une vision propositionnelle de ces logiques : la quantification disparaît), et à l'état de l'application d'autre part. Ce n'est qu'une fois la structure du dialogue calculée que les intentions des interlocuteurs sont analysées (elles sont réévaluées après chaque énoncé, elles peuvent donc varier sans que la structure du dialogue soit remise en cause).

Dans le système OZONE que nous décrivons plus loin, cette idée de structuration linguistique du dialogue est mise en œuvre de façon plus légère que dans la SDRT, et s'appuie sur ce que nous appelons des *domaines de référence*.

3.2 Gestion du dialogue avec l'utilisateur

Parmi les fonctions principales du gestionnaire de dialogue, une autre fonction consiste à assurer la collaboration entre l'utilisateur et les applications. Le gestionnaire de dialogue doit décider du « quoi dire », du « quoi faire », en fonction de la représentation de l'état du dialogue ou de l'état des applications. En ce qui concerne l'état du dialogue, son rôle est d'inférer à partir des énoncés les intentions sous-jacentes aux actes de langage. En ce qui concerne l'état des applications, son rôle est de traduire des intentions. Le point principal est l'ancrage dans la structuration linguistique : toute réponse à l'utilisateur exploite le contenu de cette structuration, de manière à conserver une certaine cohérence linguistique. C'est particulièrement le cas lors de la mention d'une entité déjà mentionnée par l'un ou l'autre des interlocuteurs : cette mention préalable peut amener le gestionnaire de dialogue à produire une anaphore. C'est également le cas si l'utilisateur a produit un acte de langage indirect : bien qu'il ait identifié le véritable acte de langage, le gestionnaire du dialogue doit générer une réponse cohérente par rapport à l'acte de surface.

3.3 Architecture résultante

De ce qui précède on peut distinguer clairement trois rôles principaux du gestionnaire de dialogue : la gestion d'une structure linguistique construite à partir des énoncés de l'utilisateur et permettant la génération d'énoncés cohérents ; la gestion et la planification des buts en collaboration avec l'utilisateur ; la gestion des applications. Ces trois rôles sont à l'origine d'autant

de modules, respectivement : module linguistique ; module collaboratif ; module applicatif. Plus précisément, le module linguistique a pour fonction d'une part d'interpréter les énoncés de l'utilisateur pour reconstruire à chaque étape l'état du dialogue, d'autre part de générer des énoncés linguistiquement cohérents avec cet état du dialogue. Il gère le terrain commun. Il ne fait aucune inférence sur les croyances de quiconque. Comme nous l'avons vu précédemment, le module collaboratif prend en charge les buts et les demandes de l'utilisateur, et le module applicatif gère la réalisation de la tâche par les applications. Ces modules font l'objet d'agents communicants dans l'architecture globale. La communication entre ces trois agents se fait avec des actes de langage comparables à FIPA-ACL.

4 Illustration avec le démonstrateur OZONE

4.1 Architecture globale

La figure 1 montre le découpage en agents choisi pour le démonstrateur du projet OZONE. L'architecture est issue de celle du projet MIAMM, qui a fait l'objet de recherches sur des protocoles de communication basés sur SOAP et destinés à faire transiter des contenus sémantiques. Compte tenu de la multimodalité (entrée vocale associée à des gestes, avec des dispositifs à retour d'effort dans MIAMM, et sur l'écran tactile d'un TabletPC dans OZONE), ces contenus sémantiques sont multimodaux et ont nécessité la spécification d'un langage de représentation : *MultiModal Interface Language* ou MMIL (Kumar, Romary, 2002). Les trois principaux types d'actes de langage considérés sont l'assertion, la question et l'ordre. MMIL repose sur les deux notions d'événement et de participant, et les actes de langage sont assimilés à des événements de type particulier. Dans l'architecture OZONE, la communication entre agents se fait en MMIL.

