

DAFT : langage de requêtes formel 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 visant à la réalisation d'un agent conversationnel assistant, nous cherchons à élaborer un langage de requêtes structuré de manière à faciliter la compréhension par le système des requêtes langagières issues des utilisateurs. La conception de cette représentation intermédiaire 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 été fortement améliorée, la majorité des systèmes 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 qu'on soit conscient de leurs limites [3]. Pourtant, la population des utilisateurs a 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 dont l'interface correspond de moins en moins à la représentation cognitive qu'ils en ont : l'assistance devient donc cruciale. Mais l'offre disponible actuellement (généralement sous forme de manuels papier ou de pages web) est inadaptée car :

- impersonnelle (désincarnée et ignorant les spécificités de l'utilisateur)

¹source : enquête INSEE, Mars 2005

²source : enquête INSEE, Mai 2006

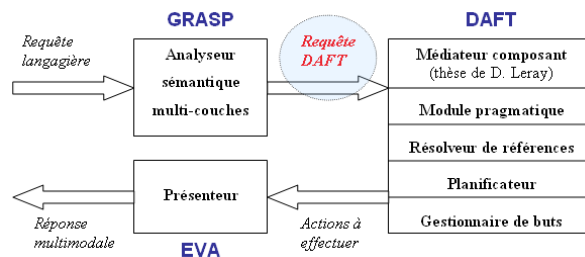


Figure 1 : Architecture globale du projet DAFT

- uniquement réactive (incapable d'anticiper les problèmes)
- statique (ne tenant pas compte du contexte dynamique d'utilisation)

1.2 Contexte du stage : le projet DAFT et les agents conversationnels

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 triple problème par l'utilisation d'un agent conversationnel animé, doté de capacités d'introspection au sein du modèle de l'application pour répondre aux questions de l'utilisateur 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 l'utilisateur s'exprime le mieux.

Depuis le développement d'ELIZA [13] à la fin des années 60, le panorama des agents conversationnels s'est considérablement enrichi. Depuis la fin des années 90 et le développement d'OLGA [1] au KTH et des travaux de Cassell et al. sur REA [2], les agents conversationnels sont également souvent incarnés (visibles pour l'utilisateur sous la forme d'un avatar), animés, et dotés de synthèse vocale. De plus en plus, on tente également de les rendre multimodaux en entrée comme en sortie, et à les doter d'émotions [7].

Leurs domaines d'utilisation sont très variés : de la formation à l'utilisation de machines ou à la négociation [12] aux jeux vidéos, en passant par de véritables compagnons quotidiens - la fonction d'assistance, à laquelle nous nous intéressons, n'est donc qu'un champ particulier parmi beaucoup d'autres.

Actuellement en version 1, DAFT se compose de 3 modules distincts (cf fig. 1) et se retrouve donc à la confluence de plusieurs communautés scientifiques : Traitement Automatique des Langues pour l'analyseur linguistique, Intelligence Artificielle pour l'agent rationnel servant notamment à la résolution de références, Composants Logiciels pour la gestion du médiateur et IHM pour la partie visuelle et l'animation.

1.3 Problématique

Mon stage porte sur la création d'un langage formel (par opposition à la langue naturelle) de requêtes, 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.

Par rapport à la version précédente, le langage à construire se distingue donc par :

- une **approche corpus** basée sur une annotation ethno-méthodologique pour un langage mieux adapté,
- des **schémas** plus complexes, permettant de mieux capturer la richesse sémantique des requêtes de l'utilisateur.

2. ÉTUDE DU CORPUS DAFT

2.1 Présentation du corpus

Notre étude a été menée à partir d'un corpus de requêtes d'assistance en langue naturelle constitué de 5 189 phrases fin mars 2006³. Elles ont été recueillies par le biais d'un agent WebLEA⁴, essentiellement via le site internet du GT ACA⁵, ce qui explique la forte proportion de requêtes liées à la navigation et à la recherche d'informations sur un site. Le reste des phrases correspond à des requêtes d'assistance pour l'utilisation d'applets développées dans le cadre du projet Interviews⁶ : Coco le compteur et un jeu de tours de Hanoi. Notons que nous travaillons exclusivement sur des requêtes *isolées*, c'est-à-dire issues d'interactions dialogiques limitées à un seul tour (ce n'est donc pas du dialogue homme-machine au sens traditionnel). Ainsi, même si le corpus contient des 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.

