Meta
Journal des traducteurs
Translators' Journal

META

Le dialogue homme-machine : un système de traduction automatique spécifique

Pierre-André Buvet et Laurent Tromeur

Volume 55, numéro 1, mars 2010

Le parcours du sens : d'une langue à l'autre — Mélanges offerts à André Clas
The Way of Meaning: From a Language to Another — Collection of Articles Offered to André Clas

URI : https://id.erudit.org/iderudit/039602ar
DOI : https://doi.org/10.7202/039602ar

Aller au sommaire du numéro

Éditeur(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (imprimé)
1492-1421 (numérique)

Découvrir la revue

Citer cet article

Buvet, P.-A. & Tromeur, L. (2010). Le dialogue homme-machine : un système de traduction automatique spécifique. *Meta*, *55*(1), 58–70. https://doi.org/10.7202/039602ar

Résumé de l'article

Nous décrivons un système de dialogue homme-machine conçu comme un système de traduction bilingue qui a la particularité de mettre en relation une langue naturelle et un langage informatique. Le système de dialogue homme-machine doit servir d'interface entre un utilisateur et une plateforme Web qui développe automatiquement des applications informatiques. Dans un premier temps, nous rappelons les particularités d'un système de dialogue homme-machine et nous présentons le système d'interaction en langue naturelle que nous utilisons. Dans un deuxième temps, nous détaillons le mode de fonctionnement du système de transfert. Dans un troisième temps, nous précisons le traitement linguistique qui permet au système d'interpréter des instructions rédigées par les usagers et de les traduire en langage-machine.

Le dialogue homme-machine : un système de traduction automatique spécifique

PIERRE-ANDRÉ BUVET[*]
LDI (UMR 7187, CNRS) – Université Paris 13, Paris, France
pabuvet@ldi.univ-paris13.fr

LAURENT TROMEUR
LDI (UMR 7187, CNRS) – Université Paris 13, Paris, France
Société Ontomantics, Orléans, France
ltromeur@ldi.univ-paris13.fr

RÉSUMÉ

Nous décrivons un système de dialogue homme-machine conçu comme un système de traduction bilingue qui a la particularité de mettre en relation une langue naturelle et un langage informatique. Le système de dialogue homme-machine doit servir d'interface entre un utilisateur et une plateforme Web qui développe automatiquement des applications informatiques. Dans un premier temps, nous rappelons les particularités d'un système de dialogue homme-machine et nous présentons le système d'interaction en langue naturelle que nous utilisons. Dans un deuxième temps, nous détaillons le mode de fonctionnement du système de transfert. Dans un troisième temps, nous précisons le traitement linguistique qui permet au système d'interpréter des instructions rédigées par les usagers et de les traduire en langage-machine.

ABSTRACT

We describe a human-machine dialog system designed as a transfer system which establishes a relationship between a natural language and an artificial language. The human-machine dialog system is intended to be used as an interface between a user and a Web platform which automatically develops applications. We first discuss the human-machine dialog system. We then describe how the transfer system works. We finally depict the natural language processing which enables the system to interpret and translate the user instructions.

MOTS-CLÉS/KEYWORDS

dialogue homme-machine, traduction automatique, représentation métalinguistique
human-machine dialog, machine translation, metalinguistic representation

Nous décrivons comment fonctionne un système de transfert entre une langue naturelle et un langage machine dans le cadre du développement automatique d'applications. Nous rappelons les particularités d'un système de dialogue homme-machine d'une façon générale puis nous présentons le système d'interaction en langue naturelle que nous utilisons. Nous précisons ensuite le mode de fonctionnement de ce système et la nature de ses différentes composantes. Nous expliquons en dernier lieu quel traitement linguistique permet le transfert entre des instructions rédigées en langue naturelle et des requêtes écrites en langage informatique.

1. Le projet Ontomantics-LDI

Nous décrivons rapidement les principales particularités des différents types de dialogue homme-machine ; nous précisons dans quel cadre s'inscrit notre travail et présentons les objectifs généraux de notre projet.

1.1. État de la question

De nombreux travaux de recherche en Traitement automatique des Langues (TAL) et en Intelligence artificielle (IA) portent sur le dialogue homme-machine (Sabah 1989 ; Pierrel et Sabah 1991). De plus, il s'agit d'un secteur d'activité central dans les Industries de la Langue (IL – Pierrel 2000). Les formulaires, les menus interactifs, les invites de commandes, etc., permettent à l'utilisateur de donner des instructions à un ordinateur. De tels modes de communication sont parfois insuffisants pour exprimer les besoins précis de l'utilisateur du fait de leur relative rigidité. Le dialogue homme-machine se conçoit comme un moyen de remédier à ces insuffisances. Il s'agit d'un système permettant une interaction entre un humain et un système dans un cadre restreint (Glass, Polifroni *et al.* 2000). Idéalement, c'est une technologie qui donne à l'utilisateur la possibilité de saisir des instructions en langue naturelle afin d'exprimer plus finement ses besoins. Sa mise en œuvre est rendue difficile par le caractère complexe de la langue.

