

Langage formel et heuristiques pour des agents conversationnels assistants

François Bouchet

bouchet@limsi.fr

Directeur de thèse : Jean-Paul Sansonnet

Présentation au groupe AMI

7 décembre 2006

Plan

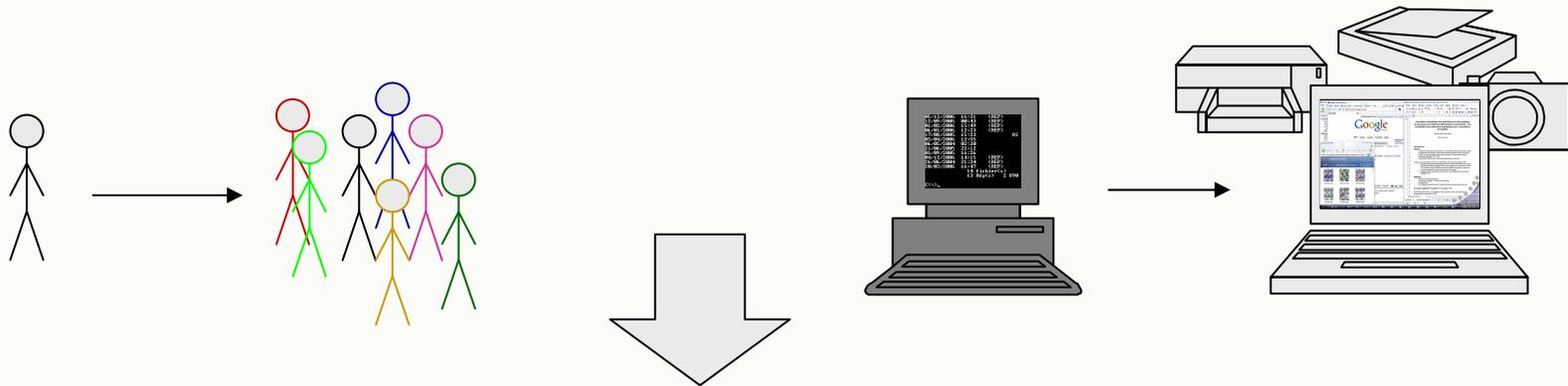
- Contexte :
 - ◆ L'assistance
 - ◆ Les agents conversationnels (animés)
 - ◆ Le projet Daft

- Travaux de Master Recherche :
 - ◆ Etude du corpus Daft
 - ◆ Proposition d'un langage de requêtes

- Sujet de thèse :
 - ◆ Production automatique de requêtes formelles
 - ◆ Interprétation pragmatique des requêtes formelles à l'aide d'heuristiques

Le besoin d'assistance

- La population d'**utilisateurs** d'outils informatiques a considérablement évolué en 30 ans :
 - ◆ En **nombre** : la moitié des foyers français ont au moins un ordinateur.
 - ◆ En **diversité** : informatique d'informaticiens → informatique pour tous.
- Les **machines** sont plus puissantes (loi de Moore) et ont élargi ainsi :
 - ◆ Leur **champ d'application** : traitement de texte, multimédia, jeux vidéos, recherche d'information...
 - ◆ Leur **champ d'interaction** : imprimante, scanner, PDA, appareil photo...



Inadéquation croissante entre le système et la représentation cognitive qu'en ont les utilisateurs.

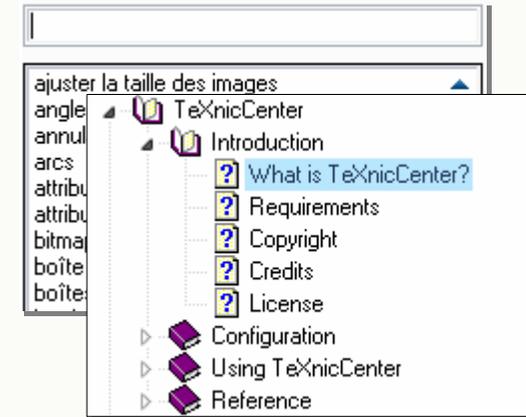
Evolution du traitement de l'assistance (1/3)

- Années 80 : le mythe de la GUI transparente :
 - ◆ Principe de la **Manipulation Directe** [Schneidermann, 83] : manipulation des entités internes à l'application via une GUI conviviale et intuitive.
 - ◆ Principe de Transparence : pas besoin d'aide car les usagers occasionnels s'**autoforment** (« learning by doing »).
 - ◆ Mais les **connaissances** des applications acquises ainsi sont :
 - **imparfaites** (conception erronée du fonctionnement),
 - **limitées** (faible partie des opérations disponibles connue et à plus forte raison utilisée),ruinant le mythe de l'autoformation [Duffy et al 89 ; Carroll & McKendree 87 ; Fisher et al 85 ; Streitz 88].



