

Conception d'un langage de requêtes pour un agent conversationnel assistant

François Bouchet
LIMSI-CNRS
BP 133, 91403 Orsay, France
Université Paris XI
francois.bouchet@limsi.fr

Résumé

Dans le cadre du projet DAFT qui a pour objectif la réalisation d'un agent conversationnel assistant capable de raisonnement sur la structure et le fonctionnement d'une application, nous cherchons à élaborer un langage de requêtes structuré dans lequel seront transformées les requêtes langagières des utilisateurs pour être ensuite exploitées par le module de raisonnement.

La conception de cette représentation intermédiaire fondée sur l'identification d'actes de dialogue se base sur l'étude réalisée lors de ce stage d'un corpus de requêtes d'assistance recueilli à cet effet.

Mots Clés

Agents conversationnels, fonction d'assistance, actes de dialogue, corpus de requêtes.

1. INTRODUCTION

1.1 Contexte du projet : l'assistance

C'est dans les années 70 que furent mis au point la plupart des concepts de base en interface homme-machine. Et si l'ergonomie a depuis lors été fortement améliorée, la majorité des systèmes actuels emploie toujours une interface de type WIMP (Windows Icons Menus Pointer) imaginée au PARC par Xerox et popularisée par Apple puis Microsoft, dont les paradigmes de base demeurent globalement inchangés bien que l'on soit conscient de leurs limites [11]. Pourtant, la population des utilisateurs a pour sa part considérablement évolué en 30 ans :

- en nombre : la moitié des foyers français sont équipés d'au moins un ordinateur¹ et la moitié des français ont déjà "surfé" sur internet².
- en diversité : on est passé d'une informatique d'informaticiens à une informatique destinée à tout le monde.

Parallèlement, suivant la loi de Moore, les machines sont devenues plus puissantes et ont donc vu s'élargir leur champ d'application (traitement de texte, multimédia, jeux vidéos, recherche d'informations...) et se sont mises à interagir avec de nombreux autres systèmes (imprimante, scanner, PDA, appareil photo...).

Ainsi, la population actuelle d'utilisateurs novices doit faire face à des systèmes complexes, dotés d'une interface qui correspond de moins en moins à la représentation cognitive

¹source : enquête INSEE, Mars 2005

²source : enquête INSEE, Mai 2006

qu'ils en ont intuitivement. En effet, la métaphore du bureau sous-jacente à ce type d'interface n'est pas forcément la plus pertinente envisageable pour des jeunes ou des personnes âgées peu familières de cet environnement. À défaut d'une révolution ergonomique improbable à court terme, la fonction d'assistance devient donc incontournable. Mais l'offre disponible actuellement (généralement sous forme de manuels papier ou de pages web) est inadaptée car :

- **impersonnelle** : l'aide est désincarnée et identique pour tous, ignorant les spécificités de l'utilisateur.
- **uniquement réactive** : elle est incapable d'anticiper les problèmes qui vont apparaître.
- **statique** : elle ne tient pas compte du contexte dynamique d'utilisation de l'application.
- **procédurale** : elle est souvent trop orientée vers la description des opérateurs et des fonctionnalités, et pas assez vers les problèmes effectifs rencontrés par les usagers (malgré l'adjonction occasionnelle de systèmes complémentaires de type FAQ³).

1.2 Contexte du stage : le projet DAFT

Initié en 2004 au sein du groupe AMI (Architectures et Modèles pour l'Interaction) du LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur) par Jean-Paul Sansonnet, le projet DAFT se propose d'étudier une solution possible à ce problème par l'utilisation d'un agent conversationnel assistant. Cet agent se voit doté de capacités d'introspection au sein du modèle de l'application dans laquelle il est intégré (par l'utilisation d'un médiateur symbolique qui représente les composants de celle-ci), dont il se sert pour répondre à des requêtes formulées par l'utilisateur du système de manière langagière. Car si les interfaces classiques offrent une certaine rapidité d'interaction, lorsqu'il est "perdu", c'est par le langage naturel que s'exprime le mieux et le plus spontanément l'utilisateur.

Ces agents conversationnels peuvent revêtir différentes apparences visuelles :

- Textuelle uniquement : c'est le cas des rubriques d'aide avec recherche par questions libres (bien que le système d'analyse derrière soit le plus souvent rudimentaire).
- Graphique : avec une représentation statique de l'agent permettant de lui donner une visibilité supérieure.
- Animée : avec une représentation visuelle de l'agent (humanoïde ou non) capable de s'animer et qui facilite la communication.

³Frequently Asked Questions - Foire Aux Questions

C'est cette forme d'agents que nous utilisons dans le projet DAFT.

Les agents conversationnels se prêtent donc parfaitement à l'assistance car ils sont (idéalement) capables de répondre aux trois besoins essentiels et complémentaires de l'utilisateur en situation d'échec ou d'égarement face à une application :

- Besoin de **compréhension** : assuré par l'**agent dialogique** qui communique avec l'utilisateur selon la modalité la plus adaptée pour résoudre son problème.
- Besoin de **crédibilité** : satisfait par l'**agent rationnel** qui analyse l'état de l'application à partir d'un modèle et apporte donc une réponse adaptée.
- Besoin de **présence** : comblé par l'**agent animé** qui rassure psychologiquement l'utilisateur novice qui sent sa présence et lui accorde inconsciemment des qualités humaines (c'est le "persona effect" de Lester [19]).

L'architecture employée a pour avantage d'établir une séparation méthodologique entre la problématique du traitement des requêtes en langue naturelle et celle liée au déploiement de composants et services. Elle offre également une certaine flexibilité en permettant de générer différents modèles de médiateur pour un même composant en fonction de l'application dans laquelle il est intégré.

Actuellement en version 1, DAFT se compose de trois modules distincts (cf fig. 1) :

- **GRASP** (Générateur de Requêtes par Analyse Sémantique Partielle) : il s'agit d'un module d'analyse linguistique au sens classique dont l'objectif est d'extraire la sémantique des requêtes d'assistance en langue naturelle pour produire des requêtes formelles à destination de l'agent rationnel. Il effectue donc une contraction sémantique de l'énoncé originel, en se basant sur des ressources linguistiques.
- **DAFT** : ce module, qui donne son nom au projet, implémente la fonction d'agent rationnel. Il traite une requête formelle en DAFT (Dialogical Agent Formal Talk) en effectuant une extension sémantique en situation en fonction de l'état actuel du dialogue et de l'application, et se charge notamment de résoudre les références (au sens de références extensionnelles associatives) figurant dans la requête, c'est-à-dire de faire le lien entre les expressions langagières et les composants de l'application à partir du modèle médiateur dont il dispose (la définition de ce modèle est l'objectif des travaux de David Leray [18]). Il sera de plus doté d'un planificateur et d'un gestionnaire de buts pour pouvoir élaborer des stratégies de raisonnement pour répondre aux questions portant sur le moyen d'atteindre un objectif.
- **EVA** : ce dernier module permet de présenter les résultats du traitement de la requête par l'agent rationnel. Cette réponse est de nature multimodale puisque l'agent peut donner sa réponse sous forme textuelle et vocale, tout en l'accompagnant de gestes déictiques et d'expressions faciales appropriées de l'avatar pour renforcer l'impact de l'information qu'il délivre. On utilise pour cela un des quatre

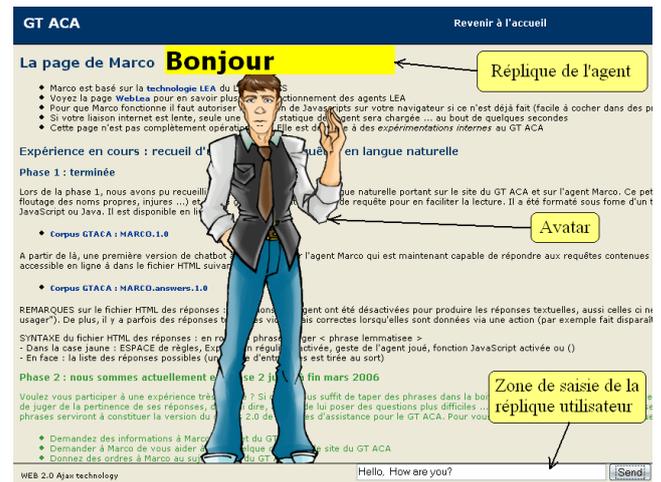


Figure 2 : L'agent WebLea (Marco) placé sur une page du site web du GT ACA

avatars WebLEA⁴, développés par Jean-Claude Martin (cf. figure 2).

Le projet se retrouve ainsi à la confluence de plusieurs communautés scientifiques : Traitement Automatique des Langues pour l'analyseur linguistique, Intelligence Artificielle pour l'agent rationnel, notamment pour la résolution de références et la représentation formelle des connaissances, Composants Logiciels pour la gestion du médiateur et Interface Homme-Machine pour tous les choix relatifs à l'avatar de l'agent.

1.3 Problématique de recherche

Mon stage porte sur la création d'un langage formel de requêtes⁵, afin d'offrir une représentation symbolique de la sémantique des requêtes de l'utilisateur. Il fait ainsi le lien entre la couche supérieure de GRASP, dont il sera le langage cible, et le module DAFT, pour lequel il constituera la base du raisonnement de l'agent rationnel pour la résolution des requêtes à partir du modèle de l'application.

Une première version de langage DAFT 1.0 a été élaborée par Karl Le Guern en 2004 [17]. Par rapport à cette version, le langage à construire se distingue par :

- **une approche corpus** : en construisant le langage après une annotation manuelle d'un corpus représentatif des requêtes d'assistance et de taille conséquente, il est ancré dans la réalité car construit en fonction des spécificités de ce corpus. Cela lui permettra de s'avérer mieux adapté aux répliques des utilisateurs finaux une fois implémenté.
- **des schémas** plus élaborés : en augmentant la complexité des schémas du langage, on espère pouvoir parvenir à capter de manière plus fine la richesse sémantique des requêtes de l'utilisateur.

La principale difficulté de ce travail est liée à la nécessité d'un compromis entre :

⁴LIMSI Embodied Agent

⁵Ce langage est qualifié de "formel" pour signifier qu'il ne s'agit pas de langue naturelle, et non au sens de la logique formelle.

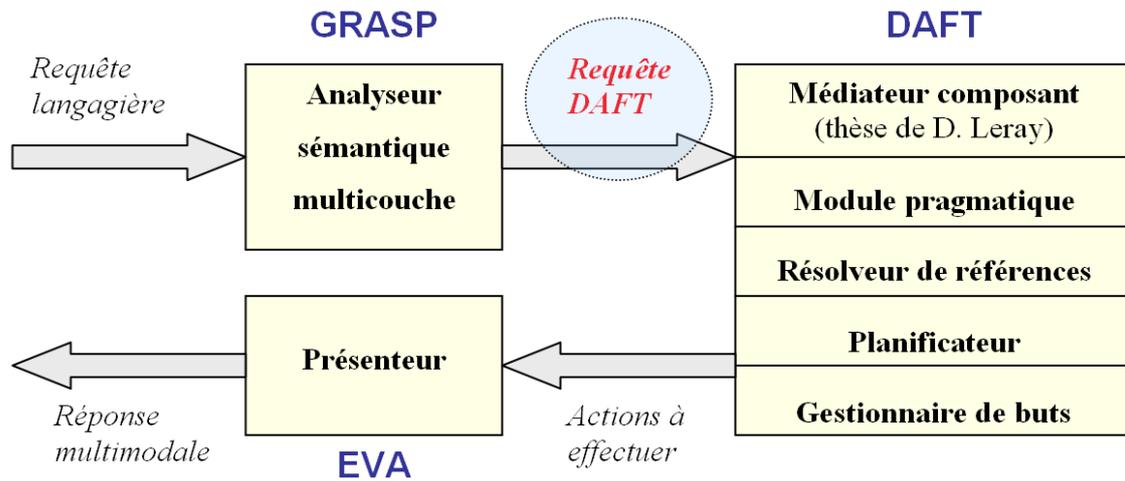


Figure 1 : Architecture globale du projet DAFT

- un langage suffisamment expressif pour que la contraction sémantique n'entraîne pas de pertes préjudiciables à la bonne compréhension de la requête telle que formulée à l'origine,
- un langage trop riche, incapable de faire ressortir le sens premier des requêtes de l'utilisateur.