Les systèmes de communication en langue naturelle de la première génération sont essentiellement dédiés à l'interrogation de bases de données ou à des tâches très spécialisées. Ils permettent de formuler des instructions en langage contrôlé. Les applications qui les intègrent sont la commande automatique de titre de transport, la réservation automatique de chambre d'hôtel, etc. Le plus remarquable est le système *HALPIN*[1]. Il est dédié à la recherche documentaire ; un graphe de tâches coordonne les différents sous-dialogues entre l'homme et la machine (Rouillard 2000).

Les systèmes d'interaction en langue naturelle de la deuxième génération sont plus élaborés. Notamment, dans le cadre des travaux sur les systèmes de question-réponse, le dialogue est une suite d'échanges entre interlocuteurs dans un contexte donné. Les systèmes de dialogue homme-machine interprètent les requêtes de l'utilisateur en fonction de la tâche à accomplir, de l'histoire du dialogue et du comportement de l'utilisateur. L'objectif est de donner à l'utilisateur les informations recherchées tout en assurant une interaction efficace et naturelle (Galibert, Illouz *et al.* 2005). *G-TAG*[2] et *COALA*[3] sont représentatifs des systèmes de la deuxième génération. Le premier prend appui sur le formalisme des grammaires d'arbres adjoints (TAG) ; il construit une représentation conceptuelle des informations à transmettre, annotée d'informations pragmatiques. Le second est fondé sur un modèle calculatoire et dynamique du dialogue homme-machine quel que soit le champ d'application. Une phase d'apprentissage lui permet d'enregistrer les expressions et les termes effectivement employés dans une situation de dialogue.

Le mode de structuration du dialogue est une question centrale pour le traitement automatique du dialogue. Idéalement, la manière de formuler des instructions ne doit faire l'objet d'aucune contrainte ; autrement dit, il faut que la machine analyse les instructions quelles que soient les formulations. Le système doit être capable de traiter des phénomènes comme les fautes de frappe, la dysorthographie, l'anaphore ou l'implicite. Ces phénomènes ne sont pas spécifiques à la langue générale, on les

observe également dans les langues spécialisées. Par conséquent, ils concernent également les systèmes d'interaction en langue naturelle portant sur des domaines spécifiques (l'alimentation, l'économie, la médecine, etc.).

Les systèmes de communication en langue naturelle doivent également être capables de formuler correctement les réponses de la machine. Un système de dialogue homme-machine fonctionne en deux temps. Dans un premier temps, l'instruction formulée par l'utilisateur est analysée afin d'être associée à une représentation. Dans un deuxième temps, le système cherche une réponse adéquate qui est ensuite exprimée en langue naturelle. Certains systèmes génèrent des textes syntaxiquement corrects à partir d'un corpus de phrases préenregistrées comportant des variables (Weizenbaum 1966). D'autres exploitent un répertoire de phrases élémentaires en rapport avec les concepts sous-jacents à l'application (Manaris et Dominick 1993).

Les performances des systèmes de dialogue homme-machine restent encore en deçà de ce qui serait nécessaire pour qu'ils soient intégrés à grande échelle dans des applications qui manipulent de l'information. Ces systèmes sont essentiellement confrontés à des difficultés de nature linguistique.

1.2. Contexte applicatif : la plateforme Ontomantics

Le système *Ontomantics* est une plateforme Web qui génère automatiquement des applications[4]. Il a été développé par l'entreprise éponyme. L'utilisation de l'interface graphique de développement ne présuppose pas la connaissance d'un langage de programmation. L'interface doit permettre à n'importe quel utilisateur de créer rapidement des applications. L'application est développée puis exécutée à partir d'un navigateur Internet ; les tâches étant effectuées depuis un serveur, elles ne nécessitent aucune installation sur le poste de l'utilisateur client. La technologie *Ontomantics* permet des gains de temps et d'argent, car elle ne fait pas appel à des ressources externes pour modéliser le besoin applicatif de l'utilisateur (achat de logiciel tiers, location des services d'un expert, etc.).

Le système est composé de trois modules : le premier crée les bases de données qu'utilisera l'application ; le deuxième crée les écrans nécessaires à la navigation à l'aide d'un système intuitif de Drag and Drop (*Glisser/Déposer*) ; le troisième crée les règles qui déterminent le comportement de l'application.

L'entreprise *Ontomantics* a comme objectif de rendre accessible le développement d'applications à des utilisateurs qui ne sont pas des informaticiens. La version actuelle de la plateforme ne permet pas d'atteindre cet objectif, car des connaissances en informatique restent nécessaires au développement d'une application. La réalisation d'un module supplémentaire, une interface homme-machine, doit permettre de remédier à cet inconvénient.