Evolution du traitement de l'assistance (2/3)

- Années 90 : les aides en ligne hypertextuelles
 - ◆ Utilisation des techniques issues du web pour des « manuels papier interactifs », FAQ et index de mots-clés.
 - ◆ Mais les aides sont **délaissées** par les novices au profit de formations, hotlines, amis experts... et le **côté intrusif** irrite (« Clippie effect »).
 - ◆ 2 grandes difficultés [Mack et al 83 ; Kearsley 98 ; Sellen&Nicol 90] :
 - Comment savoir quoi demander et où le trouver ?
 - Informations délivrées souvent non pertinentes ou non comprises.
 - ◆ Besoin crucial de **contextualisation** [Jameson 03] :
 - Adaptation aux profils variés des nouveaux usagers grand public ;
 - Adaptation en fonction de l'état courant de la tâche-usager [Capobianco & Carbonell 01].



Evolution du traitement de l'assistance (3/3)

- Années 2000 : l'apport des aides multimodales / dialogiques
 - ◆ La multimodalité permet la complémentarité et limite ainsi l'intrusion cognitive de l'aide :
 - Manipulation tâche-usager via la GUI (clavier-souris) ;
 - Aide demandée et délivrée par modalité orale (reco et synthèse de la parole) et/ou déictique.
 - ◆ La **langue naturelle non contrainte** est la **modalité optimale d'expression des problèmes** [Carbonell & Capobianco 02], même si les usagers sont capables d'adaptation à des sous-langages [Kennedy et al 88 ; Amalberti et al 93].
 - ◆ Deux limites actuelles de la modalité dialogique :
 - Coûts (en temps, en main d'œuvre) ;
 - Manque de robustesse des systèmes existants.

Agents conversationnels (animés)

Agent

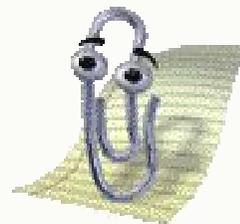
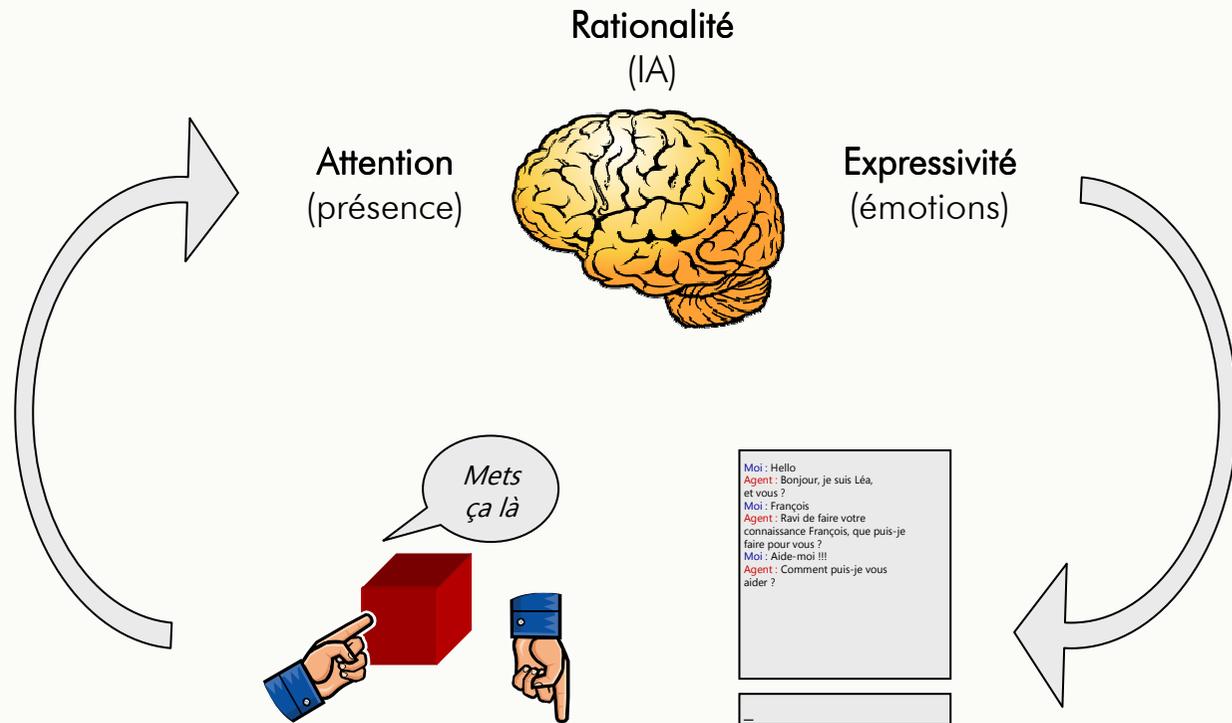
Raisonnement