Dans ce rapport, nous commencerons donc par exposer un état de l'art dans le domaine des agents conversationnels, et en particulier en ce qui concerne leur utilisation dans le cadre de l'assistance qui est celui du projet DAFT.

Dans un deuxième temps, je présenterai les résultats de l'étude du corpus DAFT dans laquelle je me suis attaché à déterminer les spécificités de celui-ci en le comparant à un corpus généraliste ainsi qu'à d'autres corpus de nature similaire. J'ai également cherché à établir une classification des différents types de requêtes le constituant et à caractériser ces catégories selon différents critères.

Dans une troisième partie, j'exposerai en détail la méthode employée pour construire la syntaxe d'un nouveau langage de requêtes palliant les défauts du modèle précédent, en m'attardant en particulier sur la justification de certains des choix effectués.

Enfin, avant de conclure, j'évaluerai le langage proposé en le confrontant au corpus à partir duquel il a été construit et en discutant des moyens de l'intégrer à la chaîne de traitement DAFT actuellement en place.

2. LES AGENTS CONVERSATIONNELS

La notion d'agent conversationnel étant au coeur du projet DAFT, nous allons dresser dans cette section un état de l'art visant à rappeler leur origine et leur évolution, à analyser les problématiques de recherche qu'ils recouvrent et à lister leurs domaines d'application, avant finalement de nous attarder plus particulièrement sur le cas des agents conversationnels assistants.

2.1 Historique et évolution

Le principe des agents conversationnels consiste à établir une relation de type dialogique en langue naturelle entre un utilisateur et un agent logiciel doté de bases de dialogue et

de connaissances spécifiques à la tâche qu'il doit accomplir. Le concept d'agent conversationnel est relativement ancien puisqu'il remonte à la fin des années 60, lorsque fut développé l'un des plus illustres en la personne d'ELIZA de Joseph Weizenbaum [30], qui jouait un rôle de "psychanalyste virtuel". Il fut suivi en 1971 du système SHRDLU de Terry Winograd qui ajoute chez l'agent la notion de "monde physique" [31].

Dans les années 90, avec le développement de projets universitaires comme OLGA [4] au KTH et suite aux travaux de Cassell et al. d'abord sur Gandalf [27], puis sur REA [9], les agents conversationnels sont devenus majoritairement des agents conversationnels animés avec l'adjonction d'un avatar représentant l'agent, doté au minimum d'une représentation graphique statique, mais aussi souvent d'expressions animées et d'une voix de synthèse leur permettant de vocaliser leurs réponses. De plus en plus, on tente également de leur donner une véritable "incarnation" (traduction du terme anglais "embodiment") en les dotant d'une personnalité propre, rendue perceptible notamment par le biais de leurs émotions [21][6]. Tous ces éléments contribuent à renforcer la présence de l'agent du point de vue de l'utilisateur grâce à l'utilisation conjointe ou dissociée de plusieurs modalités en entrée (reconnaissance vocale, des émotions, des mouvements de l'utilisateur dans l'espace, des gestes déictiques...) comme en sortie (synthèse vocale, gestes déictiques, représentation des émotions...).

2.2 Domaines de recherche

Les principaux défis concernant les agents conversationnels concernent désormais le développement de toutes les technologies qui les constituent et qui ont été réunies dans les projets pionniers mentionnés précédemment. Depuis le début des années 2000, des projets de grande envergure généralement financés par les pouvoirs publics visent donc à aller au-delà des prototypes déjà réalisés en développant des agents mieux modélisés et aux capacités de dialogue plus poussées pour pouvoir élaborer des applications orientées vers le grand public.

2.2.1 Multimodalité et émotions

Le premier axe de recherche concerne l'utilisation d'autres modalités que la langue naturelle (qu'elle soit sous forme écrite ou orale) pour que l'agent puisse à la fois percevoir davantage d'informations sur l'environnement réel dans lequel il se situe (et notamment concernant l'utilisateur) et mieux transmettre son message dans ses interventions en utilisant des signes de communication extra-verbaux (gestes et expressions du corps et du visage). De nombreux travaux tournent donc autour de la détection et la restitution d'émotions, qui sont jugés indispensables à la réussite d'une interaction homme-machine de qualité [3].

On peut notamment citer à cet égard le projet européen NICE⁶ qui visait à étudier et à développer une conversation multimodale entre des enfants et un agent virtuel représentant le conteur danois Andersen. La multimodalité portait sur l'utilisation conjointe de la parole et de gestes dans un espace en deux dimensions (en pointant sur un écran) [7]. Les agents WebLea employés dans DAFT ont d'ailleurs été développé pour des tests menés dans le cadre de ce projet. Depuis 2004, le réseau d'excellence européen HUMAIN⁷ implique une trentaine de laboratoires et vise à coordonner les recherches notamment en ce qui concerne les théories des émotions, les schémas de signes révélateurs d'émotions dans les interactions et l'utilisabilité de systèmes orientés par les émotions.

La synthèse vocale qui constitue une brique de plus en plus courante dans les agents conversationnels se doit également d'être affectée par les émotions potentielles de l'agent, et actuellement, même les meilleurs systèmes de synthèse vocale sont difficilement capables de modifier suffisamment finement la prosodie pour rendre un discours expressif naturel [8].

Dans le cadre du projet DAFT, ces deux éléments sont délégués au module LEA, et mais ils ne constituent clairement pas notre sujet d'étude.

2.2.2 Gestion du dialogue

Le second axe principal sur lequel se focalisent les recherches n'est pas spécifique aux agents conversationnels puisqu'il s'agit de l'étude des différentes stratégies possibles concernant la gestion du dialogue [24].

Historiquement, les grammaires de dialogues se fondent sur l'étude des actes de dialogues [25] dans un dialogue pris dans son ensemble. Les approches plus récentes se focalisent davantage sur la notion de paires adjacentes qui constituent le dialogue [14], qui consiste à dire qu'une question sera suivie d'une réponse, une proposition d'une acceptation ou d'un rejet. . .

Néanmoins, ces méthodes sont peu adaptables lorsque le dialogue s'éloigne du sujet d'origine, et depuis les années 90, les approches sont très souvent basées sur l'étude de schémas ou frames pour augmenter la flexibilité du système, celles-ci permettant au moins au système de saisir le thème général de la conversation s'il n'est pas capable d'en capter les détails. C'est également ce type d'approche par frames que nous employons dans le projet DAFT, comme nous le ver-

⁶Natural Interactive Communication for Edutainment - de 2002 à 2005

⁷Human-Machine Interaction Network on Emotion

rons en 4.2.1.

Un autre point important des recherches en gestion de dialogue concerne la planification : puisque « dire c'est faire » [2], il faut aller au-delà du contenu purement sémantique des phrases et identifier les buts sous-jacents et plus ou moins implicites de celui qui les prononce. Cette conception n'est cependant pas généralisable à tout type de dialogue, mais fonctionne bien dans le cas d'un système conçu pour aider à accomplir un ensemble défini de tâches, ce qui correspond à ce que nous désirons faire avec DAFT (dans le cas général, c'est-à-dire en excluant les phrases ne portant pas directement sur l'assistance, voir en 3.3.2 pour plus de détails).

Une des modélisations les plus courantes est l'architecture BDI⁸ [5] dans laquelle :

- les croyances (B) représentent les connaissances de l'agent au sujet du monde,
- les désirs (D) correspondent à l'état du monde qu'il souhaiterait atteindre,
- les intentions (I) sont différentes stratégies possibles pour rendre la réalité du monde conforme à ses désirs.

Ce modèle fut perfectionné par Traum dans le cadre du projet TRAINS-93 pour gérer plusieurs agents en intégrant la notion de croyances communes et en considérant que la perception du monde peut influencer les désirs et intentions des agents [28].

En ce qui concerne le projet DAFT, pour le moment nous ne traitons que des requêtes isolées, aussi les techniques d'analyse de requêtes actuellement employées sont-elles peut-être davantage à rapprocher des systèmes de questions-réponses que des systèmes de dialogue. En revanche, la réponse n'est pas obtenue par une recherche à l'aide de patrons syntaxiques de réponses dans un corpus de grande taille mais par une analyse du modèle de l'application en fonction de certaines heuristiques (cf section 5.4.2).

2.3 Domaines d'utilisation

Parallèlement, les agents conversationnels sont sortis des laboratoires de recherche universitaires et des solutions commerciales à base d'agents conversationnels commencent à être offertes par des entreprises spécialisées. Les domaines d'utilisation de ceux-ci sont multiples et diversifiés :

- le commerce électronique : en agissant de manière similaire à un vendeur en magasin, l'agent renseigne et rassure, augmentant ainsi le taux d'achats. Il apporte également un aspect vivant aux sites internet et renforce leur identité, aide à fidéliser les clients et à recueillir plus naturellement des informations à leur sujet.
- les services d'assistance à distance : il s'agit d'un domaine très coûteux pour les entreprises qui délocalisent souvent ce type d'activité pour en réduire le coût au maximum. Les dialogues qui ont lieu entre les opérateurs et les clients sont cependant déjà très scriptés, et il est potentiellement très avantageux de substituer un agent conversationnel (capable de gérer plusieurs sessions en parallèle) aux opérateurs humains pour répondre aux questions les plus fréquemment posées, avec éventuellement une possibilité de basculer de manière transparente sur un opérateur hu-

⁸Beliefs Desires Intentions

main si les questions s'avèrent trop pointues par rapport au champ de connaissances de l'agent. Le client peut pour sa part ainsi espérer voir son temps d'attente réduit.

- l'apprentissage et la formation à distance : l'agent peut pallier le manque de contact humain qui est le principal travers des solutions classiques d'e-learning dans lesquels l'élève se sent souvent seul face à des documents statiques ou des présentations animées peu interactives. Dans des contextes plus spécifiques, ces agents pédagogiques peuvent s'intégrer dans des mondes virtuels pour former à la pratique de tâches physiques [15] ou à la prise de décision en milieux difficiles [29].
- les jeux vidéos : transformer les personnages contrôlés par l'intelligence artificielle en véritables agents conversationnels offre très certainement de nouvelles possibilités créatives et ludiques renforçant l'aspect vivant et réaliste des univers virtuels. C'est par exemple le cas des univers virtuels dotés d'agents discutant entre eux et sur lesquels le joueur peut intervenir pour modifier le cours de l'histoire (interactive storytelling [10]).
- les assistants personnels : offrir un agent personnel adapté aux goûts et aux habitudes de son utilisateur, omniprésent par son intégration dans un périphérique mobile de type PDA ou téléphone portable, et qui l'assiste pour toutes ses démarches quotidiennes est un des objectifs les plus ambitieux promis par les agents conversationnels. Des projets de recherche appliqués en ce sens commencent à émerger : on peut citer le projet européen Companions⁹, dirigé en premier lieu vers les utilisateurs ayant le plus besoin d'assistance dans leurs tâches quotidiennes, c'est-à-dire les jeunes et les personnes âgées.

2.4 Agents Conversationnels Assistants

Comme nous l'avons vu, le domaine de l'assistance étant en plein essor et les agents constituant une réponse adéquate à ce besoin, des solutions à base d'agents conversationnels se sont multipliées dans ce domaine ces dernières années. Le système de référence dans ce domaine et qui se rapproche le plus des objectifs du projet DAFT est le système TRIPS développé par Allen, dans lequel l'utilisateur est chargé de gérer l'évacuation des habitants d'une île, et est assisté à cet effet par un agent conversationnel [1].

Bien que l'assistance apportée soit sensiblement différente dans le cas du projet DAFT qui porte sur l'interface d'une application, on retrouve les mêmes éléments dans la chaîne de traitement de l'agent et une volonté commune de disposer d'un système générique, pouvant s'adapter sans un coût trop élevé à d'autres applications. Cependant, les différences d'objectifs entre les deux projets font que les traitements à chaque étage ne peuvent être gérés de manière identique : ainsi, au niveau de l'analyse en entrée, les requêtes des utilisateurs de DAFT ne sont souvent pas grammaticalement correctes et une approche sémantique se révèle nécessaire. De même au niveau du raisonnement, les références de TRIPS sont des éléments connus du système par un nom précis, tandis que DAFT doit résoudre celles-ci en tant que références extensionnelles associatives.