2.2 Étude comparative de corpus d'assistance

En suivant la méthode de comparaison statistique de corpus établie par Ripoche [8], on peut tout d'abord comparer le corpus DAFT à quelques autres corpus de dialogues orientés tâche en fonction de la répartition des 5 types d'actes de dialogue searliens au sein de ceux-ci (cf fig. 2) : assertifs (AST), promissifs (COM), directifs (DIR), expressifs (EXP), performatifs (PRF) - la 6^e colonne (UKN) correspondant aux catégories des taxonomies d'origine qui n'ont pas pu être retranscrites dans une des catégories précédentes. Les 3 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

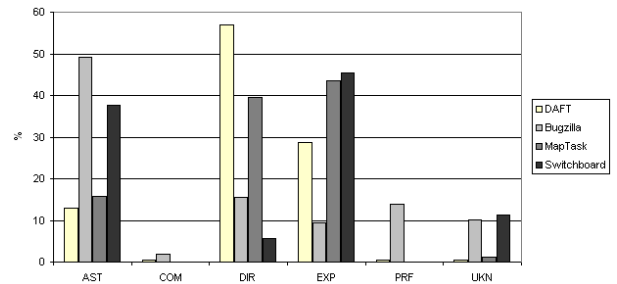


Figure 2 : Répartition des actes de dialogues searliens dans différents corpus

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 4 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. Toutefois, on observe assez nettement certaines caractéristiques distinguant le corpus DAFT des 3 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) sont intéressantes et justifient bien la nécessité de recueillir un corpus propre à notre champ applicatif. Une comparaison avec des corpus de nature différente⁷ serait également intéressante mais sort du cadre de ce stage.

2.3 Catégorisation du corpus

Nous avons dans un premier temps travaillé sur un sous-ensemble d'un dixième du corpus, soit environ 500 phrases tirées aléatoirement, afin de laisser la possibilité d'annoter l'intégralité des phrases sélectionnées (ce qui n'était pas envisageable pour l'ensemble du corpus pour un unique annotateur 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 affecte très peu la distribution, confortant ainsi la méthodologie employée.

⁷Notamment un corpus plus "généraliste" tel que Multitag fourni par P. Paroubek (LIMSILIR)

³On en compte environ 7 200 fin juin 2006

⁴LIMSIL Embodied Agent

⁵<http://www.limsi.fr/aca/>

⁶<http://www.limsi.fr/jps/interviews/index.html>

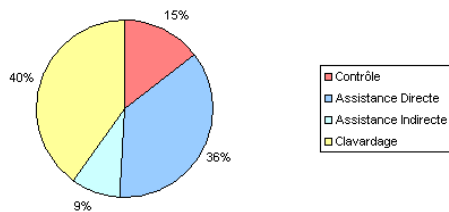


Figure 3 : Répartition du corpus en sous-corpus

Le regroupement manuel de requêtes relevant d'activités similaires a permis de distinguer assez rapidement 4 catégories de tailles inégales (cf fig. 3) 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 constitue des demandes d'aide sous-entendues, probablement perceptibles uniquement au niveau de la pragmatique.
4. **activité de clavardage** : réunissant le reste des interactions et essentiellement centrées sur l'agent ainsi que les expressions métalinguistiques et phatiques⁸ et de backchanneling⁹.

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 inmanquablement, un dialogue s'établit avec l'agent en tant qu'entité individuelle digne d'intérêt. Ce problème pourrait être en grande partie évité par la suppression de l'avatar de l'agent, mais l'utilité de sa présence physique facilite indéniablement la communication par rapport à un simple champ de texte : c'est le "persona effect" [5]. Notre étude étant centrée sur les requêtes d'assistance, on pourrait considérer ces interactions avec l'agent comme parasites et les ignorer, car elles relèvent davantage du dialogue homme-machine plus traditionnel. Néanmoins, de part leur importance quantitative (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 rajoutant un 2^e sous-ensemble d'un dixième du corpus à celui d'origine, ce qui nous a également permis de vérifier l'ensemble des mesures statistiques sur 2 ensembles indépendants.