1.3. Problématique

La société *Ontomantics* et le laboratoire *Lexiques Dictionnaires informatique (LDI)* collaborent pour développer une interface de dialogue en langue naturelle qui permettra aux utilisateurs de la plateforme de formuler leur besoin par une saisie au clavier. La formulation des instructions ne doit nécessiter aucune connaissance informatique préalable.

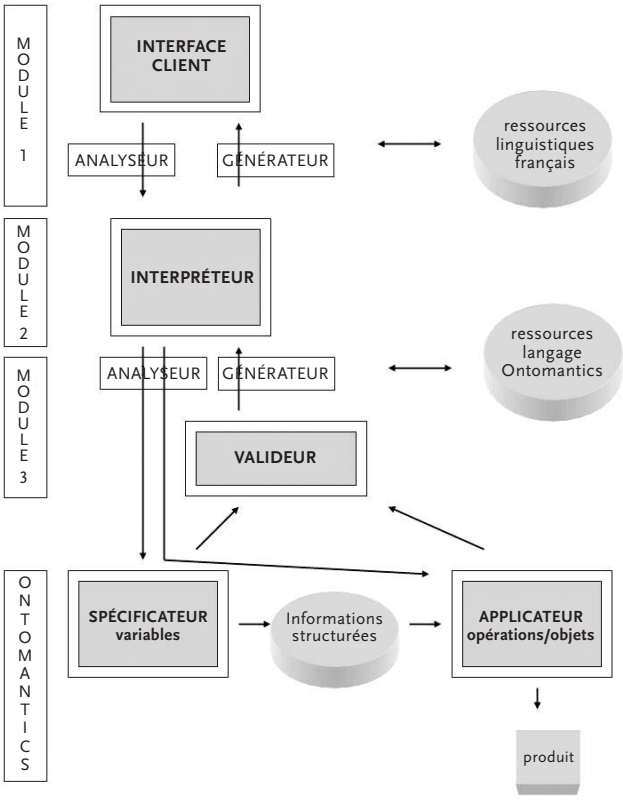
L'objectif de la collaboration est double : il s'agit, d'une part, de développer un système de transfert entre le français et le langage *Ontomantics* et, d'autre part, de mettre en place un système d'acquisition automatique de vocabulaire métier. Le dialogue homme-machine s'effectue au moyen du système de transfert qui fonctionne comme un système de traduction mais qui a comme spécificité de mettre en regard une langue naturelle et un langage machine. Il s'agit d'un système du type interprétatif-transformationnel (Buvet 2009a). Le système d'acquisition a pour finalité de fournir au système des ressources linguistiques en rapport avec le secteur d'activité de l'application développée par la plateforme *Ontomantics*. Seul le premier aspect de l'objectif est traité dans ce qui suit.

2. Mise en place du prototype

Après avoir présenté l'architecture du système d'interaction en langue naturelle (Figure 1), nous décrivons la chaîne de traitement et les spécificités de chaque module du système, et nous précisons la nature des ressources et des outils implémentés dans le système.

FIGURE 1

Architecture générale du système de transfert



2.1. Description des modules du système de transfert

Le module 1 comporte deux sous-modules : un module d'analyse du français et un module de génération du français (Blanco et Buvet 2000). Le premier sous-module permet d'associer des représentations métalinguistiques aux phrases simples, le second de produire des phrases simples à partir de représentations métalinguistiques.

Le module 3 comporte également un sous-module d'analyse et un autre de génération mais c'est le langage *Ontomantics* qui est traité. Les représentations du langage machine correspondent à des commandes en langage *SQL* (Van Der Lans 2006) lorsque les instructions se rapportent à la gestion des bases de données du système *Ontomantics*.

Le module 2, dit *interpréteur* dans la présentation schématique de l'architecture générale, permet de transférer une représentation métalinguistique d'une phrase française en une commande SQL et, réciproquement, de transférer une commande SQL en représentation d'une instruction en langage *Ontomantics*.

Le cheminement de l'information obéit à deux logiques différentes selon qu'il est orienté de l'homme vers la machine ou de la machine vers l'homme. Dans le premier cas de figure, il s'agit d'envoyer des instructions au système pour qu'il effectue une application. Dans le second cas de figure, le retour vers l'utilisateur permet de vérifier que les instructions en langage *Ontomantics* qui vont être spécifiées sont conformes aux attentes de l'utilisateur. Dans l'architecture générale, c'est la composante dite *valideur* qui est le point de départ du retour vers l'utilisateur. Celui-ci doit également permettre, le cas échéant, de demander un complément d'information.