⁹<http://www.dcs.shef.ac.uk/nlp/companions/>

Un autre exemple de projet similaire visant à définir une architecture générique permettant de dialogiser facilement des objets (du monde réel) a été mené à l'Université du Manitoba [16]. Le raisonnement se fait à l'aide de la logique de description, et fonctionne grâce à une ontologie du domaine fournie par le programmeur, avec des phrases artificiellement construites.

3. ÉTUDE DU CORPUS DAFT

3.1 Présentation du corpus

Notre étude a été menée à partir d'un corpus de requêtes d'assistance en langue naturelle recueilli sur deux ans et constitué de 5 189 phrases fin mars 2006¹⁰. Ces requêtes sont conservées en "brut" (sans aucun traitement), avec toutes les expressions orales, fautes d'orthographe, de grammaire et de syntaxe que cela suppose, comme le montre la table 1.

Soulignons le fait que nous travaillons exclusivement sur des *requêtes isolées* : ce sont des demandes ponctuelles, visant un accomplissement immédiat, par opposition à des tâches (par exemple "réserver un billet de train"), qui elles supposent une succession d'actions (et donc un dialogue les accompagnant au fur et à mesure). Ces requêtes sont donc issues d'interactions dialogiques limitées à un seul tour, et il ne s'agit donc pas du dialogue homme-machine au sens traditionnel. Ainsi, même si le corpus contient quelques répliques dénotant visiblement une réaction de l'utilisateur à une réplique de l'agent, elles sont actuellement prises en compte indépendamment de tout échange antérieur.

3.1.1 Homogénéité

Les requêtes d'assistance constituant le corpus ont été recueillies par le biais d'agents de type LEA, dans deux situations :

- des agents assistants inclus dans des applications ou applets Java qui avaient été développées dans le cadre du projet Interviews¹¹ : Coco le compteur et un jeu de tours de Hanoi [17] [18].
- des agents assistants présents sur des sites web, et en particulier, sur le site internet du GT ACA¹² (cf. figure 2), ce qui explique les nombreuses requêtes liées à la navigation et à la recherche d'informations sur un site.

Cette légère disparité de nature des requêtes peut sembler problématique a priori. Cependant, DAFT vise à s'intégrer dans le cadre du Web 2.0 employant des technologies de type AJAX¹³, c'est-à-dire un contexte dans lequel quasiment toutes les applications traditionnellement installées en local sur l'ordinateur de l'utilisateur sont en réalité situées à distance et accessibles via des pages web : du traitement de texte¹⁴ au bureau virtuel¹⁵. Dans ce contexte, le concept de navigation est par conséquent omniprésent même dans le cadre d'applications plus traditionnelles, et il n'y a donc

¹⁰On en compte près de 8 000 début septembre 2006

¹¹<http://www.limsi.fr/jps/interviews/index.html>

¹²<http://www.limsi.fr/aca/>

¹³Asynchronous JavaScript And XML

¹⁴Comme Writely (<http://www.writely.com>)

¹⁵Regroupant messagerie instantanée, client de courrier électronique, calendrier et mini-suite bureautique (<http://www.goowy.com>)

Table 1 : Quelques exemples de phrases “brut” du corpus DAFT

a plus	A quoi sers-tu ?
a+	alors là t'es complètement paumé!
ah	appelle moi simplement Sylvie
Allez, bye	as tu des informtion ?
Allez ciao.	as tu entendu parler d'une certaine expérimentation en cours? ?
Alors?	au sujet de cette page, que peux tu dire?
As-tu des amis?	avec ce corpus, tu sauras ce qu'est une anaphore ...
auf wiedersehen	à quoi penses tu ?
avec quoi ?	Bah tu viens de dire que tu pouvais remonter le moral!
à l'aide!	be ouais tu comprends pas
bah!	ben alors reponds
barre toi de là	bon j'en ai marre je me tire ...
bidule	Bon je me casse. Bye.
bon à rien!	bon y a rien à tirer de toi!!
bon week end	bon, ça va, bonne année 2006
Bon.	Bon, dis-moi plutôt ce que tu sais faire
Bonjour, Marco	bon, reviens à la page d'accueil
bonsoir	bonjourmon vieux

pas de coupure sémantique pertinente à considérer au sein de notre corpus pour l'étude que nous voulons en faire.

3.1.2 Représentativité

La taille relativement restreinte de ce corpus de quelques milliers de phrases (contre souvent quelques millions pour d'autres types de corpus) pose la question de la représentativité de celui-ci pour le contexte d'étude que représente la fonction d'assistance. Nous pouvons nous permettre de penser que notre corpus est effectivement représentatif dans la mesure où il est constitué de :

- 65% de phrases empiriques recueillies dans le contexte défini précédemment,
- 35% de phrases enrichies de structures grammaticales issues de thésaurus : « *Express ways functions* » de Molinsky et Bliss (1995), et « *Grammaire active* » du dictionnaire Français-Anglais Robert&Collins (4e édition, 1996).

Cette méthode est certes discutable puisqu'elle n'est pas purement empirique, mais elle nous permet ainsi de compenser la difficulté d'obtenir un corpus de taille plus conséquente avec les moyens dont nous disposons, en saturant au maximum celui-ci par l'utilisation de structures grammaticales qui n'apparaîtraient peut-être de manière “naturelle” qu'une ou deux fois sur un corpus d'un million de requêtes.

En revanche, il est probable que cette saturation puisse avoir un impact sur certaines statistiques du corpus, en augmentant artificiellement les proportions de phénomènes rares.

3.2 Étude comparative de corpus

La première étape de notre étude consiste à identifier les caractéristiques propres au corpus DAFT, ce que nous réalisons en le comparant à d'autres corpus.

3.2.1 Comparaison avec un corpus généraliste

Notre objectif est de démontrer que le corpus DAFT définit un sous-langage particulier de la langue naturelle. Afin de valider cette spécificité, nous avons choisi de le comparer dans un premier temps à un corpus généraliste, en l'occurrence il s'agit de Multitag qui est constitué de phrases issues

d'articles du journal *Le Monde*, de romans et d'essais.

Nous avons commencé par calibrer Multitag sur DAFT afin d'en sélectionner un sous-ensemble ayant le même nombre de flexions que dans le corpus DAFT. On peut d'ores et déjà constater la plus grande variété de vocabulaire de Multitag : 1460 phrases suffisent à avoir le même nombre de flexions différentes que 5000 phrases pour le corpus DAFT. De plus, lorsqu'on lemmatise les flexions des deux ensembles, on constate également une richesse lexicale supérieure de Multitag par rapport à DAFT puisqu'il comporte 3620 lemmes différents, contre seulement 1788 pour DAFT, c'est-à-dire plus du double. Parmi ces lemmes, 1188 sont communs aux deux corpus.

Enfin, lorsque l'on compare le nombre de lemmes inconnus introduits par un doublement de la taille du corpus avec des phrases issues de conditions similaires (agent Marco du site du GT ACA) et le nombre de lemmes inconnus dans le cas de phrases issues du corpus Multitag, la différence est une nouvelle fois très nette : alors que seuls 250 nouveaux lemmes sont introduits dans le premier cas, ils sont 4725 dans le second.

Cette simple comparaison met bien en évidence le fait que le corpus DAFT constitue un sous-langage, voire peut-être même un genre linguistique, car il est à la fois distinct (vocabulaire “métier” spécifique) et restreint (ensemble de vocabulaire fermé) d'un point de vue lexical par rapport à l'ensemble de la langue.

3.2.2 Comparaison statistique de différents corpus d'assistance

En suivant la méthode de comparaison statistique de corpus établie par Gabriel Ripoche dans sa thèse [23], on peut ensuite comparer le corpus DAFT à quelques autres corpus de dialogues orientés tâche en étudiant leur profils interactionnels. On appelle profil interactionnel d'un corpus une représentation sous forme d'histogrammes de la répartition des 5 types d'actes de dialogue searliens au sein de celui-ci (cf fig. 3). Les 6 colonnes représentent donc :

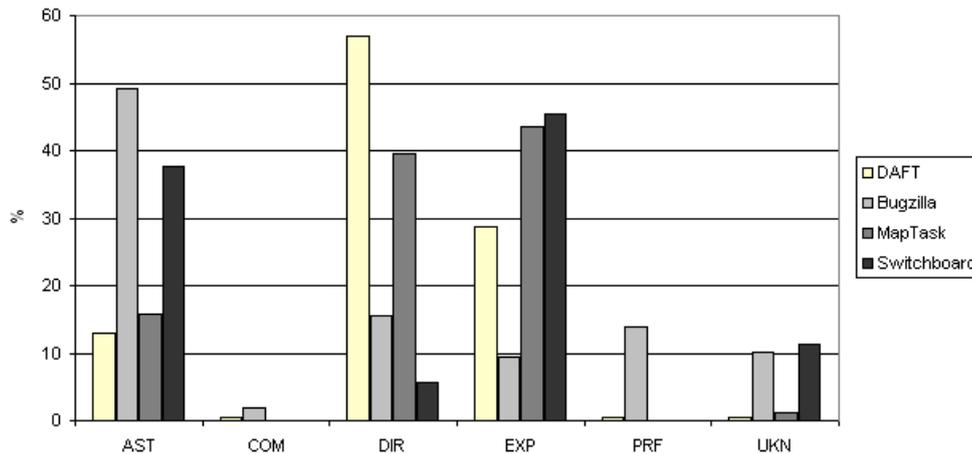


Figure 3 : Répartition des actes de dialogue searliens dans différents corpus

1. les assertifs (AST),
2. les promissifs (COM),
3. les directifs (DIR),
4. les expressifs (EXP),
5. les performatifs (PRF),
6. les inconnus (UKN), qui correspondent aux catégories des taxonomies d'origine qui n'ont pas pu être retranscrites dans une des catégories précédentes.

Les trois corpus servant de référence à cette comparaison sont Switchboard (200 000 énoncés de conversations téléphoniques orientées tâche et annotés manuellement), MapTask (128 dialogues visant à reconstruire une carte en y plaçant différents points de repère) et Bugzilla (1 200 000 commentaires répartis sur 128 000 rapports de défauts dans le cadre du développement de la suite logicielle de la Fondation Mozilla).

Les taxonomies d'actes de dialogue à partir desquelles ces quatre corpus ont été étudiés sont différentes, et la comparaison faite après conversion dans une taxonomie commune en 5 actes n'est forcément qu'approximative, par conséquent cette comparaison doit être considérée avec précaution. Par ailleurs, en annotant les requêtes du corpus DAFT, certaines distinctions n'étaient pas toujours évidentes à réaliser : ainsi, la nuance entre *expressif* et *assertif* est souvent délicate, car la plupart des déclarations de l'utilisateur sur ce qu'il perçoit comme étant l'état objectif du monde (donc *assertifs* potentiels) sont en fait fortement subjectives (et sont donc en pratique considérées comme des *expressifs*). On a donc considéré comme étant un *expressif* toute proposition mettant en jeu la perception ou le ressenti de l'utilisateur, les *assertifs* formant une classe de déclarations ne mettant nullement en jeu des notions subjectives telles que "ennuyer", "aimer", "croire"... Et il faudrait idéalement reprendre l'annotation des autres corpus de référence pour s'assurer que des choix similaires y ont été effectués. Toutefois, une fois ces précautions prises, on observe assez nettement certaines caractéristiques distinguant le corpus DAFT des trois autres :

- une présence majoritaire (57%) de *directifs*, s'expliquant par un nombre élevé d'ordres directs ou de questions à

l'agent. Bien qu'orientés tâche, les autres corpus mettent en jeu uniquement des interlocuteurs humains, et il est vraisemblable que le fait de s'adresser à la machine (même via un agent) tend à rendre les requêtes plus directes car on suppose l'agent incapable des mêmes inférences qu'un être humain.

- un nombre assez faible d'*assertifs* (13%), dénotant le fait que l'utilisateur exprime bien plus son état d'esprit (29%) par rapport à des faits que ceux-ci de manière neutre et "objective" comme par exemple dans le corpus Bugzilla.
- quelques *promissifs* sont présents (1%) mais marginaux, ce qui s'explique par la nature de la relation entre l'utilisateur et l'agent assistant, où le second est de fait soumis au premier et non l'inverse.

Ces divergences entre des corpus portant sur le même thème (l'assistance dialogique à une tâche) sont intéressantes et justifient bien la nécessité de recueillir un corpus propre à notre champ applicatif. Une comparaison exhaustive de profils interactionnels avec des corpus de nature différente (par exemple en reprenant Multitag) serait également intéressante à réaliser, mais sortait du cadre de ce stage.