⁸Pour maintenir le contact communicatif avec le locuteur

⁹Pour marquer son accord aux propos du locuteur et l'inciter à continuer

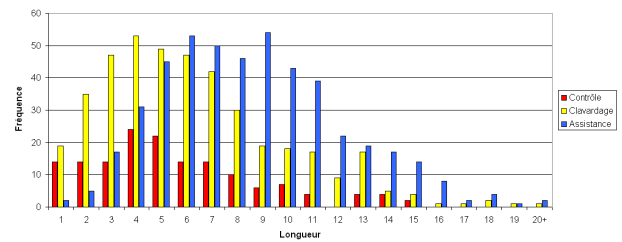


Figure 4 : Répartition des phrases par longueur dans les 4 sous-corpus

2.4 Première caractérisation des sous-corpus

Ces 4 sous-corpus dont on peut observer quelques exemples dans la table 1 présentent clairement une difficulté croissante pour être mis sous forme de requêtes formelles :

1. les requêtes de contrôle sont essentiellement prédictives (plus des 3/4) et courtes (longueur moyenne : 5,4 mots).
2. les requêtes d'assistance directes et indirectes sont dans l'ensemble plus longues (longueur moyenne : 9 mots) et ont majoritairement 2 ou 3 modalités (ex. de modalités pour la phrase 6 : « avoir peur », « exister un moyen de »).
3. les requêtes de dialogue sont très diverses (longueur moyenne : 6 mots), allant des plus courtes (phrase 9) nécessitant des modalités bien à part pour les traiter, à des phrases longues (phrase 12) et mettant en jeu une très grande variété de prédicats différents contrairement aux requêtes de contrôle (ex. phrase 11 : « chanter »). Cela signifie que l'intégration de ces requêtes nécessitera une extension importante du langage à établir.

La répartition des phrases en fonction de leur longueur (cf fig. 4) n'est visiblement pas suffisante pour distinguer les 4 sous-corpus, mais nous verrons par la suite (cf section 4.2) une méthode plus efficace basée sur l'étude des requêtes du langage élaboré.

3. ÉLABORATION DU MODÈLE DE LANGAGE DE REQUÊTES

3.1 Modèle naïf

Dans une première version (stage de K. Le Guern [4]), l'analyse des requêtes se faisait selon un filtre en 3 couches distinctes et uniques :

- Modalités (notées M) : ce sont des actes de dialogues, au sens large et non searlien [10], et sont inspirées des catégories de l'inventaire réalisé par Molinsky&Bliss [6].
- Prédicats (notés P) : il s'agit de verbes d'action (ex : modifier, déplacer...) ou de prédication (ex : être prêt/rouge...).
- Références (notées R_i) : elles correspondent généralement aux groupes nominaux des requêtes et sont à prendre au sens de références extensionnelles associatives¹⁰.

Ce qui fournissait une analyse sémantique des phrases de la forme : $M [P (R_1, R_2, \dots)]$

Néanmoins, cette approche montre rapidement ses limites sur des requêtes plus complexes comme la n° 6 de la table 1,

¹⁰Objet de 2 thèses rien qu'au sein du groupe AMI (S. Gérard 2003 et G. Pitel 2004)

Table 1 : 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 -

et on se rend rapidement compte qu'une unique modalité n'est pas suffisante et qu'un modèle plus élaboré, dans lequel plusieurs modalités sont capables de s'articuler entre elles, est nécessaire.

3.2 Modèle multicouche hiérarchisé

3.2.1 Syntaxe

Dans ce nouveau modèle, 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 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*,

$c_1 - c_m$ & $c'_1 - c'_l$ sont les intitulés des champs tels que décrits dans les schémas des modalités et des prédicats,

$R_1 - R_l$ sont des *références* issues de la requête langagière.

Les modalités et les prédicats sont définis en fonction de *schémas*, qui se basent essentiellement sur leur structure acoustique [11] à la manière de ce que propose FrameNet.¹¹ Néanmoins, les spécifications du langage formant un document d'une quarantaine de pages, nous donnerons ici une vue macroscopique en présentant les intitulés de l'ensemble

¹¹Ressources lexicales en langue anglaise pour une analyse de phrases selon des "frames"

des schémas définis avant d'en examiner plus précisément quelques-uns et en analysant certains des choix effectués.

Avec cette nouvelle syntaxe, la phrase 6, non traitable par l'approche en 3 couches, peut désormais s'écrire comme indiqué sur la ligne en-dessous d'elle dans la table 1.

3.2.2 Les modalités M_i

Comme le montre la table 2, les modalités peuvent être regroupées en 8 catégories. 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 seront plutôt sur les couches intérieures précédant les prédicats (ex : phrase 5). Les structures de leurs schémas sont également de complexités inégales, de NEG ou CHECK qui ne prennent qu'un argument à des schémas plus complexes comme OBLIGATION(of=[pers], by=[pers], todo=[act]), même si tous les champs du schéma ne sont pas nécessairement remplis simultanément.