Conversationnel

Interaction

Animé

Personnification

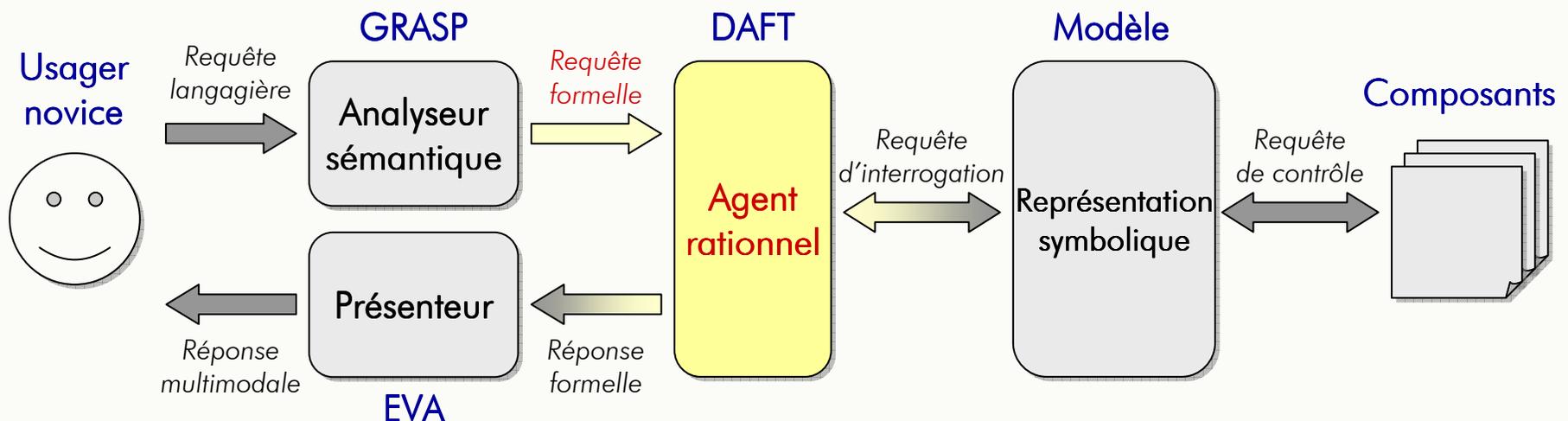


Le projet Daft

■ Objectifs :

- ◆ **Agents Conversationnels Animés** dédiés à la **fonction d'assistance** :
 - Pour : les utilisateurs novices s'exprimant en langue naturelle non contrainte
 - Par : un raisonnement sur la structure et le fonctionnement des applications (aide contextuelle)

■ Schéma d'architecture générale :



Plan

- Contexte :
 - ◆ L'assistance
 - ◆ Les agents conversationnels (animés)
 - ◆ Le projet Daft

- Travaux de Master Recherche :
 - ◆ Etude du corpus Daft
 - ◆ Proposition d'un langage de requêtes

- Sujet de thèse :
 - ◆ Production automatique de requêtes formelles
 - ◆ Interprétation pragmatique des requêtes formelles à l'aide d'heuristiques

Problématique du stage de M2R

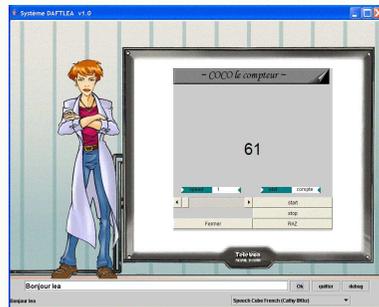
- Etude du corpus Daft
 - ◆ **Constitution** d'un corpus
 - Pourquoi un corpus ?
 - ◆ **Couverture** du corpus
 - Est-il représentatif du domaine étudié ?
 - ◆ **Spécificités** du corpus
 - Est-il différent d'autres corpus disponibles ?
 - ◆ **Catégorisation** du corpus par ses activités
 - Est-il uniforme ?
 - ◆ **Caractérisation** des activités
 - Quelles sont leurs distinctions ?
- Proposition d'un langage de requêtes
 - ◆ Basé sur l'étude de corpus
 - ◆ Permettant d'exprimer une sémantique plus fine que la version précédente

Constitution du corpus Daft

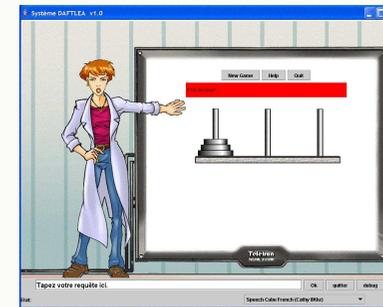
- On ne s'intéresse qu'à un domaine restreint de la langue : les Systèmes de Traitement de Requêtes d'Assistance, sur des **requêtes isolées** (pas de succession de dialogues).
- 8000 requêtes recueillies entre juin 2004 et août 2006.
- 2 méthodes complémentaires :
 - ◆ Des requêtes réelles d'utilisateurs (2/3) → **empirisme**
 - ◆ Des structures dialogiques génériques issues de thésaurus employées en contexte (1/3) → **couverture**

qui permettent de garantir un **corpus suffisant**, c'est-à-dire représentatif du domaine étudié.