2.2. Outils linguistiques et technologies informatiques

Les procédures d'analyse et de génération du français reposent, d'une part, sur l'exploitation de dictionnaires morphosyntaxiques (Mathieu-Colas 2009) et de dictionnaires syntactico-sémantiques (Buvet 2009a ; Gross 1992 ; Mathieu-Colas 1994) et, d'autre part, sur l'utilisation de grammaires locales (Gross 1995 ; Maurel 1993).

Les dictionnaires syntactico-sémantiques mentionnés sont conçus pour le TAL. Autrement dit, ils ont une microstructure normalisée qui permet l'exploitation informatique des descripteurs associés à chaque vedette. Il y a trois types de dictionnaires selon que la macrostructure concerne des prédicats, des arguments élémentaires ou des actualisateurs.

Une grammaire locale décrit le contexte d'une unité lexicale donnée en tant qu'ensemble de configurations de mots[5]. Elle est représentée par un graphe comportant : un nœud initial, un nœud final et un ensemble de nœuds intermédiaires, et des arcs qui relient les nœuds en fonction des configurations de mots de la grammaire locale. Les nœuds intermédiaires sont associés à des mots, à des lemmes, à des catégories grammaticales ou à d'autres informations métalinguistiques.

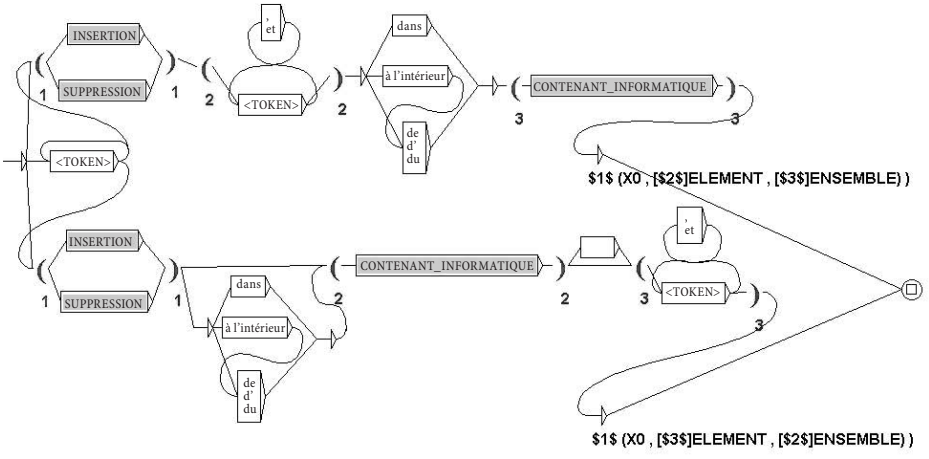
Un automate à états finis est un outil informatique qui permet de définir une grammaire locale et d'analyser une séquence de mots, et de décider si la séquence de mots analysée correspond à l'une des configurations de la grammaire locale. Un transducteur à états finis est un automate à états finis qui permet d'associer une nouvelle information à de l'information reconnue. Qu'il s'agisse d'automates ou de transducteurs, les informations métalinguistiques enregistrées dans les graphes au

niveau des nœuds sont celles qui sont encodées dans les dictionnaires électroniques associés aux graphes.

Le graphe ci-dessous indique la façon dont les instructions en français sont converties en commandes SQL.

FIGURE 2

Conversion des instructions en commandes SQL



Un script écrit en langage *Perl*[6] permet de générer dans la plateforme *Ontomantics* le code informatique de l'action correspondant à une instruction de l'utilisateur. Le script exécute l'intégralité de la chaîne de traitement : il lance l'invite de commande permettant à l'utilisateur d'entrer sa requête, puis les exécutables *Unitex*[7] qui permettent de générer les représentations métalinguistiques à partir de transducteurs à états finis et de générer le code *Ontomantics*.

3. Formulation d'une instruction : de la langue naturelle au langage SQL

Nous établissons en premier lieu que l'information traitée dans une base de données est fondée sur une relation partie-tout. Nous discutons ensuite des commandes SQL qui permettent de manipuler de l'information dans une base de données ; nous montrons que les commandes INSERT INTO et DELETE FROM correspondent à diverses instructions en langue naturelle faisant appel à des prédicats dits de <déplacement>, dont deux des trois arguments sont caractérisés par une relation méronymique. Nous précisons en dernier lieu comment la théorie des trois fonctions primaires associe les instructions mentionnées à des représentations métalinguistiques qui sont ensuite converties en commande SQL.

3.1. Méronymie et base de données

La gestion des bases de données repose sur différentes sortes d'opérations qui permettent de manipuler de l'information structurée. La structuration de l'information est fondée sur une double articulation :

1) celle entre une base et ses tables ;
2) celle entre une table et ses champs[8].

L'information contenue dans une base figure dans au moins une table qui comporte au moins un champ.