3.3 Catégorisation du corpus

3.3.1 Méthodologie

Nous avons dans un premier temps travaillé sur un sous-ensemble d'un dixième du corpus, soit environ 500 phrases tirées aléatoirement, afin d'annoter l'intégralité des phrases sélectionnées (ce qui n'était pas envisageable pour l'ensemble du corpus pour un unique annotateur disponible lors de ce stage).

Cette restriction peut évidemment entraîner la perte de phénomènes de langue peu représentés, mais une rapide comparaison de fréquence des verbes menée sur l'ensemble du corpus et sur ce sous-ensemble montre que cette réduction d'échelle n'affecte que très peu la distribution, confortant ainsi la méthodologie employée.

Assez rapidement, à la lecture d'extraits du corpus, deux catégories de requêtes (de **contrôle**, et d'**assistance**) émergeaient clairement et permettaient de classer environ la moitié des phrases. Bien que la distinction entre ces deux caté-

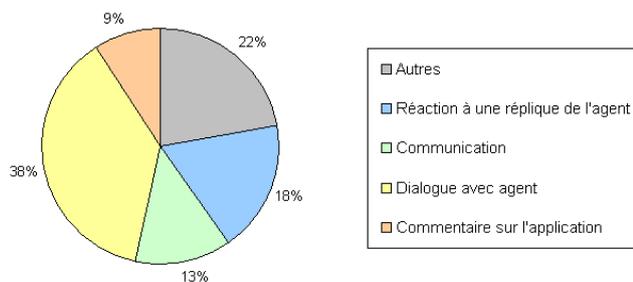


Figure 4 : Sous-catégories du sous-corpus de clavardage

gories soit évidente dans la majorité des cas, on peut noter quelques exceptions où il n'est pas trivial de comprendre si l'objectif de l'utilisateur est d'exécuter une action ou d'apprendre le moyen de la réaliser par lui-même. Considérons la requête suivante : « Je veux la documentation de l'application » : est-ce un ordre demandant d'afficher la documentation comme l'utilisateur aurait pu le faire en cliquant sur le bouton d'aide approprié (équivalent à « Aller à la documentation »), ou bien une demande implicite d'aide pour accéder à celle-ci (« Comment faire pour afficher la documentation ? »). De même dans le cas de « Je voudrais trier la liste » : s'agit-il de savoir comment faire pour la trier ou d'effectuer ce tri ?

Par ailleurs, les requêtes restantes à cette étape du travail furent regroupées dans une catégorie un peu bâtarde dite de **clavardage**, car elles étaient essentiellement axées sur des remarques faites à l'avatar représentant l'agent. Ce terme me semblait toutefois trop générique, et comme le nombre de phrases s'y retrouvant était important (un peu moins de la moitié), j'ai souhaité réaliser un nouveau découpage de cette catégorie pour mieux cerner les différents phénomènes de langue qu'elle regroupait. Le sous-ensemble de clavardage peut donc se diviser en 5 sous-catégories, dont les proportions se répartissent comme indiqué sous forme de secteurs dans la figure 4 :

1. **Communication** : il s'agit de fonctions communicatives dénotant des situations bien précise, au nombre de 4 :
 - **amorçage** - début de conversation : *“bonjour”, “hello, comment ça va”, “je m'appelle Jean-Paul”*...
 - **prise de congé** - fin de conversation : *“bye”, “ciao bella”, “bon je me casse”*...
 - **vérification de présence** - besoin de se rassurer sur le bon fonctionnement du système : *“Marco?”, “il y a quelqu'un?”, “tu dors ou quoi?”*...
 - fin d'aide - l'utilisation du programme continue, mais l'assistance est suspendue tant que l'agent n'est pas de nouveau sollicité : *“j'ai plus besoin de toi”, “je vais essayer de me débrouiller toute seule”, “vas t'en!”*...
2. **Dialogue avec agent** : ce sont des interactions de type “chat” où l'agent est considéré en tant qu'interlocuteur. On peut dénoter que les 5 types de comportements :
 - **ordre** - c'est le contrôle de l'agent : *“ça serait mieux si on se tutoyait”, “tais-toi!”, “ferme-la”*...

- **question** - c'est l'assistance concernant l'agent : *“es-tu prof?”, “as-tu peur des humains?”, “puis-je te faire confiance?”*...
- **compliment** - commentaire positif sur l'agent : *“tu es très drôle”, “j'adore ton accent”, “merci de m'avoir supporté si longtemps”*...
- **critique** - commentaire négatif sur l'agent : *“je hais tes remarques”, “bon à rien”, “t'es limite chiant”*¹⁶...
- **menace** : *“répète un peu ça pour voir!”, “ne me force pas à te tuer!”*...

3. **Réaction à une réplique de l'agent** : ces requêtes sont faites par l'utilisateur uniquement suite à une réplique de l'agent, et 5 types de réactions ont pu être recensées :

- **accord** - la réponse est comprise et acceptée : *“OK”, “ça va”, “merci”, “parfait!”*...
- **désaccord** - la réponse ne convient pas à l'utilisateur, et il en faudra une autre ou justifier celle-ci : *“non!”, “je proteste!”, “pas question”*...
- **incrédulité** - la réponse ne convainc pas, et l'utilisateur attend une confirmation et éventuellement une justification de celle-ci : *“ah bon?”, “ça m'étonnerait”, “avec toi on n'est jamais sûr de rien”*...
- **incompréhension** - la réponse est trop complexe ou sans lien apparent avec la requête initiale : *“j'ai bien peur de ne rien comprendre”, “je n'ai pas compris ton explication”*...
- **insistance** - la réponse ne convient pas et l'utilisateur ne voit pas d'alternative : *“réponds-moi stp”, “faut-y aller”*...

4. **Commentaire sur l'application** : *“la présentation est élégante”, “je n'aime pas du tout les tout petits boutons rouges”, “le site AMI est très bien dans l'ensemble mais certains détails mériteraient d'être revus”*...

5. **Autres** : ce sont des phrases difficilement classables car sans rapport avec le cadre de l'application ou incompréhensibles hors contexte : *“ça me stresse”, “c'est la dactylo”, “ou des blagues?”*...

Certaines requêtes constituant les commentaires sur l'application étaient humainement interprétables comme des demandes d'aide. L'assistance étant notre domaine d'étude, j'ai donc décidé de les faire figurer à un niveau supérieur en distinguant :

1. les requêtes d'**assistance directe**, telles que définies précédemment,
2. ces requêtes dites d'**assistance indirecte**, qui constituent donc un quatrième sous-ensemble principal du corpus

Notons que comme le sous-corpus de clavardage n'est pas l'objet essentiel de notre étude, maintenant que nous avons défini les sous-catégories qui le constituent, nous ne les détaillerons plus dans la suite de cette étude où il sera considéré dans son ensemble.

¹⁶Il s'agit de phrases réellement recueillies en contexte, et qui s'avèrent être souvent assez impolies.

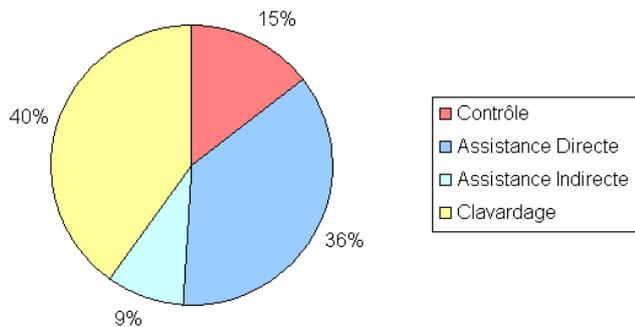


Figure 5 : Répartition du corpus en sous-corpus

3.3.2 Résultats et analyse

Le regroupement manuel de requêtes relevant d'activités similaires a finalement permis de distinguer 4 catégories de tailles inégales (cf fig. 5) qui constituent 4 "sous-corpus" correspondant à 4 types d'activités distinctes :

1. **activité de contrôle** : constitué de *commandes* à l'agent afin qu'il interagisse lui-même avec l'application pour les exécuter.
2. **activité d'assistance directe** : regroupant des *demandes d'aide* plus ou moins précises.
3. **activité d'assistance indirecte** : formé de *commentaires* sur l'application qui constituent des demandes d'aide sous-entendues, probablement perceptibles uniquement au niveau de la pragmatique.
4. **activité de clavardage** : réunissant le reste des interactions essentiellement centrées sur l'agent ainsi que les expressions métalinguistiques, phatiques¹⁷ et de backchanneling¹⁸.

L'existence du sous-corpus de contrôle montre bien qu'à partir du moment où l'utilisateur est face à un agent capable d'introspection sur le modèle de l'application, il s'attend également à ce que celui-ci soit également capable d'agir sur elle. Le sous-corpus de dialogue s'explique quant à lui par la présence visuelle d'un avatar représentant l'agent dans le monde virtuel personnalisant l'interface de dialogue, et immanquablement, un dialogue s'établit avec l'agent en tant qu'entité individuelle digne d'intérêt. Ce problème pourrait être en grande partie évité par l'utilisation d'agents conversationnels uniquement textuels, mais l'intérêt de celui-ci est d'une autre nature (cf. 1.2). Notre étude étant centrée sur les requêtes d'assistance, on pourrait considérer ces interactions avec l'agent comme parasitaires et les ignorer, car elles relèvent davantage du véritable dialogue homme-machine. Néanmoins, au regard de leur importance quantitative non négligeable (40%), nous avons décidé de les conserver, tout du moins dans une certaine mesure que nous préciserons ultérieurement.

Par ailleurs, dans la mesure où ce sont les requêtes d'assistance qui nous intéressent le plus et que nous ne disposions plus avec ce découpage d'un nombre suffisamment important de requêtes de ce type, la suite de l'étude (et notamment la conception du langage) s'est faite en menant la suite

¹⁷Pour maintenir le contact communicatif avec le locuteur

¹⁸Pour marquer son accord aux propos du locuteur et l'inciter à continuer

de l'étude en parallèle sur un deuxième sous-ensemble d'un dixième du corpus. Cette procédure nous a également permis de vérifier l'ensemble des mesures statistiques sur deux ensembles indépendants (en pratique, les résultats d'ordre statistique sur ces deux sous-ensembles se sont avérés très proches, validant ainsi notre hypothèse de travail sur des requêtes sélectionnées aléatoirement dans l'ensemble du corpus disponible).

3.4 Caractérisation des sous-corpus

Ces quatre sous-corpus, dont on peut voir quelques exemples dans la table 2, présentent clairement une difficulté croissante pour être mis sous forme de requêtes formelles¹⁹ :

1. les requêtes de contrôle sont essentiellement de nature prédicative (exemples des prédicats des phrases 1 et 2 : « glisser », « donner »).
2. les requêtes d'assistance directes et indirectes ont majoritairement deux ou trois modalités (exemples de modalités pour la phrase 6 : « avoir peur », « exister un moyen de »).
3. les requêtes de dialogue sont très diverses, allant des courtes (phrase 9) nécessitant des modalités bien à part pour les traiter, à des phrases longues (phrase 12) mettant en jeu une très grande variété de prédicats différents contrairement aux requêtes de contrôle (phrase 11 : « chanter »). Leur vocabulaire est également beaucoup plus varié. Cela signifie que la compréhension de ces requêtes par le système nécessitera une extension importante du langage à établir.

Les quatre sous-corpus que nous avons caractérisé l'ont été uniquement par le travail manuel d'annotation, mais il serait bien évidemment souhaitable de pouvoir automatiser une telle opération de classification. En effet, elle pourrait constituer une étape de prétraitement de la requête pour pouvoir avoir la possibilité d'analyser par la suite de manière différente des phrases recouvrant les activités propres à chaque sous-corpus. Nous nous sommes notamment inspirés pour ce travail des méthodes suggérées par Ripoche [23].

3.4.1 Longueur des requêtes langagières

Il apparaît qu'en moyenne les requêtes de contrôle sont globalement assez courtes (5,4 mots/phrased), les requêtes d'assistance (directe et indirecte) dans l'ensemble plus longues (9 mots/phrased) tandis que les requêtes de clavardage ont une longueur intermédiaire (6 mots/phrased) (cf. table 3). Néanmoins, comme le montre la figure 6, bien que leur répartition pourrait être approximée par une loi normale, elle est très dispersée puisqu'elles ont toutes un écart type de l'ordre de 3,5.