Les environnements de recueil du corpus



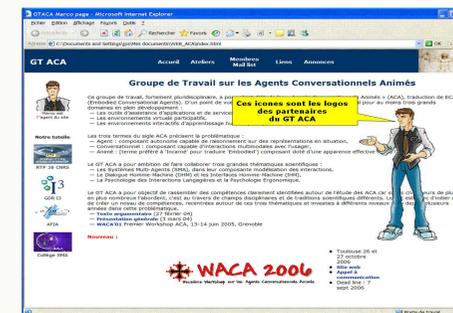
Application Java : compteur dont l'utilisateur contrôle la vitesse et le démarrage (thread)
Modèle : construit en parallèle de l'application
Public : restreint (~2/3 étudiants en informatique et ~1/3 d'utilisateurs novices)



Application Java : jeu de tours de Hanoi (fonctionnement modal)
Modèle : construit a posteriori, par filtrage automatique
Public : restreint (~2/3 étudiants en informatique et ~1/3 d'utilisateurs novices)



Site web : Version active du site du groupe AMI du LIMSI
Modèle : construit a posteriori, manuellement
Public : restreint (~2/3 étudiants en informatique et ~1/3 d'utilisateurs novices)



Site web : Site internet du GT ACA
Modèle : construit a posteriori, manuellement
Public : Non contrôlé, ouvert à tous sur le web

Une vue du corpus Daft

...

a plus

a+

ah

Allez, bye

Allez ciao.

Alors ?

As-tu des amis?

auf viedersen

à l'aide !

bah!

barre toi de là

bidule

bon à rien !

bon week end

Bon.

Bonjour

Bonsoir

...

...

À quoi sers-tu ?

alors **là** t'es **completement** paumé !

appelle moi simplement Sylvie

as tu des **informtion**?

as tu entendu parler d'une expérimentation en cours ??

au sujet de cette page, que **peut tu** dire ?

avec ce corpus, tu sauras ce qu'est une **anaphore** ...

à quoi penses tu?

be ouais tu comprends pas

ben alors reponds

bon j'en ai marre je me tire ...

Bon je me casse. Bye.

bon y a rien **â** tirer de toi !!

bon, ça va, bonne année 2006

Bon, dis-moi plutôt ce que tu sais faire

bon, reviens à **l apage** d'accueil

bonjourmon vieux

...

Études comparatives de corpus

Daft : 8k requêtes d'assistance

Corpus de textes "généralistes"

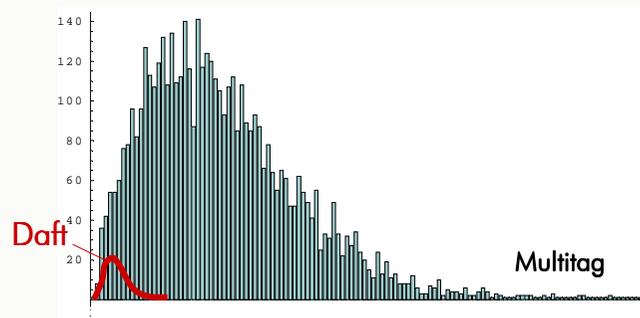
Corpus de dialogues humains
"orientés tâche"

Multitag [Paroubek, 2000] : phrases issues d'articles du journal Le Monde, de romans et d'essais.

Switchboard [Jurafsky et al., 1998] : 200k conversations téléphoniques.

MapTask [Carletta et al., 1996] : 128 dialogues (reconstruction d'une carte).

Bugzilla [Ripoche, 2006]: 1,2M de commentaires de rapport de bugs issus de 128k rapports de défauts.