Une base de données équivaut à une liste d'items informatifs s'il s'agit d'une base avec une seule table dotée d'un seul champ. Elle présente alors peu d'intérêt dans la mesure où elle n'autorise qu'un nombre limité d'opérations. Il en est de même lorsqu'une base comporte plusieurs tables dotées d'un champ unique. En revanche, le traitement de l'information est amélioré si la base est constituée de plus d'une table dotée de plus d'un champ. La qualité du traitement de l'information dépend de la conception de la base de données ; notamment, les relations entre les tables sont primordiales pour l'amélioration de la structuration de l'information.

Les champs contiennent de l'information proprement dite, de la méta-information ou de l'information logistique. La méta-information est une information sur l'information ; typiquement, une définition dans une base de données lexicale. L'information logistique est une information inhérente à la structuration de l'information, typiquement un identifiant attribué à une information.

Les bases de données servent à stocker et traiter des informations. Un langage de requête, typiquement le langage *SQL*, prend en charge la gestion des bases de données. Il permet d'effectuer deux sortes de requêtes : d'une part, des requêtes qui permettent d'effectuer toutes sortes de manipulations relatives à l'administration d'une base de données ; d'autre part, des requêtes qui permettent de traiter l'information contenue dans la base et d'afficher le résultat du traitement.

La fonction de stockage de l'information justifie que l'on interprète une table d'une base de données comme un contenant dont le contenu est de l'information. Autrement dit, une relation méronymique caractérise le rapport entre une table et les informations dont elle fait état. Cette relation est fondamentale dans la mesure où elle est linguistiquement marquée dans les instructions formulées par l'utilisateur de la plateforme *Ontomantics*.

3.2. Formulations en langue naturelle

Différentes commandes SQL servent de point de départ pour générer des instructions spécifiques au langage *Ontomantics*, voir plus haut. Parmi celles-ci, il y a les commandes INSERT INTO et DELETE FROM. La première permet d'ajouter une information et les informations qui lui sont associées dans une table déjà structurée (c'est-à-dire une table telle que les différents champs ont été prédéfinis). La seconde autorise, entre autres, l'opération inverse ; une information et les informations qui lui sont associées sont enlevées d'une table déjà structurée.

Dans le contexte applicatif envisagé, les commandes INSERT INTO et DELETE FROM permettent de générer des tâches d'addition ou de suppression des coordonnées d'une personne dans une liste (voir plus haut). En amont, la réalisation de ces tâches présuppose que l'utilisateur de l'application formule des énoncés en langue naturelle tels que ceux qui apparaissent dans les première et troisième colonnes du tableau 1, par exemple *Insérer Luc Dupont dans la liste des inscrits* ou *Effacer Luc Dupont de la liste des inscrits*.

Toutes ces formulations font référence à des prédicats d'<adjonction> et de <suppression>[9]. Certains prédicats donnent lieu uniquement à des emplois verbaux (*mettre*) alors que d'autres donnent lieu à des emplois verbaux et nominaux (*insérer* / *insertion*, *mettre à jour* / *mise à jour*). Ces prédicats sont tous du type triadique (Buvet 2009b).

Aucun emploi ne mentionne l'argument qui fait référence à l'application et occupe la position sujet dans une construction canonique, *L'application supprime Luc Dupont de la liste des utilisateurs*, car les formulations sont nécessairement des injonctions correspondant à des phrases à l'infinitif, *supprimer Luc Dupont de la liste des utilisateurs*, au mode impératif, *supprime Luc Dupont de la liste des utilisateurs*, ou des phrases comportant également des prédicats de <volition>, *Je voudrais supprimer Luc Dupont de la liste des utilisateurs*.

Lorsque les emplois prédicatifs sont des verbes, les deux autres arguments occupent indifféremment les positions de premier complément et de second complément ; par contre, ils conservent leur mode de rection quelle que soit leur position. Avec le verbe *supprimer*, le groupe nominal *Luc Dupont* est toujours transitif direct et le groupe nominal *la liste des utilisateurs* transitif indirect en *de* : *supprimer Luc Dupont de la liste des utilisateurs* / *supprimer de la liste des utilisateurs Luc Dupont*.

Lorsque les emplois prédicatifs sont des noms, les positions des arguments sont fixes : l'un des deux arguments suit immédiatement le substantif prédicatif auquel il est relié par la préposition *de,* et l'autre vient ensuite précédé de la même préposition que celle qui le précède quand l'emploi prédicatif est un verbe (*insérer Luc Dupont dans la liste des utilisateurs* / *procéder à l'insertion de Luc Dupont dans la liste des utilisateurs*).

Les prépositions communes aux emplois verbaux et aux emplois nominaux varient selon qu'elles se rapportent à des arguments relatifs à des contenants (*liste des utilisateurs*), ou à leurs contenus (*Luc Dupont*) : *Mettre à jour la liste des utilisateurs (avec + *dans) Luc Dupont* / *Mettre Luc Dupont (*avec + dans) la liste des utilisateurs*. Elles dépendent également de la classe sémantique du prédicat : *Insérer Luc Dupont (dans + *de) la liste des utilisateurs* / *Ôter Luc Dupont (*dans + de) la liste des utilisateurs*.