3.2.3 Les prédicats P_i

On vérifie que leur fréquence suit une loi de Zipf (cf fig. 5) et on pourrait donc supposer que leur nombre est virtuellement illimité. Néanmoins, si l'on se concentre sur les sous-corpus de contrôle et d'assistance, ces 53 prédicats constituent une base assez complète d'actions génériques dans le cadre d'une application ou d'un site web. Leurs schémas sont de complexités variées, allant de schémas vides (ex : Biper()) à des schémas comportant 4 ou 5 champs, particulièrement pour les actions les plus fréquentes telles que Modifier(objet, propriété, valeur, manière).

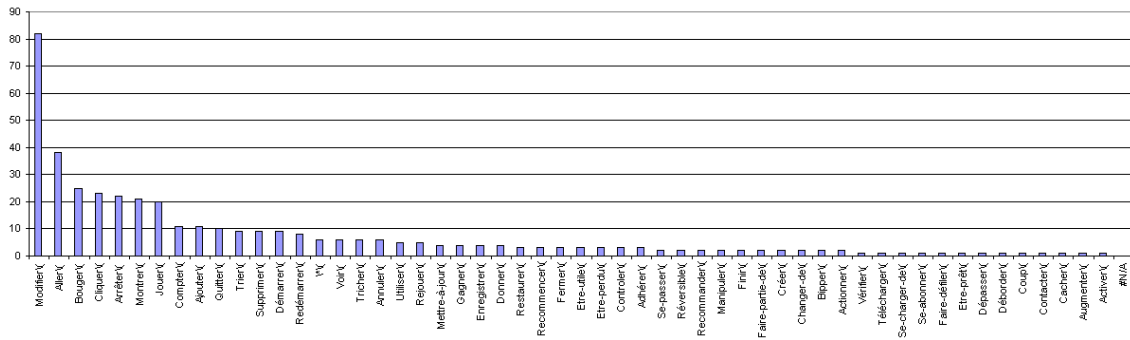


Figure 5 : Répartition en fréquence des 53 prédicats P_i DAFT dans l'ensemble du corpus

Table 2 : Liste des 39 modalités M_i DAFT

Informations sur une entité			
INFOS	MEANING	TYPE	VALUE
PROPERTY	FUNCTION	ROLE	LIMIT
FUNCTIONING			
Capacités, droits et devoirs			
POSSIBILITY	KNOWLEDGE	OBLIGATION	WILL
PROBABILITY			
Identification d'entités			
EXISTENCE	OTHER	OBJECT	
Relations inter-entités			
DIFFERENCE	ORDER		
Circonstancielle			
WAY	EFFECT	REASON	PLACE
MOMENT	NUMBER		
Sentiments			
LIKE	FEAR	BOTHER	DOUBT
SURPRISE	REGRET	HAPPY	
Assistance			
PROBLEM	MISTAKE	HELP	TELL
Marqueurs			
CHECK	ASK	NEG	

3.2.4 Les fonctions communicatives

Les fonctions communicatives sont à placer au niveau des modalités dans l'ordre d'imbrication. Elles s'en distinguent cependant par le fait qu'elles ne peuvent contenir de prédicats ou d'autres fonctions communicatives, et n'ont été introduites que pour prendre en compte une partie du sous-corpus de clavardage. Les classes correspondant à ces fonctions sont au nombre de 4 : ACK, PING, HELLO et BYE, et la structure de leurs schémas étant très simple, nous ne nous y attarderons pas.

3.2.5 Les références

Une analyse plus fine des références fait partie des tâches restant à réaliser d'ici la fin du stage. Elles sont pour l'instant le plus souvent gardées sous leur forme brute, même si des champs ont été prévus dans les schémas de prédicats en vue de leur découpage (ex. de la phrase 1).

3.2.6 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. On peut citer l'exemple de la modalité 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* dans l'application,
2. si l'agent a les *capacités* et les connaissances nécessaires pour réaliser cette action,
3. si le sujet est *autorisé* à réaliser cette action à l'instant de la demande.

En effet, selon les maximales de quantités de Grice, 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 3 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 en prendre note) et dès lors la nuance sémantique est caduque.

4. ÉVALUATION

4.1 Couverture du corpus par le langage

Le traitement du corpus s'est donc fait en 2 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. Cette couverture est cependant inégale (cf table 3), 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 compte¹².

¹²Cela 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.