Comparaison de corpus orientés tâches

■ Méthodologie de l'étude :

- ◆ Justification : Méthode de comparaison de corpus annotés avec des taxonomies différentes.
- ◆ Basée sur la notion de **profil interactionnel** [Ripoche, 2005]

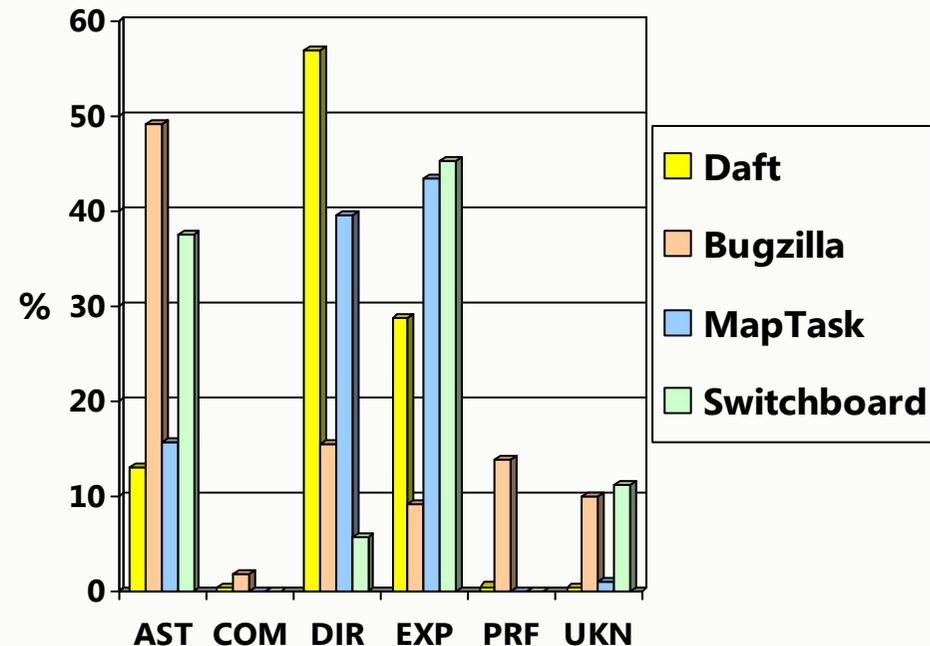
■ Définition :

« Représentation sous forme d'histogramme de la répartition des classes d'actes de dialogue dans un corpus donné. »

■ Résultats :

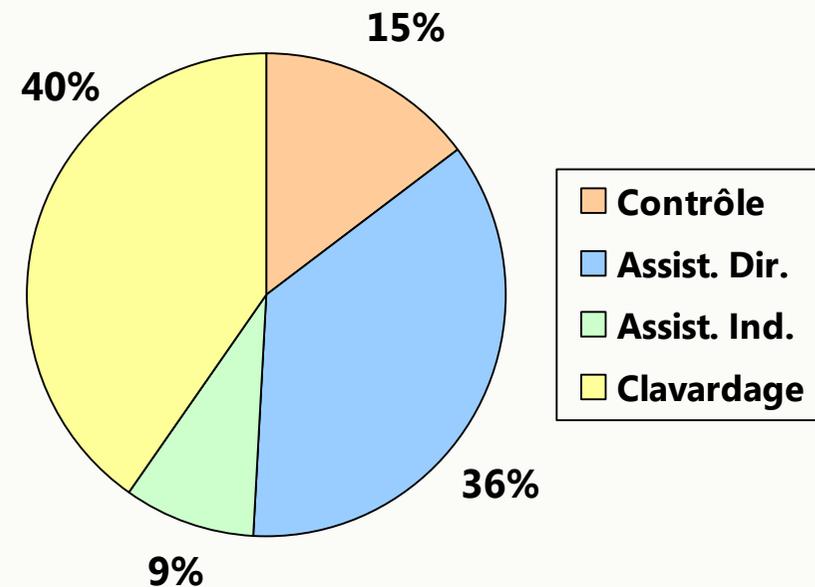
- ◆ Une majorité de directifs (57%) : distinction entre DHM et DHH.
- ◆ Plus d'expressifs que d'assertifs : l'utilisateur exprime plus ses sentiments que des faits « objectifs ».
- ◆ Promissifs marginaux : soumission de l'agent à l'utilisateur, et non l'inverse.

■ Conclusion : le corpus est **spécifique**, donc **nécessaire**.