L'interprétation des formulations dépend de la nature sémantique des prédicats, de l'identification de leurs arguments et de la compréhension de leur rôle. Elle est fondée sur une analyse syntactico-sémantique du lexique.

3.3. Traitement syntactico-sémantique des formulations

Les trois fonctions primaires sont la fonction prédicative, la fonction argumentale et la fonction actualisatrice. Elles permettent d'analyser les unités linguistiques sur le plan syntactico-sémantique et d'expliquer leur rôle dans la construction d'un énoncé. Par exemple, la phrase *Remplir cette liste avec ton nom* est analysée comme suit : la fonction prédicative s'applique au verbe ; la fonction argumentale s'applique aux deux noms ; la fonction actualisatrice s'applique à la marque de l'infinitif, aux deux prédéterminants et à la préposition.

La formation d'une phrase à partir de la combinatoire des différentes unités linguistiques implique une organisation hiérarchique des différents constituants

phrastiques. La structuration de la phrase est en grande partie fondée sur des relations de dépendance et de solidarité[10]. Pour ce qui est de la structure prédicat-argument, ce sont les prédicats qui prédominent structurellement les arguments, car les occurrences des arguments sont subordonnées à celles des prédicats. Par exemple, les compléments du verbe *remplir* sont obligatoirement deux groupes nominaux dont l'un fait office de contenant et l'autre de contenu. Par contre, les deux groupes nominaux peuvent se combiner avec d'autres verbes qui n'imposent pas une telle relation entre eux : *Écris sous cette liste ce nom*. La relation de dépendance entre le verbe *remplir* et les substantifs *liste* et *nom* démontre la position prédominante du prédicat verbal sur les arguments nominaux.

L'organisation hiérarchique de la structure prédicat-argument conduit à distinguer la fonction actualisatrice selon qu'elle concerne l'actualisation des prédicats ou bien celle des arguments. La marque temporelle étant imputable au seul verbe, on l'analyse comme un actualisateur prédicatif. On analyse le déterminant démonstratif comme un actualisateur argumental parce qu'il se combine avec les arguments nominaux. Cependant, il ne dépend pas de sa seule combinatoire avec les noms ; la combinatoire des noms avec le verbe explique également l'occurrence de l'article défini (Kleiber 1981). La relation de dépendance caractérise la combinatoire des actualisateurs avec des prédicats ou des arguments ; une phrase est bien formée parce que l'occurrence d'un actualisateur implique celle d'un prédicat ou celle d'un argument et réciproquement. Ces observations découlent du caractère indispensable de l'actualisation pour rendre une phrase grammaticale (Gross et Vivès 1986). Les actualisateurs prédicatifs sont en relation de solidarité uniquement avec les prédicats alors que les actualisateurs argumentaux le sont avec les prédicats et les arguments.

Les unités linguistiques n'ont pas les mêmes propriétés sémantiques selon la fonction primaire qui les caractérise : les prédicats et les arguments relèvent de la signification lexicale, les actualisateurs de la signification grammaticale (Blanco et Buvet 2004). On peut exprimer la différence entre les deux sortes de signification sur le plan quantitatif : le cardinal de l'ensemble des valeurs relatives à la signification lexicale est un grand nombre (des centaines de milliers), mais la fréquence d'occurrences de chacune de ces valeurs est faible ; le cardinal de l'ensemble des valeurs relatives à la signification grammaticale est un petit nombre (moins d'une cinquantaine), mais la fréquence d'occurrences de chacune de ces valeurs est élevée.

La relation méronymique caractérise des arguments en termes de signification grammaticale. L'un est un élément, l'autre l'ensemble auquel appartient cet élément[11]. Dans les différentes formulations du tableau 1, la relation méronymique est imputable à la nature de la préposition, à celle des arguments et à celle du prédicat. Ces trois facteurs permettent de préciser lequel des deux arguments correspond à un élément ou à un ensemble.

L'organisation hiérarchique des différents constituants rapportés à une fonction primaire peut être formalisée par une représentation fonctionnelle, dite représentation métalinguistique (Buvet 2001) : le prédicat équivaut à une fonction dont les variables sont les arguments. Les actualisateurs argumentaux sont indiqués dans cette représentation par des indices. Le tableau ci-dessous associe chaque énoncé en langue naturelle à une représentation métalinguistique qui stipule la structure prédicat-argument sous-jacente et la signification grammaticale en rapport avec la relation méronymique entre les deux arguments.