La longueur des phrases ne constitue donc pas à elle seule un paramètre de discrimination suffisant entre les sous-corpus.

3.4.2 Profils interactionnels

Un autre paramètre discriminant éventuel pourrait être les profils interactionnels, définis en 3.2.2. On distingue en effet certaines différences de profils interactionnels assez nettes entre les sous-corpus et par rapport au profil générique du

¹⁹Les notions de prédicats et de modalités, uniquement illustrées ici par des exemples, seront plus amplement définies dans la section suivante.

Table 2 : Quelques exemples de requêtes du corpus DAFT

N°	Sous-corpus	Phrase
1	Contrôle	<i>glisser le disque de droite à gauche</i>
		Bouger(objet="le disque",origine="droite",destination="gauche")
2	Contrôle	<i>Donne moi la liste des projets</i>
		Montrer(objet="la liste des projets")
3	Assist. dir.	<i>c quoi le GT ACA</i>
		ASK(INFOS(about="le GT ACA"))
4	Assist. dir.	<i>comment faire pour gagner ?</i>
		ASK(WAY(goal=Gagner()))
5	Assist. dir.	<i>d'après toi, y a t-il des fonctions d'annulation dans cette application ?</i>
		ASK(KNOWLEDGE(of=s,about=EXISTENCE(time=2,of=FUNCTION(doing=Annuler(),in="cette appli..."))))
6	Assist. dir.	<i>j'ai peur qu'il n'y ait pas moyen de changer la taille de la police qui est bien trop petite</i>
		FEAR(agent=u,fear=NEG(POSSIBILITY(todo=Modifier(objet="la police...trop petite",propriété="taille"))))
7	Assist. ind.	<i>le bouton "fermer" est semblable au bouton "quitter"</i>
		NEG(DIFFERENCE(between="le bouton 'fermer'",and="le bouton 'quitter'"))
8	Assist. ind.	<i>probablement que le bouton NEXT ne marche pas correctement</i>
		PROBABILITY(degree=2,of=PROBLEM(with="le bouton NEXT"))
9	Clavardage	<i>Absolument</i>
		ACK()
10	Clavardage	<i>bonjour je suis Arthur</i>
		HELLO(),VALUE(object=u,property="nom",value="Arthur")
11	Clavardage	<i>sais-tu chanter ?</i>
		- non traitée -
12	Clavardage	<i>Ce que je préfère chez toi c'est tes beaux cheveux roux</i>
		- non traitée -

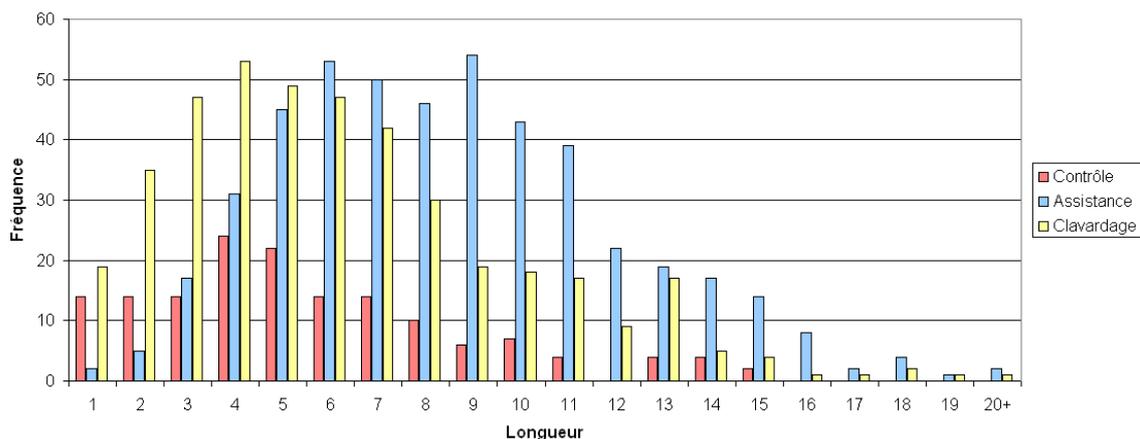


Figure 6 : Répartition des phrases par longueur dans les 4 sous-corpus

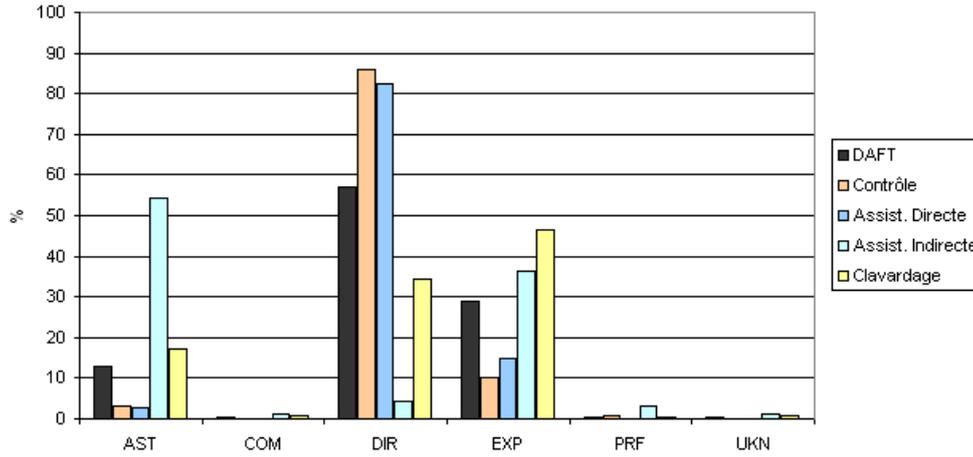


Figure 7 : Répartition des actes de dialogue searliens dans les différents sous-corpus

Table 3 : Longueur moyenne des phrases des 4 sous-corpus

Sous-corpus	Longueur moyenne	Ecart-type
Contrôle	5,4	3,36
Assist. dir.	8,0	3,54
Assist. ind.	9,9	3,30
Clavardage	6,0	3,62

corpus DAFT (rappelé en gris foncé), notamment pour distinguer l’assistance directe (avec une forte majorité de directives et quelques expressifs) de l’assistance indirecte (une majorité d’assertifs et des expressifs). En revanche, les profils interactionnels des sous-corpus de contrôle et d’assistance directe sont assez similaires. Cette méthode présente donc un certain intérêt mais ne peut-être utilisée de manière unique. De plus, d’un point de vue pratique, l’automatisation de détection de ces actes de dialogue n’est pas non plus triviale.

3.4.3 Conclusion

Ni l’étude de la longueur des phrases, ni celle de leurs profils interactionnels ne sont suffisantes pour identifier de manière automatique l’appartenance d’une phrase à une activité particulière. Il faudra donc probablement envisager d’autres paramètres et trouver la meilleure combinaison de ceux-ci si l’on souhaite disposer d’un classifieur automatique. D’autre part, nous verrons par la suite (cf section 5.3) qu’il existe une méthode plus efficace basée sur l’étude des requêtes du langage élaboré, mais son utilisation ne peut être envisageable que pour une classification a posteriori des requêtes.

4. ÉLABORATION DU MODÈLE DE LANGAGE DE REQUÊTES

4.1 Modèle naïf

En l’absence de corpus de requêtes au début du projet, une première version du langage DAFT basée sur un petit ensemble de 300 phrases avait été proposé par Karl Le Guern [17]. Dans cette version, l’analyse des requêtes était réalisée selon un filtre en 3 couches distinctes et uniques :

- **Modalités** (notées M) : ce sont des actes de dialogue, au sens large et non searlien [25], et sont inspirées des catégories de l’inventaire réalisé par Molinsky et Bliss [20].
- **Prédicats** (notés P) : il s’agit de verbes d’action (ex : modifier, déplacer...) ou de prédication (ex : être prêt...).
- **Références** (notées R_i) : elles correspondent généralement aux groupes nominaux des requêtes et sont à prendre au sens de références extensionnelles associatives²⁰. Dans la suite, toute mention faite aux références le sera dans ce sens là.

L’analyse sémantique des phrases était donc fourni sous la forme : $M [P (R_1, R_2, \dots)]$

Néanmoins, cette approche montre ses limites sur des requêtes plus complexes comme la n° 6 de la table 2. Ainsi, on se rend rapidement compte qu’une unique modalité n’est généralement pas suffisante pour traiter les requêtes réellement saisies par l’utilisateur et qu’un modèle plus élaboré, dans lequel plusieurs modalités sont capables de s’articuler entre elles, s’avère effectivement nécessaire.

4.2 Modèle multicouche hiérarchisé

4.2.1 Syntaxe générale

Dans ce nouveau modèle (nommé DAFT 2.0 pour le distinguer de la version naïve), je considère la possibilité pour les modalités de s’imbriquer entre elles selon certaines règles afin d’augmenter l’expressivité du langage. Avec cette structuration “en pelures d’oignon”, on peut avoir plusieurs **modalités** imbriquées dans un ordre précis, contenant un niveau unique de **prédicats** qui peuvent être coordonnés entre eux et ayant chacun une ou plusieurs **références** correspondantes.

Les requêtes DAFT 2.0 s’écrivent alors sous la forme :

$$M_1(\dots M_n(c_1 = P_1(c'_1 = R_1, \dots, c'_l = R_l), \dots, c_m = P_m(\dots)) \dots)$$

Expression dans laquelle :

$M_1 - M_n$ sont des **modalités**,

$P_1 - P_m$ sont des **prédicats**,

²⁰Le traitement de la référence extensionnelle associative a fait l’objet de deux thèses récentes rien qu’au sein du groupe AMI : [12] et [22].

$c_1 - c_m$ & $c'_1 - c'_l$ sont les intitulés des champs **typés** tels que décrits dans les schémas des modalités et des prédicats, $R_1 - R_l$ sont des **références** issues de la requête langagière.

Et grâce à cette nouvelle syntaxe, la phrase 6, qui n'était pas traitable par l'approche précédente en trois couches, peut désormais s'écrire comme indiqué sur la ligne en-dessous d'elle dans la table 2.

Les modalités et les prédicats sont définis en fonction de **schémas**, qui se basent essentiellement sur leur structure actantielle [26] à la manière de ce que propose FrameNet²¹. Ces structures sont décrites en détail dans les spécifications du langage qui forment un document d'une quarantaine de pages²². Ce document sera résumé ici en une vue d'ensemble présentant les intitulés des schémas définis, leur signification et des exemples de fragments de requêtes issues du corpus qui y correspondent. Par ailleurs, nous examinerons plus précisément quelques-uns de ces schémas afin d'illustrer leur fonctionnement et de justifier une partie des choix de conception effectués.

4.2.2 Les types de champs

Tous les champs de modalités et de prédicats sont typés, c'est-à-dire qu'ils ne peuvent être remplis que par une référence ayant un type compatible. L'utilisation de types présente clairement des avantages non négligeables :

- augmenter la finesse des schémas en effectuant un filtrage automatique des références capables de remplir les champs du schéma détecté.
- faire un choix entre plusieurs schémas possibles si les types des champs ne correspondent pas aux types des références de la requête langagière.

Néanmoins, ceci suppose évidemment que lors de l'étape d'analyse des requêtes langagières on soit capable de déterminer le(s) type(s) des références ; l'utilisation de listes des occurrences de chaque type dans le corpus devrait dans un premier temps suffire à typer la majorité des références, mais une heuristique plus élaborée sera sans doute nécessaire par la suite.

Il faut également noter que la plupart des modalités disposent aussi d'un type global (généralement **a** ou **o** - cf. table 4) pour la signification) qui sert pour les imbrications d'un schéma dans l'autre. Prenons l'exemple de la phrase "y a-t-il un minimum pour la vitesse ?" qui s'écrit en DAFT 2.0 `ASK(EXISTENCE(time=2, of=VALUE(object="vitesse", asked="min")))`

`EXISTENCE` dispose d'un champ `of` qui attend un élément de type **o**, par conséquent, le type global de `VALUE` est **o**. Précisons toutefois qu'il a bien évidemment fallu se baser sur plus d'un exemple pour vérifier cette hypothèse.