Catégorisation du corpus Daft

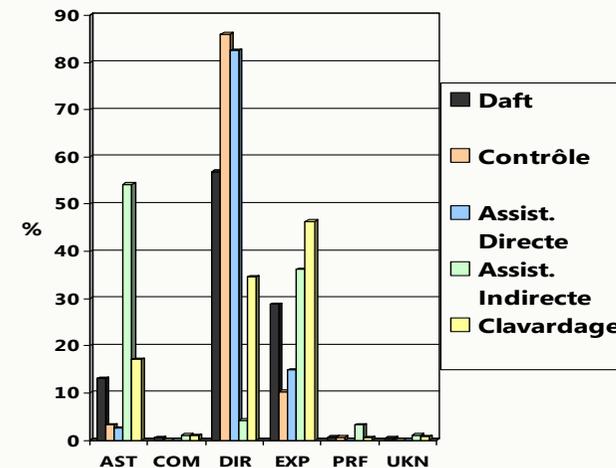
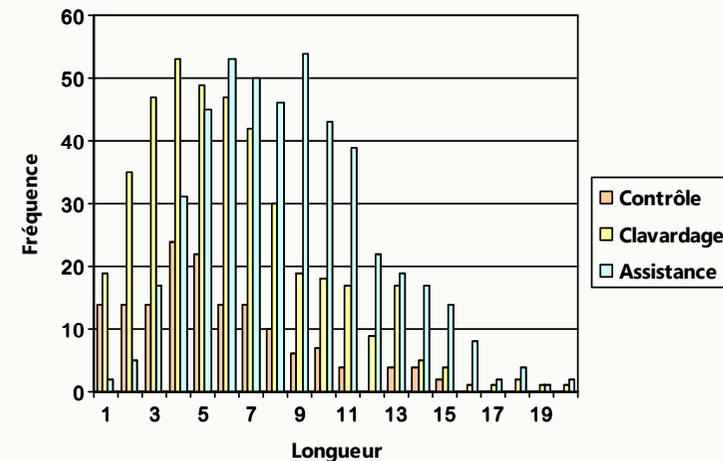
- Méthodologie :
Annotation manuelle des requêtes issues de deux sous-ensembles au 1/10^e du corpus.
- Résultats :
 - ◆ **Contrôle**
Commandes prédictives via l'agent
 - ◆ **Assistance directe**
Demandes d'aide explicites
 - ◆ **Assistance indirecte**
Commentaires sous-entendant une demande d'aide (pragmatique)
 - ◆ **Clavardage**
Interactions utilisateur-agent



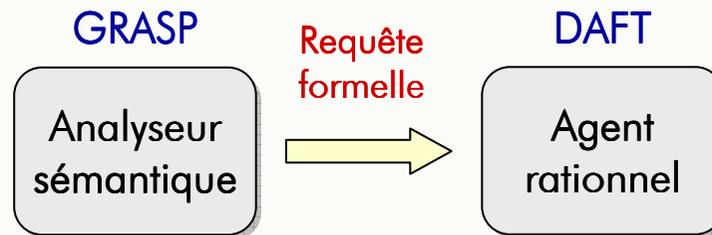
- Conclusion :
4 activités distinctes
→ **4 sous-corpus**

Caractérisation des activités

- Selon la **longueur** des requêtes :
 - ◆ Écart-types trop importants et moyennes trop proches.
- Selon les **profils interactionnels** :
 - ◆ Permet de distinguer uniquement assistance directe/indirecte, et automatisation non triviale .
- Ces activités sont :
 - ◆ De **complexité** de traitement **croissante**.
 - ◆ Facilement distinguables par un annotateur humain,
 - ◆ **Difficiles à classifier automatiquement**.



Définition d'un langage de requêtes formelles



- Le langage formel DAFT sert :
 - ◆ Pour la représentation de la sémantique des requêtes des usagers.
 - ◆ De support aux raisonnements de l'agent rationnel.
- Besoin d'un langage :
 - ◆ Capable d'exprimer une sémantique fine (structures complexes).
 - ◆ Ancré dans la réalité (phrases issues du corpus).

Le langage DAFT 2.0

- Les requêtes sont de la forme :

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

- ♦ **Modalités** (M_i) \approx actes de dialogue (« obligation », « avoir peur de », etc.)
- ♦ **Prédicats** (P_i) = verbes d'action ou de prédication (« modifier », « être prêt », etc.)
- ♦ **Références** (R_i) \approx GN (« le petit bouton », etc.) – Réf. Extensionnelles Associatives

- Exemples de requêtes :

Glisser le disque de droite à gauche	<code>Bouger(objet="le disque",origine="droite", destination="gauche")</code>
d'après toi, y a t-il des fonctions d'annulation dans cette application ?	<code>ASK(KNOWLEDGE(of=s, about=EXISTENCE(time=2, of=FUNCTION(doing=Annuler(), in="cette appli..."))))</code>
j'ai peur qu'il n'y ait pas moyen de changer la taille de la police qui est bien trop petite	<code>FEAR(agent=u, fear=NEG(POSSIBILITY(todo=Modifier(objet="la police...trop petite",propriété="taille"))))</code>

Le langage DAFT 2.0

■ 39 modalités :

- ◆ INFOS
- ◆ MEANING
- ◆ TYPE
- ◆ VALUE
- ◆ PROPERTY
- ◆ FUNCTION
- ◆ ROLE
- ◆ FUNCTIONING
- ◆ LIMIT
- ◆ POSSIBILITY
- ◆ KNOWLEDGE
- ◆ OBLIGATION
- ◆ WILL
- ◆ PROBABILITY
- ◆ EXISTENCE
- ◆ OTHER
- ◆ OBJECT
- ◆ DIFFERENCE
- ◆ ORDER
- ◆ WAY
- ◆ EFFECT
- ◆ REASON
- ◆ PLACE
- ◆ MOMENT
- ◆ NUMBER
- ◆ LIKE
- ◆ FEAR
- ◆ BOTHER
- ◆ DOUBT
- ◆ SURPRISE
- ◆ REGRET
- ◆ HAPPY
- ◆ PROBLEM
- ◆ MISTAKE
- ◆ HELP
- ◆ TELL
- ◆ ASK
- ◆ CHECK
- ◆ NEG