TABLEAU 1

Ajout ou suppression de coordonnées personnelles

Commande insert into		**Commande delete from**	
Instruction en langue naturelle	***Représentation métalinguistique***	**Instruction en langue naturelle**	***Représentation métalinguistique***
Insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*Effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Insère Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*Efface Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
J'aimerais insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*J'aimerais effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
J'aimerais pouvoir insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*J'aimerais pouvoir effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je veux insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*Je veux effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je veux pouvoir insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*Je veux pouvoir effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je voudrais insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*Je voudrais effacer de la liste des inscrits Luc Dupont.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je voudrais pouvoir insérer Luc Dupont dans la liste des inscrits.	*insérer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*	*Je voudrais pouvoir effacer de la liste des inscrits Luc Dupont.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Insérer dans la liste des inscrits Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Effacer de la liste des inscrits Luc Dupont.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Insère dans la liste des inscrits Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Efface de la liste des inscrits Luc Dupont.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
J'aimerais insérer dans la liste des inscrits Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*J'aimerais effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
J'aimerais pouvoir insérer dans la liste des inscrits Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*J'aimerais pouvoir effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je veux insérer Luc Dupont dans la liste des inscrits.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Je veux effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je veux pouvoir insérer Luc Dupont dans la liste des inscrits.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Je veux pouvoir effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je voudrais insérer Luc Dupont dans la liste des inscrits.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Je voudrais effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Je voudrais pouvoir insérer Luc Dupont dans la liste des inscrits.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Je voudrais pouvoir effacer Luc Dupont de la liste des inscrits.*	*effacer ([Luc Dupont]ÉLÉMENT, [liste des inscrits]ENSEMBLE)*
Mettre à jour la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Effacer Luc Dupont de la liste des inscrits.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*

Mets à jour la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Effacer Luc Dupont de la liste des inscrits.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*
J'aimerais mettre à jour la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*J'aimerais effacer de la liste des inscrits Luc Dupont.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*
J'aimerais pouvoir mettre à jour la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*J'aimerais pouvoir effacer de la liste des inscrits Luc Dupont.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*
Je voudrais mettre à jour la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Je voudrais effacer de la liste des inscrits Luc Dupont.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*
Je voudrais pouvoir mettre à jour la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*Je voudrais pouvoir effacer de la liste des inscrits Luc Dupont.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*
J'aimerais pouvoir remplir la liste des inscrits avec Luc Dupont.	*insérer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*	*J'aimerais pouvoir effacer de la liste des inscrits Luc Dupont.*	*effacer ([liste des inscrits]ENSEMBLE, [Luc Dupont]ÉLÉMENT)*
…	…	…	…

Ce tableau montre comment un grand nombre d'instructions formulées en langue naturelle a un nombre très inférieur de représentations métalinguistiques. Dans la chaîne de traitement, ce sont les représentations métalinguistiques des instructions qui sont ensuite transposées en commandes SQL ; la transposition s'effectue dans le module 2 dit de transfert. L'ensemble des énoncés en langue naturelle sont donc rapportés à l'une des commandes SQL suivantes : INSERT INTO et DELETE FROM.

4. Perspectives

En l'état actuel de nos travaux, la chaîne de traitement est opérationnelle du début à la fin. Notre objectif à moyen terme est d'augmenter, dans le contexte applicatif envisagé, le nombre d'instructions analysées afin de les transposer en langage machine.

Parallèlement, nous travaillons à la mise en place d'un système d'acquisition semi-automatique de vocabulaire spécialisé à partir de corpus. L'objectif est d'accroître les ressources linguistiques exploitées par le système car ses performances sont subordonnées à la qualité des analyses des instructions formulées par les utilisateurs.

NOTES

* Laboratoire Lexiques Dictionnaires Informatique UMR 7187.

1. Hyperdialogue avec un Agent en Langage proche de l'Interaction naturelle.
2. Danlos Laurence (1994) : Génération de textes dans le formalisme G-TAG inspiré des grammaires d'arbres adjoints. *Rapport technique TALANA*, 1.
3. À propos du système COALA : <http://www-lium.univ-lemans.fr/~lehuen/recherche/these/index.html>, consulté le 1er septembre 2009.
4. <http://www.ontomantics.com/>, consulté le 1er septembre 2009.
5. Le recours à des grammaires locales diminue le temps de traitement de l'information textuelle, car il permet de ne pas effectuer une analyse syntaxique robuste. Cette dernière n'est nécessaire que lorsque l'interprétation d'un texte est fondée sur l'identification des structures prédicat-argument.
6. <http://www.perl.org>, consulté le 1er septembre 2009.
7. <http://www-igm.univ-mlv.fr/~unitex/>, consulté le 1er septembre 2009.