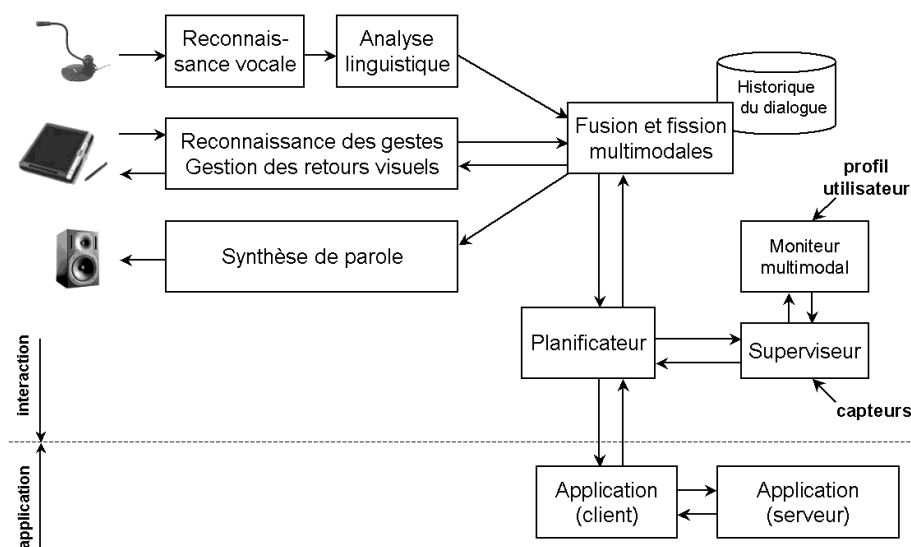


Figure 1: L'architecture du démonstrateur OZONE.

Le gestionnaire du dialogue comprend ici deux agents communicants. Celui nommé « fusion et fission multimodales », intégrant l'historique du dialogue, est chargé de l'interprétation contextuelle des énoncés de l'utilisateur, de leur gestion dans une structuration linguistique, et de la génération multimodale, c'est-à-dire de la fission de l'information à générer dans les canaux

acoustique et visuel. Il s'agit donc du module linguistique décrit plus haut. L'autre, nommé « planificateur », se charge des aspects tactiques, à la fois avec l'application (qui est ici unique) et l'utilisateur. Les capacités de planification sont en effet factorisées dans l'implantation OZONE. L'agent résultant correspond donc à la fusion des modules collaboratif et applicatif.

D'autre part, l'application est ici divisée en deux agents, car l'une des préoccupations du projet OZONE était de gérer les applications à travers un réseau (en tant que *WebServices*).

4.2 Structuration linguistique à l'aide de domaines de référence

Plutôt que des SDRS, nous avons choisi dans les projets MIAMM et OZONE de gérer des domaines de référence. Les travaux que nous menons dans le domaine de l'interprétation de la référence aux objets dans le dialogue homme-machine montrent que tout acte de référence passe par l'activation d'un domaine de référence (Landragin, 2003). Cette notion modélise la focalisation dans un espace attentionnel. Elle permet de justifier l'utilisation de descriptions discriminantes dans des ensembles d'objets plus réduits que le contexte complet. Un domaine de référence est un sous-ensemble contextuel, de nature linguistique, visuelle, gestuelle ou encore liée au déroulement de l'interaction ou à la tâche applicative. De manière simplifiée, il s'agit d'un groupe d'objets dans lequel les composants de l'expression référentielle permettent d'une part d'extraire un référent, d'autre part de préparer un support pour l'interprétation d'une future référence. Ainsi, « *le triangle rouge* » contient deux propriétés dont la combinaison doit être discriminante dans un domaine de référence qui doit, pour justifier cette expression et dans la mesure du possible, comprendre un ou plusieurs triangles non rouges. Après l'interprétation de cette expression, le domaine de référence est structuré avec une première distinction entre les triangles et les autres formes, et une deuxième distinction entre le triangle rouge et les autres triangles. Une expression ultérieure telle que « *les autres triangles* » sera interprétée dans le même domaine, dénotant ainsi une continuité dans la chaîne de référence et, d'une manière générale, dans le dialogue. Cette notion de domaine consiste à regrouper dans un cadre unifié les contraintes provenant de toutes les facettes du contexte. C'est dans ce point que réside son avantage par rapport à la SDRT : conçu pour le dialogue multimodal, le modèle des domaines de référence facilite la confrontation et l'intégration d'informations hétérogènes (linguistiques, visuelles, gestuelles) dans des structures homogènes. Cette intégration permet de prendre en compte une grande variété d'aspects contextuels lors de la résolution des références et de l'interprétation des énoncés.