Les types ont été définis au fur et à mesure de la retranscription des phrases en fonction de la nature des références, et leur nombre s'est assez rapidement stabilisé à 11, tels qu'ils sont définis dans la table 4. Les exemples donnés dans cette table le sont à titre d'illustration du type uniquement, mais

²¹Ressources lexicales en langue anglaise pour une analyse de phrases selon des "frames"

²²Document disponible à l'adresse :

[http://fbouchet.vorty.net/doc/Spécifications DAFT 2.0.pdf](http://fbouchet.vorty.net/doc/Spécifications_DAFT_2.0.pdf)

il n'y a pas une équivalence systématique entre un mot et un type : le contexte compte. Par exemple, le mot "couleur" peut être un type ou une propriété :

"quelles sont les couleurs disponibles ?" s'écrit en DAFT : `ASK(OBJECT(type="couleur", property="disponible", multiple=1))`

"existe-t-il un truc de couleur rouge ?" s'écrit en DAFT : `ASK(EXISTENCE(time=2, of=OBJECT(type="truc", property="couleur", value="rouge")))`

Il faudra alors être capable de choisir en fonction du contexte, éventuellement à l'aide d'informations d'ordre grammaticale fournies par GRASP : dans cet exemple, l'utilisation de "couleur" en tant que nom fait référence à un type, tandis que dans le second cas, "de couleur rouge" se comporte comme un qualificatif de "truc".

4.2.3 Les modalités M_i

Comme le montre les tables 5 et 6, les modalités définies sont au nombre de 39, et elles peuvent être regroupées en fonction de leur *structure* ou de leur *signification* en 8 grandes catégories. Soulignons toutefois que les modalités ont été créées au fur et à mesure de l'annotation, et les 8 catégories n'ont été qu'un moyen d'organisation a posteriori.

Ces modalités se distinguent les unes des autres selon plusieurs critères :

- Leur profondeur dans la liste formée par la requête DAFT : sans détailler les règles d'imbrication, on peut noter que les modalités de sentiments se situent globalement plutôt dans les couches extérieures des requêtes (ex : phrase 6), tandis que les modalités d'informations sont plutôt sur les couches intérieures précédant les prédicats (ex : phrase 5).
- La complexité des schémas : les structures des schémas sont de complexité inégale ; les marqueurs (cf. table 6) ne prennent qu'un unique argument, tandis que des schémas plus complexes comme `OBLIGATION(of=[p], by=[p], todo=[a])` peuvent en prendre jusqu'à 4 ou 5, même si en pratique tous les champs du schéma ne sont que rarement remplis simultanément.
- La complexité d'identification du phénomène sémantique représenté : les modalités sont de subtilité diverse, et si une simple liste de mots-clé pourrait suffire à en détecter certaines comme `ROLE`, d'autres comme `PLACE` ou `MOMENT` peuvent parfois être implicites dans les requêtes langagières : "je ne trouve pas X", sous-entendu "l'endroit où est X", "je veux que le compteur stoppe à 1000", autrement dit "quand il vaut 1000".
- Leur fréquence : comme souvent dans l'étude de phénomènes linguistiques, on peut constater que les modalités ne sont pas représentées de manière égale dans le corpus, et que leur distribution dans les requêtes transcrites manuellement suit une loi de Zipf (cf fig. 8).
- Leur généricité : si les modalités circonstancielles ou les modalités de sentiments sont plutôt classiques dans ce type d'approche et relativement génériques puisqu'on les retrouve dans la plupart des systèmes de dialogue homme-

Table 5 : Liste des 39 modalités M_i DAFT 2.0

Informations sur une entité	
INFOS	(Avoir) des informations/plus d'informations sur un sujet. <i>Qu'est-ce que (c'est) o, Parle moi un peu de o, Plus d'infos / Plus d'explications sur o</i>
MEANING	Le sens de quelque chose, d'après quelqu'un. <i>p vouloir dire o, o correspondre à, signifier o</i>
TYPE	Le type d'un objet est tel type. <i>o être t, Qu'est-ce que (c'est) o</i>
VALUE	La valeur (de la propriété d'un objet / d'un objet) est valeur. <i>P être activable/visible, v être le nom de o, La valeur de o/P</i>
PROPERTY	La propriété 'propriété' d'un objet vaut une valeur. <i>P de o</i>
FUNCTION	La fonction faisant une action déterminée pour des objets donnés dans un cadre particulier, et ayant certaines propriétés. <i>fonctions de a, De fonctions P, Une fonction/commande a pour o dans l</i>
ROLE	Le rôle d'un objet / concept / personne par rapport à un référentiel. <i>A quoi correspondre/servir o, But de o, Rôle de o par rapport à o</i>
LIMIT	Limites dans un objet / des propriétés d'un objet. <i>o être limité, Y avoir (une limite / des limitations) (à / de) o</i>
FUNCTIONING	Le fonctionnement d'un objet. <i>Comment marcher/fonctionner o, Le fonctionnement/comportement de o, Ce qui se passe avec / durant c</i>
Identification d'entités	
EXISTENCE	Existence de quelque chose dans un endroit. <i>Il (n')y a (pas) / (n')y a-t-il o, Est-ce que l avoir/posséder o, Existe-t-il o</i>
OTHER	Un objet / Une action autre que celui / celle considéré(e). <i>autre chose que a/o</i>
OBJECT	Un objet donné ayant un type, des propriétés et situé en un lieu donné. Il peut servir à effectuer une action ou être objet de celle-ci.
Relations inter-entités	
DIFFERENCE	Différences entre une entité et une autre selon un critère. <i>o1 avoir le même P que o2, o1 et o2 sont différents/liés/identiques, Ce qui distingue o1 et o2 c'est, revenir au même a1 que a2</i>
ORDER	Réalisation d'un ensemble d'actions étape par étape, la dernière d'entre elles dans l'ordre d'énumération étant considérée comme le but de la procédure. <i>De a1 puis de a2, a1 d'abord avant de/pour a2</i>
Capacités, droits et devoirs	
POSSIBILITY	Possibilités offertes à quelqu'un par une autorité (de réaliser quelque chose / au sujet d'un objet) pour quelqu'un. <i>que pouvoir p1 pour p2, être possible/impossible/dangereux/permis/autorisé/capable/à même de a</i>
KNOWLEDGE	(La connaissance / Le savoir / La compréhension / La capacité) de quelqu'un ou quelque chose (pour faire une action / au sujet d'un objet), avec un degré de certitude. <i>p avoir un tuyau pour a, p avoir connaissance de o, p ne pas être sûr de o</i>
OBLIGATION	Obligation d'une personne par une autre à faire quelque chose / d'un objet. <i>p1 être forcé/obligé (par p2) de a, p obliger à a, p interdire de a</i>
WILL	Volonté de quelqu'un qu'une action soit réalisée par une personne / d'un objet. <i>p vouloir a, p attendre de p que a, laisser p a</i>
PROBABILITY	Probabilité d'une action / d'un objet à un certain degré. <i>Se pouvoir que a, être fort probable que a, ne faire aucun doute que a</i>

Table 6 : Liste des 39 modalités M_i DAFT 2.0 - suite

Circonstancielles	
WAY	Moyen d'atteindre un but. <i>Comment faire (pour) a, La meilleure solution pour a, Un moyen de a, De quelle façon a</i>
EFFECT	L'effet d'une action / d'un objet est une autre action. <i>Ce qui arrive/se passe si/quand a, Sur quoi débouche a, a1 si/quand a2, a1 provoquer a2</i>
REASON	La raison/cause de quelque chose. <i>pourquoi a, le fait que a, car a</i>
PLACE	Emplacement de quelque chose / où faire quelque chose. <i>Où pouvoir être / trouver o, ne pas trouver o</i>
MOMENT	Instant où une action se fait / où un événement a lieu. <i>quand a, lorsque a, à l'instant où a</i>
NUMBER	Nombre d'éléments dans un espace donné. <i>nombre de o, combien de o</i>
Sentiments	
LIKE	Un agent aime un objet aimé (quelqu'un ou quelque chose). <i>p aimer/préférer o/c, o être moche/ridicule/nul</i>
FEAR	Un agent a peur d'une crainte. <i>p avoir (bien) peur (de/que) o/a</i>
BOTHER	Le gêné est incommodé par une gêne. <i>p être ennuyé/dérangé (de/que/par) o/a</i>
DOUBT	Un doute d'une personne au sujet de quelque chose (absence de certitude). <i>p avoir un/des doute(s) au sujet de o, o étonner p, Ah bon ?</i>
SURPRISE	Un agent est surpris par quelque chose. <i>Être bizarre/surprenant/étonnant/étrange o, p être (vraiment) surpris / étonné que o</i>
REGRET	Un agent a des regrets au sujet de quelque chose. <i>p regretter de o, Être dommage/embêtant que o</i>
HAPPY	Un agent est heureux au sujet de quelque chose. <i>p être ravi de a, p se désoler que o, p être déçu que o</i>
Assistance	
PROBLEM	Situation non satisfaisante, quelque chose ne se passe pas comme attendu. Problème d'une personne avec un objet ou pour faire une action. <i>p avoir du mal à a, problème de o, o planter, o être en panne/fou</i>
MISTAKE	Acte involontaire/inadapté. Un responsable commet une erreur sur une action ou un objet. <i>a sans le faire exprès, p se tromper de o, p avoir fait une bêtise</i>
HELP	Un demandeur réclame de l'aide à un assistant, (pour effectuer une action / au sujet de quelque chose), en faisant quelque chose (éventuellement). <i>p1 aider p2 à a, p1 conseiller p2 pour a, ça aiderait p1 que p2 faire a</i>
TELL	Un locuteur s'adresse à un interlocuteur pour lui parler d'un sujet ou lui exposer une action à réaliser. <i>p1 expliquer (à) p2, p1 dire à p2 de a, p1 décrire o à p2</i>
Marqueurs	
CHECK	Marqueur de demande de vérification d'une information. <i>...c'est ça ?, est-on bien sûr que... , o ?, o être bien de type t ?</i>
ASK	Marqueur d'interrogation, pour toute interrogation qui n'est pas un CHECK. <i>Quel est le type de o ?</i>
NEG	Marqueur de négation d'une action ou d'une modalité : il peut potentiellement précéder n'importe quel schéma et peut donc se trouver à n'importe quelle couche dans l'imbrication. Sa position dans la requête formelle est à prendre en compte pour en retrouver le sens.

Table 7 : Liste des 52 actions DAFT 2.0

Actionner	Actionner un objet
Activer	Activer un objet
Adhérer	Une personne devient membre d'un objet
Ajouter	Ajouter un objet à un endroit d'une certaine manière (ex : "sans risque")
Aller	Aller en un lieu à un instant donné (ex : "tout de suite") et d'une manière donnée (ex : "rapidement") OU aller à un instant antérieur (ex : "le coup d'avant")
Annuler	Annuler une action antérieure
Arrêter	[Arrêter/Continuer] [de faire une action/un objet] à l'instant où quelque chose, pour une durée définie
Bipper	Émettre un son
Bouger	Bouger un objet ou une personne d'un endroit à un autre (ou au sein d'un endroit) d'une certaine manière
Cacher	Cacher un objet
Changer-de	Changer d'objet
Cliquer	Une personne clique sur un objet
Compter	Compter d'une certaine manière
Contacter	Contacter une personne
Contrôler	Contrôler un objet / les propriétés d'un objet
Coup	Faire un coup d'un endroit à un autre
Créer	Créer un objet
Démarrer	Démarrer un objet
Dépasser	Un objet / une propriété d'un objet dépasse une certaine valeur
Donner	Un donneur donne un objet à un receveur
Enregistrer	Enregistrer un objet à un endroit
Etre-perdu	Une personne est perdue
Etre-prêt	Une personne est prête
Etre-utile	Un objet est utile
Faire-défiler	Faire défiler un objet
Faire-partie-de	Un objet fait partie d'un contenant
Fermer	Fermer un objet
Finir	Atteindre la fin, sous-entendu en réitérant une action précédente
Gagner	Gagner d'une certaine manière (ex : "facilement")
Jouer	Le joueur joue quelque chose (ex : "le prochain coup") pour une personne, à un instant t, d'une certaine manière (ex : "avec plus de disques")
Manipuler	Le manipulateur manipule un objet d'une certaine manière
Mettre-à-jour	Mettre un jour un objet par une personne
Modifier	Modifier un objet ou une propriété, d'une certaine manière ou pour atteindre une valeur donnée
Montrer	Une personne montre quelque chose à quelqu'un d'une certaine manière
Quitter	Quitter un objet (sous-entendu l'application/le programme)
Recommander	Recommander un objet
Recommencer	Recommencer un concept (ou une action?) (ex : "rejouer une nouvelle partie")
Redémarrer	Redémarrer un objet ou une personne (l'agent)
Rejouer	Rejouer un concept (ex : "une nouvelle partie")
Restaurer	Restaurer un objet à une valeur
Réversible	Réversibilité d'une action / d'un concept (ex : "les actions sur les tables")
Se-abonner	S'abonner à un objet
Se-charger-de	Un responsable se charge d'une action
Se-passer	Ce qui se passe/s'est passé à un instant donné
Supprimer	Supprimer un objet d'une certaine manière
Télécharger	Télécharger un objet
Tricher	Tricher d'une certaine manière (ex : "un tout petit peu")
Trier	Trier un objet d'une manière (ex : "par bureau, en sens inverse")
Utiliser	Utiliser un objet de l'appli (ex : "Coco") ou de contrôle (ex : "la souris")
Vérifier	Vérifier un objet
Voir	Une personne voit quelque chose
*	Action générique (ex : "refaire quelque chose")

4.2.7 Compromis “subtilité-simplicité”

Bien que le langage ainsi défini soit relativement expressif, il a fallu à plusieurs reprises abandonner certaines nuances linguistiques non essentielles pour l’obtention d’une réponse adéquate, car il ne faut en effet pas perdre de vue que l’analyse est réalisée dans un contexte de réalisation de tâches. C’est un problème qui s’est posé tant au niveau des prédicats que des modalités.