8. Une table est un assemblage de lignes et de colonnes. Chaque colonne correspond à un type d'information et chaque ligne à un ensemble d'occurrences des différents types d'informations en rapport avec les colonnes.
9. La liste des prédicats d'<adjonction> comprend des verbes comme *ajouter*, *adjoindre*, *mettre* et des noms comme *insertion*, *mise à jour*, *adjonction*, etc. Celle des prédicats de <suppression>, des verbes comme *enlever*, *ôter*, des noms comme *enlèvement*, *suppression*.
10. Une relation de dépendance concerne la situation suivante : une occurrence d'un constituant implique une occurrence d'un autre constituant mais une occurrence de ce dernier n'implique pas une occurrence du constituant dont il dépend. Une relation de solidarité concerne la situation suivante : une occurrence d'un constituant implique une occurrence d'un autre constituant et réciproquement (Harris 1976).
11. Le rapport élément/ensemble permet de subsumer toutes sortes de relations méronymiques : partie/tout, membre/collection, contenu/contenant, etc. (Buvet 2001).

RÉFÉRENCES

Blanco, Xavier et Buvet, Pierre-André (2000) : De l'analyse syntactico-sémantique du lexique à la traduction automatique. *BULAG*. 25:69-87.

Blanco, Xavier et Buvet, Pierre-André (2004) : Verbes supports et significations grammaticales. Implications pour la traduction espagnol-français. *Lingvisticae Investigationes*. 27(2):327-342.

Buvet, Pierre-André (2001) : Représentations métalinguistiques de phrases à partir de transducteurs. *Revue informatique et Statistique dans les Sciences humaines*. 36:86-99.

Buvet, Pierre-André (2009a) : Quelles procédures d'étiquetage pour la gestion de l'information textuelle électronique. *L'information grammaticale*. 122:40-48.

Buvet, Pierre-André (2009b) : Des mots aux emplois : la représentation lexicographique des prédicats. *Le Français moderne*. 77(1):83-96.

Galibert, Olivier, Illouz, Gabriel et Rosset, Sophie (2005) : RITEL : dialogue homme-machine à domaine ouvert. *In* : *Actes du colloque TALN 2005*. (TALN 2005, Dourdan, 6-10 juin 2005). 439-444.

Glass, James, Polifroni, Joseph, Seneff, Stephanie *et al.* (2000) : Data collection and performance evaluation of spoken dialogue systems : the MIT experience. *In* : *Proceedings of ICSLP'00*. (ICSLP 2000, Pékin, octobre 2000). Vol. 4, 1-4.

Gross, Gaston (1992) : Reconnaissance des emplois à l'aide d'un dictionnaire électronique. *Études de linguistique appliquée*. 85/86:89-97.

Gross, Gaston et Vivès, Robert (1986) : Les constructions nominales et l'élaboration d'un lexique-grammaire. *Langue française*. 69:5-27.

Gross, Maurice (1995) : Une grammaire locale de l'expression des sentiments. *Langue française*. 105(1):70-87.

Harris, Zellig-S. (1976) : *Notes du cours de syntaxe*. Paris : Le Seuil.

Kleiber, Georges (1981) : Problèmes de référence. Descriptions définies et noms propres. *Recherches linguistiques n° VI – Études publiées par le Centre d'Analyse syntaxique de l'Université de Metz*. Paris : Klincksieck.

Manaris, Bill Z. et Dominick, Wayne D. (1993) : Nalige : a User Interface for the Development of Natural Language Interface. *International Journal of Man-Machine Studies*. 38(6):891-921.

Mathieu-Colas, Michel (1994) : *Les mots à trait d'union. Problèmes de lexicographie informatique*. CNRS-InaLF, Paris : Didier Erudition.

Mathieu-Colas, Michel (2009) : *Morfetik* : une ressource lexicale pour le TAL. *Cahiers de lexicologie*. 94:137-146.

Maurel, Denis (1993) : Reconnaissance automatique d'un groupe nominal prépositionnel. Exemple des adverbes de date. *Lexique*. 11:147-161.

Pierrel, Jean-Marie, dir. (2000) : *Ingénierie des langues*. Paris : Hermès.

Pierrel, Jean-Marie et Sabah, Gérard (1991) : Dialogue en langage naturel écrit ou oral : bilan des approches du CRIN et du LIMSI. *In* : *Actes Communication Homme-Machine*. (EC2 – Greco PRC CHM, Toulouse, 29-30 janvier 1991). 91-111.

Rouillard, José (2000) : *Hyperdialogue sur Internet. Le système HALPIN*. Thèse de doctorat. Grenoble : Université Grenoble I.

Sabah, Gérard (1989) : *L'intelligence artificielle et le langage*. 2 volumes. Paris : Hermès.

van der Lans, Rick F. (2006) : *Introduction to SQL : Mastering the Relational Database Language*. Upper Saddle River, NJ : Addison-Wesley.

Weizenbaum, Joseph (1966) : ELIZA – a computer program for the study of natural language communication between man and machine. *CACM*. 9:36-45.