Table 3 : Taux de couverture des 4 sous-corpus par le langage DAFT

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

*(hors dialogue)

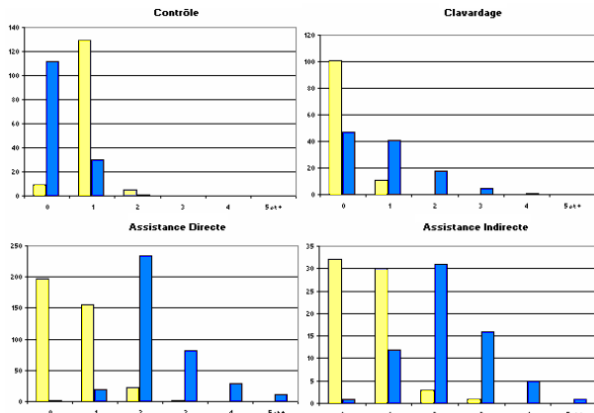


Figure 6 : Nombre de schémas par phrase dans les 4 sous-corpus (clair = prédicats, foncé = modalités)

Le fait d'avoir travaillé indépendamment sur 2 extraits au 1/10^e nous a permis de vérifier le caractère générique du langage proposé puisque peu de nouveaux schémas de modalités et de fonctions communicatives ont été introduits. Néanmoins, une étude plus détaillée sur ce sujet reste à faire (voir les perspectives en conclusion).

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

La caractérisation des sous-corpus a priori (cf section 2.4) étant assez peu concluante, nous avons tenté une nouvelle caractérisation basée sur l'étude de la proportion de schémas de prédicats et de modalités (cf fig. 6). On constate cette fois une très nette différence entre les 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).

5. CONCLUSION

5.1 Bilan

J'ai montré que la nature de l'interaction homme-machine permettait de distinguer le corpus DAFT d'autres corpus d'assistance, et qu'il pouvait être divisé en 4 pour refléter les différentes activités de l'utilisateur. Bien que certainement non unique, ce découpage arbitraire est validé par une étude

de caractérisation a posteriori, et m'a aidé à concevoir de manière incrémentale un langage de requêtes.

5.2 Perspectives

D'ici la fin du stage, plusieurs tâches restent encore à accomplir pour valider le langage proposé et présenté ici. Il conviendra notamment de :

- évaluer la couverture du langage sur une nouvelle tranche de 1/10^e du corpus DAFT, pour étudier la robustesse du langage actuel en déterminant notamment le nombre de prédicats (et éventuellement de modalités le cas échéant) à ajouter lors d'une augmentation de 50% du corpus (l'ensemble actuellement étudié couvrant 1/5^e).
- étudier plus précisément la structure des références parfois complexes (ex : « certains paramètres de configuration de l'application ») qui sont pour l'instant souvent laissées sous forme brute dans les requêtes. On envisagera notamment de les étudier par le biais d'observateurs, selon l'approche développée dans le projet Interviews [9].

Le langage DAFT défini lors de ce stage servant à la communication entre les étages GRASP et DAFT (cf fig. 1), 2 possibilités de poursuite se dessinent :

- travailler en amont sur le module d'analyse sémantique GRASP en ajoutant une couche au-dessus de l'analyse grammaticale actuelle pour générer les requêtes en DAFT,
- se charger de la résolution des requêtes évoquée en 1.3, notamment en donnant un sens à des opérateurs aussi abstraits pour l'application que "avoir peur de", en s'appuyant sur une ontologie des verbes d'action¹³.

6. REFERENCES

- [1] 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.
- [2] J. CASSELL, T. BICKMORE, M. BILLINGHURST, L. CAMPBELL, K. CHANG, H. VILHJÁLMSSON 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.
- [3] D. GENTNER et J. NIELSEN : The Anti-Mac interface. *Commun. ACM*, 39(8):70-82, août 1996.
- [4] K. LE GUERN : Définition d'une architecture de médiateur pour des agents conversationnels animés. Mémoire de Master, Univ. Paris XI, sept. 2004.
- [5] 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.
- [6] S. J. MOLINSKY et B. BLISS : *Inventory of functions and conversation strategies*, p. 177-187. Prentice Hall, jan. 1994.
- [7] C. PELACHAUD et I. POGGI : Subtleties of facial expressions in embodied agents. *J. Vis. Comput. Anim.*, 13:301-312, 2002.
- [8] G. RIPOCHE : *Sur les traces de Bugzilla*. Thèse de doctorat, Univ. Paris XI, juin 2006.

¹³A l'image de la thèse de K. Chibout (LIMSI-LIR 1999)

- [9] J.-P. SANSONNET : Notes sur VQL 0.4. Rap. tech., LIMSI-CNRS, mai 2001.
- [10] J. R. SEARLE : *Speech Acts : An essay in the Philosophy of language*. Cambridge University Press, new édn, jan. 1969.
- [11] L. TESNIÈRE : *Élément de syntaxe structurale*. Klincksieck, 1959.
- [12] 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.
- [13] 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.