■ 53 prédicats :

- ◆ Actionner
- ◆ Activer
- ◆ Adhérer
- ◆ Ajouter
- ◆ Aller
- ◆ Annuler
- ◆ Arrêter
- ◆ Bipper
- ◆ Bouger
- ◆ Cacher
- ◆ Changer-de
- ◆ Cliquer
- ◆ Compter
- ◆ Contacter
- ◆ Contrôler
- ◆ Coup
- ◆ Créer
- ◆ Démarrer
- ◆ Dépasser
- ◆ Donner
- ◆ Enregistrer
- ◆ Etre-perdu
- ◆ Etre-prêt
- ◆ Etre-utile
- ◆ Faire-défiler
- ◆ Faire-partie-de
- ◆ Fermer
- ◆ Finir
- ◆ Gagner
- ◆ Jouer
- ◆ Manipuler
- ◆ Mettre-à-jour
- ◆ Modifier
- ◆ Montrer
- ◆ Quitter
- ◆ Recommander
- ◆ Recommencer
- ◆ Redémarrer
- ◆ Rejouer
- ◆ Restaurer
- ◆ Se-abonner
- ◆ Se-charger-de
- ◆ Se-passer
- ◆ Supprimer
- ◆ Télécharger
- ◆ Tricher
- ◆ Trier
- ◆ Utiliser
- ◆ Vérifier
- ◆ Voir

(Spécifications au 01/09/2006)

Le langage DAFT 2.0

- Chaque modalité / prédicat est doté d'un schéma d'imbrication.
- Exemple : OBLIGATION

Obligation **d'**une personne **par** une autre **à** faire quelque chose.

[p] of personne obligée
[p] by personne obligéant
[a] todo action obligatoire

- Comparaison DAFT 1.0 / DAFT 2.0 :

	DAFT 1.0	DAFT 2.0
Corpus utilisé	300 phrases construites	1000 phrases retranscrites manuellement parmi 5000 recueillies
Structure des requêtes	1 modalité, 1 prédicat, n références	n modalités imbricables, n' prédicats coordonnés, n'' références

Plan

- Contexte :
 - ◆ L'assistance
 - ◆ Les agents conversationnels (animés)
 - ◆ Le projet Daft

- Travaux de Master Recherche :
 - ◆ Etude du corpus Daft
 - ◆ Proposition d'un langage de requêtes

- **Sujet de thèse :**
 - ◆ Production automatique de requêtes formelles
 - ◆ Interprétation pragmatique des requêtes formelles à l'aide d'heuristiques

Problématique de la thèse

- Bilan du stage de M2R : une version 2.0 du langage DAFT :
 - ◆ Ancrée dans la réalité (approche corpus).
 - ◆ Permettant une expressivité fine de la sémantique des requêtes issues des usagers.

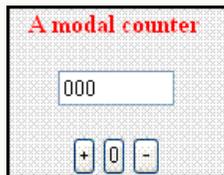
- Objectifs de la thèse :
 - ◆ Production automatique de requêtes formelles.
 - ◆ Interprétation pragmatique des requêtes formelles à l'aide d'heuristiques.
 - ◆ Un système facilement éditable pour expérimenter la pertinence des heuristiques auprès d'utilisateurs réels.

Production automatique des requêtes formelles

1. Phrase d'origine « *Peux-tu accélérer ?* »
2. Sortie de GRASP *TOCAN, THESYSTEM, TOACCELERATE, ?*
3. Transposition en DAFT 2.0 *{**POSSIBILITY()**, **PROBABILITY()**},
_person="system",
Modify(property="speed", value="+"), **ASK()***
4. Combinaison des éléments ***ASK**(**POSSIBILITY**(of="system",
todo=**Modify**(property="speed",
value="+"))))*
5. Mise sous forme canonique à l'aide de classes d'équivalence, en vue du traitement par heuristiques
Exemple :
***ASK**(**POSSIBILITY**(todo=...))
⇔ **WILL**(todo=...)*

Classes d'équivalence

- Principe : N formulations \Leftrightarrow 1 unique sens en contexte
- Différents niveaux :
 - ◆ D'ordre **grammatical** :
 - « la vitesse est changée par le curseur » \Leftrightarrow « le curseur change la vitesse »
 - ◆ D'ordre **sémantique** :
 - **ASK**(**POSSIBILITY**(todo=...)) \Leftrightarrow **WILL**(todo=...)
 - ◆ D'ordre **pragmatique** : utilisation de règles de sens commun (Grice)