4.3 Exemple

Dans l'extrait de scénario suivant, où S représente le système et U l'utilisateur, les rôles du gestionnaire de dialogue sont visibles dès les premiers énoncés. L'utilisateur veut acheter un billet de train pour aller à Paris, et utilise pour cela le système OZONE dont l'une des applications est un serveur d'informations sur les transports ferroviaires. Son premier énoncé, « *je veux aller à Paris* » (U-2) est une assertion qui sera conservée comme telle dans le terrain commun. Du contenu de cet énoncé l'agent linguistique déduit le but de l'utilisateur, aller à Paris. L'agent applicatif envoie à l'application l'ordre d'emmener l'utilisateur à Paris. Compte tenu de ses capacités, le but de l'application est alors de connaître le point de départ. Il lui faut en effet deux informations, le point de départ et le point d'arrivée, pour qu'elle puisse afficher une carte géographique avec les trajets possibles reliant ces deux points (les horaires et temps de transport

seront débattus plus tard). L'agent collaboratif transmet ce but et l'agent linguistique cherche la relation de discours appropriée. Il génère donc une question portant sur le point de départ, sans répéter le point d'arrivée ni le but initial consistant à s'y rendre. Le point d'arrivée sera exploité lors du calcul des trajets possibles en S-5. Le but initial consistant à se rendre en ce point sera exploité plus tard, en S-13 où le système incite l'utilisateur à se décider entre les deux solutions pour lesquelles il a demandé des précisions.

- S-1: *Bonjour, que puis-je faire pour vous ?*
U-2: Je veux aller à Paris.
S-3: *D'où voulez-vous partir ?*
U-4: de Meudon.
S-5: *Voici les trajets possibles (+ affichage de trajets).*
U-6: Combien de temps ça prend avec ce trajet (+ geste) ?
S-7: *Ce trajet prend deux heures et vingt minutes.*
U-8: Combien y a-t-il de changements ?
S-9: *Il y a un changement avec ce trajet.*
U-10: Combien de temps ça prend avec ce trajet (+ geste) ?
S-11: *Ce trajet prend vingt-deux minutes.*
U-12: Bien.
S-13: *Quel trajet préférez-vous ?*
U-14: Celui-ci.
S-15: *Quand voulez-vous partir ?*
U-16: Dès que possible.
S-17: *Il y a un train dans dix-sept minutes.*
U-18: ...

5 Conclusion et perspectives

Nous avons donc montré comment l'utilisateur pouvait dialoguer avec une application de type FIPA-ACL à travers un gestionnaire de dialogue composé de trois agents : un agent linguistique, un agent collaboratif et un agent applicatif. Dans le cadre du projet européen OZONE, l'état actuel de nos activités est le suivant : nous avons posé le problème de la séparation du gestionnaire de dialogue et d'une application, et nous avons distingué deux agents communicants au sein du gestionnaire de dialogue. L'implantation qui en a été faite sur la base d'une architecture de type SOAP-MMIL nous a permis de réaliser un démonstrateur autorisant un dialogue multimodal spontané. Bien entendu, ce démonstrateur présente des défauts. Tout d'abord, sa généralité est incomplète, en particulier sur le paramétrage du gestionnaire de dialogue par les applications. En effet, dans la réalisation actuelle, chaque application fournit au gestionnaire de dialogue :

- une grammaire complète pour le dialogue entre l'utilisateur et le gestionnaire de dialogue ;
- une ontologie complète pour le dialogue entre le gestionnaire de dialogue et l'application.