Ainsi, dans le cas des prédicats, on peut se poser la question de l’utilité de distinguer les actions d’augmentation et d’incrémentation (qui est un cas particulier de l’augmentation). Et si l’on choisit de les unir comme nous l’avons fait, vient alors le cas de l’unification d’augmentation/incrémentation (et leur contraire) au prédicat plus général `Modifier()`, en rajoutant une option catégorisant la nature de la modification effectuée. Néanmoins, même en unifiant le plus souvent possible, cela n’a pas empêché pas un nombre assez élevé de prédicats, principalement justifié par des actions “exotiques” n’apparaissant qu’une ou deux fois dans le corpus, mais tellement spécifiques qu’elles ne peuvent être unifiées à un autre prédicat, par exemple “s’abonner” ou “télécharger”. Ce phénomène apparaissait bien sur la courbe de répartition (cf fig. 9).

En ce qui concerne les modalités, on peut citer l’exemple de `POSSIBILITY`, qui recouvre à la fois les notions de capacité, de permission et de possibilité - notions pourtant bien distinctes d’un point de vue sémantique. Ce choix se justifie cependant dans ce contexte particulier, comme le montre l’exemple d’une requête du type « peux-tu modifier le tableau ? ». Une telle question suppose que l’utilisateur désire savoir à la fois :

1. si cette action est *prévue et implémentée* dans le composant,
2. si l’agent a les *capacités* et les connaissances nécessaires pour réaliser cette action et que le composant lui est accessible,
3. si l’utilisateur est *autorisé* à réaliser cette action lui-même,
4. s’il n’y a pas de conditions de garde liée à l’application empêchant l’action d’être réalisée à cet instant.

En effet, selon les maximes de quantités de Grice [13], la réponse de l’agent doit être “aussi informative que requis par le contexte”, or la probabilité est forte qu’en cas de réponse affirmative à cette question, l’utilisateur demande immédiatement après de la réaliser, et une réponse de type « Je ne sais pas le faire » serait très certainement frustrante pour lui, d’où la nécessité d’analyser les requêtes de ce type selon ces quatre significations simultanément, plutôt que séparément. De la même manière, il n’existe pas de modalité `DISLIKE`, pourtant sémantiquement différente de `NEG(LIKE)` (car on peut ne pas aimer une chose sans pour autant le détester), mais dans le cadre d’une interaction avec un agent assistant, c’est l’aspect pragmatique qui prime : si l’utilisateur prend la peine de dire qu’il “n’aime pas” une chose, c’est pour qu’une action soit réalisée (changer cette chose ou tout au moins stocker cette information) et dès lors la nuance sémantique est caduque.

De manière générale, on propose donc un système “sous-déterminé” qui généralement unit au maximum les actions et

Table 8 : Taux de couverture des 4 sous-corpus par le langage DAFT 2.0

Sous-corpus	Ctrl	Ass. dir.	Ass. ind.	Clav.*
Couverture	92,3%	96,7%	70,2%	82,35%

*(hors dialogue)

les modalités, tout en gardant intérieurement (dans certains champs) les nuances. Ce sera le rôle de l’agent rationnel que d’interpréter (lorsqu’il en est capable) ces nuances en fonction du contexte qu’il identifiera.

5. VALIDATION

5.1 Couverture du corpus par le langage

Le traitement du corpus s’est donc fait en deux sous-ensembles au 1/10^e, soit un total de 1075 phrases traitées. Parmi elles, 698 ont été transcrites en requêtes respectant la syntaxe définie au sein du langage DAFT 2.0. Cette couverture, bien que globalement très bonne, est cependant inégale (cf table 8), et une partie du sous-corpus de clavardage concernant le dialogue avec l’agent (202 phrases) qui ne relève pas directement de l’assistance n’a pas été prise en compte²³.

5.2 Robustesse du langage

Il est important de vérifier que le langage que nous avons défini est suffisamment robuste, c’est-à-dire que l’ajout de nouvelles requêtes d’assistance dans le corpus n’entraînera ni une remise en question de la syntaxe adoptée, ni un ajout de nombreux prédicats supplémentaires pour pouvoir les modéliser. Pour vérifier ceci, nous avons pris une nouvelle tranche de 1/10^e du corpus DAFT que nous avons partiellement retranscrite en DAFT 2.0 (uniquement sur les sous-corpus de contrôle et d’assistance). On a ainsi pu estimer qu’il faudrait éventuellement ajouter 2 modalités de quantification (`TOO`, `ENOUGH`) et 4 prédicats d’action (`Bloquer`, `Communiquer`, `Choisir`, `Agir`) pour retranscrire ces phrases. Il s’agit donc d’ajouts relativement minimes (de l’ordre de 5%) pour une augmentation de 50% de la taille du corpus, et surtout, la structure même de la syntaxe proposée n’a pas été mise en défaut par ces nouvelles phrases. Cela ne signifie évidemment pas que toutes les requêtes seront aisément retranscrites en DAFT 2.0, puisque les taux de couverture ne sont pas à 100%, mais cela confirme au moins que le travail sur deux sous-ensembles au 1/10^e du corpus n’a pas biaisé la conception du langage.

5.3 Caractérisation des sous-corpus par étude de leurs requêtes DAFT

La caractérisation des sous-corpus a priori (cf section 3.4) n’avait pas permis de trouver de paramètres suffisamment significatifs pour permettre un traitement a priori capable de déterminer l’appartenance d’une phrase à l’un des sous-corpus. Maintenant que le langage DAFT 2.0 a été défini, on peut tenter une nouvelle caractérisation basée sur l’étude de la proportion de schémas de prédicats et de modalités (cf fig. 10). On constate cette fois une très nette différence entre les sous-corpus :

²³Cela ne veut pas dire pour autant qu’elles ne peuvent être mises sous forme de requêtes DAFT : leur structure est souvent proche des requêtes d’assistance mais il faudrait étendre le vocabulaire des prédicats.

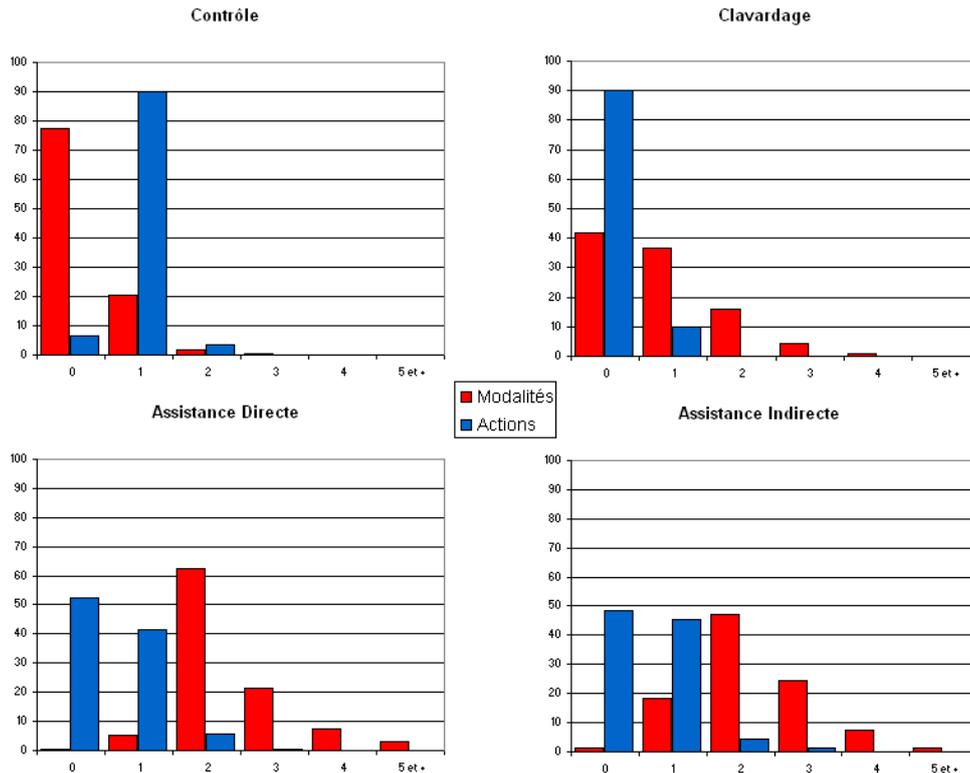


Figure 10 : Nombre de schémas par phrase dans les 4 sous-corpus

- le contrôle se caractérise par un nombre élevé de prédicats par phrase (0.97) et peu de modalités (0.22).
- l'assistance directe contient beaucoup de modalités (2.41) et un nombre moyen de prédicats (0.54).
- l'assistance indirecte diffère peu de l'assistance directe avec 0.59 prédicats par phrase et légèrement moins de modalités (2.22), avec toutefois une part plus forte de phrases à modalité unique (18% contre 5%).
- le clavardage (réduit aux phatiques) est extrêmement pauvre en prédicats (0.08) et en revanche très riche en fonctions communicatives (62% des phrases traitées en ont au moins une).

Cela confirme les intuitions que l'on pouvait avoir de part la manière dont on a distingué manuellement ces 4 sous-corpus, mais il n'y a probablement pas moyen d'exploiter ces résultats d'un point de vue pratique dans l'application.

5.4 Intégration au système existant

5.4.1 En amont : clés sémantiques de GRASP et alignement d'ontologies

L'objectif étant de faire générer par le module GRASP actuel des requêtes en DAFT 2.0, j'ai été amené à vérifier les possibilités d'intégration à celui-ci, en particulier en comparant les prédicats d'action avec les clés sémantiques actuellement attachées aux mots ou expressions lors de l'analyse. Il s'agissait donc de chercher à trouver les classes équivalentes entre les deux ontologies sémantiques.