- « active le compteur »
- « démarre l'appli » (*appli = compteur*)
- « vas y » (*action par défaut = lancer le compteur*)
- « dommage que le compteur soit bloqué »

\Leftrightarrow **WILL**(of=user,
todo=**Start**(object="compteur"))

Réactions pragmatiques

■ Définition de réactions pour les classes

- ◆ $\text{INFORM}_{\text{user}} [*P(x)] \rightarrow \text{INTENT}_{\text{syst}} [\neg P(x)]$
- ◆ $\text{REQUEST}_{\text{user}} [A(x,y)] \rightarrow \text{INTENT}_{\text{syst}} [\text{EXECUTE}[A(x,y)]]$
- ◆ $\text{JUDGE}_{\text{user}} [*J(x)] \rightarrow \text{INTENT}_{\text{syst}} [\text{rev}(J,x) (x)]$

**: péjoratif*

P: proposition

A: action

J: jugement

rev: opérateur de réversion

« tu parles trop fort »

→ $\text{JUDGE}_{\text{user}} [*TOOMUCH(\text{system.sound.level})]$

→ $\text{rev}(\text{TOOMUCH}, x:\text{NUM}) = \text{EXECUTE}[\text{DECREASE}(x, 20\%)]$

■ Génération de réponses à l'utilisateur

« j'aime pas tes yeux »

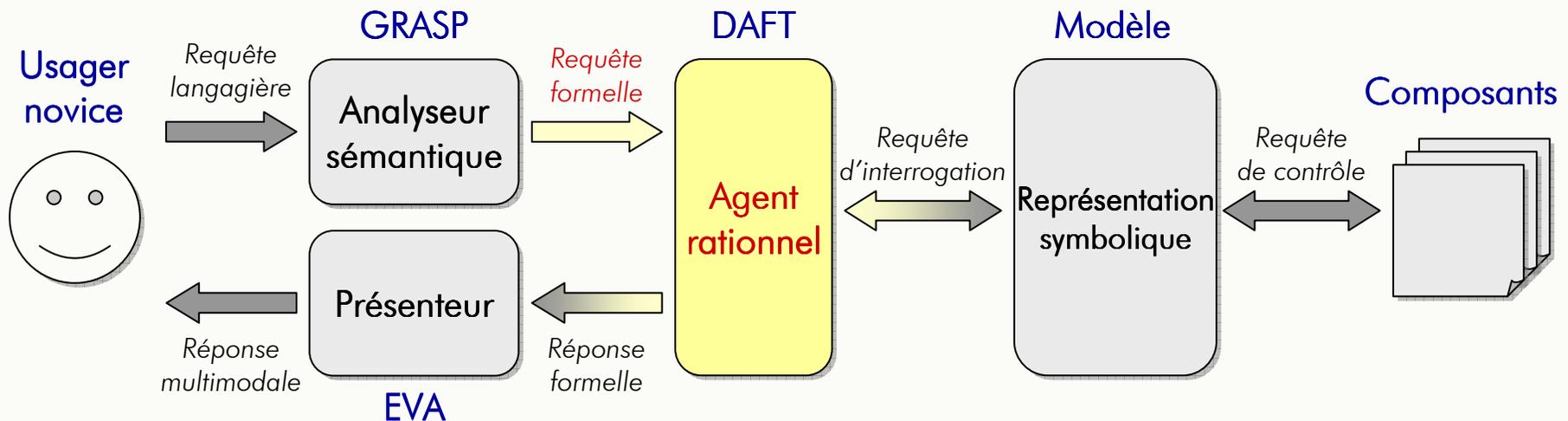
→ $\text{DISLIKE}_{\text{user}} [\text{LOOK}(\text{system.bodypart.eye})]$

→ $\text{INTENT}_{\text{syst}} [\text{rev}(J) (x)]$

→ $\text{ANSWER}_{\text{syst}} [< \text{No modifying operator for 'LOOK'} >]$

« Désolé, je ne puis modifier mon apparence »

Conclusion



■ Stage M2R :

- ◆ **Etude** de corpus :
 - Spécifique et couvrant.
 - 3 grandes activités (contrôle, assistance, clavardage).
- ◆ **Définition** d'un langage de requêtes formelles
 - Ancré dans la réalité.
 - Permettant une expressivité fine de la sémantique des requêtes.

■ Thèse :

- ◆ **Production automatique** de requêtes formelles.
- ◆ **Interprétation** pragmatique des requêtes formelles à l'aide d'**heuristiques**.
- ◆ Un système facilement éditable pour expérimenter la pertinence des heuristiques auprès d'utilisateurs réels.