La grammaire recouvre en fait un lexique et une grammaire de type *Lexicalized Tree Adjoining Grammar* (LTAG), incluant des traits sémantiques pour ancrer l'analyse sémantique sur l'analyse syntaxique. Elle est fournie en totalité par l'application, et le système n'est utilisable que pour des phrases restant dans la couverture définie au préalable.

Pour résoudre ce problème et améliorer la généralité, plusieurs pistes sont envisageables. Il

s'agit de scinder la grammaire et l'ontologie, avec, dans les deux cas, une partie gérée par le gestionnaire du dialogue et indépendante des applications, et une partie apportée par l'application et spécifique à elle. Une grammaire correspondrait ainsi au langage usuel, et chaque application apporterait sa grammaire spécialisée. De même, une ontologie comprendrait des concepts et des états de haut niveau indépendants des applications, et chaque application apporterait son ontologie spécialisée.

Références

- Asher N., Lascarides A. (2003), *Logics of Conversation*, Cambridge University Press.
- Chaib-draa B., Levesque P. (1996), Hierarchical Model and Communication By Signs, Signals and Symbols in Multiagent Environments, *Journal of Experimental and Theoretic Artificial Intelligence*, Vol. 8, 7-20.
- Clark H.H., Schaefer E.F. (1989), Contributing to Discourse, *Cognitive Science*, Vol. 13, 259-294.
- Cohen P.R., Levesque, H.J. (1995), Communicative Actions for Artificial Agents, In *Proceedings of the International Conference on multi-agents Systems*, AAAI Press.
- Denecke M., Waibel A. (1999), Integrating Knowledge Sources for the Specification of a Task-Oriented System, In *Workshop Proceedings "Knowledge And Reasoning in Practical Dialogue Systems" of IJ-CAI'99*.
- FIPA, Foundation for Intelligent Physical Agents, <http://www.fipa.org>.
- Grosz B.J., Sidner C.L. (1986), Attention, Intentions and the Structure of Discourse, *Computational Linguistics*, Vol. 12(3), 175-204.
- Kumar S., Huber M.J., McGee D.R., Cohen P.R., Levesque H.J. (2000), Semantics of Agent Communication Languages for Group Interaction, In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI'00)*, Austin, 42-47.
- Kumar A., Romary L. (2002), A Comprehensive Framework for Multimodal Meaning Representation, In *Proceedings of the International Workshop on computational Semantics (IWCS-5)*, Tilburg, Netherlands.
- Landragin F. (2003), Modélisation de la communication multimodale. Vers une formalisation de la pertinence, Thèse de l'Université Henri Poincaré de Nancy.
- Litman D.J., Allen J.F. (1990), Discourse Processing and Commonsense Plans, In Cohen P.R., Morgan J., Pollack M.E. (Eds.), *Intentions in Communication*, MIT Press, 365-388.
- MIAMM, Multidimensional Information Access using Multiple Modalities, IST-2000-29487 European Project, <http://www.miamm.org>, 2001-2004.
- OZONE (O₃), Offering an Open and Optimal roadmap towards consumer oriented ambient intelligence, <http://www.extra.research.philips.com/euprojects/ozone>, IST-2000-30026 European Project, 2001-2004.
- Quignard M. (2000), Modélisation cognitive de l'argumentation dialoguée, Thèse de l'Université Joseph Fourier de Grenoble.
- Roulet E., Auchlin A., Moeschler J., Rubattel C., Schelling M. (1985), *L'articulation du discours en français contemporain*, Bern, Lang.
- Sadek, D. (1990), Logical Task Modelling for Man-Machine Dialogue, In *Proceedings of AAAI 1990*, 970-975.
- Searle J.R. (1969), *Speech acts: An Essay in the Philosophy of Language*, Cambridge University Press.