GRASP définit des clés sémantiques, c'est-à-dire des éléments de sens principaux, à partir desquels on peut en dé-

river d'autres par antonymie, méronymie... Ces clés sémantiques définies à partir de tous les mots et expressions présents dans le corpus ont été regroupées par Jean-Paul Sansonnet en 308 classes sémantiques distinctes, formant 24 thèmes: *Determinants, Pronouns, Grammar, Meta Attributes, Slot Attributes, Relative Attributes, Mental States, Mental Process, Dialog Acts, Actions Control, Basic Actions, Event Control, Process Control, Process Attributes, Time Attributes & Domains, Space Positions, Ordinality, Quantity, Data Types & Structures, Widgets, Things, Language, Concepts, Ad Hocs.*

Ces clés regroupent sous forme de lemmes à la fois : le mot racine (R), l'action liée (A), la propriété liée (P), et leurs antonymes (préfixés par -). Ainsi, la clé `VISIBILITY` est définie comme :

```
VISIBILITY, ISA[Relative Attribute] {
  R["visibilite"],
  A["voir", "see"],
  P["visible", "perceptible", "discernable"],
  -A["cacher", "disparaitre"],
  -P["cacher", "invisible", "illisible"]
}
```

Bien que les deux ontologies que nous avons définies sont basées sur une étude du même corpus, les deux classifications ne peuvent pas être alignées simplement l'une avec l'autre : certaines actions sont séparées dans l'ontologie de DAFT 2.0 et regroupées dans les clés sémantiques de GRASP, et inversement. Certaines actions s'alignent parfaitement, par

exemple `Faire-partie-de()` s'aligne avec l'intégralité du contenu de la clé sémantique `CONTENT` :

```
CONTENT, ISA[Meta Attribute] {
  R["contenu"],
  P["appartenir", LOC[VV, "appartenir", "faire",
"partie", "de"]],
  DOM["partie", "bloc", "element", "membre",
"participer"],
  A["contenir", "comporter"]
}
```

Dans d'autres cas, l'alignement d'une action DAFT ne se fait qu'avec certaines parties d'une clé sémantique GRASP. C'est le cas de `Cacher()`, qui s'aligne avec les sous-clés `-A` et `-P` de `VISIBILITY` (présentée précédemment).

Enfin, dans plusieurs cas il y a une superposition d'actions DAFT au sein d'une même clé GRASP qu'il faudra alors envisager d'éclater. Ainsi, la clé `HELP` contient des mots rattachés à la modalité `AIDER` ainsi qu'à l'action `Recommander()` :

```
HELP, ISA[Dialog Act] {
  R["aide", "assistance", "conseil",
"recommandation", "indication", "renseignement",
"service", "faveur", "renfort", "soutien",
"suggestion", "support"],
  A["help", "aider", "cooperer", "assister",
"conseiller", "preconiser", "suggerer", "proposer",
"recommander"],
  AN["conseilleur", "assistant"]
}
```

Bien que basées sur le même corpus, l'ontologie des actions et modalités DAFT 2.0 et celle des clés sémantiques de GRASP ne sont pas compatibles, et un travail d'adaptation sera nécessaire des deux côtés lorsqu'on intégrera le langage défini lors du stage à la chaîne de traitement globale.

5.4.2 En aval : lien avec des heuristiques

Le langage DAFT 2.0 devra également être fortement lié aux heuristiques de l'agent rationnel permettant d'interpréter les requêtes, l'objectif étant d'avoir pour chaque heuristique d'interprétation une correspondance bijective avec un ensemble de schémas de modalités et de prédicats. La définition de ces heuristiques restant à réaliser, elles n'ont pas pu être prises en compte lors de la création du langage, cependant, l'analyse très pragmatique qui est actuellement faite des requêtes nous laisse espérer qu'il n'y aura probablement pas besoin de modifications trop profondes du langage et qu'il s'agira principalement de définir des classes d'équivalence entre requêtes.

Par exemple, `NEG(KNOWLEDGE(...))` et `WILL(...)` ayant en contexte un sens proche, seront éventuellement unifiés au sein d'une même classe : si l'utilisateur dit à l'agent qu'il ne sait pas (faire) quelque chose, c'est vraisemblablement afin que celui-ci lui explique ou le fasse à sa place. Par conséquent, une heuristique simpliste en pseudo-langage pourrait être de la forme :

```
If (NEG(KNOWLEDGE(x)) || WILL(x)) {
  If (x is Action) Then Do(x)
  Else Tell(x.value)
}
```

6. CONCLUSION

6.1 Bilan

A travers une étude approfondie du corpus DAFT, j'ai pu montrer que :

- la nature particulière de l'interaction homme-machine permettait de le distinguer d'autres corpus de dialogue de tâches et généraux par une répartition différente des actes de dialogue searliens (en particulier une proportion plus importante de directives et plus faible d'assertifs au profit des expressifs).

- il pouvait être divisé en quatre sous-corpus distincts qui reflètent les quatre principales activités de l'utilisateur lorsqu'il interagit avec l'agent conversationnel assistant (contrôle de l'agent, demande d'assistance directe ou indirecte et discussion avec l'agent). Bien que certainement non unique, ce découpage arbitraire est validé par une étude de caractérisation a posteriori.

Grâce à cette étude, j'ai pu concevoir de manière incrémentale une version plus élaborée du langage de requêtes DAFT dans laquelle une requête peut être formée de plusieurs modalités imbriquées encadrant plusieurs prédicats articulés entre eux et faisant appel à un certain nombre de références extraites de la requête sous forme langagière. En utilisant le corpus, j'ai ainsi retranscrit plusieurs centaines de phrases, ce qui m'a permis de définir des schémas élaborés et typés pour toutes les modalités et prédicats rencontrés.

6.2 Perspectives

Puisque le langage DAFT défini lors de ce stage sert à la communication entre les étages GRASP et DAFT (cf fig. 1), la poursuite du travail consistera logiquement à développer la chaîne de traitement selon ces deux directions :

- en travaillant en amont sur le module d'analyse sémantique GRASP pour lui ajouter une couche prenant place au-dessus de l'analyse grammaticale actuelle, afin qu'il puisse générer en sortie des requêtes en DAFT 2.0. L'analyse actuellement effectuée par GRASP fournit des résultats sous forme de boîtes imbriquées pour représenter la structure grammaticale de la phrase, et le passage à une forme DAFT 2.0 nécessitera donc encore un effort non négligeable.
- en se chargeant de la résolution des requêtes évoquée en introduction (cf section 1.3). L'idée que nous comptons développer se base sur la conception d'heuristiques capables de donner un sens pratique pour l'application à des opérateurs abstraits, en s'appuyant notamment sur une ontologie des verbes d'action²⁴ et des regroupements de schémas en classes d'équivalence tel qu'évoqué en 5.4.2.

7. REMERCIEMENTS

Nos remerciements vont à Patrick Paroubek (LIMSI-LIR) pour nous avoir fourni des extraits du corpus Multitag, ainsi qu'aux multiples interlocuteurs de Marco sur le site du GT ACA pour leurs "dons cognitifs" qui nous furent précieux pour l'obtention d'un corpus DAFT représentatif de la fonction d'assistance que nous souhaitions étudier.

²⁴A l'image de la thèse de K. Chibout (LIMSI-LIR 1999)

8. REFERENCES

- [1] J. F. ALLEN et G. FERGUSON : Trips : an integrated intelligent problem-solving assistant. In *AAAI '98/IAAI '98 : Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, p. 567–572, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.
- [2] J. AUSTIN : *How to Do Things with Words*. Oxford University Press, 1962.
- [3] J. BATES : The role of emotion in believable agents. *Commun. ACM*, 37(7):122–125, juil. 1994.
- [4] J. BESKOW, K. ELENUS et S. MCGLASHAN : OLGA - a dialogue system with an animated talking agent. In *EUROSPEECH '97 - 5th European Conf. on Speech Comm. and Tech.*, p. 1651–1654, sept. 1997.
- [5] M. E. BRATMAN, D. J. ISRAEL et M. E. POLLACK : Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4):349–355, 1988.
- [6] S. BUISINE, S. ABRILIAN, R. NIEWIADOMSKI, J.-C. MARTIN, L. DEVILLERS et C. PELACHAUD : Perception of blended emotions : From video corpus to expressive agent. In J. GRATCH, M. YOUNG, R. AYLETT, D. BALLIN et P. OLIVIER, édés : *Proceedings of IVA 2006*, vol. 4133 de *Lecture Notes in Computer Science*, p. 93–106. Springer, août 2006.
- [7] S. BUISINE et J.-C. MARTIN : Children's and adults' multimodal interaction with 2d conversational agents. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 1240–1243, Portland, Oregon, USA, avr. 2005. ACM Press.
- [8] M. BULUT, S. S. NARAYANAN et A. K. SYRDAL : Expressive speech synthesis using a concatenative synthesizer. In *Proceedings of ICSLP 2002*, Denver, Colorado, USA, oct. 2002.
- [9] J. CASSELL, T. BICKMORE, M. BILLINGHURST, L. CAMPBELL, K. CHANG, H. VILHJLÁMSSON et H. YAN : Embodiment in conversational interfaces : Rea. In *CHI '99 : Proceedings of the SIGCHI conf. on Human factors in comp. syst.*, p. 520–527, New York, NY, USA, 1999. ACM Press.
- [10] M. CAVAZZA, F. CHARLES et S. J. MEAD : Interacting with virtual characters in interactive storytelling. In *AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, p. 318–325, New York, NY, USA, 2002. ACM Press.
- [11] D. GENTNER et J. NIELSEN : The Anti-Mac interface. *Commun. ACM*, 39(8):70–82, août 1996.
- [12] S. GÉRARD : *Un modèle cognitif pour l'interprétation des expressions référentielles dans le cadre d'un système générique de dialogue Homme-Machine*. Thèse de doctorat, Univ. Paris XI, juin 2003.
- [13] H. P. GRICE : Logic and Conversation. *Syntax and semantics*, 3:41–58, 1975.
- [14] G. JEFFERSON : *Studies in Social Interaction*, chap. Side Sequences, p. 294–338. New York : Free Press, 1972.
- [15] W. L. JOHNSON, J. W. RICKEL et J. C. LESTER : Animated pedagogical agents : Face-to-face interaction in interactive learning environments. *Int. Journal of AI in Education*, 11:47–78, 2000.
- [16] C. KEMKE : Natural language control of agent systems, or "how to talk to your toaster". In *Proceedings of International Symposium on INnovations in Intelligent SysTems and Applications (INISTA) 2005*, Istanbul, Turkey, juin 2005.
- [17] K. LE GUERN : Définition d'une architecture de médiateur pour des agents conversationnels animés. Mémoire de Master, Univ. Paris XI, sept. 2004.
- [18] D. LERAY : Daft swing : une famille de composants dialogiques pour un agent conversationnel assistant. Mémoire de Master, Univ. Paris XI, sept. 2005.
- [19] J. C. LESTER, S. A. CONVERSE, S. H. KAHLER, S. T. BARLOW, B. A. STONE et R. S. BHOGAL : The Persona Effect : Affective impact of animated pedagogical agents. In *CHI '97 : Proceedings of the SIGCHI conf. on Human factors in comp. syst.*, p. 359–366, New York, NY, USA, mars 1997. ACM Press.
- [20] S. J. MOLINSKY et B. BLISS : *Inventory of functions and conversation strategies*, p. 177–187. Prentice Hall, jan. 1994.
- [21] C. PELACHAUD et I. POGGI : Subtleties of facial expressions in embodied agents. *J. Vis. Comput. Anim.*, 13:301–312, 2002.
- [22] G. PITEL : *MICO : La notion de construction située pour un modèle d'interprétation et de traitement de la référence pour le dialogue finalisé*. Thèse de doctorat, Univ. Paris XI, sept. 2004.
- [23] G. RIPOCHE : *Sur les traces de Bugzilla*. Thèse de doctorat, Univ. Paris XI, juin 2006.
- [24] G. SABAH, J. VIVIER, A. VILNAT, J.-M. PIERREL, L. ROMARY et A. NICOLLE : *Machine, Langage et Dialogue*. Figures de l'interaction. L'Harmattan, Paris, jan. 1998.
- [25] J. R. SEARLE : *Speech Acts : An essay in the Philosophy of language*. Cambridge University Press, new éd'n, jan. 1969.
- [26] L. TESNIÈRE : *Éléments de syntaxe structurale*. Klincksieck, 1959.
- [27] K. R. THÓRISSON : Gandalf : An embodied humanoid capable of real-time multimodal dialogue with people. In *First ACM International Conference on Autonomous Agents*, p. 536–537, Marina del Rey, California, USA, fév. 1997. ACM Press.
- [28] D. TRAUM : Conversational agency : The trains-93 dialogue manager. In *Proceedings of the Twente Workshop on Language Technology : Dialogue Management in Natural Language Systems (TWLT 11)*, p. 1–11, 1996.
- [29] D. TRAUM, S. MARSELLA, J. RICKEL, J. GRATCH et W. SWARTOUT : Toward a new generation of virtual humans for interactive experiences. *IEEE Intell. Syst.*, 17(4):32–38, juil. 2002.
- [30] J. WEIZENBAUM : ELIZA – a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, jan. 1966.
- [31] T. WINOGRAD : *Understanding Natural Language*. Academic Press, New York, 1972.